

TCP/IPX Connection Mib Specification

Status of this Memo

This document defines an Experimental Protocol for the Internet community. This does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

IESG Note:

Internet Engineering Steering Group comment from the Area Director for Transport Services: Please note well that this memo is an individual product of the author. Implementation experience, particularly on the effectiveness of the protocols in dual-stack environments, is needed.

1. Introduction

Traditionally, TCP and UDP runs over IP. STD 17, RFC 1213 defines TCP connection MIB object and UDP listener object assuming just that. For TCP and UDP running over IPX, tcpConnTable and udpTable objects from RFC 1213 cannot be used since they define the address to be of type IpAddress. As such, we need to define new objects that can properly describe TCP and UDP connections over IPX.

New MIB objects, tcpIpxConnTable, udpIpxTable, tcpUnspecConnTable and udpUnspecTable are presented in this paper, to be used in place of tcpConnTable and udpListenerTable when TCP and UDP are running over IPX.

2. Objects

```
TCPIPX-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    OBJECT-TYPE
```

```
    FROM RFC-1212;
```

```
-- IPX address type.
```

```
-- First 4 octets are the network numbers and the last 6
```

```
-- octets are the node numbers. In ascii, it is represented
```

[illegible]

```

    tcpIpxConnState
        INTEGER,
    tcpIpxConnLocalAddress
        IpAddress
    tcpIpxConnLocalPort
        INTEGER (0..65535),
    tcpIpxConnRemAddress
        IpAddress,
    tcpIpxConnRemPort
        INTEGER (0..65535)
}

tcpIpxConnState 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)
    }
    ACCESS  read-write
    STATUS  mandatory
    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)."
```

::= { tcpIpConnEntry 1 }

tcpIpConnLocalAddress OBJECT-TYPE

```

    SYNTAX  IpAddress
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The local IPX address for this TCP connection.
        In the case of a connection in the listen state
        which is willing to accept connections for any
        interface, the value 00000000:000000000000 is
        used. See tcpUnspecConnTable for connections in
        the listen state which is willing to accept
        connects for any IP interface associated with
        the node."
```

::= { tcpIpConnEntry 2 }

```

-- NetworkAddress defined in SMI only include IP currently,
-- so we can't use it to represent both IP and IPX address.
```

tcpIpConnLocalPort OBJECT-TYPE

```

    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The local port number for this TCP connection."
```

::= { tcpIpConnEntry 3 }

tcpIpConnRemAddress OBJECT-TYPE

```

    SYNTAX  IpAddress
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The remote IPX address for this TCP connection."
```

::= { tcpIpConnEntry 4 }

tcpIpConnRemPort OBJECT-TYPE

```

    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The remote port number for this TCP connection."
```

::= { tcpIpConnEntry 5 }

```

-- the UDP Listener table

-- The UDP listener table contains information about this
-- entity's UDP end-points on which a local application is
-- currently accepting datagrams.

udpIpXTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF UdpIpXEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "A table containing UDP listener information."
    ::= { tcpXUdp 1 }

udpIpXEntry OBJECT-TYPE
    SYNTAX      UdpIpXEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "Information about a particular current UDP
        listener."
    INDEX       { udpIpXLocalAddress, udpIpXLocalPort }
    ::= { udpIpXTable 1 }

UdpIpXEntry ::=
    SEQUENCE {
        udpIpXLocalAddress
        IpXAddress
        udpIpXLocalPort
        INTEGER (0..65535)
    }

udpIpXLocalAddress OBJECT-TYPE
    SYNTAX      IpXAddress
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The local IPX address for this UDP listener.  In
        the case of a UDP listener which is willing to
        accept datagrams for any interface, the value
        00000000:000000000000 is used.  See
        udpUnspecTable for UDP listener which is
        willing to accept datagrams from any network
        layer."
    ::= { udpIpXEntry 1 }

udpIpXLocalPort OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)

```

```

ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "The local port number for this UDP listener."
 ::= { udpIpXEntry 2 }

-- the TCP/UNSPEC Connection table

-- The TCP/UNSPEC connection table contains information
-- about this entity's existing TCP connections over
-- unspecified network.
-- Since the network is unspecified, the network
-- address is also unspecified. Hence, this
-- connection table does not include any network
-- address.

tcpUnspecConnTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF TcpIpXConnEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "A table containing information specific on
         TCP connection over unspecified network layer."

    ::= { tcpXtcp 2 }

tcpUnspecConnEntry OBJECT-TYPE
    SYNTAX      TcpUnspecConnEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "Information about a particular current TCP
         connection over unspecified network layer.  An
         object of this type is transient, in that it
         ceases to exist when the connection makes
         transition beyond LISTEN state, or when (or
         soon after) the connection makes transition
         to the CLOSED state,"

    INDEX       { tcpUnspecConnLocalPort }
    ::= { tcpUnspecConnTable 1 }

TcpUnspecConnEntry ::=
    SEQUENCE {
        tcpUnspecConnState
            INTEGER,
        tcpUnspecConnLocalPort
    }

```

```

        INTEGER (0..65535),
    }

tcpUnspecConnState OBJECT-TYPE
    SYNTAX  INTEGER {
                closed(1),
                listen(2),
                deleteTCB(12)
            }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The state of this TCP connection.

        Since the TCP connection can belong to this table
        only when its state is less than SYN_SENT, only
        closed and listen state apply.

        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)."
```

::= { tcpUnspecConnEntry 1 }

```

tcpUnspecConnLocalPort OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The local port number for this TCP connection."
    ::= { tcpUnspecConnEntry 2 }
```

```

-- the UDP Listener table

-- The UDP listener table contains information about this
-- entity's UDP end-points over unspecified network layer,
-- on which a local application is currently accepting
-- datagrams.  If network layer is unspecified, the network
-- address is also unspecified.  Hence, this table does not
-- include any network address.

udpUnspecTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF UdpUnspecEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A table containing UDP listener information."
    ::= { tcpxUdp 2 }

udpUnspecEntry OBJECT-TYPE
    SYNTAX  UdpUnspecEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "Information about a particular current UDP
        listener."
    INDEX   { udpUnspecLocalPort }
    ::= { udpUnspecTable 1 }

UdpUnspecEntry ::=
    SEQUENCE {
        udpUnspecLocalPort
            INTEGER (0..65535)
    }

udpUnspecLocalPort OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The local port number for this UDP listener."
    ::= { udpUnspecEntry 1 }

END

```


Acknowledgement

The author would like to thank following folks and others for their assistance: Greg Minshall, Dave Piscitello.

Security Considerations

Security issues are not discussed in this memo.

Author's Address

Tae Sung
Novell, Inc.
2180 Fortune Drive
San Jose, California, 95131

Phone: (408)577-8439
EMail: tae@novell.Com

