Network Working Group
Request for Comments: 2012
Updates: 1213
Category: Standards Track

K. McCloghrie, Editor
Cisco Systems

November 1996

SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2

Status of this Memo

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

IESG Note:

The IP, UDP, and TCP MIB modules currently support only IPv4. These three modules use the IpAddress type defined as an OCTET STRING of length 4 to represent the IPv4 32-bit internet addresses. (See RFC 1902, SMI for SNMPv2.) They do not support the new 128-bit IPv6 internet addresses.

Table of Contents

1. Introduction

A management system contains: several (potentially many) nodes, each with a processing entity, termed an agent, which has access to management instrumentation; at least one management station; and, a management protocol, used to convey management information between the agents and management stations. Operations of the protocol are carried out under an administrative framework which defines authentication, authorization, access control, and privacy policies.

Management stations execute management applications which monitor and control managed elements. Managed elements are devices such as hosts, routers, terminal servers, etc., which are monitored and controlled via access to their management information.

Management information is viewed as a collection of managed objects, residing in a virtual information store, termed the Management Information Base (MIB). Collections of related objects are defined in MIB modules. These modules are written using a subset of OSI's Abstract Syntax Notation One (ASN.1) [1], termed the Structure of Management Information (SMI) [2].

This document is the MIB module which defines managed objects for managing implementations of the Transmission Control Protocol (TCP) [3].

The managed objects in this MIB module were originally defined using the SNMPv1 framework as a part of MIB-II [4]. This document defines the same objects for TCP using the SNMPv2 framework.

2. Definitions

```
TCP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Integer32, Gauge32,
    Counter32, IpAddress, mib-2        FROM SNMPv2-SMI
    MODULE-COMPLIANCE, OBJECT-GROUP    FROM SNMPv2-CONF;

tcpMIB MODULE-IDENTITY
    LAST-UPDATED "9411010000Z"
    ORGANIZATION "IETF SNMPv2 Working Group"
    CONTACT-INFO
            "        Keith McCloghrie

             Postal: Cisco Systems, Inc.
                     170 West Tasman Drive
                     San Jose, CA  95134-1706
                     US

             Phone:  +1 408 526 5260
             Email:  kzm@cisco.com"
```

```
    DESCRIPTION
            "The MIB module for managing TCP implementations."
    REVISION      "9103310000Z"
    DESCRIPTION
            "The initial revision of this MIB module was part of MIB-
            II."
    ::= { mib-2 49 }

-- the TCP group

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

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 [5]
                }
    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
    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 upon the
            algorithm used to determine the retransmission timeout.  In
            particular, when the timeout algorithm is rsre(3), an object
            of this type has the semantics of the LBOUND quantity
            described in RFC 793."
    ::= { tcp 2 }

tcpRtoMax OBJECT-TYPE
    SYNTAX      Integer32
    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 upon the
            algorithm used to determine the retransmission timeout.  In
            particular, when the timeout algorithm is rsre(3), an object
            of this type has the semantics of the UBOUND quantity
            described in RFC 793."
    ::= { tcp 3 }

tcpMaxConn OBJECT-TYPE
    SYNTAX      Integer32
    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 TCP connections have made a direct
            transition to the SYN-SENT state from the CLOSED state."
    ::= { 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."
    ::= { tcp 6 }

tcpAttemptFails OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of times 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 TCP
            connections have made a direct transition to the LISTEN
            state from the SYN-RCVD state."
    ::= { tcp 7 }
```

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

```
    ::= { tcp 12 }


-- the TCP Connection table

-- The TCP connection table contains information about this
-- entity's existing TCP connections.

tcpConnTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF TcpConnEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table containing TCP connection-specific information."
    ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
    SYNTAX      TcpConnEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A conceptual row of the tcpConnTable 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   { tcpConnLocalAddress,
              tcpConnLocalPort,
              tcpConnRemAddress,
              tcpConnRemPort }
    ::= { tcpConnTable 1 }

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

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      current
    DESCRIPTION
            "The state of this TCP connection.

            The only value which 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 this has the effect of deleting the TCB
            (as defined in RFC 793) of the corresponding connection on
            the managed node, 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      current
    DESCRIPTION
            "The local IP address for this TCP connection.  In the case
            of a connection in the listen state which is 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      INTEGER (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The local port number for this TCP connection."
    ::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE
```

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

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

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)."
    ::= { tcp 14 }

tcpOutRsts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of TCP segments sent containing the RST flag."
    ::= { tcp 15 }

-- conformance information

tcpMIBConformance OBJECT IDENTIFIER ::= { tcpMIB 2 }

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


-- compliance statements

tcpMIBCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "The compliance statement for SNMPv2 entities which
            implement TCP."
    MODULE  -- this module
```

```
        MANDATORY-GROUPS { tcpGroup
                           }
    ::= { 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    current
    DESCRIPTION
            "The tcp group of objects providing for management of TCP
            entities."
    ::= { tcpMIBGroups 1 }

END
```

3. Acknowledgements

This document contains a modified subset of RFC 1213.

4. References

[1] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, (December, 1987).

[2] McCloghrie, K., Editor, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, Cisco Systems, January 1996.

[3] Postel, J., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", STD 7, RFC 793, DARPA, September 1981.

[4] McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, March 1991.

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

5. Security Considerations

Security issues are not discussed in this memo.

6. Editor's Address

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
US

Phone: +1 408 526 5260
EMail: kzm@cisco.com