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

TCP/IPX-MIB DEFINITIONS ::= BEGIN

IMPORTS

OBJECT-TYPE
   FROM RFC-1212;

-- IPX address type.
-- First 4 octests are the network numbers and the last 6
-- octests are the node numbers. In ascii, it is represented
-- as hex digits, as in: nnnnnnnn:mmmmmmmmmmmm

IpXAddress ::= OCTET STRING (size (10))

-- TCP/IPX MIB object identifiers

novell OBJECT IDENTIFIER ::= { enterprises 23 }
mibDoc OBJECT IDENTIFIER ::= { novell 2 }
tcppx OBJECT IDENTIFIER ::= { mibDoc 29 }
tcppxTcp OBJECT IDENTIFIER ::= { tcpx 1 }
tcppxUdp OBJECT IDENTIFIER ::= { tcpx 2 }

-- the TCP/IPX Connection table

-- The TCP/IPX connection table contains information
-- about this entity’s existing TCP connections over
-- IPX.

tcpIpxConnTable OBJECT-TYPE
SYNTAX  SEQUENCE OF TcpIpxConnEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
"A table containing information specific on
TCP connection over IPX network layer."
::= { tcpxTcp 1 }

tcpIpxConnEntry OBJECT-TYPE
SYNTAX  TcpIpxConnEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
"Information about a particular current TCP
connection over IPX An object of this type is
transient, in that it ceases to exist when (or
soon after) the connection makes the transition
to the CLOSED state."
INDEX  { tcpIpxConnLocalAddress,
tcpIpxConnLocalPort,
tcpIpxConnRemAddress,
tcpIpxConnRemPort }
::= { tcpIpxConnTable 1 }

TcpIpxConnEntry ::=  
SEQUENCE {

Sung

tcpIpxConnState
INTEGER,
tcpIpxConnLocalAddress
IpxAddress
tcpIpxConnLocalPort
INTEGER (0..65535),
tcpIpxConnRemAddress
IpxAddress,
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)."
::= { tcpIpxConnEntry 1 }

tcpIpxConnLocalAddress OBJECT-TYPE
SYNTAX  IpxAddress
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."
::= { tcpIpxConnEntry 2 }

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

tcpIpxConnLocalPort OBJECT-TYPE
SYNTAX  INTEGER (0..65535)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
"The local port number for this TCP connection."
::= { tcpIpxConnEntry 3 }

tcpIpxConnRemAddress OBJECT-TYPE
SYNTAX  IpxAddress
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
"The remote IPX address for this TCP connection."
::= { tcpIpxConnEntry 4 }

tcpIpxConnRemPort OBJECT-TYPE
SYNTAX  INTEGER (0..65535)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
"The remote port number for this TCP connection."
::= { tcpIpxConnEntry 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
}
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