Network Working Group
Request for Comments: 4022
Obsoletes: 2452, 2012
Category: Standards Track

R. Raghunarayan, Ed.
Cisco Systems
March 2005

Management Information Base for the Transmission Control Protocol (TCP)

Status of This Memo

Copyright Notice

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects used for implementations of the Transmission Control Protocol (TCP) in an IP version independent manner. This memo obsoletes RFCs 2452 and 2012.

Table of Contents

1. The Internet-Standard Management Framework

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

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

2. Overview

The current TCP-MIB defined in this memo consists of two tables and a group of scalars:

- The tcp group of scalars includes two sets of objects:

  o Parameters of a TCP protocol engine. These include parameters such as the retransmission algorithm in use (e.g., vanj [VANJ]) and the retransmission timeout values.

  o Statistics of a TCP protocol engine. These include counters for the number of active/passive opens, input/output segments, and errors. Discontinuities in the stats are identified identified via the sysUpTime object, defined in [RFC3418].

- The tcpConnectionTable provides access to status information for all TCP connections handled by a TCP protocol engine. In addition, the table reports identification of the operating system level processes that handle the TCP connections.

- The tcpListenerTable provides access to information about all TCP listening endpoints known by a TCP protocol engine. And as with the connection table, the tcpListenerTable also reports the identification of the operating system level processes that handle this listening TCP endpoint.

2.1. Relationship to Other MIBs

This section discusses the relationship of this TCP-MIB module to other MIB modules.

2.1.1. Relationship to RFC1213-MIB

TCP related MIB objects were originally defined as part of the RFC1213-MIB defined in RFC 1213 [RFC1213]. The TCP related objects of the RFC1213-MIB were later copied into a separate MIB module and published in RFC 2012 [RFC2012] in SMIv2 format.

The previous versions of the TCP-MIB both defined the tcpConnTable, which has been deprecated basically for two reasons:

(1) The tcpConnTable only supports IPv4.

The current approach in the IETF is to write IP version neutral MIBs, based on the InetAddressType and InetAddress constructs defined in [RFC4001], rather than to have different definitions for various version of IP. This reduces the amount of overhead when new objects are introduced, as there is only one place to add them. Hence, the approach taken in [RFC2452], of having separate tables, is not continued.

(2) The tcpConnTable mixes listening endpoints with connections.

It turns out that connections tend to have a different behaviour and management access pattern than listening endpoints. Therefore, splitting the original tcpConnTable into two tables allows for the addition of specific status and statistics objects for listening endpoints and connections.

2.1.2. Relationship to IPV6-TCP-MIB

The IPV6-TCP-MIB defined in RFC 2452 has been moved to Historic status because the approach of having separate IP version specific tables is not followed anymore. Implementation of RFC 2452 is no longer suggested.

2.1.3. Relationship to HOST-RESOURCES-MIB and SYSAPPL-MIB

The tcpConnectionTable and the tcpListenerTable report the identification of the operating system level process that handles a connection or a listening endpoint. The value is reported as an Unsigned32, which is expected to be the same as the hrSWRunIndex of the HOST-RESOURCES-MIB [RFC2790] (if the value is smaller than 2147483647) or the sysApplElmtRunIndex of the SYSAPPL-MIB [RFC2287]. This allows management applications to identify the TCP connections that belong to an operating system level process, which has proven to be valuable in operational environments.

3. Definitions

```
TCP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Integer32, Unsigned32,
    Gauge32, Counter32, Counter64, IpAddress, mib-2
                                        FROM SNMPv2-SMI
    MODULE-COMPLIANCE, OBJECT-GROUP     FROM SNMPv2-CONF
    InetAddress, InetAddressType,
    InetPortNumber                      FROM INET-ADDRESS-MIB;

tcpMIB MODULE-IDENTITY
    LAST-UPDATED "200502180000Z"  -- 18 February 2005
    ORGANIZATION
           "IETF IPv6 MIB Revision Team
            http://www.ietf.org/html.charters/ipv6-charter.html"
    CONTACT-INFO
           "Rajiv Raghunarayan (editor)

            Cisco Systems Inc.
            170 West Tasman Drive
            San Jose, CA 95134

            Phone: +1 408 853 9612
            Email: <raraghun@cisco.com>

            Send comments to <ipv6@ietf.org>"
    DESCRIPTION
           "The MIB module for managing TCP implementations.

            Copyright (C) The Internet Society (2005). This version
            of this MIB module is a part of RFC 4022; see the RFC
            itself for full legal notices."
    REVISION      "200502180000Z"  -- 18 February 2005
    DESCRIPTION
           "IP version neutral revision, published as RFC 4022."
    REVISION      "9411010000Z"
    DESCRIPTION
           "Initial SMIv2 version, published as RFC 2012."
    REVISION      "9103310000Z"
    DESCRIPTION
           "The initial revision of this MIB module was part of
            MIB-II."
    ::= { mib-2 49 }

-- the TCP base variables group
```

```
tcp      OBJECT IDENTIFIER ::= { mib-2 6 }

-- Scalars

tcpRtoAlgorithm OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),    -- none of the following
                    constant(2), -- a constant rto
                    rsre(3),     -- MIL-STD-1778, Appendix B
                    vanj(4),     -- Van Jacobson's algorithm
                    rfc2988(5)   -- RFC 2988
                }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The algorithm used to determine the timeout value used for
            retransmitting unacknowledged octets."
    ::= { tcp 1 }

tcpRtoMin OBJECT-TYPE
    SYNTAX     Integer32 (0..2147483647)
    UNITS      "milliseconds"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The minimum value permitted by a TCP implementation for
            the retransmission timeout, measured in milliseconds.
            More refined semantics for objects of this type depend
            on the algorithm used to determine the retransmission
            timeout; in particular, the IETF standard algorithm
            rfc2988(5) provides a minimum value."
    ::= { tcp 2 }

tcpRtoMax OBJECT-TYPE
    SYNTAX     Integer32 (0..2147483647)
    UNITS      "milliseconds"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The maximum value permitted by a TCP implementation for
            the retransmission timeout, measured in milliseconds.
            More refined semantics for objects of this type depend
            on the algorithm used to determine the retransmission
            timeout; in particular, the IETF standard algorithm
            rfc2988(5) provides an upper bound (as part of an
            adaptive backoff algorithm)."
    ::= { tcp 3 }
```

```
tcpMaxConn OBJECT-TYPE
    SYNTAX     Integer32 (-1 | 0..2147483647)
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The limit on the total number of TCP connections the entity
            can support.  In entities where the maximum number of
            connections is dynamic, this object should contain the
            value -1."
    ::= { tcp 4 }

tcpActiveOpens OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of times that TCP connections have made a direct
            transition to the SYN-SENT state from the CLOSED state.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 5 }

tcpPassiveOpens OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of times TCP connections have made a direct
            transition to the SYN-RCVD state from the LISTEN state.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 6 }

tcpAttemptFails OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of times that TCP connections have made a direct
            transition to the CLOSED state from either the SYN-SENT
            state or the SYN-RCVD state, plus the number of times that
            TCP connections have made a direct transition to the
            LISTEN state from the SYN-RCVD state.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
```

```
    ::= { tcp 7 }

tcpEstabResets OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of times that TCP connections have made a direct
            transition to the CLOSED state from either the ESTABLISHED
            state or the CLOSE-WAIT state.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 8 }

tcpCurrEstab OBJECT-TYPE
    SYNTAX     Gauge32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of TCP connections for which the current state
            is either ESTABLISHED or CLOSE-WAIT."
    ::= { tcp 9 }

tcpInSegs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of segments received, including those
            received in error.  This count includes segments received
            on currently established connections.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 10 }

tcpOutSegs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of segments sent, including those on
            current connections but excluding those containing only
            retransmitted octets.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
```

```
    ::= { tcp 11 }

tcpRetransSegs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of segments retransmitted; that is, the
            number of TCP segments transmitted containing one or more
            previously transmitted octets.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 12 }

tcpInErrs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of segments received in error (e.g., bad
            TCP checksums).

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 14 }

tcpOutRsts OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of TCP segments sent containing the RST flag.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 15 }

-- { tcp 16 } was used to represent the ipv6TcpConnTable in RFC 2452,
-- which has since been obsoleted.  It MUST not be used.

tcpHCInSegs OBJECT-TYPE
    SYNTAX     Counter64
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of segments received, including those
            received in error.  This count includes segments received
```

```
            on currently established connections.  This object is
            the 64-bit equivalent of tcpInSegs.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 17 }

tcpHCOutSegs OBJECT-TYPE
    SYNTAX     Counter64
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of segments sent, including those on
            current connections but excluding those containing only
            retransmitted octets.  This object is the 64-bit
            equivalent of tcpOutSegs.

            Discontinuities in the value of this counter are
            indicated via discontinuities in the value of sysUpTime."
    ::= { tcp 18 }


-- The TCP Connection table

tcpConnectionTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF TcpConnectionEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "A table containing information about existing TCP
            connections.  Note that unlike earlier TCP MIBs, there
            is a separate table for connections in the LISTEN state."
    ::= { tcp 19 }

tcpConnectionEntry OBJECT-TYPE
    SYNTAX     TcpConnectionEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "A conceptual row of the tcpConnectionTable containing
            information about a particular current TCP connection.
            Each row of this table is transient in that it ceases to
            exist when (or soon after) the connection makes the
            transition to the CLOSED state."
    INDEX   { tcpConnectionLocalAddressType,
              tcpConnectionLocalAddress,
              tcpConnectionLocalPort,
              tcpConnectionRemAddressType,
```

```
                tcpConnectionRemAddress,
                tcpConnectionRemPort }
    ::= { tcpConnectionTable 1 }

TcpConnectionEntry ::= SEQUENCE {
        tcpConnectionLocalAddressType   InetAddressType,
        tcpConnectionLocalAddress       InetAddress,
        tcpConnectionLocalPort          InetPortNumber,
        tcpConnectionRemAddressType     InetAddressType,
        tcpConnectionRemAddress         InetAddress,
        tcpConnectionRemPort            InetPortNumber,
        tcpConnectionState              INTEGER,
        tcpConnectionProcess            Unsigned32
    }

tcpConnectionLocalAddressType OBJECT-TYPE
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The address type of tcpConnectionLocalAddress."
    ::= { tcpConnectionEntry 1 }

tcpConnectionLocalAddress OBJECT-TYPE
    SYNTAX     InetAddress
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The local IP address for this TCP connection.  The type
            of this address is determined by the value of
            tcpConnectionLocalAddressType.

            As this object is used in the index for the
            tcpConnectionTable, implementors should be
            careful not to create entries that would result in OIDs
            with more than 128 subidentifiers; otherwise the information
            cannot be accessed by using SNMPv1, SNMPv2c, or SNMPv3."
    ::= { tcpConnectionEntry 2 }

tcpConnectionLocalPort OBJECT-TYPE
    SYNTAX     InetPortNumber
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The local port number for this TCP connection."
    ::= { tcpConnectionEntry 3 }

tcpConnectionRemAddressType OBJECT-TYPE
```

```
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The address type of tcpConnectionRemAddress."
    ::= { tcpConnectionEntry 4 }

tcpConnectionRemAddress OBJECT-TYPE
    SYNTAX     InetAddress
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The remote IP address for this TCP connection.  The type
            of this address is determined by the value of
            tcpConnectionRemAddressType.

            As this object is used in the index for the
            tcpConnectionTable, implementors should be
            careful not to create entries that would result in OIDs
            with more than 128 subidentifiers; otherwise the information
            cannot be accessed by using SNMPv1, SNMPv2c, or SNMPv3."
    ::= { tcpConnectionEntry 5 }

tcpConnectionRemPort OBJECT-TYPE
    SYNTAX     InetPortNumber
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The remote port number for this TCP connection."
    ::= { tcpConnectionEntry 6 }

tcpConnectionState OBJECT-TYPE
    SYNTAX     INTEGER {
                    closed(1),
                    listen(2),
                    synSent(3),
                    synReceived(4),
                    established(5),
                    finWait1(6),
                    finWait2(7),
                    closeWait(8),
                    lastAck(9),
                    closing(10),
                    timeWait(11),
                    deleteTCB(12)
                }
    MAX-ACCESS read-write
    STATUS     current
```

```
    DESCRIPTION
           "The state of this TCP connection.

            The value listen(2) is included only for parallelism to the
            old tcpConnTable and should not be used.  A connection in
            LISTEN state should be present in the tcpListenerTable.

            The only value that may be set by a management station is
            deleteTCB(12).  Accordingly, it is appropriate for an agent
            to return a 'badValue' response if a management station
            attempts to set this object to any other value.

            If a management station sets this object to the value
            deleteTCB(12), then the TCB (as defined in [RFC793]) of
            the corresponding connection on the managed node is
            deleted, resulting in immediate termination of the
            connection.

            As an implementation-specific option, a RST segment may be
            sent from the managed node to the other TCP endpoint (note,
            however, that RST segments are not sent reliably)."
    ::= { tcpConnectionEntry 7 }

tcpConnectionProcess OBJECT-TYPE
    SYNTAX     Unsigned32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The system's process ID for the process associated with
            this connection, or zero if there is no such process.  This
            value is expected to be the same as HOST-RESOURCES-MIB::
            hrSWRunIndex or SYSAPPL-MIB::sysApplElmtRunIndex for some
            row in the appropriate tables."
    ::= { tcpConnectionEntry 8 }

-- The TCP Listener table

tcpListenerTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF TcpListenerEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "A table containing information about TCP listeners.  A
            listening application can be represented in three
            possible ways:

            1. An application that is willing to accept both IPv4 and
               IPv6 datagrams is represented by
```

```
                a tcpListenerLocalAddressType of unknown (0) and
                a tcpListenerLocalAddress of ''h (a zero-length
                octet-string).

             2. An application that is willing to accept only IPv4 or
                IPv6 datagrams is represented by a
                tcpListenerLocalAddressType of the appropriate address
                type and a tcpListenerLocalAddress of '0.0.0.0' or '::'
                respectively.

             3. An application that is listening for data destined
                only to a specific IP address, but from any remote
                system, is represented by a tcpListenerLocalAddressType
                of an appropriate address type, with
                tcpListenerLocalAddress as the specific local address.

             NOTE: The address type in this table represents the
             address type used for the communication, irrespective
             of the higher-layer abstraction.  For example, an
             application using IPv6 'sockets' to communicate via
             IPv4 between ::ffff:10.0.0.1 and ::ffff:10.0.0.2 would
             use InetAddressType ipv4(1))."
    ::= { tcp 20 }

tcpListenerEntry OBJECT-TYPE
    SYNTAX     TcpListenerEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "A conceptual row of the tcpListenerTable containing
            information about a particular TCP listener."
    INDEX   { tcpListenerLocalAddressType,
              tcpListenerLocalAddress,
              tcpListenerLocalPort }
    ::= { tcpListenerTable 1 }

TcpListenerEntry ::= SEQUENCE {
        tcpListenerLocalAddressType       InetAddressType,
        tcpListenerLocalAddress           InetAddress,
        tcpListenerLocalPort              InetPortNumber,
        tcpListenerProcess                Unsigned32
    }

tcpListenerLocalAddressType OBJECT-TYPE
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
```

```
            "The address type of tcpListenerLocalAddress.  The value
             should be unknown (0) if connection initiations to all
             local IP addresses are accepted."
    ::= { tcpListenerEntry 1 }

tcpListenerLocalAddress OBJECT-TYPE
    SYNTAX     InetAddress
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "The local IP address for this TCP connection.

             The value of this object can be represented in three
             possible ways, depending on the characteristics of the
             listening application:

             1. For an application willing to accept both IPv4 and
                IPv6 datagrams, the value of this object must be
                ''h (a zero-length octet-string), with the value
                of the corresponding tcpListenerLocalAddressType
                object being unknown (0).

             2. For an application willing to accept only IPv4 or
                IPv6 datagrams, the value of this object must be
                '0.0.0.0' or '::' respectively, with
                tcpListenerLocalAddressType representing the
                appropriate address type.

             3. For an application which is listening for data
                destined only to a specific IP address, the value
                of this object is the specific local address, with
                tcpListenerLocalAddressType representing the
                appropriate address type.

             As this object is used in the index for the
             tcpListenerTable, implementors should be
             careful not to create entries that would result in OIDs
             with more than 128 subidentifiers; otherwise the information
             cannot be accessed, using SNMPv1, SNMPv2c, or SNMPv3."
    ::= { tcpListenerEntry 2 }

tcpListenerLocalPort OBJECT-TYPE
    SYNTAX     InetPortNumber
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "The local port number for this TCP connection."
    ::= { tcpListenerEntry 3 }
```

```
tcpListenerProcess OBJECT-TYPE
    SYNTAX     Unsigned32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The system's process ID for the process associated with
            this listener, or zero if there is no such process.  This
            value is expected to be the same as HOST-RESOURCES-MIB::
            hrSWRunIndex or SYSAPPL-MIB::sysApplElmtRunIndex for some
            row in the appropriate tables."
    ::= { tcpListenerEntry 4 }


-- The deprecated TCP Connection table

tcpConnTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF TcpConnEntry
    MAX-ACCESS not-accessible
    STATUS     deprecated
    DESCRIPTION
           "A table containing information about existing IPv4-specific
            TCP connections or listeners.  This table has been
            deprecated in favor of the version neutral
            tcpConnectionTable."
    ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
    SYNTAX     TcpConnEntry
    MAX-ACCESS not-accessible
    STATUS     deprecated
    DESCRIPTION
           "A conceptual row of the tcpConnTable containing information
            about a particular current IPv4 TCP connection.  Each row
            of this table is transient in that it ceases to exist when
            (or soon after) the connection makes the transition to the
            CLOSED state."
    INDEX   { tcpConnLocalAddress,
              tcpConnLocalPort,
              tcpConnRemAddress,
              tcpConnRemPort }
    ::= { tcpConnTable 1 }

TcpConnEntry ::= SEQUENCE {
        tcpConnState         INTEGER,
        tcpConnLocalAddress  IpAddress,
        tcpConnLocalPort     Integer32,
        tcpConnRemAddress    IpAddress,
        tcpConnRemPort       Integer32
```

```
    }

tcpConnState OBJECT-TYPE
    SYNTAX     INTEGER {
                    closed(1),
                    listen(2),
                    synSent(3),
                    synReceived(4),
                    established(5),
                    finWait1(6),
                    finWait2(7),
                    closeWait(8),
                    lastAck(9),
                    closing(10),
                    timeWait(11),
                    deleteTCB(12)
                }
    MAX-ACCESS read-write
    STATUS     deprecated
    DESCRIPTION
           "The state of this TCP connection.

            The only value that may be set by a management station is
            deleteTCB(12).  Accordingly, it is appropriate for an agent
            to return a 'badValue' response if a management station
            attempts to set this object to any other value.

            If a management station sets this object to the value
            deleteTCB(12), then the TCB (as defined in [RFC793]) of
            the corresponding connection on the managed node is
            deleted, resulting in immediate termination of the
            connection.

            As an implementation-specific option, a RST segment may be
            sent from the managed node to the other TCP endpoint (note,
            however, that RST segments are not sent reliably)."
    ::= { tcpConnEntry 1 }

tcpConnLocalAddress OBJECT-TYPE
    SYNTAX     IpAddress
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The local IP address for this TCP connection.  In the case
            of a connection in the listen state willing to
            accept connections for any IP interface associated with the
            node, the value 0.0.0.0 is used."
    ::= { tcpConnEntry 2 }
```

```
tcpConnLocalPort OBJECT-TYPE
    SYNTAX     Integer32 (0..65535)
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The local port number for this TCP connection."
    ::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE
    SYNTAX     IpAddress
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The remote IP address for this TCP connection."
    ::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE
    SYNTAX     Integer32 (0..65535)
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The remote port number for this TCP connection."
    ::= { tcpConnEntry 5 }

-- conformance information

tcpMIBConformance OBJECT IDENTIFIER ::= { tcpMIB 2 }

tcpMIBCompliances OBJECT IDENTIFIER ::= { tcpMIBConformance 1 }
tcpMIBGroups      OBJECT IDENTIFIER ::= { tcpMIBConformance 2 }

-- compliance statements

tcpMIBCompliance2 MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
           "The compliance statement for systems that implement TCP.

            A number of INDEX objects cannot be
            represented in the form of OBJECT clauses in SMIv2 but
            have the following compliance requirements,
            expressed in OBJECT clause form in this description
            clause:

            -- OBJECT      tcpConnectionLocalAddressType
            -- SYNTAX      InetAddressType { ipv4(1), ipv6(2) }
            -- DESCRIPTION
            --     This MIB requires support for only global IPv4
```

```
            --     and IPv6 address types.
            --
            -- OBJECT      tcpConnectionRemAddressType
            -- SYNTAX      InetAddressType { ipv4(1), ipv6(2) }
            -- DESCRIPTION
            --     This MIB requires support for only global IPv4
            --     and IPv6 address types.
            --
            -- OBJECT      tcpListenerLocalAddressType
            -- SYNTAX      InetAddressType { unknown(0), ipv4(1),
            --                               ipv6(2) }
            -- DESCRIPTION
            --     This MIB requires support for only global IPv4
            --     and IPv6 address types.  The type unknown also
            --     needs to be supported to identify a special
            --     case in the listener table: a listen using
            --     both IPv4 and IPv6 addresses on the device.
            --
          "
    MODULE  -- this module
        MANDATORY-GROUPS { tcpBaseGroup, tcpConnectionGroup,
                           tcpListenerGroup }
        GROUP       tcpHCGroup
        DESCRIPTION
           "This group is mandatory for systems that are capable
            of receiving or transmitting more than 1 million TCP
            segments per second.  1 million segments per second will
            cause a Counter32 to wrap in just over an hour."
        OBJECT      tcpConnectionState
        SYNTAX      INTEGER { closed(1), listen(2), synSent(3),
                              synReceived(4), established(5),
                              finWait1(6), finWait2(7), closeWait(8),
                              lastAck(9), closing(10), timeWait(11) }
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required, nor is support for the value
            deleteTCB (12)."
    ::= { tcpMIBCompliances 2 }

tcpMIBCompliance MODULE-COMPLIANCE
    STATUS     deprecated
    DESCRIPTION
           "The compliance statement for IPv4-only systems that
            implement TCP.  In order to be IP version independent, this
            compliance statement is deprecated in favor of
            tcpMIBCompliance2.  However, agents are still encouraged
            to implement these objects in order to interoperate with
            the deployed base of managers."
```

```
    MODULE  -- this module
        MANDATORY-GROUPS { tcpGroup }
        OBJECT      tcpConnState
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required."
    ::= { tcpMIBCompliances 1 }


-- units of conformance

tcpGroup OBJECT-GROUP
    OBJECTS   { tcpRtoAlgorithm, tcpRtoMin, tcpRtoMax,
                tcpMaxConn, tcpActiveOpens,
                tcpPassiveOpens, tcpAttemptFails,
                tcpEstabResets, tcpCurrEstab, tcpInSegs,
                tcpOutSegs, tcpRetransSegs, tcpConnState,
                tcpConnLocalAddress, tcpConnLocalPort,
                tcpConnRemAddress, tcpConnRemPort,
                tcpInErrs, tcpOutRsts }
    STATUS     deprecated
    DESCRIPTION
           "The tcp group of objects providing for management of TCP
            entities."
    ::= { tcpMIBGroups 1 }

tcpBaseGroup OBJECT-GROUP
    OBJECTS   { tcpRtoAlgorithm, tcpRtoMin, tcpRtoMax,
                tcpMaxConn, tcpActiveOpens,
                tcpPassiveOpens, tcpAttemptFails,
                tcpEstabResets, tcpCurrEstab, tcpInSegs,
                tcpOutSegs, tcpRetransSegs,
                tcpInErrs, tcpOutRsts }
    STATUS     current
    DESCRIPTION
           "The group of counters common to TCP entities."
    ::= { tcpMIBGroups 2 }

tcpConnectionGroup OBJECT-GROUP
    OBJECTS    { tcpConnectionState, tcpConnectionProcess }
    STATUS     current
    DESCRIPTION
           "The group provides general information about TCP
            connections."
    ::= { tcpMIBGroups 3 }

tcpListenerGroup OBJECT-GROUP
    OBJECTS    { tcpListenerProcess }
```

```
    STATUS     current
    DESCRIPTION
           "This group has objects providing general information about
            TCP listeners."
    ::= { tcpMIBGroups 4 }

tcpHCGroup OBJECT-GROUP
    OBJECTS    { tcpHCInSegs, tcpHCOutSegs }
    STATUS     current
    DESCRIPTION
           "The group of objects providing for counters of high speed
            TCP implementations."
    ::= { tcpMIBGroups 5 }

END
```

4. Acknowledgements

This document contains a modified subset of RFC 1213 and updates RFC 2012 and RFC 2452. Acknowledgements are therefore due to the authors and editors of these documents for their excellent work. Several useful comments regarding usability and design were also received from Kristine Adamson. The authors would like to thank all these people for their contribution to this effort.

5. References

5.1. Normative References

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DARPA, September 1981.

[RFC2287] Krupczak, C. and J. Saperia, "Definitions of System-Level Managed Objects for Applications", RFC 2287, February 1998.

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

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

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

[RFC2790] Waldbusser, S. and P. Grillo, "Host Resources MIB", RFC 2790, March 2000.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

5.2. Informative References

[RFC1213] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1213, March 1991.

[RFC2012] McCloghrie, K., Ed., "SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2", RFC 2012, November 1996.

[RFC2452] Daniele, M., "IP Version 6 Management Information Base for the Transmission Control Protocol", RFC 2452, December 1998.

[RFC2988] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000.

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

[RFC3418] Presuhn, R., Ed., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", RFC 3418, December 2002.

[VANJ] Jacobson, V., "Congestion Avoidance and Control", SIGCOMM 1988, Stanford, California.

6. Security Considerations

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

- The tcpConnectionState and tcpConnState objects have a MAX-ACCESS clause of read-write, which allows termination of an arbitrary connection. Unauthorized access could cause a denial of service.

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:

- The tcpConnectionTable and the tcpConnTable contain objects providing information about the active connections on the device, the status of these connections, and the associated processes. This information may be used by an attacker to launch attacks against known/unknown weakness in certain protocols/applications. In addition, access to the connection table could also have privacy implications, as it provides detailed information on active connections.

- The tcpListenerTable and the tcpConnTable contain objects providing information about listeners on an entity. For example, the tcpListenerLocalPort and tcpConnLocalPort objects can be used to identify what ports are open on the machine and what attacks are likely to succeed, without the attacker having to run a port scanner.

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 module.

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

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, 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.

7. Contributors

This document is an output of the IPv6 MIB revision team, and contributors to earlier versions of this document include:

Bill Fenner, AT&T Labs -- Research
EMail: fenner@research.att.com

Brian Haberman
EMail: brian@innovationslab.net

Shawn A. Routhier, Wind River
EMail: shawn.routhier@windriver.com

Juergen Schoenwalder, TU Braunschweig
EMail: schoenw@ibr.cs.tu-bs.de

Dave Thaler, Microsoft
EMail: dthaler@windows.microsoft.com

This document updates parts of the MIBs from several documents. RFC 2012 has been the base document for these updates, and RFC 2452 was the first document to define the managed objects for implementations of TCP over IPv6.

RFC 2012:

Keith McCloghrie, Cisco Systems (Editor)
EMail: kzm@cisco.com

RFC 2452:

Mike Daniele, Compaq Computer Corporation
EMail: daniele@zk3.dec.com

Editor's Address

Rajiv Raghunarayan
Cisco Systems Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

EMail: raraghun@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgement

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