# IDPR MIB

# Definitions of Managed Objects for the Inter-Domain Policy Routing Protocol (Version 1)

# Contents

| <u>1</u> | Historical Perspective   | 2    |
|----------|--|------|
| <u>2</u> | Objects  | 3    |
|          | 2.1 Format of Definitions :: :: :: :: :: :: :: :: :: :: :: :: :: | 4    |
| <u>3</u> | Overview   | 4    |
|          | 3.1 Domain Structure :: :: :: :: :: :: :: :: :: :: :: :: ::      | 4    |
|          | 3.2 IDPR Functions : :: :: :: :: :: :: :: :: :: :: :: ::         | 5    |
|          | 3.3 IDPR MIB : :: :: :: :: :: :: :: :: :: :: :: ::               | 7    |
| <u>4</u> | Object Definitions   | 7    |
| <u>5</u> | Trap Messages  | 43   |
| <u>6</u> | Security Considerations  | 46   |
| <u>7</u> | Acknowledgements   | 46   |
| <u>8</u> | References   | 46   |
| <u>9</u> | Editor's Address   | 47   |
| In       | iternet Draft IDPR MIB October 1                                 | 1993 |

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet Drafts.

This Internet Draft will be submitted to the RFC editor as an experimental extension to the SNMP MIB. Distribution of this Internet Draft is unlimited. Please send comments to IDPR working group (idpr-wg@bbn.com).

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a ``working draft'' or ``work in progress''.

Please check the 1id-abstracts.txt listing contained in the internet-drafts shadow directories on nic.ddn.mil, nnsc.nsf.net, nic.nordu.net, ftp.nisc.sri.com, or munnari.oz.au to learn the current status of any Internet Draft.

#### Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing the Inter-Domain Policy Routing Protocol [10, 11].

This memo does not specify a standard for the Internet community. However, after experimentation, if sufficient consensus is reached in the Internet community, then a subsequent revision of this document may be incorporated into the Internet-standard MIB.

#### **<u>1</u>** Historical Perspective

As reported in RFC 1052, ``IAB Recommendations for the Development of Internet Network Management Standards'' [1], a two-prong strategy for network management of TCP/IP-based internets was undertaken. In the short-term, the Simple Network Management Protocol (SNMP), defined in RFC 1067, was to be used to manage nodes in the Internet community. In the long-term, the use of the OSI network management framework was to be examined. Two documents were produced to define the management information: RFC 1065, which defined the Structure of Management Information (SMI), and RFC 1066, which defined the Management Information Base (MIB). Both of these documents were designed so as to be compatible with both the SNMP and the OSI network management framework.

This strategy was quite successful in the short-term: Internet-based

network management technology was fielded, by both the research and commercial communities, within a few months. As a result of this,

R. A. Woodburn, Editor

[Page 2]

IDPR MIB

portions of the Internet community became network manageable in a timely fashion.

As reported in <u>RFC 1109</u>, Report of the Second Ad Hoc Network Management Review Group [2], the requirements of the SNMP and the OSI network management frameworks were more different than anticipated. As such, the requirement for compatibility between the SMI/MIB and both frameworks was suspended. This action permitted the operational network management framework, based on the SNMP, to respond to new operational needs in the Internet community by producing MIB-II.

In May of 1990, the core documents were elevated to "Standard Protocols" with "Recommended" status. As such, the Internet-standard network management framework consists of: Structure and Identification of Management Information for TCP/IP-based internets, RFC 1155 [3], which describes how managed objects contained in the MIB are defined; Management Information Base for Network Management of TCP/IP-based internets, which describes the managed objects contained in the MIB, RFC 1156 [4]; and, the Simple Network Management Protocol, RFC 1157 [5], which defines the protocol used to manage these objects. Consistent with the IAB directive to produce simple, workable systems in the short-term, the list of managed objects defined in the Internet-standard MIB was derived by taking only those elements which are considered essential. However, the SMI defined three extensibility mechanisms: one, the addition of new standard objects through the definitions of new versions of the MIB; two, the addition of widely-available but non-standard objects through the experimental subtree; and three, the addition of private objects through the enterprises subtree. Such additional objects can not only be used for vendor-specific elements, but also for experimentation as required to further the knowledge of which other objects are essential.

This memo defines extensions to the MIB using the second method. It contains definitions of managed objects used for experimentation. After experimentation, if sufficient consensus is reached in the Internet community, then a subsequent revision of this memo may be placed in the Internet-standard MIB.

#### 2 Objects

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) [7] defined in the SMI. In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the OBJECT DESCRIPTOR, to also refer to the object type.

The syntax of an object type defines the abstract data structure corresponding to that object type. The ASN.1 language is used for this purpose. However, the SMI [3] purposely restricts the ASN.1 constructs

R. A. Woodburn, Editor

[Page 3]

Internet Draft

IDPR MIB

which may be used. These restrictions are explicitly made for simplicity.

The encoding of an object type is simply how that object type is represented using the object type's syntax. Implicitly tied to the notion of an object type's syntax and encoding is how the object type is represented when being transmitted on the network.

The SMI specifies the use of the basic encoding rules of ASN.1  $[\underline{8}]$ , subject to the additional requirements imposed by the SNMP.

# **<u>2.1</u>** Format of Definitions

<u>Section 6</u> contains the specification of all object types contained in this MIB module. The object types are defined using the conventions defined in the SMI, as amended by the extensions specified in [9].

#### <u>3</u> Overview

The Inter-Domain Policy Routing Protocol (IDPR) is a routing protocol for use between Administrative Domains. The objective of IDPR is to construct routes between source and destination administrative domains that provide user traffic with the service requested within the constraints stipulated by the domains transited.

With IDPR, we introduce a new network-layer Internet protocol based on source-specified routing between administrative domains and a new Internet addressing structure based on an administrative domain hierarchy. However, for IDPR version 1, we recommend implementing a proper subset of the complete architecture proposed in [11]. The recommended subset provides the majority of the features of IDPR and comprises the virtual gateway protocol, the domain status distribution protocol, the route synthesis procedure, the path setup protocol, and the message forwarding procedure.

# 3.1 Domain Structure

The IDPR architecture has been designed to accommodate an Internet with tens of thousands of administrative domains collectively containing hundreds of thousands of local networks. Inter-domain policy routes are constructed using information about the policy attributes of, and the connectivity between, administrative domains. The intra-domain details, gateways and networks traversed, of an inter-domain policy route are the responsibility of intra-domain routing and are thus outside the scope of inter-domain routing. An Administrative Domain (AD) is a collection of contiguous networks managed by a single administrative authority that places policy restrictions on transit traffic, defines policy requirements for locally-generated traffic, and selects the intra-domain addressing

R. A. Woodburn, Editor

[Page 4]

Internet Draft

schemes and routing procedures. Each administrative domain has a unique identity within the Internet.

Virtual Gateways (VGs) are the only IDPR-recognized connecting points between adjacent administrative domains. Each virtual gateway is a collection of directly-connected policy gateways in two adjacent domains, whose existence has been sanctioned by the authorities in both domains. The domain authorities may agree to establish more than one virtual gateway between the two domains. For each such virtual gateway, the two authorities together assign a virtual gateway identifier, unique within the set of virtual gateways connecting the two domains. To produce a virtual gateway identifier unique within its domain, the domain authority concatenates the mutually assigned identifier together with the adjacent domain's identifier.

Policy Gateways (PGs) are the physical gateways within a virtual gateway. Each policy gateway enforces policy restrictions on transit traffic as stipulated by the domain's administrative authority. A single policy gateway may belong to multiple virtual gateways. Within a domain, two policy gateways are neighbors if they are in different virtual gateways. Within a virtual gateway, two policy gateways are peers if they are in the same domain and are adjacent if they are in different domains. Adjacent policy gateways are directly connected if they are the only Internet-addressable entities attached to the connecting medium. Note that this definition implies that not only point-to-point links but also networks may serve as direct connections between adjacent policy gateways. The authority for a given domain assigns to each of its policy gateways an identifier, unique within that domain.

# 3.2 IDPR Functions

Inter-domain policy routing comprises the following functions:

- 1. Collecting and distributing connectivity and policy information pertaining to transit domains.
- 2. Synthesizing and selecting policy routes based on the connectivity and policy information associated with the transit domains and on the service requirements associated with the source.
- 3. Setting up paths across the Internet, using the policy routes synthesized.
- 4. Forwarding messages across and between administrative domains along the established paths.
- 5. Maintaining databases of transit domain connectivity and policy,

inter-domain policy routes, configured global information such as network-address/domain-identifier mappings, and configured local information such as the policy gateways within an administrative domain.

R. A. Woodburn, Editor

[Page 5]

IDPR MIB

As described in [11], a source domain controls synthesis and selection of policy routes to destination domains, while intermediate domains on a specific source-selected policy route determine whether or not the route is consistent with their transit policies. The route synthesis procedure uses domain status information in the form of advertised transit policies and adjacencies, to generate policy routes between source and destination domains. The source then selects policy routes from those provided by route synthesis, according to its own service requirements. Policy routes require a path setup procedure during which policy gateways in intermediate domains verify whether or not they will carry traffic traveling along the path, and contingent upon route acceptance, form an association between the path identifier and the previous and next policy gateways use the path identifier carried in data messages to forward data traffic along the path.

Several different entities are responsible for performing the IDPR functions. Policy gateways collect and distribute status information about their administrative domains, participate in path setup, and forward data messages along established paths. Path agents act on behalf of hosts to select policy routes and to set up and manage paths. Special-purpose servers maintain the routing databases which are distributed with partial redundancy throughout the Internet. Each special-purpose server within an administrative domain has a unique identifier, assigned by the domain authority. Route servers are responsible for both the domain status (connectivity and policy) database and the route database. Also, route servers synthesize policy routes using domain status information and source traffic requirements. Name servers are responsible for the

domain-name/network-address/domain-identifier database. Configuration servers are responsible for databases of configured information that apply to policy gateways, path agents, and route servers in the given administrative domain and inform these entities of configuration changes.

Both route servers and name servers are organized hierarchically, where a server's position in the hierarchy determines the extent of its database. At the top are global servers that maintain information about all Internet domains; at the bottom are local servers that maintain information about a particular domain, its neighbors, and other frequently visited domains, this last type of information usually obtained from higher-level servers. Hierarchical database organization releases hosts and gateways from the burden of maintaining information about large portions of the Internet, most of which they will never use.

In IDPR version 1, each policy gateway performs all IDPR functions, including those of the path agent and the special-purpose servers. Aggregating all routing functions into policy gateways simplifies implementation; one need only install IDPR protocols in policy gateways. Moreover, it simplifies communication between routing functions, as all functions reside within each policy gateway. We also note that IDPR version 1 supports only a flat route server hierarchy; each policy gateway contains a global route server. We are presently investigating approaches to making efficient use of hierarchically-organized routing information.

R. A. Woodburn, Editor

[Page 6]

Internet Draft

IDPR MIB

Given the size of the current Internet (on the order of 100 administrative domains) and the type of policies supported in IDPR version 1 (access restrictions only), we believe that policy gateways have adequate processing and memory resources to synthesize policy routes and set up paths as well as to forward messages and participate in intra-domain routing. Meanwhile, we are developing autonomous route servers and configuration servers, separate from policy gateways, so that IDPR can accommodate larger numbers of administrative domains and policies in the future.

## 3.3 IDPR MIB

These objects are used to control and manage an IDPR  $[\underline{11}]$  implementation.

This MIB is broken into 10 sections. The first section describes system variables. The remaining sections describe tables and individual IDPR sub-protocols.

### **<u>4</u>** Object Definitions

```
RFCxxxx-MIB DEFINITIONS ::= BEGIN
```

IMPORTS

experimental, IpAddress, Counter, TimeTicks FROM <u>RFC-1155</u> OBJECT-TYPE FROM <u>RFC-1212</u> DisplayString FROM <u>RFC-1213</u>;

-- This MIB module uses the extended OBJECT-TYPE macro as -- defined in [9]. The local BitField type is included -- to improve readability.

The BitField type is an OCTET STRING of four octets.
The first octet contains bit positions 0-7. The second
octet contains bit positions 8-15. The third octet
contains bit positions 16-23. The fourth octet contains
bit positions 24-31.

BitField ::= OCTET STRING (Size 4)

idpr OBJECT IDENTIFIER ::= { experimental 28 }

-- 5.1 IDPR System Object Definitions

idprsys OBJECT IDENTIFIER ::= { idpr 1 }

| idprAD | OBJECT-TYPE |
|--------|-------------|
| SYNTAX | INTEGER     |
| ACCESS | read-only   |
| STATUS | mandatory   |

R. A. Woodburn, Editor

[Page 7]

IDPR MIB

```
DESCRIPTION
          "The AD identifier of the entity being queried."
     ::= { idprsys 1 }
idprId
              OBJECT-TYPE
     SYNTAX
              INTEGER
    ACCESS read-only
    STATUS
              mandatory
     DESCRIPTION
          "The entity identifier of the entity being queried."
     ::= { idprsys 2 }
idprType
              OBJECT-TYPE
     SYNTAX
              BitField
    ACCESS read-only
    STATUS
              mandatory
     DESCRIPTION
          "The functions supported by the local entity as
          defined by the bitfields below.
               Bit position 0 PG flag
               Bit position 1 RS flag."
     ::= { idprsys 3 }
idprADRep
              OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
              mandatory
     STATUS
     DESCRIPTION
          "The entity identifier of the AD representative
           as perceived by the local entity."
     ::= { idprsys 4 }
idprUpTime
              OBJECT-TYPE
    SYNTAX
              TimeTicks
     ACCESS
              read-only
    STATUS
              mandatory
     DESCRIPTION
          "The amount of time since IDPR was last
           initialized. The value is expressed in
           TimeTicks (10 millisecond) units."
     ::= { idprsys 5 }
idprPolicyState
                  OBJECT-TYPE
              INTEGER(0..255)
    SYNTAX
     ACCESS
              read-write
     STATUS
              optional
     DESCRIPTION
          "Reading this object returns the current policy
```

state for the entity. A domain administrator may choose to associate source and transit policies with one or more policy states. A transit or source policy thus associated becomes active only when one of its policy state values matches the current policy state. Otherwise,

R. A. Woodburn, Editor

[Page 8]

```
the policy is dormant and is not distributed
                in RID messages or used in route computation.
                Writing a new value to this object sets the
                new policy state, which may cause a different
                set of policies to take effect."
          ::= { idprsys 6 }
     idprDomainName
                            OBJECT-TYPE
         SYNTAX
                    DisplayString (Size 64)
         ACCESS
                    read-only
         STATUS
                    mandatory
          DESCRIPTION
               "The name of the domain as represented by a
                printable string."
          ::= { idprsys 7 }
-- 5.2 IDPR CMTP Object Definitions
     idprcmtp
                    OBJECT IDENTIFIER ::= { idpr 2 }
     idprcmtpVer
                    OBJECT-TYPE
          SYNTAX
                    BitField
         ACCESS
                   read-only
         STATUS
                   mandatory
          DESCRIPTION
               "The CMTP versions supported by this PG.
                    Bit Position 0 == Version 1
                    Bit Position 1 == Version 2
                    ...and so on...
                    Bit Position n == Version n+1
                where n \le 31."
          ::= { idprcmtp 1 }
     idprcmtpIns
                    OBJECT-TYPE
         SYNTAX
                    Counter
         ACCESS
                    read-only
          STATUS
                    mandatory
          DESCRIPTION
               "The number of CMTP messages received by the
                local entity."
          ::= { idprcmtp 2 }
     idprcmtpOuts
                    OBJECT-TYPE
                    Counter
          SYNTAX
         ACCESS
                    read-only
          STATUS
                    mandatory
          DESCRIPTION
```

"The number of CMTP messages issued by the local entity." ::= { idprcmtp 3 }

idprcmtpRetries OBJECT-TYPE

R. A. Woodburn, Editor

[Page 9]

```
SYNTAX
              Counter
    ACCESS read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of retransmissions of CMTP messages
          by the local entity."
    ::= { idprcmtp 4 }
idprcmtpAckIns OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
              mandatory
    STATUS
    DESCRIPTION
         "The number of CMTP ACKs received by the local
          entity."
    ::= { idprcmtp 5 }
idprcmtpAckOuts
                   OBJECT-TYPE
    SYNTAX
              Counter
              read-only
    ACCESS
    STATUS
              mandatory
    DESCRIPTION
         "The number of CMTP ACKs issued by the local
          entity."
    ::= { idprcmtp 6 }
idprcmtpNakIns OBJECT-TYPE
    SYNTAX Counter
              read-only
    ACCESS
    STATUS
              mandatory
    DESCRIPTION
         "The total number of CMTP NAKs received by the
          local entity."
    ::= { idprcmtp 7 }
idprcmtpNakOuts
                   OBJECT-TYPE
    SYNTAX Counter
              read-only
    ACCESS
    STATUS
              mandatory
    DESCRIPTION
         "The number of CMTP NAKs issued by the local
          entity."
    ::= { idprcmtp 8 }
idprcmtpTimeouts
                   OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
              mandatory
    STATUS
```

DESCRIPTION
 "The number of CMTP messages that failed to
 be acknowledged in a timely manneer, causing
 delivery of the message to timeout."
 ::= { idprcmtp 9 }

R. A. Woodburn, Editor

[Page 10]

```
idprcmtpUnkAuths
                   OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of CMTP protocol messages received
          with an unrecognized authentication type."
     ::= { idprcmtp 10 }
idprcmtpUnkProtos
                   OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of CMTP protocol messages received
          with an invalid IDPR sub-protocol number."
     ::= { idprcmtp 11 }
idprcmtpBadAuths
                   OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of CMTP protocol messages that
          failed message authentication tests."
     ::= { idprcmtp 12 }
idprcmtpBadLengths OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of CMTP protocol messages received
          with an invalid length field."
     ::= { idprcmtp 13 }
idprcmtpBadTimes
                   OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
    STATUS
              mandatory
     DESCRIPTION
         "The number of CMTP protocol messages received
          with a bad timestamp."
     ::= { idprcmtp 14 }
```

-- 5.3 IDPR VGP Object Definitions

idprvgp OBJECT IDENTIFIER ::= { idpr 3 }

idprvgpPer OBJECT-TYPE SYNTAX INTEGER ACCESS read-only

R. A. Woodburn, Editor

[Page 11]

```
STATUS
               mandatory
     DESCRIPTION
          "The default update interval for the up/down
           protocol in milliseconds. This object
           corresponds to the vgp_per_sec configuration
           parameter described in [<u>12</u>], but normalized
           to millisecond units."
     ::= { idprvgp 1 }
idprvgpPGInt
               OBJECT-TYPE
     SYNTAX
               INTEGER
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The retransmission interval for acknowledged
           VGP message types in millisecond units. This
           object corresponds to the vgp_int_usec system
           configuration parameter [12] divided by 1000,
           and rounded to the nearest millisecond."
     ::= { idprvgp 2 }
idprvgpPGRet
               OBJECT-TYPE
    SYNTAX
               INTEGER
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The number of retransmissions to be sent
           before a message is classified as undeliverable.
           This object corresponds to the vgp_ret system
           configuration parameter specified in [12]."
     ::= { idprvgp 3 }
idprvgpSTSInt OBJECT-TYPE
     SYNTAX
               INTEGER
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The delay (hold time) in seconds from the time a
           VG representative receives a PG connect message
           until it issues a PG status message for that VG.
           For systems which do not support more than 2 PGs
           per VG, this object is not applicable and should
           return the value 0."
     ::= { idprvgp 4 }
idprvgpUpDownIns
                    OBJECT-TYPE
               Counter
     SYNTAX
               read-only
     ACCESS
```

STATUS mandatory
DESCRIPTION
 "The number of up/down messages received."
::= { idprvgp 5 }

idprvgpUpDownInErrs OBJECT-TYPE

R. A. Woodburn, Editor

[Page 12]

```
SYNTAX
              Counter
    ACCESS
              read-only
    STATUS
              mandatory
     DESCRIPTION
          "The number of invalid up/down messages received.
          Up/down messages improperly addressed are
          tabulated by this object, along with other
          generic errors."
     ::= { idprvgp 6 }
idprvgpUpDownOuts
                   OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of up/down messages sent."
     ::= { idprvgp 7 }
idprvgpPGConIns
                   OBJECT-TYPE
              Counter
    SYNTAX
    ACCESS
              read-only
    STATUS
              optional
    DESCRIPTION
         "The number of PG connect messages received."
     ::= { idprvgp 8 }
idprvgpPGConOuts
                   OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
    STATUS
              optional
     DESCRIPTION
         "The number of PG connect messages sent. This
          object should return the value 0 if PG connect
          messages are not supported."
     ::= { idprvgp 9 }
idprvqpPGPlcyIns
                   OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
    STATUS
              optional
     DESCRIPTION
         "The number of PG policy messages received."
     ::= { idprvgp 10 }
idprvgpPGPlcyOuts
                   OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
              optional
    STATUS
```

DESCRIPTION
 "The number of PG policy messages sent. This
 object should return the value 0 if PG policy
 messages are not supported."
 ::= { idprvgp 11 }

R. A. Woodburn, Editor

[Page 13]

```
idprvgpVGConIns
                   OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of VG connect messages received."
     ::= { idprvgp 12 }
idprvgpVGConOuts
                   OBJECT-TYPE
    SYNTAX
              Counter
              read-only
    ACCESS
    STATUS
              mandatory
    DESCRIPTION
         "The number of VG connect messages sent."
     ::= { idprvgp 13 }
idprvgpVGPlcyIns
                   OBJECT-TYPE
    SYNTAX
              Counter
              read-only
    ACCESS
              optional
    STATUS
    DESCRIPTION
         "The number of VG policy messages received."
     ::= { idprvgp 14 }
idprvgpVGPlcyOuts
                   OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
              optional
    STATUS
     DESCRIPTION
         "The number of VG policy messages sent. This
          object should return the value 0 if VG policy
          messages are not supported."
     ::= { idprvgp 15 }
idprvgpPGTab
              OBJECT-TYPE
    SYNTAX
              SEQUENCE OF IdprPGEntry
    ACCESS
              not-accessible
    STATUS
              mandatory
     DESCRIPTION
         "The table listing other policy gateways known
          by this entity. The table contains PGs in
          local and adjacent domains."
     ::= { idprvgp 16 }
idprPGEntry
              OBJECT-TYPE
    SYNTAX
              IdprPGEntry
    ACCESS
              not-accessible
    STATUS
              mandatory
```

DESCRIPTION
 "Table entry information about a an adjacent, peer,
 or neighbor PG."
INDEX { idprPGAD, idprPGId }
 ::= { idprvgpPGTab 1 }

R. A. Woodburn, Editor

[Page 14]

IdprPGEntry ::= SEQUENCE { idprPGAD INTEGER, idprPGId INTEGER, idprPGStatus INTEGER, idprPGType INTEGER, idprPGTrans Counter, idprPGUpDownPer INTEGER } idprPGAD **OBJECT-TYPE** SYNTAX INTEGER ACCESS read-only STATUS mandatory DESCRIPTION "The AD identifier for the PG." ::= { idprPGEntry 1 } idprPGId **OBJECT-TYPE** SYNTAX INTEGER ACCESS read-only STATUS mandatory DESCRIPTION "The entity identifier for the PG." ::= { idprPGEntry 2 } idprPGStatus **OBJECT-TYPE** SYNTAX INTEGER { up(1), down(2) } ACCESS read-only STATUS mandatory DESCRIPTION "Operational status of PG. Reading this object returns the operational state of a PG." ::= { idprPGEntry 3 } idprPGType **OBJECT-TYPE** SYNTAX INTEGER { adjacent(1), neighbor(2), peer(3)

}
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The type of PG entity"
::= { idprPGEntry 4 }

R. A. Woodburn, Editor

[Page 15]

```
idprPGTrans
              OBJECT-TYPE
    SYNTAX Counter
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
         "The number of times a PG has transitioned state."
     ::= { idprPGEntry 5 }
idprPGUpDownPer
                   OBJECT-TYPE
     SYNTAX INTEGER
    ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
         "This object returns the negotiated (or default)
          interval for exchanging up/down messages."
     ::= { idprPGEntry 6 }
idprvgpVGTab
              OBJECT-TYPE
     SYNTAX
              SEQUENCE OF IdprVGEntry
    ACCESS
              not-accessible
    STATUS
              mandatory
    DESCRIPTION
         "The table of known VGs"
     ::= { idprvgp 17 }
idprVGEntry
              OBJECT-TYPE
    SYNTAX
              IdprVGEntry
    ACCESS
              not-accessible
    STATUS
              mandatory
    DESCRIPTION
         "Information regarding a single VG."
     INDEX { idprVGAdj, idprVGId }
     ::= { idprvgpVGTab 1 }
IdprVGEntry ::= SEQUENCE {
     idprVGAdj
         INTEGER,
     idprVGId
         INTEGER,
     idprVGStatus
         Counter,
     idprVGVGRep
         INTEGER
}
idprVGAdj
             OBJECT-TYPE
     SYNTAX INTEGER
```

ACCESS read-only STATUS mandatory DESCRIPTION "The adjacent AD that the VG connects to." ::= { idprVGEntry 1 }

R. A. Woodburn, Editor

[Page 16]

```
idprVGId
               OBJECT-TYPE
     SYNTAX
               INTEGER
    ACCESS
              read-only
     STATUS
               mandatory
     DESCRIPTION
          "The entity identifier of the VG."
     ::= { idprVGEntry 2 }
idprVGStatus
               OBJECT-TYPE
     SYNTAX
               INTEGER {
                 up(1),
                 down(2)
               }
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The operational status of the entity."
     ::= { idprVGEntry 3 }
idprVGVGRep
               OBJECT-TYPE
     SYNTAX
               INTEGER
    ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The PG acting as the VG representative for the
          VG as seen by this PG."
     ::= { idprVGEntry 4 }
idprvgpPGMap
               OBJECT-TYPE
     SYNTAX
               SEQUENCE OF IdprPGMapEntry
               not-accessible
     ACCESS
     STATUS
               mandatory
     DESCRIPTION
          "A table that returns the association (or mapping)
           between PG and VG entities. This table is used
           to determine the VGs that PGs belongs to."
     ::= { idprvgp 18 }
idprPGMapEntry OBJECT-TYPE
     SYNTAX
               IdprPGMapEntry
     ACCESS
               not-accessible
     STATUS
              mandatory
     DESCRIPTION
          "Information for a single VG."
     INDEX { idprPGMapAD, idprPGMapId,
             idprPGMapAdj, idprPGMapVG }
     ::= { idprvgpPGMap 1 }
```

IdprPGMapEntry ::= SEQUENCE {
 idprPGMapAD
 INTEGER,
 idprPGMapId
 INTEGER,
 idprPGMapAdj

R. A. Woodburn, Editor

[Page 17]

```
INTEGER,
         idprPGMapVG
             INTEGER
    }
    idprPGMapAD OBJECT-TYPE
         SYNTAX INTEGER
         ACCESS read-only
         STATUS mandatory
         DESCRIPTION
             "The AD identifier of the PG."
         ::= { idprPGMapEntry 1 }
    idprPGMapId
                  OBJECT-TYPE
         SYNTAX INTEGER
         ACCESS read-only
         STATUS mandatory
         DESCRIPTION
             "The entity identifier of the PG."
         ::= { idprPGMapEntry 2 }
    idprPGMapAdj
                  OBJECT-TYPE
         SYNTAX INTEGER
         ACCESS read-only
         STATUS mandatory
         DESCRIPTION
             "The adjacent AD of the VG."
         ::= { idprPGMapEntry 3 }
   idprPGMapVG OBJECT-TYPE
         SYNTAX INTEGER
         ACCESS read-only
         STATUS mandatory
         DESCRIPTION
             "A VG to which this PG belongs."
         ::= { idprPGMapEntry 4 }
-- 5.4 IDPR RID Object Definitions
    idprrid OBJECT IDENTIFIER ::= { idpr 4 }
    idprridConfigPer
                      OBJECT-TYPE
         SYNTAX INTEGER
         ACCESS
                  read-only
         STATUS
                  mandatory
         DESCRIPTION
              "The interval in seconds between the periodic
```

issuance of configuration updates."
::= { idprrid 1 }

idprridDynamicPer OBJECT-TYPE SYNTAX INTEGER

R. A. Woodburn, Editor

[Page 18]

```
read-only
     ACCESS
     STATUS
              mandatory
     DESCRIPTION
          "The interval in seconds between the periodic
           issuance of dynamic message updates."
     ::= { idprrid 2 }
idprridInt
              OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "The retransmission interval of configuration and
           dynamic updates on a millisecond timescale."
     ::= { idprrid 3 }
idprridRet
              OBJECT-TYPE
     SYNTAX
              INTEGER
     ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "The maximum number of retransmission attempts
           of configuration or dynamic updates by the
           CMTP module. An update that cannot be delivered
           in idprridRet tries is declared undeliverable."
     ::= { idprrid 4 }
idprridTimNew OBJECT-TYPE
     SYNTAX
              INTEGER
     ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "The maximum wait time in seconds before
           generating a 'correcting' update as a result
           of receiving an old update."
     ::= { idprrid 5 }
idprridConfigLif
                     OBJECT-TYPE
              INTEGER
     SYNTAX
     ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "The lifetime of configuration message updates
          measured in seconds. Previously received
           configuration messages are invalidated when
           the lifetime of the configuration message expires."
     ::= { idprrid 6 }
```
idprridDynamicLif OBJECT-TYPE SYNTAX INTEGER ACCESS read-only STATUS mandatory DESCRIPTION "The lifetime of received dynamic message updates

R. A. Woodburn, Editor

[Page 19]

```
in seconds. Previously received dynamic messages
           are invalidated when the lifetime of the dynamic
           message expires."
     ::= { idprrid 7 }
idprridConfigIns
                    OBJECT-TYPE
    SYNTAX
               Counter
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The number of configuration message updates
           received by the local entity."
     ::= { idprrid 8 }
idprridConfigOuts
                    OBJECT-TYPE
     SYNTAX
               Counter
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The number of configuration message updates
           issued by the local entity. Note that an request
           to flood a config message update by the local
           entity is recorded as a single message issuance
           event. Furthermore, transit forwarding of config
           message updates which are issued by other entities
           are not included in this counter."
     ::= { idprrid 9 }
idprridDynamicIns
                    OBJECT-TYPE
     SYNTAX
               Counter
               read-only
     ACCESS
     STATUS
               mandatory
     DESCRIPTION
          "The number of dynamic updates received by the
           local Entity."
     ::= { idprrid 10 }
idprridDynamicOuts OBJECT-TYPE
     SYNTAX
               Counter
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The number of dynamic updates issued by the local
           entity. Note that an attempt to flood a Dynamic
           update by the local entity is recorded as a single
           out message event. Furthermore, transit forwarding
           of Dynamic updates which are issued by other entities
           are not recorded by this counter."
```

::= { idprrid 11 }

-- 5.5 IDPR RSQP Object Definitions

R. A. Woodburn, Editor

[Page 20]

Internet Draft

IDPR MIB

idprrsqp OBJECT IDENTIFIER ::= { idpr 5 } idprrsqpRegIns OBJECT-TYPE SYNTAX Counter ACCESS read-only STATUS mandatory DESCRIPTION "The number of route request messages received. This object contains a valid counter only if full route server functionality is supported." ::= { idprrsqp 1 } idprrsqpResps OBJECT-TYPE SYNTAX Counter ACCESS read-only STATUS mandatory DESCRIPTION "The number of responses returned by the route server. This object contains valid data only if full route server functionality is supported." ::= { idprrsqp 2 } idprrsqpNegResps **OBJECT-TYPE** SYNTAX Counter ACCESS read-only STATUS mandatory DESCRIPTION "The number of negative route responses returned by the route server. Negative responses are returned if no route could be found. This object contains valid data only if full route server functionality is supported." ::= { idprrsqp 3 } idprrsqpReqInt OBJECT-TYPE SYNTAX INTEGER ACCESS read-only STATUS mandatory DESCRIPTION "The retransmission interval for route server requests measured in seconds" ::= { idprrsqp 4 } idprrsqpReqRet OBJECT-TYPE SYNTAX INTEGER ACCESS read-only STATUS mandatory DESCRIPTION

"The number of retransmissions for route server requests before timing out the request" ::= { idprrsqp 5 }

## idprrsqpRidReqIns OBJECT-TYPE SYNTAX Counter

R. A. Woodburn, Editor

[Page 21]

```
ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "The number of requests received by a route server
          for RID configuration or dynamic information."
     ::= { idprrsqp 6 }
idprrsqpRidReqOuts OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
              mandatory
     STATUS
     DESCRIPTION
          "The number of responses issued by a route server
           for RID configuration or dynamic information."
     ::= { idprrsqp 7 }
idprrsqpRSTab
              OBJECT-TYPE
    SYNTAX
              SEQUENCE OF IdprRSEntry
    ACCESS
              not-accessible
     STATUS
              mandatory
     DESCRIPTION
          "The table of other known route servers."
     ::= { idprrsqp 8 }
idprRSEntry
              OBJECT-TYPE
    SYNTAX
              IdprRSEntry
    ACCESS
              not-accessible
    STATUS
              mandatory
     DESCRIPTION
          "Information for a single RS."
     INDEX { idprRSAD, idprRSId }
     ::= { idprrsqpRSTab 1 }
IdprRSEntry ::= SEQUENCE {
     idprRSAD
          INTEGER,
     idprRSId
          INTEGER,
     idprRSSend
          INTEGER,
     idprRSAdvert
          INTEGER,
     idprRSLastSend
          TimeTicks,
     idprRSLastRecv
         TimeTicks
}
```

idprRSAD OBJECT-TYPE SYNTAX INTEGER ACCESS read-only STATUS mandatory DESCRIPTION "The AD identifier for the RS."

R. A. Woodburn, Editor

[Page 22]

```
::= { idprRSEntry 1 }
idprRSId
               OBJECT-TYPE
     SYNTAX
               INTEGER
     ACCESS
               read-only
     STATUS
               mandatory
    DESCRIPTION
          "The entity identifier of the RS."
     ::= { idprRSEntry 2 }
idprRSSend
              OBJECT-TYPE
               INTEGER {
     SYNTAX
                 ignore(0),
                 send(1)
               }
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The value of this field indicates whether or not
           the RS should be included in the list of RSs to
           which updates are sent."
     ::= { idprRSEntry 3 }
idprRSAdvert
               OBJECT-TYPE
     SYNTAX
               INTEGER {
                 ignore(0),
                 advert(1)
               }
     ACCESS
               read-only
               mandatory
     STATUS
     DESCRIPTION
          "The value of this field indicates whether or not
           the RS should be included in the list of RSs
           which are advertized from this AD."
     ::= { idprRSEntry 4 }
idprRSLastSend OBJECT-TYPE
     SYNTAX
               TimeTicks
    ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The last time an update was sent to this RS."
     ::= { idprRSEntry 5 }
idprRSLastRecv OBJECT-TYPE
     SYNTAX
               TimeTicks
    ACCESS
               read-only
               mandatory
     STATUS
```

DESCRIPTION
 "The last time an update was received from this RS."
 ::= { idprRSEntry 6 }

R. A. Woodburn, Editor

[Page 23]

IDPR MIB

```
-- 5.6 IDPR PSP Object Definitions
    idprpsp
                   OBJECT IDENTIFIER ::= { idpr 6 }
    idprpspInt
                   OBJECT-TYPE
         SYNTAX
                   INTEGER
         ACCESS
                   read-only
         STATUS
                   mandatory
         DESCRIPTION
              "The retransmission interval for path setup
               messages, in units of seconds."
         ::= { idprpsp 1 }
    idprpspRet
                   OBJECT-TYPE
         SYNTAX
                   INTEGER
         ACCESS
                   read-only
         STATUS
                   mandatory
         DESCRIPTION
              "The number of retransmissions of a path setup
               message before the message is declared
               undeliverable."
         ::= { idprpsp 2 }
    idprpspSetupIns
                        OBJECT-TYPE
         SYNTAX
                   Counter
         ACCESS
                   read-only
         STATUS
                   mandatory
         DESCRIPTION
              "The number of path setup messages received
               by a PG."
         ::= { idprpsp 3 }
    idprpspSetupOuts
                        OBJECT-TYPE
         SYNTAX Counter
         ACCESS
                   read-only
         STATUS
                   mandatory
         DESCRIPTION
              "The number of path setup messages issued
               by a PG."
         ::= { idprpsp 4 }
    idprpspTeardownIns OBJECT-TYPE
         SYNTAX
                   Counter
         ACCESS
                   read-only
         STATUS
                   mandatory
         DESCRIPTION
              "The number of path teardown messages received
               by a PG."
```

::= { idprpsp 5 }

idprpspTeardownOuts OBJECT-TYPE SYNTAX Counter ACCESS read-only STATUS mandatory

R. A. Woodburn, Editor

[Page 24]

```
DESCRIPTION
         "The number of path teardown messages issued
          by a PG."
     ::= { idprpsp 6 }
idprpspRefuseIns OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
    STATUS mandatory
    DESCRIPTION
         "The number of path refuse messages received
          by a PG."
     ::= { idprpsp 7 }
idprpspRefuseOuts OBJECT-TYPE
    SYNTAX
              Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of path refuse messages issued
          by a PG."
     ::= { idprpsp 8 }
idprpspAcceptIns
                   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
         "The number of path accept messages received
          by a PG."
     ::= { idprpsp 9 }
idprpspAcceptOuts OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of path accept messages issued
          bv a PG."
     ::= { idprpsp 10 }
idprpspRepairIns OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The number of path repair messages received
          by a PG."
```

::= { idprpsp 11 }

idprpspRepairOuts OBJECT-TYPE SYNTAX Counter ACCESS read-only STATUS mandatory

R. A. Woodburn, Editor

[Page 25]

```
DESCRIPTION
         "The number of path repair messages issued
          by a PG."
     ::= { idprpsp 12 }
idprpspErrorIns
                 OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
     DESCRIPTION
         "The number of path error messages received
          by a PG."
     ::= { idprpsp 13 }
idprpspErrorOuts OBJECT-TYPE
    SYNTAX Counter
    ACCESS
              read-only
    STATUS mandatory
     DESCRIPTION
         "The number of path error messages issued
          by a PG."
     ::= { idprpsp 14 }
- -
-- PCP Path Table
- -
idprpspPathTab OBJECT-TYPE
    SYNTAX SEQUENCE OF IdprPathEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
         "The table IDPR paths."
     ::= { idprpsp 15 }
idprPathEntry OBJECT-TYPE
    SYNTAX IdprPathEntry
     ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
         "Information for a single IDPR path."
     INDEX { idprPathAD, idprPathEnt, idprPathId }
     ::= { idprpspPathTab 1 }
IdprPathEntry ::= SEQUENCE {
     idprPathAD
         INTEGER,
     idprPathEnt
         INTEGER,
```

idprPathId INTEGER, idprPathDst INTEGER, idprPathPrevPGAD INTEGER,

R. A. Woodburn, Editor

[Page 26]

idprPathPrevPGId INTEGER, idprPathPrevVGAD INTEGER, idprPathPrevVGId INTEGER, idprPathNextPGAD INTEGER, idprPathNextPGId INTEGER, idprPathNextVGAD INTEGER, idprPathNextVGId INTEGER, idprPathRefCnt INTEGER, idprPathState INTEGER, idprPathMsgs Counter, idprPathMsgsMax INTEGER, idprPathBytes Counter, idprPathBytesMax INTEGER, idprPathExpire INTEGER, idprPathAuth INTEGER, idprPathVer BitField, idprPathTos INTEGER, idprPathUCI INTEGER, idprPathTrType INTEGER } idprPathAD **OBJECT-TYPE** SYNTAX INTEGER ACCESS read-only STATUS mandatory DESCRIPTION "The administrative domain of the PG that

```
originated the path."
::= { idprPathEntry 1 }
```

idprPathEnt OBJECT-TYPE SYNTAX INTEGER ACCESS read-only STATUS mandatory DESCRIPTION

R. A. Woodburn, Editor

[Page 27]

```
"The entity identifier of the PG that
           originated the path."
     ::= { idprPathEntry 2 }
idprPathId
              OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS read-only
    STATUS
              mandatory
    DESCRIPTION
         "The unique identifier assigned to a path
          by the originating PG."
     ::= { idprPathEntry 3 }
idprPathDst
              OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The destination or terminating AD for the path."
     ::= { idprPathEntry 4 }
idprPathPrevPGAD OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The AD of the previous hop PG. If a path entered
           through another PG in the same domain, the value
           returned by this object will be the same as the AD
           for the PG being gueried."
     ::= { idprPathEntry 5 }
idprPathPrevPGId
                    OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The entity identifier of the previous hop PG."
     ::= { idprPathEntry 6 }
idprPathPrevVGAD
                   OBJECT-TYPE
    SYNTAX
              INTEGER
              read-only
    ACCESS
              mandatory
    STATUS
     DESCRIPTION
         "The AD of the previous hop VG. This object
           refers to the AD of entry VG, which corresponds
           to the AD immediately before the current AD
```

along a path." ::= { idprPathEntry 7 } idprPathPrevVGId OBJECT-TYPE SYNTAX INTEGER ACCESS read-only

R. A. Woodburn, Editor

[Page 28]

IDPR MIB

```
mandatory
     STATUS
     DESCRIPTION
          "The identifier of the previous hop VG."
     ::= { idprPathEntry 8 }
idprPathNextPGAD OBJECT-TYPE
    SYNTAX
               INTEGER
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The AD of the next hop PG. If a path is exiting
           through another PG in the same domain, the value
           returned by this object will be the same as the AD
           for the PG being gueried."
     ::= { idprPathEntry 9 }
idprPathNextPGId
                     OBJECT-TYPE
    SYNTAX
               INTEGER
     ACCESS
               read-only
               mandatory
     STATUS
     DESCRIPTION
          "The entity identifier of the next hop PG."
     ::= { idprPathEntry 10 }
idprPathNextVGAD
                    OBJECT-TYPE
    SYNTAX
               INTEGER
               read-only
     ACCESS
               mandatory
     STATUS
     DESCRIPTION
          "The AD of the next hop VG. This object refers
           to the AD of the exit VG, which corresponds to
           the AD after the current AD along a path."
     ::= { idprPathEntry 11 }
idprPathNextVGId
                    OBJECT-TYPE
    SYNTAX
               INTEGER
               read-only
     ACCESS
     STATUS
               mandatory
     DESCRIPTION
          "The identifier of the next hop VG."
     ::= { idprPathEntry 12 }
idprPathRefCnt OBJECT-TYPE
     SYNTAX
               INTEGER
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The number of address map entries referencing
```

a path." ::= { idprPathEntry 13 } idprPathState OBJECT-TYPE SYNTAX INTEGER { idle (1), -- the path is no longer being used

R. A. Woodburn, Editor

[Page 29]

```
-- a setup request has been received
          start (2),
                          -- a setup has been sent
          setup (3),
          acceptwait (4), -- an ACK has been received
          active (5),
                         -- the accept has been received
          dying (6)
                         -- teardown has been received/sent
     }
    ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "The current state of the path as enumerated above."
     ::= { idprPathEntry 14 }
idprPathMsgs
              OBJECT-TYPE
     SYNTAX
              Counter
     ACCESS
              read-only
              mandatory
     STATUS
     DESCRIPTION
          "The number of messages (i.e., packets) that has
           been forwarded on this path."
     ::= { idprPathEntry 15 }
idprPathMsgsMax
                   OBJECT-TYPE
     SYNTAX
              INTEGER
     ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "The maximum number of data messages
           (i.e., packets) that can be forwarded before
           a path expires."
     ::= { idprPathEntry 16 }
idprPathBytes OBJECT-TYPE
     SYNTAX
              Counter
     ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "The number of bytes of data forwarded through
          a path thus far."
     ::= { idprPathEntry 17 }
idprPathBytesMax
                    OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
     DESCRIPTION
          "The total allowable number of bytes of data
           that can be forwarded on a path before the
           path expires."
```

::= { idprPathEntry 18 }

idprPathExpire OBJECT-TYPE SYNTAX INTEGER ACCESS read-only STATUS mandatory

R. A. Woodburn, Editor

[Page 30]

```
DESCRIPTION
          "The amount of time remaining (in seconds) before
          a path expires."
     ::= { idprPathEntry 19 }
idprPathAuth
              OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The authentication protocol used for a path."
     ::= { idprPathEntry 20 }
idprPathVer
              OBJECT-TYPE
    SYNTAX
              BitField
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
          "The version of the IDPR data encapsulation protocol
           supported by a path.
              Bit Position 0 == Version 1
              Bit Position 1 == Version 2
               ...and so on...
              Bit Position n == Version n+1
          where n <= 31."
     ::= { idprPathEntry 21 }
-- The idprPathTos and idprPathUCI objects were originally
-- specified with a syntax of OCTET STRING (Size 8). The
-- syntax has been changed to INTEGER (0..255), because we
-- assume that a path is uniquely associated with only one
-- TOS or UCI value. However, the value 0 is special and
-- denotes that a path supports traffic regardless of TOS
-- or UCI specifications.
idprPathTos
              OBJECT-TYPE
              INTEGER (0..255)
     SYNTAX
    ACCESS read-only
     STATUS
              mandatory
     DESCRIPTION
          "The type of service (TOS) associated with a path.
          Only packets with a matching the TOS are
          forwarded through the path, with one exception.
          A TOS value of 0 indicate that a path supports
          all types of service, regardless of specification.
           If a path is used to transport IP packets, this
           object refers to the 8-bit binary-encoded TOS
```

field near the beginning of the IP header."
::= { idprPathEntry 22 }

idprPathUCI OBJECT-TYPE SYNTAX INTEGER (0..255) ACCESS read-only

R. A. Woodburn, Editor

[Page 31]

STATUS mandatory DESCRIPTION "The user class identifier (UCI) associated with the path. Only packets matching the UCI returned by this object are transported over the path. A value of 0 indicates that a path supports all user classes, regardless of specification. A non-zero value indicates that only packets matching the UCI are capable of using the path." ::= { idprPathEntry 23 } idprPathTrType OBJECT-TYPE SYNTAX INTEGER (0..255) ACCESS read-only STATUS optional DESCRIPTION "The traffic type associated with the path. Only packets with a matching traffic type are transported over the path, with one exception. A value of 0 indicates that a path supports all traffic types, regardless of specification. A non-zero value indicates that only packets matching the traffic type are capable of using the path. If a path is used to transport IP packets, this object refers to the traffic type encoded as specified in IP option 144 (decimal)." ::= { idprPathEntry 24 } -- PCP Path Route Table. This table returns the routes -- taken by active IDPR paths. The route for each path -- is specified as the set of VGs through which the path -- traverses. The table index field is used to select -- information about each hop of path route. - --- In the present specification, routes are provided -- only for paths that originate from the local domain. -- Transit or terminating paths are not included in the -- path route table. - idprpspPathRouteTab OBJECT-TYPE SEQUENCE OF IdprPathRouteEntry SYNTAX ACCESS not-accessible STATUS mandatory DESCRIPTION "IDPR path route table. This table returns

the routes taken by active IDPR paths. Each
route is returned as the next hop AD and entry
VG number, which can be thought of equivalently
as the set of VGs through which the path
traverses."
::= { idprpsp 16 }

R. A. Woodburn, Editor

[Page 32]

```
idprPathRouteEntry OBJECT-TYPE
     SYNTAX
               IdprPathRouteEntry
     ACCESS
               not-accessible
     STATUS
               mandatory
     DESCRIPTION
          "IDPR path route table entry. Each table entry
           corresponds a hop along the routing path. The
           index for this table corresponds to the path
           identifier plus a table index that denotes a
           path hop count from the source."
     INDEX { idprPathRouteAD, idprPathRouteEnt,
             idprPathRouteId, idprPathRouteIndex }
     ::= { idprpspPathRouteTab 1 }
IdprPathRouteEntry ::= SEQUENCE {
     idprPathRouteAD
          INTEGER,
     idprPathRouteEnt
          INTEGER,
     idprPathRouteId
          INTEGER,
     idprPathRouteIndex
          INTEGER,
     idprPathRouteVGAD
          INTEGER,
     idprPathRouteVGId
          INTEGER,
     idprPathRouteADCmp
          INTEGER,
     idprPathRouteTrnPlcy
          INTEGER
}
-- Note that the first four objects in the table
-- are used as a table index. The index consists
-- of the path identifier along with an index which
-- serves as a hop count field.
- -
                    OBJECT-TYPE
idprPathRouteAD
     SYNTAX
               INTEGER
     ACCESS
               read-only
               mandatory
     STATUS
     DESCRIPTION
          "This object returns the AD identifier part of
           the path identifier. The object gives the AD
           that initiated path setup."
```

::= { idprPathRouteEntry 1 }

idprPathRouteEnt OBJECT-TYPE SYNTAX INTEGER ACCESS read-only STATUS mandatory

R. A. Woodburn, Editor

[Page 33]

```
DESCRIPTION
          "This object returns the entity identifier part
          of the path identifier."
     ::= { idprPathRouteEntry 2 }
idprPathRouteId
                    OBJECT-TYPE
    SYNTAX
            INTEGER
    ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "This object returns the local identifier part
          of the path identifier."
     ::= { idprPathRouteEntry 3 }
idprPathRouteIndex
                      OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "This object returns the path route table hop
           index. The first hop of the routing path is
           assigned the index value 1. The 'n'th hop is
           given the value n."
     ::= { idprPathRouteEntry 4 }
idprPathRouteVGAD
                    OBJECT-TYPE
     SYNTAX
              INTEGER
     ACCESS
              read-only
     STATUS
              mandatory
     DESCRIPTION
          "This object returns the administrative domain
           for the hop being referenced."
     ::= { idprPathRouteEntry 5 }
idprPathRouteVGId
                    OBJECT-TYPE
     SYNTAX
              INTEGER
              read-only
     ACCESS
     STATUS
              mandatory
     DESCRIPTION
          "This object returns the entry VG sub-identifier
           for the hop. The pair that is given by
           idprPathRouteAD and idprPathRouteVGId identify
           the VG through which a path traverses."
     ::= { idprPathRouteEntry 6 }
idprPathRouteADCmp
                     OBJECT-TYPE
     SYNTAX
              INTEGER
    ACCESS
              read-only
```

STATUS mandatory DESCRIPTION "This object returns the domain component that advertised the RID information which led to the selection of the VG as part of the path. The value returned corresponds to the ``CMP''

R. A. Woodburn, Editor

[Page 34]

```
field provided with path setup messages."
          ::= { idprPathRouteEntry 7 }
     idprPathRouteTrnPlcy
                             OBJECT-TYPE
          SYNTAX
                   INTEGER
          ACCESS
                   read-only
          STATUS
                   mandatory
          DESCRIPTION
               "This object returns the transit policy used to
                satisfy the path request. The value returned
                corresponds to the ``TP'' field provided with
                path setup messages."
          ::= { idprPathRouteEntry 8 }
-- 5.7 IDPR Data Object Definitions
     idprdata
                   OBJECT IDENTIFIER ::= { idpr 7 }
     idprdataVer
                   OBJECT-TYPE
                   BitField
          SYNTAX
          ACCESS
                   read-only
          STATUS
                   mandatory
          DESCRIPTION
               "The versions of the data encapsulation protocol
                supported by the PG being queried. One or more
                data protocol versions may be supported concurrently.
                    Bit Position 0 == Version 1
                   Bit Position 1 == Version 2
                    ...and so on...
                    Bit Position n == Version n+1
              where n \le 31."
          ::= { idprdata 1 }
     idprdataErrs
                   OBJECT-TYPE
         SYNTAX
                   Counter
                   read-only
          ACCESS
          STATUS
                   mandatory
          DESCRIPTION
               "The number of messages received which could
                not be processed."
          ::= { idprdata 2 }
                        OBJECT-TYPE
     idprdataUnkPaths
          SYNTAX
                   Counter
         ACCESS
                   read-only
          STATUS
                   mandatory
          DESCRIPTION
```

"The number of data messages received by this
entity referencing an unknown path."
::= { idprdata 3 }

idprdataUnkProtos OBJECT-TYPE

R. A. Woodburn, Editor

[Page 35]

```
SYNTAX
                   Counter
          ACCESS
                   read-only
          STATUS
                   mandatory
          DESCRIPTION
               "The number of data packets received that had an
                invalid protocol field. Each IDPR data message
                has an eight bit field which specifies the network
                layer protocol carried in the data message; e.g.,
                IP is one of several possible network layer
                protocol types."
          ::= { idprdata 4 }
     idprdataBadLengths OBJECT-TYPE
          SYNTAX
                   Counter
                   read-only
          ACCESS
          STATUS
                   mandatory
          DESCRIPTION
               "The number of data packets received with an
                invalid length field."
          ::= { idprdata 5 }
     idprdataBadAuths
                        OBJECT-TYPE
          SYNTAX
                   Counter
          ACCESS
                   read-only
          STATUS
                   mandatory
          DESCRIPTION
               "The number of data packets received that failed
                authentication checks. This object contains
                a valid counter only if authentication is used
                by the data protocol."
          ::= { idprdata 6 }
-- 5.8 IDPR Address Mapping Table Object Definitions
                   OBJECT IDENTIFIER ::= { idpr 8 }
     idpraddr
     idpraddrTab
                   OBJECT-TYPE
                   SEQUENCE OF IdprAddrEntry
          SYNTAX
          ACCESS
                   not-accessible
          STATUS
                   mandatory
          DESCRIPTION
               "The address mapping table that translates
                source and destination addresses into IDPR
                paths."
          ::= { idpraddr 1 }
     idpraddrEntry OBJECT-TYPE
```

SYNTAX IdprAddrEntry ACCESS not-accessible STATUS mandatory DESCRIPTION "Information for a single IDPR Address Map entry."

R. A. Woodburn, Editor

[Page 36]

## IDPR MIB

```
INDEX { idpraddrSrc, idpraddrDst, idpraddrTos,
             idpraddrProto, idpraddrSrcPort,
             idpraddrDstPort, idpraddrTrType }
     ::= { idpraddrTab 1 }
IdprAddrEntry ::= SEQUENCE {
     idpraddrSrc
          IpAddress,
     idpraddrDst
          IpAddress,
     idpraddrSrcMask
          IpAddress,
     idpraddrDstMask
          IpAddress,
     idpraddrTos
          INTEGER,
     idpraddrProto
          INTEGER,
     idpraddrSrcPort
          INTEGER,
     idpraddrDstPort
          INTEGER,
     idpraddrUCI
          INTEGER,
     idpraddrTrType
          INTEGER,
     idpraddrPathAD
          INTEGER,
     idpraddrPathEnt
          INTEGER,
     idpraddrPathId
          INTEGER,
     idpraddrState
          INTEGER,
     idpraddrPriority
          INTEGER
}
idpraddrSrc
               OBJECT-TYPE
     SYNTAX
               IpAddress
     ACCESS
               read-only
     STATUS
               mandatory
     DESCRIPTION
          "The source address for the address map entry."
     ::= { idpraddrEntry 1 }
idpraddrDst
               OBJECT-TYPE
     SYNTAX
               IpAddress
```
ACCESS read-only STATUS mandatory DESCRIPTION "The destination address for the address map entry" ::= { idpraddrEntry 2 }

R. A. Woodburn, Editor

[Page 37]

```
idpraddrSrcMask
                   OBJECT-TYPE
    SYNTAX IpAddress
    ACCESS
             read-only
    STATUS
              mandatory
    DESCRIPTION
         "The source address mask for the address map entry."
    ::= { idpraddrEntry 3}
idpraddrDstMask
                   OBJECT-TYPE
    SYNTAX IpAddress
              read-only
    ACCESS
    STATUS
              mandatory
    DESCRIPTION
         "The destination address mask for the address
          map entry."
    ::= { idpraddrEntry 4 }
idpraddrTos OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS
              mandatory
    DESCRIPTION
         "The TOS for the address map entry."
    ::= { idpraddrEntry 5}
idpraddrProto OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
    STATUS mandatory
    DESCRIPTION
         "The IP protocol for the address map entry."
    ::= { idpraddrEntry 6 }
                   OBJECT-TYPE
idpraddrSrcPort
    SYNTAX INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The higher layer (transport) protocol source port
          for the address map entry."
    ::= { idpraddrEntry 7 }
idpraddrDstPort
                   OBJECT-TYPE
              INTEGER
    SYNTAX
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The higher layer (transport) protocol destination
```

port for the address map entry."
::= { idpraddrEntry 8 }

idpraddrUCI OBJECT-TYPE SYNTAX INTEGER (0..255) ACCESS read-only

R. A. Woodburn, Editor

[Page 38]

```
mandatory
    STATUS
     DESCRIPTION
         "The user class identifier associated with the
          address map entry."
     ::= { idpraddrEntry 9 }
idpraddrTrType OBJECT-TYPE
    SYNTAX
              INTEGER (0..255)
    ACCESS
              read-only
    STATUS
              optional
    DESCRIPTION
         "The traffic type IP option associated with the
          address map. The traffic type specification is
          an abstract integer tag which is used to associate
          user data packets of a given type with a path."
     ::= { idpraddrEntry 10 }
idpraddrPathAD OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The path AD identifier for the address map entry."
     ::= { idpraddrEntry 11 }
idpraddrPathEnt
                   OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
    DESCRIPTION
         "The path entity identifier for the address
           map entry."
     ::= { idpraddrEntry 12 }
idpraddrPathId OBJECT-TYPE
    SYNTAX
              INTEGER
    ACCESS
              read-only
    STATUS
              mandatory
     DESCRIPTION
         "The local path identifier for the address map
          entry."
     ::= { idpraddrEntry 13 }
idpraddrState OBJECT-TYPE
    SYNTAX
              INTEGER {
         idle (1), -- no request pending, no path assigned
         waiting (2), -- request pending, no path assigned
         active (3) -- path assigned
```

}
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The state of the address map entry. An address
 map entry is in effect only when it is in the

R. A. Woodburn, Editor

[Page 39]

active(3) state." ::= { idpraddrEntry 14 } idpraddrPriority OBJECT-TYPE SYNTAX INTEGER (0..255) ACCESS read-only STATUS optional DESCRIPTION "The packet priority level associated with the address map entry." ::= { idprPathEntry 15 } -- 5.9 Source Policy Table idprsrcplcy OBJECT IDENTIFIER ::= { idpr 9 } idprsrcplcyTab OBJECT-TYPE SYNTAX SEQUENCE OF IdprSrcPlcyEntry ACCESS not-accessible STATUS mandatory DESCRIPTION "Table of source policies defined for this AD." ::= { idprsrcplcy 1 } idprSrcPlcyEntry OBJECT-TYPE SYNTAX IdprSrcPlcyEntry ACCESS not-accessible STATUS mandatory DESCRIPTION "Table entry for source policies for this AD." INDEX { idprSrcPlcyId } ::= { idprsrcplcyTab 1 } IdprSrcPlcyEntry ::= SEQUENCE { idprSrcPlcyId INTEGER, idprSrcPlcyStatus INTEGER, idprSrcPlcyInfoSyntax INTEGER, idprSrcPlcyInfo OCTET STRING } idprSrcPlcyId OBJECT-TYPE SYNTAX INTEGER

ACCESS read-write STATUS mandatory DESCRIPTION "The identifier of the source policy." ::= { idprSrcPlcyEntry 1 }

R. A. Woodburn, Editor

[Page 40]

```
OBJECT-TYPE
idprSrcPlcyStatus
     SYNTAX
               INTEGER {
                 active(1),
                 inactive(2)
               }
               read-write
     ACCESS
     STATUS
               optional
     DESCRIPTION
           "Status of the source policy. A source policy
            may be defined but may not be active. An
            inactive source policy is ignored during path
            computation.
            For implementations that support table write
            operations, writing the value inactive(2)
            for an active source policy causes the policy
            to become inactive. Changing the state of the
            source policy to active(1) re-enables the
            source policy."
     ::= { idprSrcPlcyEntry 2 }
idprSrcPlcyInfoSyntax
                         OBJECT-TYPE
     SYNTAX
               INTEGER {
                 displaystring(1),
                 opaque(2),
                 other(3)
               }
     ACCESS
               read-write
     STATUS
               mandatory
     DESCRIPTION
          "This object gives the syntax of the
           idprSrcPlcyInfo object when it is read. If
           the syntax is displaystring(1), the source
           policy is returned as a DisplayString following
           the syntax recommended by the IDPR configuration
           guide. The value opaque(2) indicates that an
           ASN.1 syntax is wrapped inside the Opaque data
           type. The value other(3) is used for all
           other syntaxes."
     ::= { idprSrcPlcyEntry 3 }
idprSrcPlcyInfo
                    OBJECT-TYPE
               OCTET STRING
    SYNTAX
     ACCESS
               read-write
     STATUS
               mandatory
     DESCRIPTION
          "Source policy specification. A new source
```

policy is supported by writing the new policy specification to this object." ::= { idprSrcPlcyEntry 4 }

R. A. Woodburn, Editor

[Page 41]

IDPR MIB

```
-- 5.10 Transit Policy Table
    idprtrnplcy OBJECT IDENTIFIER ::= { idpr 10 }
    idprtrnplcyTab OBJECT-TYPE
          SYNTAX
                   SEQUENCE OF IdprTrnPlcyEntry
         ACCESS
                   not-accessible
         STATUS
                   mandatory
          DESCRIPTION
               "Table of transit policies known to this AD. This
               table contains both transit policies defined
               locally, as well as those received from other
               domains via configuration messages."
          ::= { idprtrnplcy 1 }
    idprTrnPlcyEntry
                        OBJECT-TYPE
         SYNTAX IdprTrnPlcyEntry
         ACCESS
                   not-accessible
         STATUS
                   mandatory
         DESCRIPTION
              "Table entry for transit policies for this AD."
          INDEX { idprTrnPlcyAD, idprTrnPlcyId }
          ::= { idprtrnplcyTab 1 }
    IdprTrnPlcyEntry ::= SEQUENCE {
          idprTrnPlcyAD
              INTEGER,
          idprTrnPlcyEnt
              INTEGER,
          idprTrnPlcyId
              INTEGER,
          idprTrnPlcyStatus
              INTEGER,
          idprTrnPlcyInfoSyntax
              INTEGER,
          idprTrnPlcyInfo
              OCTET STRING
    }
    idprTrnPlcyAD OBJECT-TYPE
         SYNTAX
                   INTEGER
         ACCESS
                   read-write
                   mandatory
         STATUS
         DESCRIPTION
              "The AD that advertised the transit policy."
          ::= { idprTrnPlcyEntry 1 }
```

idprTrnPlcyEnt OBJECT-TYPE

SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
 "The entity that advertised the transit policy."
::= { idprTrnPlcyEntry 2 }

R. A. Woodburn, Editor

[Page 42]

```
idprTrnPlcyId OBJECT-TYPE
     SYNTAX
               INTEGER
     ACCESS
               read-write
     STATUS
              mandatory
     DESCRIPTION
          "The identifier of the transit policy."
     ::= { idprTrnPlcyEntry 3 }
idprTrnPlcyStatus
                      OBJECT-TYPE
     SYNTAX
               INTEGER {
                 active(1),
                 inactive(2)
               }
    ACCESS
               read-write
     STATUS
               optional
     DESCRIPTION
          "Status of the transit policy. Transit policies
           that are advertised and used for route computation
           are marked as active. Transit policies received
           from remote domains are always marked as being in
           the active state.
           Note that a local domain's transit policies may
           be in the active or inactive state. Inactive
           transit policies are not including in RID
           configuration updates and are not used by route
           computation.
           For implementations that support table write
           operations, writing the value inactive(2) to a
           defined transit policy puts the policy into a
           state where it is no longer advertised.
           Depending on implementation, changing the status
           to inactive(2) may not necessarily remove the
           entry from the transit policy table. Changing
           the state to active(1) may re-enable the transit
           policy."
     ::= { idprTrnPlcyEntry 4 }
                          OBJECT-TYPE
idprTrnPlcyInfoSyntax
    SYNTAX
               INTEGER {
                 displaystring(1),
                 opaque(2),
                 other(3)
               }
     ACCESS
               read-write
```

STATUS mandatory
DESCRIPTION
 "This object specifies the syntax of the
 idprTrnPlcyInfo object. If the syntax is
 displaystring(1), the transit policy
 specification is returned as a DisplayString

R. A. Woodburn, Editor

[Page 43]

```
following the syntax recommended by the IDPR
           configuration guide. The value opaque(2)
           indicates that an ASN.1 syntax is wrapped
           inside the Opaque data type. The value other(3)
           is used for all other syntaxes."
     ::= { idprTrnPlcyEntry 5 }
idprTrnPlcyInfo
                   OBJECT-TYPE
    SYNTAX
              OCTET STRING
    ACCESS
              read-write
    STATUS
              mandatory
    DESCRIPTION
          "Transit policy specification. A new transit
           policy is supported by writing the new policy
           specification to this object. Reading from
           this object will return the current transit
           policy specification."
     ::= { idprTrnPlcyEntry 6 }
```

END

#### **<u>5</u>** Trap Messages

This section describes the trap messages that are issued by the network management system.

```
RFCxxxx-TRAP DEFINITIONS ::= BEGIN
IMPORTS
     experimental, IpAddress, Counter, TimeTicks
          FROM <u>RFC-1</u>155
     OBJECT-TYPE
          FROM RFC-1212
     TRAP-TYPE
          FROM RFC-1215;
idprADRepChange
                     TRAP-TYPE
     ENTERPRISE
                    idpr
     VARIABLES {
          idprAD,
                               -- AD and id of entity
                                -- issuing trap message.
          idprId,
          idprADRep
                                -- New AD representative.
     }
     DESCRIPTION
          "This trap is issued when the AD representative
           for a domain has changed. It is issued in
           response to the local domain becoming partitioned."
```

```
::= 1
```

# idprPGStateChange TRAP-TYPE ENTERPRISE idpr VARIABLES {

R. A. Woodburn, Editor

[Page 44]

idprPGAD, -- AD and id of remote entity idprPGId, -- affected by state change. idprPGStatus, -- New PG reachability state. -- PG type. idprPGType idprPGTrans, -- Number of transitions. } DESCRIPTION "This trap is issued when the VGP Pair-PG protocol has declared a peer entity as having transitioned between reachable and unreachable states. The peer entities may be in the same domain (i.e., IDPR neighbor or peer entities) or in an adjoining domain (i.e., adjacent entities). The variable binding list identifies the remote entity for which the state has changed." ::= 2 idprVGRepChange TRAP-TYPE ENTERPRISE idpr VARIABLES { idprVGAdj, -- VG identifier for which idprVGId, -- there is a new AD rep. idprVGVGRep -- New VG representative. } DESCRIPTION "This trap is issued when the VG representative, has changed, and is issued only if the IDPR implementation supports multiple peer PGs in a VG." ::= 3 idprpspError TRAP-TYPE ENTERPRISE idpr VARIABLES { -- Path identifier idprPathAD, idprPathEnt, -- which experienced idprPathId -- the path error. } DESCRIPTION "This trap message is issued when a path encounters an abnormal, un-recoverable error resulting in tear down of the path. The typical cause of such errors include one or more PGs along a path becoming unreachable. Tear down of paths due to path lifetime or usage expirations are considered to be normal events and do not result in trap messages being issued."

```
::= 4
```

idprtrnPlcyExpired TRAP-TYPE ENTERPRISE idpr VARIABLES { idprTrnPlcyAD, -- Domain and entity that

R. A. Woodburn, Editor

[Page 45]

```
idprTrnPlcyEnt, -- issued the transit policy.
idprTrnPlcyId -- Transit policy number.
}
DESCRIPTION
    "A RID configuration message has expired.
Failure to receive periodic RID configuration
messages imply that a domain may no longer
be reachable or that the transit policies
associated with it are no longer valid."
::= 5
```

END

R. A. Woodburn, Editor

[Page 46]

Internet Draft

### **<u>6</u>** Security Considerations

Security issues are not discussed in this memo.

## 7 Acknowledgements

We would like to acknowledge the assistance of all the members of the IDPR Working Group and particularly the following individuals:

Ken Carlberg, SAIC Dan Lee, SRI Mark Sleeper, Sparta Martha Steenstrup, BBN

#### 8 References

[1] V. Cerf, IAB Recommendations for the Development of Internet Network Management Standards, Internet Working Group Request for Comments 1052. Network Information Center, SRI International, Menlo Park, California, April 1988.

[2] V. Cerf, Report of the Second Ad Hoc Network Management Review Group, Internet Working Group Request for Comments 1109. Network Information Center, SRI International, Menlo Park, California, August 1989.

[3] M.T. Rose and K. McCloghrie, Structure and Identification of Management Information for TCP/IP-based internets, Internet Working Group Request for Comments 1155. Network Information Center, SRI International, Menlo Park, California, May 1990.

[4] K. McCloghrie and M.T. Rose, Management Information Base for Network Management of TCP/IP-based internets, Internet Working Group Request for Comments 1156. Network Information Center, SRI International, Menlo Park, California, May 1990.

[5] J.D. Case, M.S. Fedor, M.L. Schoffstall, and J.R. Davin, Simple Network Management Protocol, Internet Working Group Request for Comments <u>1157</u>. Network Information Center, SRI International, Menlo Park, California, May 1990.

[6] M.T. Rose (editor), Management Information Base for Network Management of TCP/IP-based internets, Internet Working Group Request for Comments 1158, Network Information Center, SRI International, Menlo Park, California, May 1990.

[7] Information processing systems - Open Systems Interconnection -

Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, December 1987.

R. A. Woodburn, Editor

[Page 47]

Internet Draft

# IDPR MIB

[8] Information processing systems - Open Systems Interconnection -Specification of Basic Encoding Rules for Abstract Notation One (ASN.1), International Organization for Standardization. International Standard 8825, December 1987.

[9] M.T. Rose, K. McCloghrie (editors), Towards Concise MIB Definitions, Internet Working Group Request for Comments 1212, Network Information Center, SRI International, Menlo Park, California, September 1990.

[10] M. Lepp and M. Steenstrup, An Architecture for Inter-Domain Policy Routing, Internet Draft, July 1990.

[11] ORWG, Inter-Domain Policy Routing Protocol Specification and Usage, Internet Draft, July 1991.

[12] H. Bowns and M. Steenstrup, Inter-Domain Policy Routing Configuration and Usage, Internet Draft, July 1991.

# 9 Editor's Address

IDPR Working Group c/o Robert A. Woodburn Sparta, Inc. 7926 Jones Branch Drive Suite 900 McLean, VA 22102-3303 (703) 448-0210

EMAIL: idpr-wg@bbn.com or woody@sparta.com

R. A. Woodburn, Editor

[Page 48]