Network Working Group
INTERNET-DRAFT
Category: Standards Track

Barr Hibbs
(no affiliation)
Glenn Waters
Nortel Networks
February 2004

Dynamic Host Configuration Protocol for IPv4 (DHCPv4) Server MIB

<draft-ietf-dhc-server-mib-10.txt>
Saved Friday, February 06, 2004, 4:33:38 PM

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Copyright Notice

Copyright (C), 2003, The Internet Society. All Rights Reserved.

Abstract

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in the Internet Community. In particular, it defines objects used for the management of Dynamic Host Configuration Protocol for IPv4 (DHCPv4) and Bootstrap Protocol (BOOTP) servers.

Table of Contents

1. Introduction

This memo is a product of the DHCP Working Group and defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes a set of extensions that DHCPv4 and Bootstrap Protocol (BOOTP) servers implement. Many implementations support both DHCPv4 and BOOTP within a single server and hence this memo describes the MIB for both DHCPv4 and BOOTP servers.

This memo does not cover DHCPv4/BOOTP client nor relay agent MIB extensions: these are possibly the subjects of future investigation [see discussion in section 3.1.] Also excluded from this MIB extension in the interest of simplicity are DHCP Dynamic DNS Updating, Failover, Authentication, and Load Balancing: these functions and features could be subjects of future MIB extensions. Provision is also made for Standards-Track additions to the DHCP Message Type (option 61.)

This memo is based on the Internet-standard Network Management Framework as defined by documents [RFC2578, RFC2579, RFC2580].

Objects defined in this MIB allow access to and control of DHCP Server Software. Servers MAY also provide additional management capabilities using the Applications MIB [RFC2287].

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

2. 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, [RFC2578], STD 58, [RFC2579] and STD 58, [RFC2580].

3. Overview

In the tradition of the Simple Network Management Protocol (SNMP), the minimum number of objects possible is defined in this MIB, while still providing as rich a set of management information as possible. An object is left out of this MIB when it can be derived from other objects that are provided. Further to the tradition of the SNMP, computationally intense operations are left to the domain of the management station. Thus, this MIB provides a set of objects from which other management information can be derived.

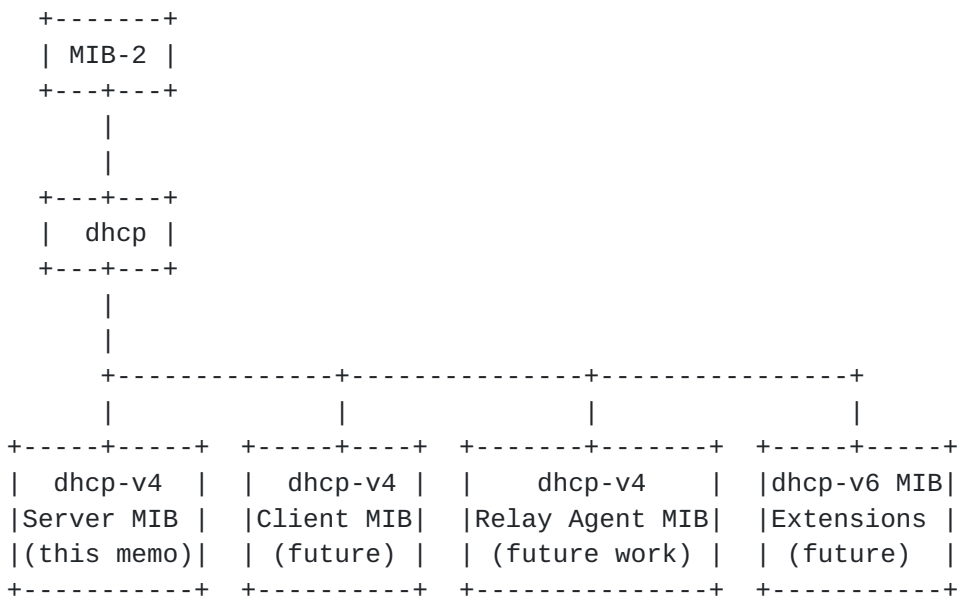
Provision for future extensions to cover DHCPv4 clients and relay agents, and DHCPv6 extensions are implied by the top-level structure illustrated in section 3.1.1.

The examples provided in sections 3.3 through 3.5 are not meant to be comprehensive but are illustrative of the potential uses of the objects defined by this MIB.

3.1. Relationship to Other MIBs

3.1.1. DHCP MIB Extensions

The DHCP MIB extensions will be the "dhcp" branch of the standard MIB-2 tree, as illustrated by the following diagram:



The DHCP MIBs share a common branching point but will be independently defined by individual memos.

3.1.2. Host System MIB Extensions

The Host System MIB [RFC1123] provides for information, command, and control of the host computer system on which a DHCP server resides. The DHCP Server MIB specifically does not include any objects that may be accessible using the Host System MIB.

3.1.3. DHCP Client MIB Extensions

Development of this set of MIB extensions is a natural path given the increasing interest in desktop and client system management. It will share a common branch point in the MIB tree with the other DHCP MIB Extensions, and may use many of the same textual conventions.

3.1.4. DHCP Relay Agent MIB Extensions

If this set of MIB extensions is ever developed, it will share a common branch point in the MIB tree with the other DHCP MIB Extensions, and will use many of the same textual conventions.

3.1.5. DHCPv6 MIB Extensions

If this set of MIB extensions is ever developed, it will share a common branch point in the MIB tree with the other DHCP MIB Extensions, and will likely use very different textual conventions as the protocol differs significantly from DHCPv4.

3.2. Textual Conventions Introduced in this MIB

One conceptual data type has been introduced in this document. No changes to the SMI or SNMP are necessary to support this convention.

3.2.1. Dhcpv4PhysicalAddress

This data type contains the type of hardware address represented by MacAddress, as defined for ARP messages, the length in octets of MacAddress, and the actual layer 1 hardware address.

3.3. BOOTP and DHCP Counter Groups

This section describes some of the management information that can be derived from the objects provided in the counter groups.

In this context, a "valid" packet is one which has an identifiable message type and has passed all format and validation checks that the DHCP server implements. Not all servers validate received packets in the same way, so there will be differences in the counts reported by different servers. It is appropriate to simply accept the server's notion of what constitutes a valid packet.

The total number of valid DHCP packets received by the server is computed as:

```
(dhcpv4CountDiscovers + dhcpv4CountRequests +
dhcpv4CountReleases + dhcpv4CountDeclines + dhcpv4CountInforms
+ dhcpv4CountLeaseQueries)
```

The total number of valid packets (BOOTP and DHCP) received is computed as:

(total number of valid DHCP packets) + bootpCountRequests)

The total number of packets received is computed as:

(total number of valid packets) + bootpCountInvalids + dhcpv4CountInvalids

Similar to the received computations, the total number of DHCP packets sent by the server is computed as:

dhcpv4CountOffers + dhcpv4CountAcks + dhcpv4CountNaks

The number of packets (BOOTP and DHCP) sent by the server is computed as:

(total number of DHCP packets sent) + (bootpCountReplies)

3.3.1. Discontinuities

Hosts for DHCP servers, and the DHCP servers themselves, are generally quite reliable but occasionally counter values may be discontinuous between successive GETs. There are several cases of interest:

- Server fails and is restarted, resetting all counters to zero.
- Server fails and is restarted, with counters in an unknown state.
- Server fails and is restarted, with counters restored to some previously checkpointed value.

The first case MAY occur when a server and agent are incapable of restarting to a "last known good" state, and a manager MUST be capable of recognizing this case. There is little a manager can do other than recognize a reset to zero has occurred and continue from the point of restart.

The second case is currently the Best Current Practice for SNMP managers. Because of the uncertain state of counters following a server restart, a manager MUST discard data from the outage interval and restart its calculations.

The third case SHOULD arise only if agents take periodic snapshots at different intervals than responding to a GET request. While the counts reported in the first GET response following the outage were accurate at some time, they MAY NOT be completely current. If this occurs, the manager MAY have to accept that data has been lost, perhaps discarding accumulated data, and continue.

3.3.2. Counter Rollover

Counter objects increment at different rates. It can be expected that some counter will reach its maximum value and rollover to zero while others are nowhere near their maximum value. When a counter's value at time t2 is less than its value at time t1, the manager SHOULD initially assume that a discontinuity has occurred and perform appropriate data validation to determine if the value has rolled over. If a single rollover has occurred, the value used in calculation SHOULD be:

```
[(maximum value of counter) - (value at time t1)] + (value at
time t2) + 1
```

Managers SHOULD be alert to the possibility of a counter rolling over more than once during the sampling interval. If this is likely to occur (due to very short leases, very large numbers of clients, network topology, and the presence of unreliable clients or intermediate network equipment) a manager SHOULD carefully examine each detected counter rollover to determine if the data can be used or should be discarded.

3.4. Server Configuration Group

The server configuration group contains objects that describe the client host configuration information that is held in the server to be offered to requesting clients. Some of the configuration information is static (e.g., a statically configured IPv4 address) and some of the configuration is dynamic (e.g., an assigned DHCP lease). The intent of the server configuration group is to be able to read the server's configuration.

The configuration information defines a minimal set of information that most servers should be able to provide. Each row of the dhcpv4ServerSubnetTable lists the subnet address, the subnet mask, and the shared network name that is equivalent to the subnet. Equivalence is defined as more than one subnet being present on the same network segment as some other subnet.

The dhcpv4ServerRangeTable lists the start and end IPv4 addresses of the ranges and the subnet of which the range is a member. The dhcpv4ServerRangeInUse object indicates the amount of the range that is currently in use, either through dynamic allocation or being reserved. The range size can be computed as:

```
dhcpv4ServerRangeStartAddress - dhcpv4ServerRangeEndAddress +
1
```

The dhcpv4ServerClientTable provides information about the static and dynamic addresses that the server contains in its configuration. Addresses can be:

- Static, in which case they are predefined though the server's configuration. Static addresses may or may not have been previously served by the server;

- Dynamic, in which case the server has served the addresses and it is currently in active use by a host;

- Expired, in which case the server had previously assigned the address, but the lease time has expired and is retained by the server for possible future use by the same client;

- Configuration-reserved, in which case the address is not available for the server to allocate to a client. A configuration-reserved address is one that has been reserved by the administrator. An example of a configuration-reserved address is an address that is assigned to a client, not through DHCP (e.g., statically assigned), and the address is within a DHCP range; and

- Server-reserved, in which case the server has taken the address out of use. Examples of server-reserved addresses are those that have been declined (i.e., through a DHCPDECLINE) by a client or those that have responded to an ICMP echo before they were assigned.

The protocol used to allocate the address can be determined from the "dhcpv4ServerClientServedProtocol" object. This object indicates whether the address has never been served, or whether BOOTP or DHCP was used to allocate the address.

4. Definitions

```
DHCP-SERVER-MIB DEFINITIONS ::= BEGIN

IMPORTS
   Counter64, Gauge32, Unsigned32, MODULE-IDENTITY, OBJECT-TYPE,
   OBJECT-IDENTITY, NOTIFICATION-TYPE, mib-2
      FROM SNMPv2-SMI

   TEXTUAL-CONVENTION, DateAndTime FROM SNMPv2-TC

   SnmpAdminString FROM SNMP-FRAMEWORK-MIB

   InetAddressIPv4, InetAddressPrefixLength
   FROM INET-ADDRESS-MIB

   MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
   FROM SNMPv2-CONF;


dhcp  MODULE-IDENTITY
   LAST-UPDATED   "200402061633Z"
   ORGANIZATION
      "IETF DHC Working Group
      General Discussion: dhcwg@ietf.org
      Subscribe: http://www1.ietf.org/mailman/listinfo/dhcwg
      Archive: http://www1.ietf.org/mailman/listinfo/dhcwg
      Chair: Ralph Droms, rdroms@cisco.com"
   CONTACT-INFO
      "         Richard Barr Hibbs
      Postal:  952 Sanchez Street
               San Francisco, California 94114-3362
               USA
         Tel:  +1-(415)-648-3920
         Fax:  +1-(415)-648-9017
      E-mail:  rbhibbs@pacbell.net

               Glenn Waters
      Postal:  Nortel Networks, Inc.
               310-875 Carling Avenue
               Ottawa, Ontario K1S 5P1
               Canada
         Tel:  +1-(613)-798-4925
      E-mail:  gww@NortelNetworks.com"
   DESCRIPTION
      "The MIB module for entities implementing the server side of
      the Bootstrap Protocol (BOOTP) and the Dynamic Host
      Configuration protocol (DHCP) for Internet Protocol version
```

```
        4(IPv4).  This MIB does not include support for Dynamic DNS
        (DDNS) updating nor for the DHCP Failover Protocol.

        Copyright (C) The Internet Society (2003).  This version of
        this MIB module is part of RFC xxxx;  see the RFC itself for
        full legal notices."
        -- RFC Editor assigns xxxx and removes this comment

     REVISION "200402061633Z" -- 6 February 2004
     DESCRIPTION "Initial Version, published as RFC xxxx."
        -- RFC Editor assigns xxxx and removes this comment
     ::= { mib-2 9999 } -- IANA will make official assignment


  -- Textual conventions defined by this memo

  Dhcpv4PhysicalAddress ::= TEXTUAL-CONVENTION
     DISPLAY-HINT   "1d,1d,1x:1x:1x:1x:1x:1x"
     STATUS         current
     DESCRIPTION
        "A DHCP-specific encoding of the physical address type and
        physical address, intended to mirror the representation of
        physical addresses in DHCP messages.  The first octet of this
        object contains the hardware type from the 'htype' field of
        the DHCP message, the second octet of this object contains the
        hardware length from the 'hlen' field, and the remaining
        octets contain the hardware address from the 'chaddr' field."
     REFERENCE      "RFC 2131"
     SYNTAX         OCTET STRING (SIZE(18))


  -- declare top-level MIB objects

  dhcpv4Server  OBJECT-IDENTITY
     STATUS      current
     DESCRIPTION
        "DHCPv4 Server MIB objects are defined in this branch."
     ::= { dhcp 1 }

  dhcpv4ServerObjects  OBJECT-IDENTITY
     STATUS      current
     DESCRIPTION
        "DHCP Server MIB server identification objects are all defined
        in this branch."
  ::= { dhcpv4Server 1 }

  dhcpv4ServerSystem  OBJECT-IDENTITY
     STATUS      current
     DESCRIPTION
        "Group of objects that are related to the overall system."
```

```
   ::= { dhcpv4ServerObjects 1 }

   bootpCounters  OBJECT-IDENTITY
      STATUS      current
      DESCRIPTION
         "Group of objects that count various BOOTP events."
   ::= { dhcpv4ServerObjects 2 }

   dhcpv4Counters  OBJECT-IDENTITY
      STATUS      current
      DESCRIPTION
         "Group of objects that count various DHCPv4 events."
   ::= { dhcpv4ServerObjects 3 }

   dhcpv4ServerConfiguration  OBJECT-IDENTITY
      STATUS      current
      DESCRIPTION
         "Objects that contain pre-configured and dynamic configuration
         information."
   ::= { dhcpv4ServerObjects 6 }

   dhcpv4ServerNotifyObjects  OBJECT-IDENTITY
      STATUS      current
      DESCRIPTION
         "Objects that are used only in notifications."
      ::= { dhcpv4ServerObjects 7 }


   -- dhcpv4ServerSystemObjects Group

   dhcpv4ServerSystemDescr  OBJECT-TYPE
      SYNTAX      SnmpAdminString (SIZE(0..255))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A textual description of the server.  This value SHOULD
         include the full name and version identification of the
         server."
      ::= { dhcpv4ServerSystem 1 }

   dhcpv4ServerSystemObjectID  OBJECT-TYPE
      SYNTAX      OBJECT IDENTIFIER
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The vendor's authoritative identification of the network
         management subsystem contained in this entity.  This value is
         allocated within the SMI enterprise subtree (1.3.6.1.4.1) and
         provides an easy and unambiguous means for determining what
```

kind of server is being managed. For example, if vendor Ohso

```
         Soft, Inc. is assigned the subtree 1.3.6.1.4.1.4242, it might
         assign the identifier 1.3.6.1.4.1.4242.1.1 to its Ursa DHCP
         Server."
      ::= { dhcpv4ServerSystem 2 }


   -- bootpCounterObjects Group

   bootpCountRequests  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of packets received that contain a Message Type of
         1 (BOOTREQUEST) in the first octet and do not contain option
         number 53 (DHCP Message Type) in the options."
      REFERENCE
         "RFC-2131."
      ::= { bootpCounters 1 }

   bootpCountInvalids  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of packets received that do not contain a Message
         Type of 1 (BOOTREQUEST) in the first octet or are not valid
         BOOTP packets (e.g., too short, invalid field in packet
         header)."
      ::= { bootpCounters 2 }

   bootpCountReplies  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of packets sent that contain a Message Type of 2
         (BOOTREPLY) in the first octet and do not contain option
         number 53 (DHCP Message Type) in the options."
      REFERENCE
         "RFC-2131."
      ::= { bootpCounters 3 }

   bootpCountDroppedUnknownClients  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
```

```
         "The number of BOOTP packets dropped due to the server not
         recognizing or not providing service to the hardware address
         received in the incoming packet."
      ::= { bootpCounters 4 }

   bootpCountDroppedNotServingSubnet  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of BOOTP packets dropped due to the server not
         being configured or not otherwise able to serve addresses on
         the subnet from which this message was received."
      ::= { bootpCounters 5 }


   -- DHCP Counters Group

   dhcpv4CountDiscovers  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCPDISCOVER (option 53 with value 1) packets
         received."
      REFERENCE
         "RFC2131; RFC2132, section 9.6."
      ::= { dhcpv4Counters 1 }

   dhcpv4CountOffers  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCPOFFER (option 53 with value 2) packets
         sent."
      REFERENCE
         "RFC2131; RFC2132, section 9.6."
      ::= { dhcpv4Counters 2 }

   dhcpv4CountRequests  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCPREQUEST (option 53 with value 3) packets
         received."
      REFERENCE
         "RFC2131; RFC2132, section 9.6."
```

```
      ::= { dhcpv4Counters 3}
```

```
   dhcpv4CountDeclines  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCPDECLINE (option 53 with value 4) packets
         received."
      REFERENCE
         "RFC2131; RFC2132, section 9.6."
      ::= { dhcpv4Counters 4 }

   dhcpv4CountAcks  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCPACK (option 53 with value 5) packets sent."
      REFERENCE
         "RFC2131; RFC2132, section 9.6."
   ::= { dhcpv4Counters 5 }

   dhcpv4CountNaks  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCPNACK (option 53 with value 6) packets
         sent."
      REFERENCE
         "RFC2131; RFC2132, section 9.6."
      ::= { dhcpv4Counters 6 }

   dhcpv4CountReleases  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCPRELEASE (option 53 with value 7) packets
         received."
      REFERENCE
         "RFC2131; RFC2132, section 9.6."
      ::= { dhcpv4Counters 7 }

   dhcpv4CountInforms  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
```

```
         "The number of DHCPINFORM (option 53 with value 8) packets
         received."
      REFERENCE
         "RFC2131; RFC2132, section 9.6."
      ::= { dhcpv4Counters 8 }

   dhcpv4CountForcedRenews  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCPFORCERENEW (option 53 with value 9) packets
         sent."
      REFERENCE
         " RFC 3203, DHCP reconfigure extension."
      ::= { dhcpv4Counters 9 }

   dhcpv4CountInvalids  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCP packets received whose DHCP message type
         (i.e., option number 53) is not understood or handled by the
         server."
      ::= { dhcpv4Counters 10 }

   dhcpv4CountDroppedUnknownClient  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCP packets dropped due to the server not
         recognizing or not providing service to the client-id and/or
         hardware address received in the incoming packet."
      ::= { dhcpv4Counters 11 }

   dhcpv4CountDroppedNotServingSubnet  OBJECT-TYPE
      SYNTAX      Counter64
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of DHCP packets dropped due to the server not
         being configured or not otherwise able to serve addresses on
         the subnet from which this message was received."
      ::= { dhcpv4Counters 12 }


   -- DHCP Server Configuration
```

```
   -- dhcpv4ServerSharedNetObjects Group

   dhcpv4ServerSharedNetTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF Dhcpv4ServerSharedNetEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A list of shared networks that are configured in the server.
         A shared network is the logical aggregation of one or more
         subnets that share a common network segment (e.g., multi-
         tapped coaxial cable, wiring hub, or switch).  This table is
         present ONLY for those servers that organize the ranges of
         addresses available for assignment where a higher-level
         grouping (i.e., the 'shared' network) exists above ranges and
         subnets."
      ::= { dhcpv4ServerConfiguration 1 }

   dhcpv4ServerSharedNetEntry  OBJECT-TYPE
      SYNTAX      Dhcpv4ServerSharedNetEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A logical row in the dhcpv4ServerSharedNetTable."
      INDEX {
         dhcpv4ServerSharedNetName
      }
      ::= { dhcpv4ServerSharedNetTable 1}

   Dhcpv4ServerSharedNetEntry ::= SEQUENCE {
      dhcpv4ServerSharedNetName                    SnmpAdminString,
      dhcpv4ServerSharedNetFreeAddrLowThreshold    Unsigned32,
      dhcpv4ServerSharedNetFreeAddrHighThreshold   Unsigned32,
      dhcpv4ServerSharedNetFreeAddresses           Unsigned32,
      dhcpv4ServerSharedNetReservedAddresses       Unsigned32,
      dhcpv4ServerSharedNetTotalAddresses          Unsigned32
   }

   dhcpv4ServerSharedNetName  OBJECT-TYPE
      SYNTAX      SnmpAdminString (SIZE(1..100))
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The name of the shared network, which uniquely identifies an
         entry in the dhcpv4ServerSharedNetTable."
      ::= { dhcpv4ServerSharedNetEntry 1 }

   dhcpv4ServerSharedNetFreeAddrLowThreshold  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  accessible-for-notify
```

```
      STATUS      current
```

```
      DESCRIPTION
         "The low threshold for available free addresses in this shared
         network.  If the value for available free addresses in this
         shared network becomes equal to or less than this value, a
         dhcpv4ServerFreeAddressLow event is generated for this shared
         network.  No more dhcpv4ServerFreeAddressLow events will be
         generated for this subnet during this execution of the DHCP
         server until the value for available free addresses has
         exceeded the value of
         dhcpv4ServerSharedNetFreeAddrHighThreshold."
      ::= { dhcpv4ServerSharedNetEntry 2 }

   dhcpv4ServerSharedNetFreeAddrHighThreshold  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The high threshold for available free addresses in this
         shared network.  If a dhcpv4ServerFreeAddressLow event was
         generated for this subnet, and the value for available free
         addresses has exceeded the value of
         dhcpv4ServerSubnetFreeAddrHighThreshold, then a
         dhcpv4ServerFreeAddressHigh event will be generated.  No more
         dhcpv4ServerFreeAddressHigh events will be generated for this
         subnet during this execution of the DHCP server until the
         value for available free addresses becomes equal to or less
         than the value of dhcpv4ServerSubnetFreeAddrLowThreshold."
      ::= { dhcpv4ServerSharedNetEntry 3 }

   dhcpv4ServerSharedNetFreeAddresses  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The number of IPv4 addresses which are available within this
         shared network.  If the server does not count free addresses
         by shared network segment, this value will be zero."
      ::= { dhcpv4ServerSharedNetEntry 4 }

   dhcpv4ServerSharedNetReservedAddresses  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The number of IPv4 addresses which are reserved (not
         available for assignment) within this shared network.  If the
         server does not count reserved addresses by shared network
         segment, this value will be zero."
      ::= { dhcpv4ServerSharedNetEntry 5 }
```

```
   dhcpv4ServerSharedNetTotalAddresses  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The number of IPv4 addresses which are available within this
         shared network.  If the server does not count total addresses
         by shared network segment, this value will be zero."
      ::= { dhcpv4ServerSharedNetEntry 6 }


   -- dhcpv4ServerSubnetObjects Group

   dhcpv4ServerSubnetTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF Dhcpv4ServerSubnetEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A list of subnets that are configured in this server."
      ::= { dhcpv4ServerConfiguration 2 }

   dhcpv4ServerSubnetEntry  OBJECT-TYPE
      SYNTAX      Dhcpv4ServerSubnetEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A logical row in the dhcpv4ServerSubnetTable."
      INDEX {
         dhcpv4ServerSubnetAddress
      }
      ::= { dhcpv4ServerSubnetTable 1 }

   Dhcpv4ServerSubnetEntry ::= SEQUENCE {
      dhcpv4ServerSubnetAddress                InetAddressIPv4,
      dhcpv4ServerSubnetMask
         InetAddressPrefixLength,
      dhcpv4ServerSubnetSharedNetworkName      SnmpAdminString,
      dhcpv4ServerSubnetFreeAddrLowThreshold   Unsigned32,
      dhcpv4ServerSubnetFreeAddrHighThreshold  Unsigned32,
      dhcpv4ServerSubnetFreeAddresses          Unsigned32
   }

   dhcpv4ServerSubnetAddress  OBJECT-TYPE
      SYNTAX      InetAddressIPv4
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The IPv4 address of the subnet entry in the
         dhcpv4ServerSubnetTable."
```

```
    ::= { dhcpv4ServerSubnetEntry 1 }
```

```
   dhcpv4ServerSubnetMask  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The subnet mask of the subnet.  This MUST be the same as the
         value of DHCP option 1 offered to clients on this subnet."
      ::= { dhcpv4ServerSubnetEntry 2 }

   dhcpv4ServerSubnetSharedNetworkName  OBJECT-TYPE
      SYNTAX      SnmpAdminString (SIZE(1..100))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The shared subnet name (used as an index into the server
         shared subnet table) to which this subnet belongs.  This value
         will be null for servers that do not organize or describe
         networks in this manner."
      ::= { dhcpv4ServerSubnetEntry 3 }

   dhcpv4ServerSubnetFreeAddrLowThreshold  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The low threshold for available free addresses in this
         subnet.  If the value for available free addresses in this
         subnet becomes equal to or less than this value, a
         dhcpv4ServerSubnetFreeAddrLowThreshold event will be generated
         for this shared network.  No more
         dhcpv4ServerSubnetFreeAddrLowThreshold events will be
         generated for this subnet during this execution of the DHCP
         server until the value for available free addresses has
         exceeded the value of
         dhcpv4ServerSubnetFreeAddrHighThreshold."
      ::= { dhcpv4ServerSubnetEntry 4 }

   dhcpv4ServerSubnetFreeAddrHighThreshold  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The high threshold for available free addresses in this
         subnet.  If a dhcpv4ServerSubnetFreeAddrLowThreshold event has
         been generated for this subnet, and the value for available
         free addresses has exceeded the value of
         dhcpv4ServerSubnetFreeAddrHighThreshold, then a
         dhcpv4ServerFreeAddressHigh event will be generated.  No more
```

dhcpv4ServerFreeAddressHigh events will be generated for this

```
         subnet during this execution of the DHCP server until the
         value for available free addresses becomes equal to or less
         than the value of dhcpv4ServerSubnetFreeAddrLowThreshold."
      ::= { dhcpv4ServerSubnetEntry 5 }

   dhcpv4ServerSubnetFreeAddresses  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The number of free IPv4 addresses which are available in this
         subnet."
      ::= { dhcpv4ServerSubnetEntry 6 }


   -- dhcpv4ServerRangeObjects Group

   dhcpv4ServerRangeTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF Dhcpv4ServerRangeEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A list of ranges that are configured on this server."
      ::= { dhcpv4ServerConfiguration 3 }

   dhcpv4ServerRangeEntry  OBJECT-TYPE
      SYNTAX      Dhcpv4ServerRangeEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A logical row in the dhcpv4ServerRangeTable."
      INDEX {
         dhcpv4ServerRangeStartAddress,
         dhcpv4ServerRangeEndAddress
      }
      ::= { dhcpv4ServerRangeTable 1 }

   Dhcpv4ServerRangeEntry ::= SEQUENCE {
      dhcpv4ServerRangeStartAddress       InetAddressIPv4,
      dhcpv4ServerRangeEndAddress         InetAddressIPv4,
      dhcpv4ServerRangeSubnetMask         InetAddressPrefixLength,
      dhcpv4ServerRangeInUse              Gauge32,
      dhcpv4ServerRangeOutstandingOffers  Gauge32
   }

   dhcpv4ServerRangeStartAddress  OBJECT-TYPE
      SYNTAX      InetAddressIPv4
      MAX-ACCESS  not-accessible
      STATUS      current
```

DESCRIPTION

```
         "The IPv4 address of the first address in the range.  The
         value of dhcpv4ServerRangeStartAddress MUST be less than or
         equal to the value of dhcpv4ServerRangeEndAddress."
      ::= { dhcpv4ServerRangeEntry 1 }

   dhcpv4ServerRangeEndAddress  OBJECT-TYPE
      SYNTAX      InetAddressIPv4
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The IPv4 address of the last address in the range.  The value
         of dhcpv4ServerRangeEndAddress MUST be greater than or equal
         to the value of dhcpv4ServerRangeStartAddress."
      ::= { dhcpv4ServerRangeEntry 2 }

   dhcpv4ServerRangeSubnetMask  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The subnet address mask for this range."
      ::= { dhcpv4ServerRangeEntry 3 }

   dhcpv4ServerRangeInUse  OBJECT-TYPE
      SYNTAX      Gauge32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of addresses in this range that are currently in
         use.  This number includes those addresses whose lease has not
         expired and addresses which have been reserved (either by the
         server or through configuration)."
      ::= { dhcpv4ServerRangeEntry 4 }

   dhcpv4ServerRangeOutstandingOffers  OBJECT-TYPE
      SYNTAX      Gauge32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of outstanding DHCPOFFER messages for this range
         is reported with this value.  An offer is outstanding if the
         server has sent a DHCPOFFER message to a client, but has not
         yet received a DHCPREQUEST message from the client nor has the
         server-specific timeout (limiting the time in which a client
         can respond to the offer message) for the offer message
         expired."
      ::= { dhcpv4ServerRangeEntry 5 }
```

```
   -- dhcpv4ServerClientObjects Group
```

```
   dhcpv4ServerClientTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF Dhcpv4ServerClientEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "An optional list of addresses that are known by this server.
         The list MUST contain addresses that have not expired.  The
         list MUST NOT contain addresses that have never been assigned
         by the server UNLESS the lease is pre-configured in the server
         (e.g., a static lease for a host).  Expired leases MAY appear
         during the time they are 'remembered' by the server for
         subsequent assignment to the same host."
      ::= { dhcpv4ServerConfiguration 4 }

   dhcpv4ServerClientEntry  OBJECT-TYPE
      SYNTAX      Dhcpv4ServerClientEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A logical row in the dhcpv4ServerClientTable."
      INDEX {
         dhcpv4ServerClient
      }
      ::= { dhcpv4ServerClientTable 1 }

   Dhcpv4ServerClientEntry ::= SEQUENCE {
      dhcpv4ServerClient                  InetAddressIPv4,
      dhcpv4ServerClientSubnetMask        InetAddressPrefixLength,
      dhcpv4ServerClientRange          InetAddressIPv4,
      dhcpv4ServerClientLeaseType         INTEGER,
      dhcpv4ServerClientTimeRemaining     Unsigned32,
      dhcpv4ServerClientAllowedProtocol   INTEGER,
      dhcpv4ServerClientServedProtocol INTEGER,
      dhcpv4ServerClientPhysicalAddress   Dhcpv4PhysicalAddress,
      dhcpv4ServerClientClientId       OCTET STRING,
      dhcpv4ServerClientHostName       SnmpAdminString,
      dhcpv4ServerClientDomainName        SnmpAdminString
   }

   dhcpv4ServerClient  OBJECT-TYPE
      SYNTAX      InetAddressIPv4
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The IPv4 address of this entry in the
         dhcpv4ServerClientTable."
      ::= { dhcpv4ServerClientEntry 1 }
```

```
   dhcpv4ServerClientSubnetMask  OBJECT-TYPE
```

```
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The subnet mask (DHCP option 1) provided to the client
         offered this address.  The subnet, resulting from logically
         ANDing the subnet mask with the entry's IPv4 address, MUST be
         configured on this server and appear as a row in the
         dhcpSubnetTable."
      ::= { dhcpv4ServerClientEntry 2 }

   dhcpv4ServerClientRange  OBJECT-TYPE
      SYNTAX      InetAddressIPv4
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The starting IPv4 address (dhcpv4ServerRangeStartAddress
         object) of the range to which this address belongs.  If the
         address does not fall into one of the configured ranges (e.g.,
         a statically configured address on a subnet) the range MAY be
         0.0.0.0."
      ::= { dhcpv4ServerClientEntry 3 }

   dhcpv4ServerClientLeaseType  OBJECT-TYPE
      SYNTAX      INTEGER {
         static(1),
         dynamic(2),
         expired(3),
         configurationReserved(4),
         serverReserved(5)
      }
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The type of this address.  Types are:
           (1) Static addresses defined by the server configuration.
           (2) Dynamic addresses defined by the server configuration
               AND actually assigned by the server.
           (3) Expired dynamic addresses, previously assigned by the
               server, and 'remembered' for subsequent assignment to
               the same host.
           (4) Addresses reserved (i.e., not assignable) by the server
               configuration.
           (5) Addresses previously assigned by the server, but
               temporarily or permanently removed from assignable state
               for some reason, e.g., the server received an ICMP
               ECHOREPLY for the IPv4 address or a DHCPDECLINE message
               has been received for the IPv4 address."
      ::= { dhcpv4ServerClientEntry 4 }
```

```
   dhcpv4ServerClientTimeRemaining  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of seconds until the lease expires.  A value of
         4294967295 (i.e., 0xFFFFFFFF) SHOULD be used for leases that
         have a lease time which is 'infinite' and for BOOTP leases."
      ::= { dhcpv4ServerClientEntry 5 }

   dhcpv4ServerClientAllowedProtocol  OBJECT-TYPE
      SYNTAX      INTEGER {
         none(1),
         bootp(2),
         dhcp(3),
         bootpOrDhcp(4)
      }
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The type of protocol that is allowed to be used to serve this
         address.  A type of none (1) indicates that the address is not
         available to be served (e.g., a reserved address).  Type (2)
         is reserved for BOOTP only devices, while type (3) is reserved
         for DHCP only devices.  A type of bootp-or-dhcp (4) can be
         offered to any type of client."
      ::= { dhcpv4ServerClientEntry 6 }

   dhcpv4ServerClientServedProtocol  OBJECT-TYPE
      SYNTAX      INTEGER {
         none(1),
         bootp(2),
         dhcp(3)
      }
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The type of protocol that was used when this address was
         assigned.  This object will have the value of none (1) if the
         address has not been served."
      ::= { dhcpv4ServerClientEntry 7 }

   dhcpv4ServerClientPhysicalAddress  OBJECT-TYPE
      SYNTAX      Dhcpv4PhysicalAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The hardware type and hardware address of the client that has
         been assigned this lease.  The first octet of this object
```

contains the hardware type from the 'htype' field of the BOOTP

```
         packet and the remaining octets contain the hardware address
         from the 'chaddr' field of the BOOTP packet.  This object MAY
         be empty if the address has not been previously served."
      ::= { dhcpv4ServerClientEntry 8 }

   dhcpv4ServerClientClientId  OBJECT-TYPE
      SYNTAX      OCTET STRING (SIZE(0..255))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The client-id of the client that has been assigned this
         lease.  The client-id is the value specified in option 61
         (client-id option) when the lease was assigned.  This object
         MAY be empty if the lease has not been previously assigned or
         if the client-id option was not specified when the address was
         assigned."
      ::= { dhcpv4ServerClientEntry 9 }

   dhcpv4ServerClientHostName  OBJECT-TYPE
      SYNTAX      SnmpAdminString (SIZE(1..255))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The host name (DHCP option 12) the client is configured to
         use, or if no host name was configured then the host name that
         the client supplied when requesting an address.  While this
         object has a maximum size of 255 octets, a Fully-Qualified
         Domain Name (FQDN) consisting of a Host Name part and a Domain
         Name part is currently limited to 255 octets.  Therefore, the
         sum of the string lengths for this object and the
         dhcpv4ServerClientDomainName MUST be, in practice, less than
         256 octets."
      ::= { dhcpv4ServerClientEntry 10 }

   dhcpv4ServerClientDomainName  OBJECT-TYPE
      SYNTAX      SnmpAdminString (SIZE(1..255))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The domain name (DHCP option 15) assigned to the client.
         While this object has a maximum size of 255 octets, a Fully-
         Qualified Domain Name (FQDN) consisting of a Host Name part
         and a Domain Name part is currently limited to 255 octets,
         less the separator ('.') character.  Therefore, the sum of the
         string lengths for this object and the
         dhcpv4ServerClientHostName MUST be, in practice, less than 256
         octets."
      ::= { dhcpv4ServerClientEntry 11 }
```

```
   -- dhcpv4ServerNotifyObjects Group

   dhcpv4ServerNotifyDuplicateIpAddr  OBJECT-TYPE
      SYNTAX      InetAddressIPv4
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The IPv4 address found to be a duplicate.  Duplicates are
         detected by servers, which issue an ICMP ECHOREQUEST prior to
         offering an IPv4 address lease, or by a client issuing a
         gratuitous ARP message and reported through a DHCPDECLINE
         message."
      ::= { dhcpv4ServerNotifyObjects 1 }

   dhcpv4ServerNotifyDuplicateMac  OBJECT-TYPE
      SYNTAX      Dhcpv4PhysicalAddress
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The offending MAC address which caused a duplicate IPv4
         address to be detected, if captured by the server, else 00-00-
         00-00-00-00."
      ::= { dhcpv4ServerNotifyObjects 2 }

   dhcpv4ServerNotifyClientOrServerDetected  OBJECT-TYPE
      SYNTAX      INTEGER {
         client(1),
         server(2)
      }
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "Duplicate IPv4 addresses can be detected either by a server,
         using an ICMP ECHO message, or by a client using ARP.  This
         object is set by the server to (1) if the client used
         DHCPDECLINE to mark the offered address as in-use, or to (2)
         if the server discovered the address in use by some client
         before offering it."
      ::= { dhcpv4ServerNotifyObjects 3 }

   dhcpv4ServerNotifyServerStart  OBJECT-TYPE
      SYNTAX      DateAndTime
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The date and time when the server began operation."
      ::= { dhcpv4ServerNotifyObjects 4 }

   dhcpv4ServerNotifyServerStop  OBJECT-TYPE
```

```
      SYNTAX      DateAndTime
```

```
      MAX-ACCESS  accessible-for-notify
      STATUS      current
      DESCRIPTION
         "The date and time when the server ceased operation."
      ::= { dhcpv4ServerNotifyObjects 5 }


   -- Notifications

   dhcpv4ServerNotificationPrefix  OBJECT IDENTIFIER
   ::= { dhcpv4Server 2 }

   dhcpv4ServerNotifications OBJECT IDENTIFIER
   ::= { dhcpv4ServerNotificationPrefix 0 }

   dhcpv4ServerFreeAddressLow  NOTIFICATION-TYPE
      OBJECTS {
         dhcpv4ServerSharedNetFreeAddrLowThreshold,
         dhcpv4ServerSharedNetFreeAddresses
      }
      STATUS      current
      DESCRIPTION
         "This notification signifies that the number of available IPv4
         addresses for a particular shared network has fallen below the
         value of dhcpv4ServerSharedNetFreeAddrLowThreshold for that
         shared network."
   ::= { dhcpv4ServerNotifications 1 }

   dhcpv4ServerFreeAddressHigh  NOTIFICATION-TYPE
      OBJECTS {
         dhcpv4ServerSharedNetFreeAddrHighThreshold,
         dhcpv4ServerSharedNetFreeAddresses
      }
      STATUS      current
      DESCRIPTION
         "This notification signifies that the number of available IPv4
         addresses for a particular shared network has risen above the
         value of dhcpv4ServerSharedNetFreeAddrHighThreshold for that
         shared network."
   ::= { dhcpv4ServerNotifications 2 }

   dhcpv4ServerStartTime  NOTIFICATION-TYPE
      OBJECTS     { dhcpv4ServerNotifyServerStart }
      STATUS      current
      DESCRIPTION
         "This notification signifies that the server of the specified
         type has started on the host from which this notification has
         been sent."
   ::= { dhcpv4ServerNotifications 3 }
```

```
   dhcpv4ServerStopTime  NOTIFICATION-TYPE
      OBJECTS  { dhcpv4ServerNotifyServerStop }
      STATUS         current
      DESCRIPTION
         "This notification signifies that the server of the specified
         type has stopped normally on the host from which this
         notification has been sent."
   ::= { dhcpv4ServerNotifications 4 }

   dhcpv4ServerDuplicateAddress NOTIFICATION-TYPE
      OBJECTS {
         dhcpv4ServerNotifyDuplicateIpAddr,
         dhcpv4ServerNotifyDuplicateMac,
         dhcpv4ServerNotifyClientOrServerDetected
      }
      STATUS      current
      DESCRIPTION
         "This notification signifies that a duplicate IPv4 address has
         been detected.  The DHCP server can detect this condition
         through the ping-before-offer mechanism.  Alternatively, the
         client may have sent a DHCPDECLINE back to the server; this is
         assumed to be the result of the client detecting that the
         address was in use.  In either case, the DHCP server marks the
         IPv4 address as unavailable for leasing to clients.  The
         dhcpv4ServerNotifyClientOrServerDetected object indicates
         whether the client or server detected this condition."
   ::= { dhcpv4ServerNotifications 5 }



   -- Conformance

   dhcpv4ServerConformance  OBJECT-IDENTITY
      STATUS      current
      DESCRIPTION
         "DHCP server conformance objects are all defined in this
         branch."
      ::= { dhcpv4Server 3 }

   dhcpv4ServerCompliances  OBJECT IDENTIFIER
   ::= { dhcpv4ServerConformance 1 }

   dhcpv4ServerGroups  OBJECT IDENTIFIER
   ::= { dhcpv4ServerConformance 2 }


   -- Compliance groups

   dhcpv4ServerCompliance MODULE-COMPLIANCE
      STATUS      current
```

```
      DESCRIPTION
         "This group describes the requirements for conformance to the
         DHCP Server MIB.  A DHCPv4 server implementation is only
         REQUIRED to support IPv4 addresses.  In particular, this
         comment applies to the following objects with MAX-ACCESS not-
         accessible:
            dhcpv4ServerSubnet
            dhcpv4ServerRangeStart
            dhcpv4ServerRangeEnd
            dhcpv4ServerClient."
      MODULE -- this module
      MANDATORY-GROUPS {
         dhcpv4ServerSystemObjects,
         bootpCounterObjects,
         dhcpv4CounterObjects,
         dhcpv4ServerSharedNetObjects,
         dhcpv4ServerSubnetObjects,
         dhcpv4ServerRangeObjects,
         dhcpv4ServerClientObjects,
         dhcpv4ServerNotifyObjectsGroup,
         dhcpv4ServerNotificationsGroup
      }
   ::= { dhcpv4ServerCompliances 1 }


   -- Object groups

   dhcpv4ServerSystemObjects OBJECT-GROUP
      OBJECTS {
         dhcpv4ServerSystemDescr,
         dhcpv4ServerSystemObjectID
      }
      STATUS      current
      DESCRIPTION
         "Objects belonging to the dhcpv4ServerSystemObjects group."
   ::= { dhcpv4ServerGroups 1 }

   bootpCounterObjects OBJECT-GROUP
      OBJECTS {
         bootpCountRequests,
         bootpCountInvalids,
         bootpCountReplies,
         bootpCountDroppedUnknownClients,
         bootpCountDroppedNotServingSubnet
      }
      STATUS      current
      DESCRIPTION
         "Objects belonging to the bootpBounterObjects group."
   ::= { dhcpv4ServerGroups 2 }
```

```
   dhcpv4CounterObjects  OBJECT-GROUP
      OBJECTS {
         dhcpv4CountDiscovers,
         dhcpv4CountOffers,
         dhcpv4CountRequests,
         dhcpv4CountDeclines,
         dhcpv4CountAcks,
         dhcpv4CountNaks,
         dhcpv4CountReleases,
         dhcpv4CountInforms,
         dhcpv4CountForcedRenews,
         dhcpv4CountInvalids,
         dhcpv4CountDroppedUnknownClient,
         dhcpv4CountDroppedNotServingSubnet
      }
      STATUS      current
      DESCRIPTION
         "Objects belonging to the dhcpv4CounterObjects group."
   ::= { dhcpv4ServerGroups 3 }

   dhcpv4ServerSharedNetObjects  OBJECT-GROUP
      OBJECTS {
         dhcpv4ServerSharedNetFreeAddrLowThreshold,
         dhcpv4ServerSharedNetFreeAddrHighThreshold,
         dhcpv4ServerSharedNetFreeAddresses,
         dhcpv4ServerSharedNetReservedAddresses,
         dhcpv4ServerSharedNetTotalAddresses
      }
      STATUS      current
      DESCRIPTION
         "Objects belonging to the dhcpv4ServerSharedNetObjects group."
      ::= { dhcpv4ServerGroups 4 }

   dhcpv4ServerSubnetObjects  OBJECT-GROUP
      OBJECTS {
         dhcpv4ServerSubnetMask,
         dhcpv4ServerSubnetSharedNetworkName,
         dhcpv4ServerSubnetFreeAddrLowThreshold,
         dhcpv4ServerSubnetFreeAddrHighThreshold,
         dhcpv4ServerSubnetFreeAddresses
      }
      STATUS      current
      DESCRIPTION
         "Objects belonging to the dhcpv4ServerSubnetObjects group."
      ::= { dhcpv4ServerGroups 5 }

   dhcpv4ServerRangeObjects  OBJECT-GROUP
      OBJECTS {
         dhcpv4ServerRangeSubnetMask,
```

```
          dhcpv4ServerRangeInUse,
```

```
         dhcpv4ServerRangeOutstandingOffers
      }
      STATUS      current
      DESCRIPTION
         "Objects belonging to the dhcpv4ServerRangeObjects group."
      ::= { dhcpv4ServerGroups 6 }

   dhcpv4ServerClientObjects  OBJECT-GROUP
      OBJECTS {
         dhcpv4ServerClientSubnetMask,
         dhcpv4ServerClientRange,
         dhcpv4ServerClientLeaseType,
         dhcpv4ServerClientTimeRemaining,
         dhcpv4ServerClientAllowedProtocol,
         dhcpv4ServerClientServedProtocol,
         dhcpv4ServerClientPhysicalAddress,
         dhcpv4ServerClientClientId,
         dhcpv4ServerClientHostName,
         dhcpv4ServerClientDomainName
      }
      STATUS      current
      DESCRIPTION
         "Objects belonging to the dhcpv4ServerClientObjects group."
      ::= { dhcpv4ServerGroups 7 }

   dhcpv4ServerNotifyObjectsGroup  OBJECT-GROUP
      OBJECTS {
         dhcpv4ServerNotifyDuplicateIpAddr,
         dhcpv4ServerNotifyDuplicateMac,
         dhcpv4ServerNotifyClientOrServerDetected,
         dhcpv4ServerNotifyServerStart,
         dhcpv4ServerNotifyServerStop
      }
      STATUS      current
      DESCRIPTION
         "Objects belonging to the dhcpv4ServerNotifyObjects group."
      ::= { dhcpv4ServerGroups 8 }

   dhcpv4ServerNotificationsGroup  NOTIFICATION-GROUP
      NOTIFICATIONS {
         dhcpv4ServerFreeAddressLow,
         dhcpv4ServerFreeAddressHigh,
         dhcpv4ServerStartTime,
         dhcpv4ServerStopTime,
         dhcpv4ServerDuplicateAddress
      }
      STATUS      current
      DESCRIPTION
         "Notifications belonging to the dhcpv4ServerNotifications
```

group."

```
      ::= { dhcpv4ServerGroups 9 }

   END
```

5. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11.

Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

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

6. Acknowledgements

This document is the result of work undertaken by the DHCP working group. The editors would like to particularly acknowledge the development team from Carnegie-Mellon University whose work creating a private MIB for their DHCP server inspired the development of this proposal. In particular, many thanks to Ryan Troll who provided a great deal of useful feedback during the initial development of this MIB, and to Rich Woundy for his excellent suggestions that helped bring the work to closure.

7. IANA Considerations

IANA MUST fill in the value of the RFC number when it is assigned to this memo. It is represented as "xxxx" in the DESCRIPTION section of MODULE-IDENTITY.

One specific value for a MIB object requires completion before this memo can advance to RFC status. It is:

- OID value for "dhcp" -- see MODULE-IDENTITY

8. Security Considerations

There are no management objects defined in this MIB that have a MAX-ACCESS clause of read-write or read-create. Such objects may be considered sensitive or vulnerable in some environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. Therefore, if this MIB is implemented correctly, there is no risk that an intruder can alter or create any management objects of this MIB via direct SNMP SET operations.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- dhcpv4ServerRangeTable

- dhcpv4ServerClientTable

These two objects, in conjunction, provide an observer with a current view of the available and assigned addresses allocated by this server. Such knowledge can be used to manually configure a host computer with a valid IPv4 address for the network managed by the DHCP server. This could be part of either a Theft of Service scheme or a Denial of Service attack wherein rogue (pseudo-)hosts simply claim and defend IPv4 addresses either to subvert accounting for their use or to disrupt the network for legitimate hosts.

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

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is RECOMMENDED that the 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, it is RECOMMENDED to deploy SNMPv3 and to

enable cryptographic security. 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 indeed GET or SET (change/create/delete) them.

Denial of Service attacks on a DHCP server are conceivable by flooding the SNMP (sub-)agent with requests, tying up host system and server resources processing SNMP messages. The authors know of no way to wholly prevent such attacks, but have attempted to construct relatively simple tables to minimize the work required to respond to messages.

9. References

One normative reference is currently an Internet-Draft, nearly ready for Working Group Last Call. This reference MUST be updated when the draft advances to RFC status.

9.1. Normative References

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol," RFC 2131, March 1997.

[RFC2132] Alexander, S. and Droms, R., "DHCP Options and BOOTP Vendor Extensions," RFC 2132, March 1997.

[RFC2578] Case, J., McCloghrie, K., Perkins, D., Rose, M., Schoenwaelder, J., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 2578, April 1999.

[RFC2579] Case, J., McCloghrie, K., Rose, M., Schoenwaelder, J., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 2579, April 1999.

[RFC2580] Case, J., McCloghrie, K., Rose, M., Schoenwaelder, J., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 2580, April 1999.

[RFC3203], Yves T'Joens and Christian Hublet, Peter De Schrijver, "The DHCP Reconfigure Extension," July 2001

9.2. Informative References

[RFC1123] R. Braden, "Requirements for Internet Hosts -- Application and Support," RFC 1123, October 1989.

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

10. Editors' Addresses

Richard Barr Hibbs
952 Sanchez Street
San Francisco, California 94114-3362
USA

Phone: +1-(415)-648-3920
Fax: +1-(415)-648-9017
Email: rbhibbs@pacbell.net

Glenn Waters
Nortel Networks
310-875 Carling Avenue,
Ottawa, Ontario K1S 5P1
Canada

Phone: +1-(613)-798-4925
Email: gww@NortelNetworks.com

11. Full Copyright Statement

Copyright (C), 2004, The Internet Society. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING

BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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