8. Network Management

The management of NLSP™ routers uses SNMP running over the IPX™ protocol, as specified in [Wor92]. As an implementation option, the same management information can be accessed by SNMP running over IP or other Network layer protocol supported by the router.

An extensive suite of network management information has been defined for IPX, encompassing not only the IPX protocol itself but also RIP, SAP, and NLSP. The information has been defined in three separate Management Information Bases (MIBs): IPX, RIP/SAP, and NLSP.

The IPX MIB serves as the foundation for the other two MIBs. It contains the management information for the IPX protocol, as well as the common information supplied by both RIP/SAP and NLSP. The RIP/SAP and NLSP MIBs extend the groups and tables defined in the IPX MIB with their own protocol-specific information. Additionally, the NLSP MIB defines new groups that are not present in the IPX MIB.

Before the MIBs are described, you need to know how to use the MIBs in managing an NLSP routing area.

8.1. Managing an NLSP Area

Management stations can be implemented to manage an NLSP area. An NLSP management station would

- Identify potential network problems
- Isolate network faults when they occur
- Make network layer connectivity visible to the network administrator

The following subsections include guidelines for using the MIBs to implement a management station.

8.1.1. Examining Error Statistics and Counters

Within the IPX and NLSP MIBs are a series of error statistics and counters that help determine problems in an NLSP area.

If many packets are discarded because of excessive hop count (ipxAdvSysInTooManyHops), there is an error condition.

Also revealed by the MIBs are the number of routes (ipxAdvSysDestCount) and services (ipxAdvSysServCount) known. This number should be consistent between all routers in an NLSP area.

A rise in any of the following IPX MIB statistics indicates a network failure.

From the IPX system entry:

- Header Errors (ipxBasicSysInHdrErrors)
- Bad Checksums (ipxBasicSysInHdrErrors)

- Outbound Malformed Requests (ipxBasicSysInMalformedRequests)
- Open Sockets Fails (ipxBasicSysOpenSocketFails)

For IPX header compression:

- Compression Discards (ipxBasicSysInDiscards)
- Outbound Compression Discards (ipxBasicSysInOutDiscards)

The following NLSP MIB counters indicate potential problems:

- The number of corrupt LSPs detected (nlspSysCorrLSPs)

  If this value is anything other than zero, there is a problem. There are several reasons why this counter can increase:

  a) An NLSP router is generating corrupt LSPs.

  b) A data link is corrupting LSPs.

  c) LSPs are being corrupted after they are received.

  d) The system's NLSP software incorrectly believes that the LSP is corrupted, when it is not.

  Any one of these errors is very severe, and should be brought to the network administrator's attention immediately.

- The number of times the NLSP Level 1 LSP database has become overloaded (nlspSysL1DbaseOloads).

  Running out of memory is a severe problem because it compromises the ability of a system to perform the routing function. It is possible that this condition is transitory across the entire NLSP area. This might occur when a network administrator connects two areas, creating a routing area that requires too much memory. This problem can be fixed permanently by altering the routing area.

  A currently overloaded router should be brought to the network administrator's attention immediately. The number of times a router has been overloaded should also be monitored, so that the network administrator can distinguish between a transitory, correctable condition and the permanent condition of a router having insufficient memory.

- The number of times the router has attempted to exceed the NLSP maximum sequence number (nlspSysMaxSeqNums).

  This value indicates the number of times that this system's sequence number has wrapped. Because this condition should occur, at most, once every 160 years, a wrapping sequence number indicates an error condition. This error condition is probably internal to the router.

- The number of times a sequence number skip has occurred (nlspSysSeqNumSkips).

  This value indicates the number of times that another router has purged this router's LSPs. This can occur at startup under normal conditions, but thereafter this value should not increase. If this value is increasing, it is likely

that two systems have the same system ID. This is a severe error that can cause network wide disruption.

- The number of times a zero-aged copy of the router's own LSP has been received from some other node (nlspSysOwnLSPPurges).

  This value indicates the number of times that another router has originated this system's LSPs. Similar to nlspSysSeqNumSkips, this can occur normally at startup. Thereafter, it should not occur and should be treated as a severe error that can cause network wide disruption

The following NLSP MIB counters in the circuit group indicate problems:

- The number of times the NLSP LAN Level 1 Designated Router has changed on this circuit (nlspCircLANL1DesRouterChanges).

  This value can change normally over time, but each change causes a disruption in the network. If this value is rising rapidly (more than three times per day), there is a problem on the LAN segment. It can mean that routers are being restarted, or it can mean that the LAN segment is dropping a large amount of network traffic

- The number of times an NLSP neighbor state change has occurred on this circuit (nlspCircNeighChanges).

  This value can change normally over time, but each change causes a disruption in the network. If this value is rising rapidly (more than once every five minutes), there is a problem on the LAN segment.

- The number of times that an NLSP neighbor has been rejected on this circuit (nlspCircRejNeighbors).

  If this value is greater than zero, it is most likely that two NLSP systems have been incorrectly configured with nonmatching NLSP area addresses.

8.1.2. Using the Graph and the LSP Database to Locate Problems

Network topology information determined from a single NLSP system is adequate to describe the entire routing area. Information obtained from the graph table is sufficient to determine the network topology, the frame types in use, and the characteristics of WAN links. It can also be used to determine where RIP routes and SAP services are being injected. From the graph, a management station can construct a table of routers, obtain statistics from these routers, and maintain a record of the statistics.

Through periodic polling of the NLSP graph table of a single NLSP system, you can check for connectivity and quickly determine when and where links are down between systems. Broken links can then be flagged, and the frequency with which they (or LAN NLSP systems) are down can then be logged. Bouncing links and bouncing systems can also be determined by continuous polling of the graph table. Keeping a log of downed systems helps administrators to identify trouble spots in the network.

Duplicate network numbers can cause loss of connectivity. If two networks have the same network number, only one of them receives packets from any given router in the internetwork. By checking the graph table, it is possible to detect duplicate network numbers in an NLSP area. Recall, however, that zero is used as a null value (placeholder) for unnumbered RIP links. Expect multiple links having network number

zero. This is not a problem because zero is a reserved value, not used as a destination in routing decisions. It is also typical, for the network number of an external route to appear in more than one LSP.

Duplicate NLSP network numbers occur for one of two reasons:

- Incorrect configuration
- Data-link connectivity loss

In the latter case, a bridge or repeater fails, causing a network to split. Another possible cause of data-link connectivity loss is a failed network interface device on a router; the router might determine that it is the only router on a particular segment when, in fact, there are several other routers. Using the NLSP graph, it is possible to highlight a system that has a duplicate network number and provide the name of the system and the location of the system in the topology.

On the other hand, there are transient conditions that can cause a network to appear divided, when in fact it is not. When a duplicate network number is indicated by the NLSP graph, the network number should be flagged as a suspected duplicate. Checking again several minutes later eliminates typical transient conditions as a cause.

The graph tables of two different systems within a single NLSP area can be retrieved and compared for consistency. (Do not include expired LSPs in the check.) Within an NLSP area, two NLSP systems should have identical graphs except during transitions. So if system A knows of a link that system B does not, an error is indicated. You can also check using LSP headers. If system A has an LSP that system B does not, or if system A's LSP is newer, either

- B soon receives the LSP in question from A or
- There is an error

Two NLSP systems that are directly connected can be monitored to ensure that the LSP database is synchronizing. Slow convergence (propagation of the LSP database) can also be detected this way. Slow convergence can make a network inoperable, and should be brought to the network administrator's attention.

You can compare the potential paths reported by two systems to track down lost connectivity problems. For example, if system A indicates that it uses system B to reach a particular destination, system B should also have the remainder of the potential path. Otherwise, the network has not converged (which it should do within a minute), or there is a problem in the network preventing convergence between system A and system B (perhaps problems with the software).

8.1.3. Using the Neighbors Table and Statistics to Locate Problems

NLSP provides a two-way connectivity check intrinsically. However, it is possible that a system A might have only one-way connectivity to a system B. This occurs because of malfunctioning hardware or incorrect configuration. A neighbor that is continually in the "Initializing" state might be able to send packets on the indicated network, but not receive them.

On any given LAN, all systems should have the same Neighbors table (except that a system is not a neighbor of itself). If the Neighbors tables are inconsistent among

routers on a LAN, there might be a software failure. Another possible cause is insufficient processing, memory, or buffers in a router to process the information about its neighbors.

Finally, a frequently increasing number of either

- Designated Router changes (nlspCircLANL1DesRouterChanges) or
- Neighbor state changes (nlspCircNeighChanges)

indicates trouble. Perhaps a system on the network cannot process arriving Hellos, causing it to time out its neighbors. Perhaps a network has a physical connectivity problem (for example, excessive collisions or too much traffic). Perhaps a system's holding time is not high enough.

8.1.4. RIP and SAP Parameters That Must Be Consistent

RIP and SAP do not allow the same kinds of checks as NLSP. Still, it is possible to detect certain misconfigurations that cause connectivity loss. This kind of error is difficult for an administrator to isolate with current tools.

The following parameters must be identical for all systems on a LAN:

ripCircPacketSize

ripCircUpdate

sapCircPacketSize

sapCircUpdate

If any of these parameters are different, there might be a connectivity loss.

8.2. IPX Network Management Information

The IPX MIB defines the network management information for the IPX protocol and is intended for use by both routers and workstations that run IPX. Given the broad scope of this MIB, it is likely that there is some information that is not kept by some IPX implementations. In such cases, zero (or a similar value appropriate to the actual data type specified) should be returned for single items that are not supported. If the entire contents of a table are not supported, the table should be treated as containing no entries.

The primary index of every table in the IPX MIB is a system instance. This value identifies a unique instance of IPX running on a system. This system instance allows for a managed system to be running more than one instance of the IPX protocol at a time. This same instance value is used in the RIP/SAP and NLSP MIBs to associate their information with the information present in the IPX MIB. In cases where this value has read-write access in the MIB, it can be written only when adding a new entry to a table. If an attempt is made to change the system instance value of an existing entry in a table, the SNMP error code readOnly should be returned.

The IPX MIB defines five groups of network management information: the System group, the Circuit group, the Forwarding group, the Services group, and the Traps group. The groups are describe in more detail in the following sections.

8.2.1. System Group

The System group contains the general information about each instance of IPX running on a system. It contains information such as the system's IPX network number, counters of packets received and sent, and other such information. Two tables are defined in this group: the Basic System table and the Advanced System table.

Basic System Table

The Basic System table contains the basic information about an instance of the IPX protocol. There must be an entry in this table for each instance of IPX present on the managed system. In addition, given the basic nature of the information in this table, every attempt should be made to keep and return meaningful values for each item in the table.

Advanced System Table

The Advanced System table contains information that might not be kept by all IPX implementations (especially in the case of workstations), and it is acceptable for this table to contain no entries. However, much of the information in this table is applicable to IPX routers and should be kept by such systems.

8.2.2. Circuit Group

The Circuit group defines the management information that should be kept for each IPX circuit (a connection to a LAN, a WAN, or a direct link to another IPX system). The circuit information exists at a higher level than the interface information defined in MIB II. There might be more than one circuit associated with a single interface.

Some of the information contained in the Circuit table (the only table in the Circuit group) is applicable only to WAN circuits. On non-WAN circuits, zero (or a similar value appropriate to the item's data type) should be returned for such items.

Some of the items in the Circuit table have read-write access, but should only be written when adding a new circuit to the table. If an attempt is made to write new values for these items in an existing circuit, the SNMP error code readOnly should be returned. Items that can be written only when creating a new circuit contain the sentence "This value may be written only when creating a new entry in the table" in the DESCRIPTION portion of their MIB definition.

8.2.3. Forwarding Group

The Forwarding group contains the information about each IPX destination. The group contains two tables: the Destination table, and the Static Routes table.

Destination Table

The read-only Destination table contains information about every destination known to IPX. This table contains the minimum information that must be supplied by any IPX routing protocol. Meaningful values should be returned for every item in this table.

Static Routes Table

The Static Routes table is a subset of the destination that shows all static routes configured for an IPX system. The items in this table have read-write access so that static routes can be added or changed through SNMP requests.

8.2.4. Services Group

The Services group contains information about the known services. Three tables are defined in this group: the Services table, the Destination Services table, and the Static Services table.

Services Table

The Services table contains information about every service known to the system. It contains the basic information about the service that must be available, no matter what method was used to learn about the service (SAP, NLSP, static configuration, and so on). Meaningful values should be returned for every item in this table. The primary index (after the system instance, of course) for this table is the service type. This allows network management applications to easily show all services of a specific type that are available (for example, all the print servers) to a system.

Destination Services Table

The Destination Services table contains the same information kept in the Services table. There should be an entry in both of these tables for each service known to the system. The difference between the tables is in the order of the indexes. The Destination Services table is indexed in such a way that all the services provided by a specific destination can be obtained without having to read the entire table.

Static Services Table

The Static Services table is a subset of the Services and Destination Services tables that contains information about all of the services that have been statically configured for a system. This table has read-write access to allow the addition and deletion of statically configured services through network management.

8.2.5. Traps Group

Two enterprise-specific traps are defined in IPX MIB. One trap is used to notify a network management application when a new circuit comes up and the other for notification when a circuit goes down.

8.3. RIP and SAP Network Management Information

The RIP/SAP MIB defines the network management information for the RIP and SAP protocols. It is intended for use by all routers that support the two protocols. The RIP/SAP MIB extends the IPX MIB's System and Circuit groups with information that is specific to the two protocols. The tables in the RIP/SAP MIB are directly associated with the information in the IPX MIB via their primary index, the system instance.

8.3.1. System Group

The System group contains information about each instance of RIP and SAP that is running on the system. This group extends the information contained in the System group of the IPX MIB. This group contains two tables: the RIP System table and the SAP System table. For every entry in either of the two tables, there must be a corresponding entry in the IPX Basic System table with the same system instance value.

8.3.2. Circuit Group

The Circuit group extends the IPX MIB's Circuit group with additional information about RIP and SAP usage of IPX circuits. This group contains two tables: the RIP Circuit table and the SAP Circuit table. For every entry in either of the tables, there must be a corresponding entry in the IPX Circuit table with the same system instance and circuit index.

8.4. NLSP Network Management Information

The NLSP MIB defines the network management information for the NLSP routing protocol. It is intended for use by all NLSP routers. In addition to extending the IPX MIB's System, Circuit, and Forwarding Groups, the NLSP MIB also defines the NLSP Neighbors group, the NLSP Translation group, the NLSP Graph group, and the NLSP LSP group.

As with the other two MIBs, the primary index of every table is a system instance. The value of the system instance is used to link the NLSP information with the corresponding instance of IPX.

8.4.1. System Group

The NLSP System group contains information about each instance of NLSP running on a system. It contains information about the system's NLSP ID, the configurable NLSP parameters, as well as the area addresses. Three tables are defined in this group: the NLSP System table, the System Area Address table, and the Actual Area Address table.

NLSP System Table

The NLSP System table extends the IPX MIB's System table with global information for each instance of NLSP on the system. Among the numerous items contained in this table are the global configuration parameters for NLSP. The configuration parameters are defined with read-write access to allow the configuration of NLSP through network management. NLSP implementations that do not support configuration through network management should return the SNMP readOnly error code if an attempt is made to write a new value. NLSP implementations should be able to return meaningful values for all the items in this table. For every entry in this table, there must be a corresponding entry in the IPX Basic System table with the same system instance value.

System Area Address Table

The System Area Address table contains all the NLSP area addresses that were configured locally for this system.

Actual Area Address Table

The Actual Area Address table contains all the NLSP area addresses actually being used by this system.

8.4.2. Circuit Group

The NLSP Circuit group extends the IPX MIB's Circuit group with additional information about NLSP usage of IPX circuits. For every entry in the NLSP Circuit table

(the only table in this group) there must be an entry in the IPX Circuit table with the same system instance and circuit index.

8.4.3. Forwarding Group

The NLSP Forwarding group extends the Destination table from the IPX MIB's Forwarding group with additional information about those destinations that were learned through NLSP. For every entry in the NLSP Destination table, there must be an entry in the IPX Destination table with the same system instance and IPX network number.

8.4.4. NLSP Neighbors Group

The NLSP Neighbors group contains information about each neighboring NLSP router known to the system. This group consists of one table: the NLSP Neighbors table.

8.4.5. Translation Group

The Translation group contains mappings between NLSP system IDs, IPX network numbers, and router names. The group consists of three tables: the NLSP ID Mapping table, the IPX Network Number Mapping table, and the Name Mapping table.

NLSP ID Mapping Table

Given an NLSP system ID, this table allows the corresponding IPX network number and readable system name to be determined. Although every NLSP system ID always has a corresponding IPX network number, it is possible that there is no readable name associated with the ID. In such a case, a zero length string should be returned if the name is requested.

IPX Network Number Mapping Table

Given an IPX network number, this table allows the corresponding NLSP system ID and readable system name to be determined. While every IPX network number will have a corresponding NLSP system ID, it is possible that there will be no readable name associated with the network number. In such cases, a zero length string should be returned if the name is requested.

Name Mapping Table

Given a readable name for a system, this table allows the corresponding NLSP system ID and IPX network number to be determined. Both items are always known for every name in this table.

8.4.6. Graph Group

The Graph group gives a network management application access to the NLSP graph that represents the network topology. Using this group, a network management application can draw a representation of the network topology by querying only one NLSP router instead of all the routers in the network, identify which LANs are still running the RIP and SAP protocols, and show the potential paths a packet can take between systems.

The Graph group consists of five tables: the Node table, the Link table, the Path table, the Graph XRoutes table, and the Graph Services table. Each table is described in more detail next.

Node Table

The Node table contains an entry for each node in the NLSP graph. This table can be used to determine all the LANs, routers, and so on that exist in the network.

Link Table

The Link table contains an entry for each node's links. For every link between routers *X* and *Y* in the graph, there are actually two entries in this table: one representing the link from *X* to *Y* and the other from *Y* to *X*.

Path Table

The Path table can be used to determine the paths a data packet can take to reach a particular destination from the router reporting the information. The entries in this table provide the index values of the entries in the Link table that are used in the path a data packet takes. When a link's index appears in the path table, it means that link is on one of the least-cost paths from the router reporting the information. (There can be more than one when a router implements load-splitting.)

To find the paths a data packet can take from A to B, address the following network management queries to A:

a. Retrieve all entries in the Path table for B.

b. Using the values obtained from the Path table, obtain the corresponding entries from the Link table.

c. In the Link table, examine the neighbor reached from B via the link.

   c.1 If it is A, you have identified a path.

   c.2 If not, you have identified a router one step closer to A on the least-cost path.

d. For each "one step closer" router in Step c.2, repeat Steps a through c, using the "one step closer" router in place of B. With this iteration, links are concatenated into paths. The iteration is guaranteed to terminate because every step carries it closer to A.

Graph XRoutes Table

If a node in the graph is the source of one or more XRoutes, this table contains an entry for each of those XRoutes.

Graph Services Table

This table contains entries for the services provided by each node in the graph.

8.4.7. LSP Group

The LSP group provides access to the information in an NLSP router's LSP database. This is an optional group that consists of two tables: the LSP Header table and the LSP Options table.

LSP Header Table

The LSP header table contains summary information (lifetime, sequence number, checksum, and so on) about each LSP in the database, as well as an OCTET STRING containing the LSP header as it was received.

LSP Options Table

This table provides access to the options received in LSPs in a generic manner. There is an entry in this table for each option received in an LSP. This table is indexed by system instance (of course), the 8-byte LSP ID, and the ordinal position of the option within the LSP (starting with the value of 1 for the first option in each LSP). Entries in the table consist of the option code, length, and an OCTET STRING representing the value portion of the option. It is up to the network management application to actually decode the option's value.

8.5. MIB Definitions

The managed objects defined in the remainder of this section are presented in the concise MIB definition syntax specified in Reference [Ros91]. The same MIB definitions are also made available in electronic form. Every effort has been made to keep the material in this section identical to the electronic form, but in case of any discrepancy the electronic form is authoritative.

8.6. IPX MIB

```
IPX DEFINITIONS ::= BEGIN

-- This MIB defines the management information for a system using the IPX
-- protocol.  The MIB consists of four groups:
--
--   1.  System Group - contains general information about all instances
--                      of IPX on the system
--
--   2.  Circuit Group - contains information about all circuits used by
--                       IPX on the system
--
--   3.  Forwarding Group - contains generic routing information that
--                          must be provided by any IPX routing protocol.
--
--   4.  Services Group - contains information about all known services.
--
-- The MIB is designed to support multiple instances of the IPX
-- protocol on one system via a system instance identifier which is the
-- primary index for every table in this MIB.
--
-- This MIB is designed to provide a basic framework for the management
-- of systems implementing the IPX protocol.  Additional MIBs may be
-- created (especially in the area of IPX routing protocols) to contain
-- more specific information.  Whenever possible, these additional MIBs
-- should follow the format of this IPX MIB.  Information in these MIBs
-- should be linked to this MIB via the use of the system instance
-- identifier mentioned above.


IMPORTS
      enterprises, Counter
            FROM RFC1155-SMI
      OBJECT-TYPE
            FROM RFC-1212
      TRAP-TYPE
            FROM RFC-1215
      PhysAddress
            FROM RFC1213-MIB;

novell       OBJECT IDENTIFIER ::= { enterprises 23 }
mibDoc       OBJECT IDENTIFIER ::= { novell 2 }
ipx          OBJECT IDENTIFIER ::= { mibDoc 5 }

-- Groups

ipxSystem OBJECT IDENTIFIER ::= {ipx 1}
ipxCircuit OBJECT IDENTIFIER ::= {ipx 2}
ipxForwarding OBJECT IDENTIFIER ::= {ipx 3}
ipxServices OBJECT IDENTIFIER ::= {ipx 4}
ipxTraps OBJECT IDENTIFIER ::= {ipx 5}

-- Types

NetNumber ::= OCTET STRING (SIZE(4))

-- System Group
--   This group contains global information about each instance of IPX
--   running on one system.

-- Basic System Table
--   This table contains one entry for each instance of IPX running on
--   the system.  It contains the management information that should
--   be made available by all implementations of the IPX protocol.

ipxBasicSysTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF IPXBasicSysEntry
   ACCESS     not-accessible
   STATUS     mandatory
   DESCRIPTION "The IPX System table - basic information."
   ::= {ipxSystem 1}

ipxBasicSysEntry OBJECT-TYPE
   SYNTAX     IPXBasicSysEntry
   ACCESS     not-accessible
   STATUS     mandatory
   DESCRIPTION "Each entry corresponds to one instance of IPX running
               on the system."
   INDEX      {ipxBasicSysInstance}
   ::= {ipxBasicSysTable 1}

IPXBasicSysEntry ::= SEQUENCE {
                     ipxBasicSysInstance
                       INTEGER,
                     ipxBasicSysExistState
                       INTEGER,
                     ipxBasicSysNetNumber
                       OCTET STRING,
                     ipxBasicSysNode,
                       OCTET STRING,
                     ipxBasicSysName
                       OCTET STRING,
                     ipxBasicSysInReceives
                       Counter,
                     ipxBasicSysInHdrErrors
                       Counter,
                     ipxBasicSysInUnknownSockets
                       Counter,
                     ipxBasicSysInDiscards
                       Counter,
                     ipxBasicSysInBadChecksums
                       Counter,
                     ipxBasicSysInDelivers
                       Counter,
                     ipxBasicSysNoRoutes
                       Counter,
                     ipxBasicSysOutRequests
                       Counter,
                     ipxBasicSysOutMalformedRequests
                       Counter,
                     ipxBasicSysOutDiscards
                       Counter,
                     ipxBasicSysOutPackets
                       Counter,
                     ipxBasicSysConfigSockets
                       Counter,
                     ipxBasicSysOpenSocketFails
                       Counter
                   }

ipxBasicSysInstance OBJECT-TYPE
   SYNTAX     INTEGER
   ACCESS     read-write
   STATUS     mandatory
   DESCRIPTION "The unique identifier of the instance of IPX to which this
               row corresponds.  This value may be written only when
               creating a new entry in the table."
```

```
    ::= {ipxBasicSysEntry 1}

ipxBasicSysExistState OBJECT-TYPE
    SYNTAX      INTEGER {
                 off(1),
                 on(2)
                }
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The validity of this entry in the IPX system table.
                Setting this field to off indicates that this entry may be
                deleted from the system table at the IPX implementation's
                discretion."
    ::= {ipxBasicSysEntry 2}

ipxBasicSysNetNumber OBJECT-TYPE
    SYNTAX      NetNumber
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The network number portion of the IPX address of this
                system."
    ::= {ipxBasicSysEntry 3}

ipxBasicSysNode OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(6))
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The node number portion of the IPX address of this
                system."
    ::= {ipxBasicSysEntry 4}

ipxBasicSysName OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..48))
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The readable name for this system."
    ::= {ipxBasicSysEntry 5}

ipxBasicSysInReceives OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The total number of IPX packets received, including
                those received in error."
    ::= {ipxBasicSysEntry 6}

ipxBasicSysInHdrErrors OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of IPX packets discarded due to errors in
                their headers, including any IPX packet with a size less
                than the minimum of 30 bytes."
    ::= {ipxBasicSysEntry 7}

ipxBasicSysInUnknownSockets OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of IPX packets discarded because the
                destination socket was not open."
    ::= {ipxBasicSysEntry 8}

ipxBasicSysInDiscards OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of IPX packets received but discarded due to
                reasons other than those accounted for by
                ipxBasicSysInHdrErrors, ipxBasicSysInUnknownSockets,
                ipxAdvSysInDiscards, and ipxAdvSysInCompressDiscards."
    ::= {ipxBasicSysEntry 9}

ipxBasicSysInBadChecksums OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of IPX packets received with incorrect
                checksums."
    ::= {ipxBasicSysEntry 10}

ipxBasicSysInDelivers OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The total number of IPX packets delivered locally,
                including packets from local applications."
    ::= {ipxBasicSysEntry 11}

ipxBasicSysNoRoutes OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of times no route to a destination was
                found."
    ::= {ipxBasicSysEntry 12}

ipxBasicSysOutRequests OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of IPX packets supplied locally for
                transmission, not including any packets counted in
                ipxAdvForwPackets."
    ::= {ipxBasicSysEntry 13}

ipxBasicSysOutMalformedRequests OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of IPX packets supplied locally that contained
                errors in their structure."
    ::= {ipxBasicSysEntry 14}

ipxBasicSysOutDiscards OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of outgoing IPX packets discarded due to
                reasons other than those accounted for in
                ipxBasicSysOutMalformedRequests, ipxAdvSysOutFiltered,
                and ipxAdvSysOutCompressDiscards."
    ::= {ipxBasicSysEntry 15}

ipxBasicSysOutPackets OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The total number of IPX packets transmitted."
    ::= {ipxBasicSysEntry 16}
```

```
ipxBasicSysConfigSockets OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The configured maximum number of IPX sockets that may be
          open at one time."
  ::= {ipxBasicSysEntry 17}

ipxBasicSysOpenSocketFails OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of IPX socket open calls which failed."
  ::= {ipxBasicSysEntry 18}


-- Advanced System Table
--   This table contains one entry for each instance of IPX running on
--   the system.  It contains the advanced management information that
--   may not be available from all implementations of the IPX protocol.

ipxAdvSysTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF IPXAdvSysEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The IPX System table - advanced information."
  ::= {ipxSystem 2}

ipxAdvSysEntry OBJECT-TYPE
  SYNTAX     IPXAdvSysEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one instance of IPX running
          on the system."
  INDEX      {ipxAdvSysInstance}
  ::= {ipxAdvSysTable 1}

IPXAdvSysEntry ::= SEQUENCE {
                ipxAdvSysInstance
                  INTEGER,
                ipxAdvSysMaxPathSplits
                  INTEGER,
                ipxAdvSysMaxHops
                  INTEGER,
                ipxAdvSysInTooManyHops
                  Counter,
                ipxAdvSysInFiltered
                  Counter,
                ipxAdvSysInCompressDiscards
                  Counter,
                ipxAdvSysNETBIOSPackets
                  Counter,
                ipxAdvSysForwPackets
                  Counter,
                ipxAdvSysOutFiltered
                  Counter,
                ipxAdvSysOutCompressDiscards
                  Counter,
                ipxAdvSysCircCount
                  Counter,
                ipxAdvSysDestCount
                  Counter,
                ipxAdvSysServCount
                  Counter
                }

ipxAdvSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of IPX to which this
          row corresponds.  This value may be written only when
          creating a new entry in the table."
  ::= {ipxAdvSysEntry 1}

ipxAdvSysMaxPathSplits OBJECT-TYPE
  SYNTAX     INTEGER (1..32)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The maximum number of paths with equal routing metric
          value which this instance of the IPX may split
          between when forwarding packets."
  DEFVAL     { 1 }
  ::= {ipxAdvSysEntry 2}

ipxAdvSysMaxHops OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The maximum number of hops a packet may take."
  DEFVAL     { 64 }
  ::= {ipxAdvSysEntry 3}

ipxAdvSysInTooManyHops OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of IPX packets discarded due to exceeding the
          maximum hop count."
  ::= {ipxAdvSysEntry 4}

ipxAdvSysInFiltered OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of incoming IPX packets discarded due to
          filtering."
  ::= {ipxAdvSysEntry 5}

ipxAdvSysInCompressDiscards OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of incoming IPX packets discarded due to
          decompression errors."
  ::= {ipxAdvSysEntry 6}

ipxAdvSysNETBIOSPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of NETBIOS packets received."
  ::= {ipxAdvSysEntry 7}

ipxAdvSysForwPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of IPX packets forwarded."
```

```
    ::= {ipxAdvSysEntry 8}

ipxAdvSysOutFiltered OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of outgoing IPX packets discarded due to
                 filtering."
    ::= {ipxAdvSysEntry 9}

ipxAdvSysOutCompressDiscards OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of outgoing IPX packets discarded due to
                 compression errors."
    ::= {ipxAdvSysEntry 10}

ipxAdvSysCircCount OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of circuits known to this instance of IPX."
    ::= {ipxAdvSysEntry 11}

ipxAdvSysDestCount OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of currently reachable destinations known to
                 this instance of IPX."
    ::= {ipxAdvSysEntry 12}

ipxAdvSysServCount OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of services known to this instance of IPX."
    ::= {ipxAdvSysEntry 13}


-- Circuit Group
--   This group contains management information for each circuit known
--   to this system.

-- Circuit Table
--   The Circuit table contains an entry for each circuit known to the
--   system.

ipxCircTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IPXCircEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The Circuit table."
    ::= {ipxCircuit 1}

ipxCircEntry OBJECT-TYPE
    SYNTAX      IPXCircEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Each entry corresponds to one circuit known to the
                 system."
    INDEX       {
                 ipxCircSysInstance,
                 ipxCircIndex
                }
    ::= {ipxCircTable 1}

IPXCircEntry ::= SEQUENCE {
                 ipxCircSysInstance
                   INTEGER,
                 ipxCircIndex
                   INTEGER,
                 ipxCircExistState
                   INTEGER,
                 ipxCircOperState
                   INTEGER,
                 ipxCircIfIndex
                   INTEGER,
                 ipxCircName
                   OCTET STRING,
                 ipxCircType
                   INTEGER,
                 ipxCircDialName
                   OCTET STRING,
                 ipxCircLocalMaxPacketSize
                   INTEGER,
                 ipxCircCompressState
                   INTEGER,
                 ipxCircCompressSlots
                   INTEGER,
                 ipxCircStaticStatus
                   INTEGER,
                 ipxCircCompressedSent
                   Counter,
                 ipxCircCompressedInitSent
                   Counter,
                 ipxCircCompressedRejectsSent
                   Counter,
                 ipxCircUncompressedSent
                   Counter,
                 ipxCircCompressedReceived
                   Counter,
                 ipxCircCompressedInitReceived
                   Counter,
                 ipxCircCompressedRejectsReceived
                   Counter,
                 ipxCircUncompressedReceived
                   Counter,
                 ipxCircMediaType
                   INTEGER,
                 ipxCircNetNumber
                   NetNumber,
                 ipxCircStateChanges
                   Counter,
                 ipxCircInitFails
                   Counter,
                 ipxCircDelay
                   INTEGER,
                 ipxCircThroughput
                   INTEGER,
                 ipxCircNeighRouterName
                   OCTET STRING,
                 ipxCircNeighInternalNetNum
                   NetNumber
                }

ipxCircSysInstance OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-write
```

```
  STATUS      mandatory
  DESCRIPTION "The unique identifier of the instance of IPX
              to which this entry corresponds.  This value may be
              written only when creating a new entry in the table."
  ::= {ipxCircEntry 1}

ipxCircIndex OBJECT-TYPE
  SYNTAX      INTEGER
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The identifier of this circuit, unique within the
              instance of IPX.  This value may be written
              only when creating a new entry in the table."
  ::= {ipxCircEntry 2}

ipxCircExistState OBJECT-TYPE
  SYNTAX      INTEGER {
                off(1),
                on(2)
              }
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The validity of this circuit entry.  A circuit with this
              value set to off may be deleted from the table at the
              IPX implementation's discretion."
  ::= {ipxCircEntry 3}

ipxCircOperState OBJECT-TYPE
  SYNTAX      INTEGER {
                down(1),
                up(2),
                sleeping(3)
              }
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The operational state of the circuit."
  ::= {ipxCircEntry 4}

ipxCircIfIndex OBJECT-TYPE
  SYNTAX      INTEGER
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The value of ifIndex for the interface used by this
              circuit.  This value may be written only when creating
              a new entry in the table."
  ::= {ipxCircEntry 5}

ipxCircName OBJECT-TYPE
  SYNTAX      OCTET STRING (SIZE(0..48))
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The readable name for the circuit."
  ::= {ipxCircEntry 6}

ipxCircType OBJECT-TYPE
  SYNTAX      INTEGER {
                other(1),
                broadcast(2),
                ptToPt(3),
                wanRIP(4),
                unnumberedRIP(5),
                dynamic(6),
                wanWS(7)
              }
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The type of the circuit."
  ::= {ipxCircEntry 7}

ipxCircDialName OBJECT-TYPE
  SYNTAX      OCTET STRING (SIZE(0..48))
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The symbolic name used to refernce the dialing information
              used to create this circuit.  This value may be written
              only when creating a new entry in the table."
  ::= {ipxCircEntry 8}

ipxCircLocalMaxPacketSize OBJECT-TYPE
  SYNTAX      INTEGER
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The maximum size (including header), in bytes, that the
              system supports locally on this circuit."
  ::= {ipxCircEntry 9}

ipxCircCompressState OBJECT-TYPE
  SYNTAX      INTEGER {
                off(1),
                on(2)
              }
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The compression state on this circuit.  This value may
              be written only when creating a new entry in the table."
  DEFVAL      { off }
  ::= {ipxCircEntry 10}

ipxCircCompressSlots OBJECT-TYPE
  SYNTAX      INTEGER
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "The number of compression slots available on this
              circuit.  This value may be written only when creating a
              new entry in the table."
  DEFVAL      { 16 }
  ::= {ipxCircEntry 11}

ipxCircStaticStatus OBJECT-TYPE
  SYNTAX      INTEGER {
                unknown(1),
                current(2),
                changed(3),
                read(4),
                reading(5),
                write(6),
                writing(7)
              }
  ACCESS      read-write
  STATUS      mandatory
  DESCRIPTION "Indicates whether the information about static routes
              and services reached via this circuit matches that
              saved in permanent storage (current).  Setting the
              value to write when it had the value changed will write
              the currently in use information to permanent storage,
              if supported.  Setting the value to read when it had
              the value changed will replace any routes and services
              currently defined for the circuit with those read from
              permanent storage, if supported."
  ::= {ipxCircEntry 12}
```

```
ipxCircCompressedSent OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of compressed packets sent."
  ::= {ipxCircEntry 13}

ipxCircCompressedInitSent OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of compression initialization packets sent."
  ::= {ipxCircEntry 14}

ipxCircCompressedRejectsSent OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of compressed packet rejected packets sent."
  ::= {ipxCircEntry 15}

ipxCircUncompressedSent OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of packets sent without being compressed
          even though compression was turned on for this circuit."
  ::= {ipxCircEntry 16}

ipxCircCompressedReceived OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of compressed packets received."
  ::= {ipxCircEntry 17}

ipxCircCompressedInitReceived OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of compression initialization packets received."
  ::= {ipxCircEntry 18}

ipxCircCompressedRejectsReceived OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of compressed packet rejected packets received."
  ::= {ipxCircEntry 19}

ipxCircUncompressedReceived OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of packets received without having been
          compressed even though compression was turned on for
          this circuit."
  ::= {ipxCircEntry 20}

ipxCircMediaType OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(2))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The media type used on this circuit."
  ::= {ipxCircEntry 21}

ipxCircNetNumber OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The IPX network number of this circuit."
  ::= {ipxCircEntry 22}

ipxCircStateChanges OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times the circuit has changed state."
  ::= {ipxCircEntry 23}

ipxCircInitFails OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times that initialization of this
          circuit has failed."
  ::= {ipxCircEntry 24}

ipxCircDelay OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The period of time, in microseconds, that it takes to
          transmit one byte of data, excluding protocol headers,
          to a destination on the other end of the circuit, if
          the circuit is free of other traffic."
  ::= {ipxCircEntry 25}

ipxCircThroughput OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The amount of data, in bits per second, that may flow
          through the circuit if there is no other traffic."
  ::= {ipxCircEntry 26}

ipxCircNeighRouterName OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(0..48))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The name of the neighboring router on a WAN circuit."
  ::= {ipxCircEntry 27}

ipxCircNeighInternalNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The internal network number of the neighboring router
          on a WAN circuit."
  ::= {ipxCircEntry 28}


-- Forwarding Group
--   This group provides a representation of the forwarding database used
--   by all instances of IPX on the system.

-- Destination Table
--   The Destination table contains information about all known
--   destinations.  The routing information shown in this table represents
```

```
-- the path currently being used to reach the destination.

ipxDestTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF IPXDestEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The Destination table contains information about all
          known destinations."
  ::= {ipxForwarding 1}

ipxDestEntry OBJECT-TYPE
  SYNTAX     IPXDestEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one destination."
  INDEX      {
             ipxDestSysInstance,
             ipxDestNetNum
             }
  ::= {ipxDestTable 1}

IPXDestEntry ::= SEQUENCE {
                 ipxDestSysInstance
                   INTEGER,
                 ipxDestNetNum
                   NetNumber,
                 ipxDestProtocol
                   INTEGER,
                 ipxDestTicks
                   INTEGER,
                 ipxDestHopCount
                   INTEGER,
                 ipxDestNextHopCircIndex
                   INTEGER,
                 ipxDestNextHopNICAddress
                   PhysAddress,
                 ipxDestNextHopNetNum
                   NetNumber
                }

ipxDestSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of IPX
          to which this row corresponds."
  ::= {ipxDestEntry 1}

ipxDestNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The IPX network number of the destination."
  ::= {ipxDestEntry 2}

ipxDestProtocol OBJECT-TYPE
  SYNTAX     INTEGER {
                other(1),
                local(2),
                rip(3),
                nlsp(4),
                static(5)
                }
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The routing protocol from which knowledge of this
          destination was obtained."
  ::= {ipxDestEntry 3}

ipxDestTicks OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The delay in ticks to reach this destination."
  ::= {ipxDestEntry 4}

ipxDestHopCount OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of hops necessary to reach the destination."
  ::= {ipxDestEntry 5}

ipxDestNextHopCircIndex OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the circuit used to reach the
          next hop."
  ::= {ipxDestEntry 6}

ipxDestNextHopNICAddress OBJECT-TYPE
  SYNTAX     PhysAddress
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The NIC address of the next hop."
  ::= {ipxDestEntry 7}

ipxDestNextHopNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The IPX network number of the next hop."
  ::= {ipxDestEntry 8}


-- Static Routes Table
--  This table contains the information about all the static routes
--  defined.  There may be more than one static route to any given
--  destination.  Only the route currently being used will also be
--  present in the Destination Table defined above.

ipxStaticRouteTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF IPXStaticRouteEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The Static Routes table contains information about all
          destinations reached via statically configured routes."
  ::= {ipxForwarding 2}

ipxStaticRouteEntry OBJECT-TYPE
  SYNTAX     IPXStaticRouteEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one static route."
  INDEX      {
             ipxStaticRouteSysInstance,
             ipxStaticRouteCircIndex,
             ipxStaticRouteNetNum
             }
```

```
  ::= {ipxStaticRouteTable 1}

IPXStaticRouteEntry ::= SEQUENCE {
                    ipxStaticRouteSysInstance
                      INTEGER,
                    ipxStaticRouteCircIndex
                      INTEGER,
                    ipxStaticRouteNetNum
                      NetNumber,
                    ipxStaticRouteExistState
                      INTEGER,
                    ipxStaticRouteTicks
                      INTEGER,
                    ipxStaticRouteHopCount
                      INTEGER
                    }

ipxStaticRouteSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of IPX to
          which this row corresponds."
  ::= {ipxStaticRouteEntry 1}

ipxStaticRouteCircIndex OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the circuit used to
          reach the first hop in the static route."
  ::= {ipxStaticRouteEntry 2}

ipxStaticRouteNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The IPX network number of the route's destination."
  ::= {ipxStaticRouteEntry 3}

ipxStaticRouteExistState OBJECT-TYPE
  SYNTAX     INTEGER {
                 off(1),
                 on(2)
                }
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The validity of this static route.  Entries with the
          value set to off may be deleted from the table at the
          implementation's discretion."
  ::= {ipxStaticRouteEntry 4}

ipxStaticRouteTicks OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The delay, in ticks, to reach the route's destination."
  ::= {ipxStaticRouteEntry 5}

ipxStaticRouteHopCount OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The number of hops necessary to reach the destination."
  ::= {ipxStaticRouteEntry 6}

-- Services Group
--  The Services group contains management information for all known
--  services.

-- Services Table
--   This table contains the services information indexed by service
--   name and type.

ipxServTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF IPXServEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The table of services, indexed by name and type."
  ::= {ipxServices 1}

ipxServEntry OBJECT-TYPE
  SYNTAX     IPXServEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one service."
  INDEX      {
             ipxServSysInstance,
             ipxServType,
             ipxServName
             }
  ::= {ipxServTable 1}

IPXServEntry ::= SEQUENCE {
                  ipxServSysInstance
                    INTEGER,
                  ipxServType
                    OCTET STRING,
                  ipxServName
                    OCTET STRING,
                  ipxServProtocol
                    INTEGER,
                  ipxServNetNum
                    NetNumber,
                  ipxServNode
                    OCTET STRING,
                  ipxServSocket
                    OCTET STRING,
                  ipxServHopCount
                    INTEGER
                 }

ipxServSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of IPX
          to which this entry corresponds."
  ::= {ipxServEntry 1}

ipxServType OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(2))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The service type."
  ::= {ipxServEntry 2}

ipxServName OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(1..48))
```

```
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The service name."
  ::= {ipxServEntry 3}

ipxServProtocol OBJECT-TYPE
  SYNTAX      INTEGER {
                other(1),
                local(2),
                nlsp(4),
                static(5),
                sap(6)
              }
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The protocol from which knowledge of this service was
           obtained."
  ::= {ipxServEntry 4}

ipxServNetNum OBJECT-TYPE
  SYNTAX      NetNumber
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The IPX network number portion of the IPX address of the
           service."
  ::= {ipxServEntry 5}

ipxServNode OBJECT-TYPE
  SYNTAX      OCTET STRING (SIZE(6))
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The node portion of the IPX address of the service."
  ::= {ipxServEntry 6}

ipxServSocket OBJECT-TYPE
  SYNTAX      OCTET STRING (SIZE(2))
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The socket portion of the IPX address of the service."
  ::= {ipxServEntry 7}

ipxServHopCount OBJECT-TYPE
  SYNTAX      INTEGER
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The number of hops to the service."
  ::= {ipxServEntry 8}


-- Destination Services Table
--   This table contains the services information indexed by address,
--   name, and type.

ipxDestServTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF IPXDestServEntry
  ACCESS      not-accessible
  STATUS      mandatory
  DESCRIPTION "The table of services, indexed by address, name,
           and type."
  ::= {ipxServices 2}

ipxDestServEntry OBJECT-TYPE
  SYNTAX      IPXDestServEntry
  ACCESS      not-accessible
  STATUS      mandatory
  DESCRIPTION "Each entry corresponds to one service."
  INDEX       {
              ipxDestServSysInstance,
              ipxDestServNetNum,
              ipxDestServNode,
              ipxDestServSocket,
              ipxDestServName,
              ipxDestServType
             }
  ::= {ipxDestServTable 1}

IPXDestServEntry ::= SEQUENCE {
                     ipxDestServSysInstance
                       INTEGER,
                     ipxDestServNetNum
                       NetNumber,
                     ipxDestServNode
                       OCTET STRING,
                     ipxDestServSocket
                       OCTET STRING,
                     ipxDestServName
                       OCTET STRING,
                     ipxDestServType
                       OCTET STRING,
                     ipxDestServProtocol
                       INTEGER,
                     ipxDestServHopCount
                       INTEGER
                    }

ipxDestServSysInstance OBJECT-TYPE
  SYNTAX      INTEGER
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The unique identifier of the instance of IPX
           to which this entry corresponds."
  ::= {ipxDestServEntry 1}

ipxDestServNetNum OBJECT-TYPE
  SYNTAX      NetNumber
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The IPX network number portion of the IPX address of the
           service."
  ::= {ipxDestServEntry 2}

ipxDestServNode OBJECT-TYPE
  SYNTAX      OCTET STRING (SIZE(6))
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The node portion of the IPX address of the service."
  ::= {ipxDestServEntry 3}

ipxDestServSocket OBJECT-TYPE
  SYNTAX      OCTET STRING (SIZE(2))
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The socket portion of the IPX address of the service."
  ::= {ipxDestServEntry 4}

ipxDestServName OBJECT-TYPE
  SYNTAX      OCTET STRING (SIZE(1..48))
  ACCESS      read-only
  STATUS      mandatory
  DESCRIPTION "The service name."
```

```
  ::= {ipxDestServEntry 5}

ipxDestServType OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(2))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The service type."
  ::= {ipxDestServEntry 6}

ipxDestServProtocol OBJECT-TYPE
  SYNTAX     INTEGER {
               other(1),
               local(2),
               nlsp(4),
               static(5),
               sap(6)
             }
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The protocol from which knowledge of this service was
          obtained."
  ::= {ipxDestServEntry 7}

ipxDestServHopCount OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of hops to the service."
  ::= {ipxDestServEntry 8}


-- Static Services Table
--   This table contains information for all services reached via a
--   static route.

ipxStaticServTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF IPXStaticServEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The Static Services table contains information about
          all services reached via statically configured routes."
  ::= {ipxServices 3}

ipxStaticServEntry OBJECT-TYPE
  SYNTAX     IPXStaticServEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one service."
  INDEX      {
          ipxStaticServSysInstance,
          ipxStaticServCircIndex,
          ipxStaticServName,
          ipxStaticServType
         }
  ::= {ipxStaticServTable 1}

IPXStaticServEntry ::= SEQUENCE {
                    ipxStaticServSysInstance
                      INTEGER,
                    ipxStaticServCircIndex
                      INTEGER,
                    ipxStaticServName
                      OCTET STRING,
                    ipxStaticServType
                      OCTET STRING,
                    ipxStaticServExistState
                      INTEGER,
                    ipxStaticServNetNum
                      NetNumber,
                    ipxStaticServNode
                      OCTET STRING,
                    ipxStaticServSocket
                      OCTET STRING,
                    ipxStaticServHopCount
                      INTEGER
                   }

ipxStaticServSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of IPX to which
          this entry corresponds."
  ::= {ipxStaticServEntry 1}

ipxStaticServCircIndex OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The circuit used to reach this service."
  ::= {ipxStaticServEntry 2}

ipxStaticServName OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(1..48))
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The service name."
  ::= {ipxStaticServEntry 3}

ipxStaticServType OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(2))
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The service type."
  ::= {ipxStaticServEntry 4}

ipxStaticServExistState OBJECT-TYPE
  SYNTAX     INTEGER {
               off(1),
               on(2)
             }
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The validity of this static service.  Entries with the
          value set to off may be deleted from the table at the
          implementation's discretion."
  ::= {ipxStaticServEntry 5}

ipxStaticServNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The IPX network number portion of the IPX address of the
          service."
  ::= {ipxStaticServEntry 6}

ipxStaticServNode OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(6))
  ACCESS     read-write
  STATUS     mandatory
```

```
   DESCRIPTION "The node portion of the IPX address of the service."
   ::= {ipxStaticServEntry 7}

ipxStaticServSocket OBJECT-TYPE
   SYNTAX      OCTET STRING (SIZE(2))
   ACCESS      read-write
   STATUS      mandatory
   DESCRIPTION "The socket portion of the IPX address of the service."
   ::= {ipxStaticServEntry 8}

ipxStaticServHopCount OBJECT-TYPE
   SYNTAX      INTEGER
   ACCESS      read-write
   STATUS      mandatory
   DESCRIPTION "The number of hops to the service."
   ::= {ipxStaticServEntry 9}


-- Traps
--   The section describes the traps defined for IPX.

ipxTrapCircuitDown TRAP-TYPE
   ENTERPRISE    ipxTraps
   VARIABLES     {
                 ipxCircSysInstance,
                 ipxCircIndex
                }
   DESCRIPTION   "This trap signifies that the specified circuit has
                 gone down."
   ::= 1

ipxTrapCircuitUp TRAP-TYPE
   ENTERPRISE    ipxTraps
   VARIABLES     {
                 ipxCircSysInstance,
                 ipxCircIndex
                }
   DESCRIPTION   "This trap signifies that the specified circuit has
                 come up."
   ::= 2

END
```

8.7. RIP/SAP MIB

```
RIPSAP DEFINITIONS ::= BEGIN

-- This MIB defines the management information for the RIP and SAP
-- protocols running in an IPX enviroment.  It provides information in
-- addition to that contained in the IPX MIB itself.  All tables in this
-- MIB are linked to an instance of IPX via the system instance
-- identifier as defined in the IPX MIB.


IMPORTS
  enterprises, Counter
        FROM RFC1155-SMI
  OBJECT-TYPE
        FROM RFC-1212
  PhysAddress
        FROM RFC1213-MIB;

novell      OBJECT IDENTIFIER ::= {enterprises 23}
mibDoc      OBJECT IDENTIFIER ::= {novell 2}
ripsap      OBJECT IDENTIFIER ::= {mibDoc 20}


-- Groups

ripsapSystem  OBJECT IDENTIFIER ::= {ripsap 1}
ripsapCircuit OBJECT IDENTIFIER ::= {ripsap 2}


-- Types

NetNumber ::= OCTET STRING (SIZE(4))


-- System Group
--   This group contains global information about each instance of
--   RIP/SAP running on one system.

-- RIP System Table
--   This table contains an entry for each instance of RIP
--   running on the system.

ripSysTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF RIPSysEntry
   ACCESS     not-accessible
   STATUS     mandatory
   DESCRIPTION "The RIP system table."
   ::= {ripsapSystem 1}

ripSysEntry OBJECT-TYPE
   SYNTAX     RIPSysEntry
   ACCESS     not-accessible
   STATUS     mandatory
   DESCRIPTION "Each entry corresponds to one instance of RIP
            running on the system."
   INDEX      {ripSysInstance}
   ::= {ripSysTable 1}

RIPSysEntry ::= SEQUENCE {
                  ripSysInstance
                     INTEGER,
                  ripSysState
                     INTEGER,
                  ripSysIncorrectPackets
                     Counter
                  }

ripSysInstance OBJECT-TYPE
   SYNTAX     INTEGER
   ACCESS     read-write
   STATUS     mandatory
   DESCRIPTION "The unique identifier of the instance of RIP to
            which this row corresponds.  This value links the
            instance of RIP to an instance of IPX running on the
            system  (i.e. the value of ripSysInstance should be the
            same as a value of ipxSysInstance).  This value may be
            written only when creating a new entry in the table."
   ::= {ripSysEntry 1}

ripSysState OBJECT-TYPE
   SYNTAX     INTEGER {
                 off(1),
                 on(2)
                 }
   ACCESS     read-write
   STATUS     mandatory
   DESCRIPTION "Indicates the operational state of this instance of RIP."
   ::= {ripSysEntry 2}

ripSysIncorrectPackets OBJECT-TYPE
   SYNTAX     Counter
   ACCESS     read-only
   STATUS     mandatory
   DESCRIPTION "The number of times that an incorrectly formatted RIP
            packet was received."
   ::= {ripSysEntry 3}


-- SAP System Table
--   This table contains an entry for each instance of SAP
--   running on the system.

sapSysTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF SAPSysEntry
   ACCESS     not-accessible
   STATUS     mandatory
   DESCRIPTION "The SAP system table."
   ::= {ripsapSystem 2}

sapSysEntry OBJECT-TYPE
   SYNTAX     SAPSysEntry
   ACCESS     not-accessible
   STATUS     mandatory
   DESCRIPTION "Each entry corresponds to one instance of SAP
            running on the system."
   INDEX      {sapSysInstance}
   ::= {sapSysTable 1}

SAPSysEntry ::= SEQUENCE {
                  sapSysInstance
                     INTEGER,
                  sapSysState
                     INTEGER,
                  sapSysIncorrectPackets
                     Counter
                  }
```

```
sapSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of SAP to
           which this row corresponds.  This value links the
           instance of SAP to an instance of IPX running on the
           system  (i.e. the value of SApSysInstance should be the
           same as a value of ipxSysInstance).  This value may be
           written only when creating a new entry in the table."
  ::= {sapSysEntry 1}

sapSysState OBJECT-TYPE
  SYNTAX     INTEGER {
                off(1),
                on(2)
                }
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "Indicates the operational state of this instance of SAP."
  ::= {sapSysEntry 2}

sapSysIncorrectPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times that an incorrectly formatted SAP
           packet was received."
  ::= {sapSysEntry 3}


-- Circuit Group
--   This group contains RIP and SAP management information for each
--   circuit known to this system.

-- RIP Circuit Table
--   The RIP Circuit table contains an entry for the RIP information for
--   each circuit known to the system.

ripCircTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF RIPCircEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The RIP Circuit table."
  ::= {ripsapCircuit 1}

ripCircEntry OBJECT-TYPE
  SYNTAX     RIPCircEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one circuit known to the
           system."
  INDEX      {
           ripCircSysInstance,
           ripCircIndex
          }
  ::= {ripCircTable 1}

RIPCircEntry ::= SEQUENCE {
                  ripCircSysInstance
                    INTEGER,
                  ripCircIndex
                    INTEGER,
                  ripCircState
                    INTEGER,
                  ripCircPace
                    INTEGER,
                  ripCircUpdate
                    INTEGER,
                  ripCircAgeMultiplier
                    INTEGER,
                  ripCircPacketSize
                    INTEGER,
                  ripCircOutPackets
                    Counter,
                  ripCircInPackets
                    Counter
                  }

ripCircSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of RIP and IPX
           (via ipxSysInstance) to which this entry corresponds.
           This value may be written only when creating a new entry in
           the table."
  ::= {ripCircEntry 1}

ripCircIndex OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The identifier of this circuit, unique within the
           instance of RIP.  This value corresponds to the circuit
           identifier found in ipxCircIndex.  This value may be
           written only when creating a new entry in the table."
  ::= {ripCircEntry 2}

ripCircState OBJECT-TYPE
  SYNTAX     INTEGER {
                off(1),
                on(2),
                auto-on(3),
                auto-off(4)
                }
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "Indicates whether RIP information may be sent/received
           over this circuit."
  DEFVAL     { auto-off }
  ::= {ripCircEntry 3}

ripCircPace OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The maximum pace, in packets per second, at which RIP
           packets may be sent on this circuit."
  ::= {ripCircEntry 4}

ripCircUpdate OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The RIP periodic update interval, in seconds."
  DEFVAL     { 60 }
  ::= {ripCircEntry 5}

ripCircAgeMultiplier OBJECT-TYPE
```

```
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The holding multiplier for information received in RIP
          periodic updates."
  DEFVAL     { 4 }
  ::= {ripCircEntry 6}

ripCircPacketSize OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The RIP packet size used on this circuit."
  ::= {ripCircEntry 7}

ripCircOutPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of RIP packets sent on this circuit."
  ::= {ripCircEntry 8}

ripCircInPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of RIP packets received on this circuit."
  ::= {ripCircEntry 9}


-- SAP Circuit Table
--   The SAP Circuit table contains an entry for the SAP information for
--   each circuit known to the system.

sapCircTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF SAPCircEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The SAP Circuit table."
  ::= {ripsapCircuit 2}

sapCircEntry OBJECT-TYPE
  SYNTAX     SAPCircEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one circuit known to the
          system."
  INDEX      {
          sapCircSysInstance,
          sapCircIndex
         }
  ::= {sapCircTable 1}

SAPCircEntry ::= SEQUENCE {
                  sapCircSysInstance
                    INTEGER,
                  sapCircIndex
                    INTEGER,
                  sapCircState
                    INTEGER,
                  sapCircPace
                    INTEGER,
                  sapCircUpdate
                    INTEGER,
                  sapCircAgeMultiplier
                    INTEGER,
                  sapCircPacketSize
                    INTEGER,
                  sapCircGetNearestServerReply
                    INTEGER,
                  sapCircOutPackets
                    Counter,
                  sapCircInPackets
                    Counter
                 }

sapCircSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of SAP and IPX
          (via ipxSysInstance) to which this entry corresponds.
          This value may be written only when creating a new entry in
          the table."
  ::= {sapCircEntry 1}

sapCircIndex OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The identifier of this circuit, unique within the
          instance of SAP.  This value corresponds to the circuit
          identifier found in ipxCircIndex.   This value may be
          written only when creating a new entry in the table."
  ::= {sapCircEntry 2}

sapCircState OBJECT-TYPE
  SYNTAX     INTEGER {
               off(1),
               on(2),
               auto-on(3),
               auto-off(4)
              }
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "Indicates whether SAP information may be sent/received
          over this circuit."
  DEFVAL     { auto-off }
  ::= {sapCircEntry 3}

sapCircPace OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The maximum pace, in packets per second, at which SAP
          packets may be sent on this circuit."
  ::= {sapCircEntry 4}

sapCircUpdate OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The SAP periodic update interval, in seconds."
  DEFVAL     { 60 }
  ::= {sapCircEntry 5}

sapCircAgeMultiplier OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
```

```
  DESCRIPTION "The holding multiplier for information received in SAP
          periodic updates."
  DEFVAL     { 4 }
  ::= {sapCircEntry 6}

sapCircPacketSize OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The SAP packet size used on this circuit."
  ::= {sapCircEntry 7}

sapCircGetNearestServerReply OBJECT-TYPE
  SYNTAX     INTEGER {
                 no(1),
                 yes(2)
                }
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "Indicates whether to respond to SAP get nearest server
          requests received on this circuit."
  DEFVAL     { yes }
  ::= {sapCircEntry 8}

sapCircOutPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of SAP packets sent on this circuit."
  ::= {sapCircEntry 9}

sapCircInPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of SAP packets received on this circuit."
  ::= {sapCircEntry 10}

END
```

8.8. NLSP MIB

```
NLSP DEFINITIONS ::= BEGIN

-- This MIB defines the management information for the NLSP protocol
-- running in an IPX enviroment.  It provides information in addition
-- to that contained in the IPX MIB itself.  All tables in this MIB are
-- linked to an instance of IPX via the system instance identifier as
-- defined in the IPX MIB.

IMPORTS
  enterprises, Counter
        FROM RFC1155-SMI
  OBJECT-TYPE
        FROM RFC-1212
  PhysAddress
        FROM RFC1213-MIB;

novell  OBJECT IDENTIFIER ::= {enterprises 23}
mibDoc  OBJECT IDENTIFIER ::= { novell 2 }
nlsp    OBJECT IDENTIFIER ::= {mibDoc 19}


-- Groups

nlspSystem       OBJECT IDENTIFIER ::= {nlsp 1}
nlspCircuit      OBJECT IDENTIFIER ::= {nlsp 2}
nlspForwarding   OBJECT IDENTIFIER ::= {nlsp 3}
nlspNeighbors    OBJECT IDENTIFIER ::= {nlsp 4}
nlspTranslation  OBJECT IDENTIFIER ::= {nlsp 5}
nlspGraph        OBJECT IDENTIFIER ::= {nlsp 6}
nlspLSP          OBJECT IDENTIFIER ::= {nlsp 7}


-- Types

SystemID  ::= OCTET STRING (SIZE(6))
NLSPID    ::= OCTET STRING (SIZE(7))
NetNumber ::= OCTET STRING (SIZE(4))


-- System Group
--   This group contains global information about each instance of NLSP
--   running on one system.

-- System Table
--   This table contains an entry for each instance of NLSP running on
--   the system.

nlspSysTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF NLSPSysEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The NLSP system table."
  ::= {nlspSystem 1}

nlspSysEntry OBJECT-TYPE
  SYNTAX     NLSPSysEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one instance of NLSP running
           on the system."
  INDEX      {nlspSysInstance}
  ::= {nlspSysTable 1}

NLSPSysEntry ::= SEQUENCE {
                nlspSysInstance
                  INTEGER,
                nlspSysState
                  INTEGER,
                nlspSysID
                  SystemID,
                nlspSysMinNonBcastLSPTransInt
                  INTEGER,
                nlspSysMinBcastLSPTransInt
                  INTEGER,
                nlspSysMinLSPGenInt
                  INTEGER,
                nlspSysMaxLSPGenInt
                  INTEGER,
                nlspSysMaxLSPAge
                  INTEGER,
                nlspSysBcastHelloInt
                  INTEGER,
                nlspSysNonBcastHelloInt
                  INTEGER,
                nlspSysDRBcastHelloInt
                  INTEGER,
                nlspSysHoldTimeMultiplier
                  INTEGER,
                nlspSysCompSNPInt
                  INTEGER,
                nlspSysPartSNPInt
                  INTEGER,
                nlspSysWaitTime
                  INTEGER,
                nlspSysOrigL1LSPBufSize
                  INTEGER,
                nlspSysVersion
                  INTEGER,
                nlspSysCorrLSPs
                  Counter,
                nlspSysL1Overloaded
                  INTEGER,
                nlspSysL1DbaseOverloads
                  Counter,
                nlspSysMaxSeqNums
                  Counter,
                nlspSysSeqNumSkips
                  Counter,
                nlspSysTransmittedLSPs
                  Counter,
                nlspSysReceivedLSPs
                  Counter,
                nlspSysOwnLSPPurges
                  Counter,
                nlspSysVersionErrors
                  Counter,
                nlspSysIncorrectPackets
                  Counter,
                nlspSysNearestL2DefaultExists
                  INTEGER,
                nlspSysNearestL2DefaultRouter
                  SystemID,
                nlspSysResourceFailures
                  Counter,
                }

nlspSysInstance OBJECT-TYPE
```

```
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP to which this
          corresponds.  This value links the instance of NLSP to an
          instance of IPX running on the system  (i.e. the value
          of nlspSysInstance should be the same as a value of
          ipxSysInstance).  This value may be written only when
          creating a new entry in the table."
  ::= {nlspSysEntry 1}

nlspSysState OBJECT-TYPE
  SYNTAX     INTEGER {
               off(1),
               nlspLevel1Router(2)
             }
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "Indicates the operational state of this instance of NLSP."
  ::= {nlspSysEntry 2}

nlspSysID OBJECT-TYPE
  SYNTAX     SystemID
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The system ID for this instance of NLSP."
  ::= {nlspSysEntry 3}

nlspSysMinNonBcastLSPTransInt OBJECT-TYPE
  SYNTAX     INTEGER (1..30)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The minimum interval, in seconds, between transmission
          of LSPs on a non-broadcast circuit."
  DEFVAL     { 10 }
  ::= {nlspSysEntry 4}

nlspSysMinBcastLSPTransInt OBJECT-TYPE
  SYNTAX     INTEGER (1..30)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The minimum interval, in seconds, between transmission
          of LSPs on a broadcast circuit."
  DEFVAL     { 5 }
  ::= {nlspSysEntry 5}

nlspSysMinLSPGenInt OBJECT-TYPE
  SYNTAX     INTEGER (1..30)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The minimum interval, in seconds, between the generation
          of the same LSP."
  DEFVAL     { 5 }
  ::= {nlspSysEntry 6}

nlspSysMaxLSPGenInt OBJECT-TYPE
  SYNTAX     INTEGER (1..50000)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The maximum interval, in seconds, between the generation
          of the same LSP."
  DEFVAL     { 7200 }
  ::= {nlspSysEntry 7}

nlspSysMaxLSPAge OBJECT-TYPE
  SYNTAX     INTEGER (1..50000)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The value, in seconds, placed in the lifetime field of
          LSPs generated by this instance of NLSP."
  DEFVAL     { 7500 }
  ::= {nlspSysEntry 8}

nlspSysBcastHelloInt OBJECT-TYPE
  SYNTAX     INTEGER (1..100)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The interval, in seconds, at which NLSP Hellos will be
          sent on a broadcast circuit, if this system is not the
          designated router."
  DEFVAL     { 20 }
  ::= {nlspSysEntry 9}

nlspSysNonBcastHelloInt OBJECT-TYPE
  SYNTAX     INTEGER (1..100)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The interval, in seconds, at which NLSP Hellos will be
          sent on a non-broadcast circuit."
  DEFVAL     { 20 }
  ::= {nlspSysEntry 10}

nlspSysDRBcastHelloInt OBJECT-TYPE
  SYNTAX     INTEGER (1..100)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The interval, in seconds, at which the designated
          router sends NLSP Hellos on a broadcast circuit."
  DEFVAL     { 3 }
  ::= {nlspSysEntry 11}

nlspSysHoldTimeMultiplier OBJECT-TYPE
  SYNTAX     INTEGER (2..20)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The holding time multiplier used to specify the holding
          time for NLSP neighbor entries as a function of the NLSP
          Hello interval."
  DEFVAL     { 3 }
  ::= {nlspSysEntry 12}

nlspSysCompSNPInt OBJECT-TYPE
  SYNTAX     INTEGER (1..600)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The interval, in seconds, between generation of Complete
          Sequence Number Packets by a designated router on a
          broadcast circuit."
  DEFVAL     { 30 }
  ::= {nlspSysEntry 13}

nlspSysPartSNPInt OBJECT-TYPE
  SYNTAX     INTEGER (1..60)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The minimum interval, in seconds, between transmission of
          Partial Sequence Number Packets."
  DEFVAL     { 1 }
  ::= {nlspSysEntry 14}
```

```
nlspSysWaitTime OBJECT-TYPE
  SYNTAX     INTEGER (1..300)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The number of seconds to delay in the waiting state
          before entering the on state."
  DEFVAL     { 120 }
  ::= {nlspSysEntry 15}

nlspSysOrigL1LSPBufSize OBJECT-TYPE
  SYNTAX     INTEGER (512..4096)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The maximum size of Level 1 LSPs originated by
          this instance of NLSP."
  DEFVAL     { 512 }
  ::= {nlspSysEntry 16}

nlspSysVersion OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The version number of this instance of NLSP."
  ::= {nlspSysEntry 17}

nlspSysCorrLSPs OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of corrupt LSPs detected."
  ::= {nlspSysEntry 18}

nlspSysL1Overloaded OBJECT-TYPE
  SYNTAX     INTEGER {
               no(1),
               yes(2)
             }
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "Indicates whether the NLSP Level 1 database is overloaded."
  ::= {nlspSysEntry 19}

nlspSysL1DbaseOloads OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times the NLSP Level 1 LSP database has
          become overloaded."
  ::= {nlspSysEntry 20}

nlspSysMaxSeqNums OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times the router has attempted to exceed
          NLSP's maximum sequence number."
  ::= {nlspSysEntry 21}

nlspSysSeqNumSkips OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times a sequence number skip has occurred."
  ::= {nlspSysEntry 22}

nlspSysTransmittedLSPs OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of LSPs transmitted by this system."
  ::= {nlspSysEntry 23}

nlspSysReceivedLSPs OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of LSPs received by this system."
  ::= {nlspSysEntry 24}

nlspSysOwnLSPPurges OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times a zero-aged copy of the router's own
          LSP has been received from some other node."
  ::= {nlspSysEntry 25}

nlspSysVersionErrors OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times that a received NLSP packet was
          rejected because its version number was invalid."
  ::= {nlspSysEntry 26}

nlspSysIncorrectPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times that an incorrectly formatted NLSP
          packet was received."
  ::= {nlspSysEntry 27}

nlspSysNearestL2DefaultExists OBJECT-TYPE
  SYNTAX     INTEGER {
               no(1),
               yes(2)
             }
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "Indicates whether this instance of NLSP knows
          of a NLSP Level 2 router that currently can reach other
          areas using the default metric."
  ::= {nlspSysEntry 28}

nlspSysNearestL2DefaultRouter OBJECT-TYPE
  SYNTAX     SystemID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The system ID of the nearest NLSP Level 2 router that
          currently can reach other areas using the default
          metric.  The value is undefined if the value of
          nlspSysNearestL2DefaultExists is no."
  ::= {nlspSysEntry 29}

nlspSysResourceFailures OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times this instance of the NLSP has been
```

```
            unable to obtain needed resources (memory, etc.)"
    ::= {nlspSysEntry 30}


-- System Area Address Table
--   The System Area Address table contains the area addresses configured
--   for NLSP.

nlspSysAreaTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NLSPSysAreaEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The System Area Address table contains the area addresses
            configured for NLSP."
    ::= {nlspSystem 2}

nlspSysAreaEntry OBJECT-TYPE
    SYNTAX      NLSPSysAreaEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Each entry in the table corresponds to one NLSP
            System Area Address."
    INDEX       {
                nlspSysAreaSysInstance,
                nlspSysAreaNet,
                nlspSysAreaMask
                }
    ::= {nlspSysAreaTable 1}

NLSPSysAreaEntry ::= SEQUENCE {
                        nlspSysAreaSysInstance
                          INTEGER,
                        nlspSysAreaNet
                          OCTET STRING,
                        nlspSysAreaMask
                          OCTET STRING
                        }

nlspSysAreaSysInstance OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The unique identifier of the instance of NLSP and IPX
            (via ipxSysInstance) to which this row corresponds."
    ::= {nlspSysAreaEntry 1}

nlspSysAreaNet OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(4))
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The network address portion of the area address."
    ::= {nlspSysAreaEntry 2}

nlspSysAreaMask OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(4))
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The mask portion of the area address."
    ::= {nlspSysAreaEntry 3}


-- Actual Area Address Table
--   The Actual Area Address table contains the area addresses actually
--   used by NLSP.

nlspActAreaTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NLSPActAreaEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The Actual Area Address table contains the area addresses
            actually used by NLSP."
    ::= {nlspSystem 3}

nlspActAreaEntry OBJECT-TYPE
    SYNTAX      NLSPActAreaEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Each entry in the table corresponds to one NLSP
            Actual Area Address."
    INDEX       {
                nlspActAreaSysInstance,
                nlspActAreaNet,
                nlspActAreaMask
                }
    ::= {nlspActAreaTable 1}

NLSPActAreaEntry ::= SEQUENCE {
                        nlspActAreaSysInstance
                          INTEGER,
                        nlspActAreaNet
                          OCTET STRING,
                        nlspActAreaMask
                          OCTET STRING
                        }

nlspActAreaSysInstance OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The unique identifier of the instance of NLSP and IPX
            (via ipxSysInstance) to which this row corresponds."
    ::= {nlspActAreaEntry 1}

nlspActAreaNet OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(4))
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The network address portion of the area address."
    ::= {nlspActAreaEntry 2}

nlspActAreaMask OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(4))
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The mask portion of the area address."
    ::= {nlspActAreaEntry 3}


-- Circuit Group
--   This group contains the NLSP information for each circuit known
--   to this system.

-- Circuit Table
--   The Circuit table contains an entry containing the NLSP information
--   for each circuit known to the system.

nlspCircTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NLSPCircEntry
    ACCESS      not-accessible
    STATUS      mandatory
```

```
  DESCRIPTION "The Circuit table."
  ::= {nlspCircuit 1}

nlspCircEntry OBJECT-TYPE
  SYNTAX     NLSPCircEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one circuit known to the
          system."
  INDEX      {
          nlspCircSysInstance,
          nlspCircIndex
          }
  ::= {nlspCircTable 1}

NLSPCircEntry ::= SEQUENCE {
                  nlspCircSysInstance
                    INTEGER,
                  nlspCircIndex
                    INTEGER,
                  nlspCircState
                    INTEGER,
                  nlspCircPace
                    INTEGER,
                  nlspCircHelloTimer
                    INTEGER,
                  nlspCircL1DefaultCost
                    INTEGER,
                  nlspCircL1DesRouterPriority
                    INTEGER,
                  nlspCircL1CircID
                    OCTET STRING,
                  nlspCircL1DesRouter
                    SystemID,
                  nlspCircLANL1DesRouterChanges
                    Counter,
                  nlspCircNeighChanges
                    Counter,
                  nlspCircRejNeighbors
                    Counter,
                  nlspCircOutPackets
                    Counter,
                  nlspCircInPackets
                    Counter,
                  nlspCircActualMaxPacketSize
                    INTEGER,
                  nlspCircPSNPsSent
                    Counter,
                  nlspCircPSNPsReceived
                    Counter
                  }

nlspCircSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP and IPX
          (via ipxSysInstance) to which this entry corresponds.
          This value may be written only when creating a new
          entry in the table."
  ::= {nlspCircEntry 1}

nlspCircIndex OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The identifier of this circuit, unique within the
          instance of NLSP.  This value may be written
          only when creating a new entry in the table."
  ::= {nlspCircEntry 2}

nlspCircState OBJECT-TYPE
  SYNTAX     INTEGER {
               off(1),
               on(2)
             }
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "Indicates whether NLSP information may be sent/received
          over this circuit."
  DEFVAL     { on }
  ::= {nlspCircEntry 3}

nlspCircPace OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The maximum pace, in packets per second, at which NLSP
          packets may be sent on this circuit."
  ::= {nlspCircEntry 4}

nlspCircHelloTimer OBJECT-TYPE
  SYNTAX     INTEGER (1..100)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The interval, in seconds, between NLSP Hello packets
          sent on this circuit."
  DEFVAL     { 20 }
  ::= {nlspCircEntry 5}

nlspCircL1DefaultCost OBJECT-TYPE
  SYNTAX     INTEGER (1..63)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The NLSP default cost of this circuit for Level 1
          traffic."
  ::= {nlspCircEntry 6}

nlspCircL1DesRouterPriority OBJECT-TYPE
  SYNTAX     INTEGER (1..127)
  ACCESS     read-write
  STATUS     mandatory
  DESCRIPTION "The priority for becoming the NLSP LAN Level 1
          Designated Router on a broadcast circuit."
  ::= {nlspCircEntry 7}

nlspCircL1CircID OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(7))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The NLSP ID for this circuit."
  ::= {nlspCircEntry 8}

nlspCircL1DesRouter OBJECT-TYPE
  SYNTAX     SystemID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The system ID of the NLSP LAN Level 1 Designated Router
          on this circuit."
  ::= {nlspCircEntry 9}
```

```
nlspCircLANL1DesRouterChanges OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times the NLSP LAN Level 1 Designated
          Router has changed on this circuit."
  ::= {nlspCircEntry 10}

nlspCircNeighChanges OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times a NLSP neighbor state change has
          occurred on this circuit."
  ::= {nlspCircEntry 11}

nlspCircRejNeighbors OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of times that a NLSP neighbor has been
          rejected on this circuit."
  ::= {nlspCircEntry 12}

nlspCircOutPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of NLSP packets sent on this circuit."
  ::= {nlspCircEntry 13}

nlspCircInPackets OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of NLSP packets received on this circuit."
  ::= {nlspCircEntry 14}

nlspCircActualMaxPacketSize OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The actual maximum packet size (including header),
          in bytes, that has been used on this circuit."
  ::= {nlspCircEntry 15}

nlspCircPSNPsSent OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of PSNPs sent on this circuit."
  ::= {nlspCircEntry 16}

nlspCircPSNPsReceived OBJECT-TYPE
  SYNTAX     Counter
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of PSNPs received on this circuit."
  ::= {nlspCircEntry 17}


-- Forwarding Group
--   This group contains NLSP forwarding information in addition to that
--   contained in the IPX forwarding group.

-- Destination Table
--   The Destination table contains additional NLSP forwarding
--   information about all destinations learned about via NLSP.

nlspDestTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF NLSPDestEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The Destination table contains information about all
          known destinations learned about via NLSP."
  ::= {nlspForwarding 1}

nlspDestEntry OBJECT-TYPE
  SYNTAX     NLSPDestEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one destination."
  INDEX      {
             nlspDestSysInstance,
             nlspDestNetNum
             }
  ::= {nlspDestTable 1}

NLSPDestEntry ::= SEQUENCE {
                  nlspDestSysInstance
                    INTEGER,
                  nlspDestNetNum
                    NetNumber,
                  nlspDestID
                    NLSPID,
                  nlspDestEstDelay
                    INTEGER,
                  nlspDestEstThroughput
                    INTEGER,
                  nlspDestNextHopID
                    NLSPID,
                  nlspDestCost
                    INTEGER
                  }

nlspDestSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP and IPX
          (via ipxSysInstance) to which this row corresponds."
  ::= {nlspDestEntry 1}

nlspDestNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The IPX network number of the destination."
  ::= {nlspDestEntry 2}

nlspDestID OBJECT-TYPE
  SYNTAX     NLSPID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The destination NLSP ID (6-octet system ID plus 1-octet
          pseudo-node ID)."
  ::= {nlspDestEntry 3}

nlspDestEstDelay OBJECT-TYPE
```

```
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The estimated delay, in microseconds, to reach the
                 destination."
    ::= {nlspDestEntry 4}

nlspDestEstThroughput OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The estimated throughput, in bits per second, to the
                 destination."
    ::= {nlspDestEntry 5}

nlspDestNextHopID OBJECT-TYPE
    SYNTAX      NLSPID
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The NLSP ID (6-octet system ID plus 1-octet pseudo-node
                 ID) of the next hop."
    ::= {nlspDestEntry 6}

nlspDestCost OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The total path default cost to reach this destination."
    ::= {nlspDestEntry 7}


-- NLSP Neighbors Group
--   This group contains management information for each neighboring
--   NLSP router known to the system.

-- NLSP Neighbors Table
--   This table contains an entry for each neighboring NLSP router
--   known to the system.

nlspNeighTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NLSPNeighEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The NLSP Neighbors table."
    ::= {nlspNeighbors 1}

nlspNeighEntry OBJECT-TYPE
    SYNTAX      NLSPNeighEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Each entry corressponds to one neighboring NLSP router
                 known to the system."
    INDEX       {
                  nlspNeighSysInstance,
                  nlspNeighCircIndex,
                  nlspNeighIndex
                }
    ::= {nlspNeighTable 1}

NLSPNeighEntry ::= SEQUENCE {
                     nlspNeighSysInstance
                       INTEGER,
                     nlspNeighCircIndex
                       INTEGER,
                     nlspNeighIndex
                       INTEGER,
                     nlspNeighState
                       INTEGER,
                     nlspNeighNICAddress
                       PhysAddress,
                     nlspNeighSysType
                       INTEGER,
                     nlspNeighSysID
                       SystemID,
                     nlspNeighName
                       OCTET STRING,
                     nlspNeighUsage
                       INTEGER,
                     nlspNeighHoldTimer
                       INTEGER,
                     nlspNeighRemainingTime
                       INTEGER,
                     nlspNeighPriority
                       INTEGER
                     }

nlspNeighSysInstance OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The unique identifier of the instance of NLSP and IPX
                 (via ipxSysInstance) to which this row corresponds."
    ::= {nlspNeighEntry 1}

nlspNeighCircIndex OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The identifier of the parent circuit of this neighbor
                 within this instance of the NLSP and IPX."
    ::= {nlspNeighEntry 2}

nlspNeighIndex OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The identifier for this NLSP neighbor entry, unique
                 within the parent circuit."
    ::= {nlspNeighEntry 3}

nlspNeighState OBJECT-TYPE
    SYNTAX      INTEGER {
                  initializing(1),
                  up(2),
                  failed(3),
                  down(4)
                }
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The state of the connection to the neighboring NLSP
                 router."
    ::= {nlspNeighEntry 4}

nlspNeighNICAddress OBJECT-TYPE
    SYNTAX      PhysAddress
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The NIC Address of the neighboring NLSP router."
    ::= {nlspNeighEntry 5}
```

```
nlspNeighSysType OBJECT-TYPE
  SYNTAX     INTEGER {
               unknown(1),
               nlspLevel1Router(2)
             }
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The type of the neighboring NLSP router."
  ::= {nlspNeighEntry 6}

nlspNeighSysID OBJECT-TYPE
  SYNTAX     SystemID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The neighboring NLSP router's system ID."
  ::= {nlspNeighEntry 7}

nlspNeighName OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(0..48))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The readable name for the neighboring NLSP router."
  ::= {nlspNeighEntry 8}

nlspNeighUsage OBJECT-TYPE
  SYNTAX     INTEGER {
               undefined(1),
               level1(2)
             }
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The usage of the connection to the neighboring NLSP
          router."
  ::= {nlspNeighEntry 9}

nlspNeighHoldTimer OBJECT-TYPE
  SYNTAX     INTEGER (1..65535)
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The initial holding time, in seconds, for this NLSP
          neighbor entry as specified in the NLSP Hello packet."
  ::= {nlspNeighEntry 10}

nlspNeighRemainingTime OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The remaining time to live, in seconds, for this NLSP
          neighbor entry."
  ::= {nlspNeighEntry 11}

nlspNeighPriority OBJECT-TYPE
  SYNTAX     INTEGER (1..127)
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The priority of the neighboring NLSP router for
          becoming the LAN Level 1 Designated router if the value
          of nlspNeighSysType is nlspLevel1Router."
  ::= {nlspNeighEntry 12}


-- Translation Group
--   The translation group contains tables providing mappings between
--   network numbers, NLSP system IDs, and router names.

-- NLSP ID Mapping Table
--   This table maps NLSP system IDs to router names and IPX network
--   numbers.

nlspIDMapTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF NLSPIDMapEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "This table maps NLSP system IDs to router names and
          IPX network numbers."
  ::= {nlspTranslation 1}

nlspIDMapEntry OBJECT-TYPE
  SYNTAX     NLSPIDMapEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry maps one NLSP system ID to its corresponding
          router name and IPX network number."
  INDEX      {
          nlspIDMapSysInstance,
          nlspIDMapID
         }
  ::= {nlspIDMapTable 1}

NLSPIDMapEntry ::= SEQUENCE {
                     nlspIDMapSysInstance
                       INTEGER,
                     nlspIDMapID
                       NLSPID,
                     nlspIDMapServerName
                       OCTET STRING,
                     nlspIDMapNetNum
                       NetNumber
                    }

nlspIDMapSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP and IPX
          (via ipxSysInstance) to which this row corresponds."
  ::= {nlspIDMapEntry 1}

nlspIDMapID OBJECT-TYPE
  SYNTAX     NLSPID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The NLSP ID (6-octet sytem ID plus the pseudo-node ID)."
  ::= {nlspIDMapEntry 2}

nlspIDMapServerName OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(0..48))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The readable name corresponding to this NLSP ID."
  ::= {nlspIDMapEntry 3}

nlspIDMapNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The IPX network number corresponding to this NLSP ID."
  ::= {nlspIDMapEntry 4}
```

```
-- IPX Network Number Mapping Table
--   This table maps IPX network numbers to router names and NLSP IDs.

nlspNetMapTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF NLSPNetMapEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "This table maps IPX network numbers to router names
          and NLSP IDs."
  ::= {nlspTranslation 2}

nlspNetMapEntry OBJECT-TYPE
  SYNTAX     NLSPNetMapEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry maps one IPX network number to its
          corresponding router name and NLSP ID."
  INDEX      {
          nlspNetMapSysInstance,
          nlspNetMapNetNum
         }
  ::= {nlspNetMapTable 1}

NLSPNetMapEntry ::= SEQUENCE {
                    nlspNetMapSysInstance
                      INTEGER,
                    nlspNetMapNetNum,
                      NetNumber,
                    nlspNetMapServerName
                      OCTET STRING,
                    nlspNetMapID,
                      NLSPID
                   }

nlspNetMapSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP and IPX
          (via ipxSysInstance) to which this row corresponds."
  ::= {nlspNetMapEntry 1}

nlspNetMapNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The IPX network number."
  ::= {nlspNetMapEntry 2}

nlspNetMapServerName OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(0..48))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The router name corresponding to the IPX network number."
  ::= {nlspNetMapEntry 3}

nlspNetMapID OBJECT-TYPE
  SYNTAX     NLSPID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The NLSP ID corresponding to the IPX network number."
  ::= {nlspNetMapEntry 4}


-- Name Mapping Table
--   This table maps router names to their corresponding IPX network
--   number and NLSP ID.

nlspNameMapTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF NLSPNameMapEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "This table maps router names to the corresponding IPX
          network number and NLSP ID."
  ::= {nlspTranslation 3}

nlspNameMapEntry OBJECT-TYPE
  SYNTAX     NLSPNameMapEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry maps one router name to its corresponding
          IPX network number and NLSP ID."
  INDEX      {
          nlspNameMapSysInstance,
          nlspNameMapServerName
         }
  ::= {nlspNameMapTable 1}

NLSPNameMapEntry ::= SEQUENCE {
                    nlspNameMapSysInstance
                      INTEGER,
                    nlspNameMapServerName
                      OCTET STRING,
                    nlspNameMapNetNum
                      NetNumber,
                    nlspNameMapID
                      NLSPID
                   }

nlspNameMapSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP and IPX
          (via ipxSysInstance) to which this row corresponds."
  ::= {nlspNameMapEntry 1}

nlspNameMapServerName OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(0..48))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The readable name for this system."
  ::= {nlspNameMapEntry 2}

nlspNameMapNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The IPX network number corresponding to the router name."
  ::= {nlspNameMapEntry 3}

nlspNameMapID OBJECT-TYPE
  SYNTAX     NLSPID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The NLSP ID corresponding to the router name.  This value
          is undefined if the value of nlspSysState is off."
  ::= {nlspNameMapEntry 4}
```

```
-- Graph Group
--   The Graph group provides a representation of the network topology.
--   The group is optional.

-- Node Table
--   The node table contains an entry for each node in the graph.

nlspNodeTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF NLSPNodeEntry
   ACCESS      not-accessible
   STATUS      mandatory
   DESCRIPTION "The node table contains an entry for each node in the
               graph."
   ::= {nlspGraph 1}

nlspNodeEntry OBJECT-TYPE
   SYNTAX      NLSPNodeEntry
   ACCESS      not-accessible
   STATUS      mandatory
   DESCRIPTION "Each entry corresponds to one graph node."
   INDEX       {
               nlspNodeSysInstance,
               nlspNodeID
               }
   ::= {nlspNodeTable 1}

NLSPNodeEntry ::= SEQUENCE {
                  nlspNodeSysInstance
                     INTEGER,
                  nlspNodeID
                     NLSPID,
                  nlspNodeNetNum
                     NetNumber,
                  nlspNodeType
                     INTEGER,
                  nlspNodeEstDelay
                     INTEGER,
                  nlspNodeEstThroughput
                     INTEGER,
                  nlspNodeMaxPacketSize
                     INTEGER,
                  nlspNodeCost
                     INTEGER,
                  nlspNodeOverload
                     INTEGER,
                  nlspNodeReachable
                     INTEGER
                  }

nlspNodeSysInstance OBJECT-TYPE
   SYNTAX      INTEGER
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "The unique identifier of the instance of NLSP and IPX
               (via ipxSysInstance) to which this row corresponds."
   ::= {nlspNodeEntry 1}

nlspNodeID OBJECT-TYPE
   SYNTAX      NLSPID
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "The NLSP ID for this node."
   ::= {nlspNodeEntry 2}

nlspNodeNetNum OBJECT-TYPE
   SYNTAX      NetNumber
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "The IPX network number of this node."
   ::= {nlspNodeEntry 3}

nlspNodeType OBJECT-TYPE
   SYNTAX      INTEGER {
                  unknown(1),
                  nlspLevel1Router(2),
                  nlspLevel2Router(3),
                  router(4),
                  network(5)
                  }
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "The type of system the node represents."
   ::= {nlspNodeEntry 4}

nlspNodeEstDelay OBJECT-TYPE
   SYNTAX      INTEGER
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "The estimated delay, in microseconds, to reach the
               destination represented by this node."
   ::= {nlspNodeEntry 5}

nlspNodeEstThroughput OBJECT-TYPE
   SYNTAX      INTEGER
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "The estimated throughput, in bits per second, to the
               destination represented by this node."
   ::= {nlspNodeEntry 6}

nlspNodeMaxPacketSize OBJECT-TYPE
   SYNTAX      INTEGER
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "The maximum packet size, in bytes, that can be sent to
               the destination represented by this node."
   ::= {nlspNodeEntry 7}

nlspNodeCost OBJECT-TYPE
   SYNTAX      INTEGER
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "The cost to reach this node."
   ::= {nlspNodeEntry 8}

nlspNodeOverload OBJECT-TYPE
   SYNTAX      INTEGER {
                  no(1),
                  yes(2)
                  }
   ACCESS      read-only
   STATUS      mandatory
   DESCRIPTION "Indicates whether this node is overloaded."
   ::= {nlspNodeEntry 9}

nlspNodeReachable OBJECT-TYPE
   SYNTAX      INTEGER {
                  no(1),
                  yes(2)
                  }
```

```
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Indicates whether the destination represented by this
                 node is reachable."
    ::= {nlspNodeEntry 10}


-- Link Table
--   This table contains the entries for all of the links in the graph.

nlspLinkTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NLSPLinkEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The Link table contains entries for all of the links in
                 the graph."
    ::= {nlspGraph 2}

nlspLinkEntry OBJECT-TYPE
    SYNTAX      NLSPLinkEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Each entry corresponds to one link."
    INDEX       {
                 nlspLinkSysInstance,
                 nlspLinkNLSPID,
                 nlspLinkIndex
                }
    ::= {nlspLinkTable 1}

NLSPLinkEntry ::= SEQUENCE {
                    nlspLinkSysInstance
                      INTEGER,
                    nlspLinkNLSPID
                      NLSPID,
                    nlspLinkIndex
                      INTEGER,
                    nlspLinkNeighNLSPID
                      NLSPID,
                    nlspLinkFromNeighCost
                      INTEGER,
                    nlspLinkMaxPacketSize
                      INTEGER,
                    nlspLinkThroughput
                      INTEGER,
                    nlspLinkDelay
                      INTEGER,
                    nlspLinkMediaType
                      OCTET STRING,
                    nlspLinkToNeighCost
                      INTEGER
                   }

nlspLinkSysInstance OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The unique identifier of the instance of NLSP and IPX
                 (via ipxSysInstance) to which this row corresponds."
    ::= {nlspLinkEntry 1}

nlspLinkNLSPID OBJECT-TYPE
    SYNTAX      NLSPID
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The NLSP ID (6-byte system ID plus 1-octet pseudo-node
                 ID) of the node to which this link belongs."
    ::= {nlspLinkEntry 2}

nlspLinkIndex OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The unique value identifying the link within the node."
    ::= {nlspLinkEntry 3}

nlspLinkNeighNLSPID OBJECT-TYPE
    SYNTAX      NLSPID
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The NLSP ID (6-byte system ID plus 1-octet pseudo-node
                 ID) of the neighboring node."
    ::= {nlspLinkEntry 4}

nlspLinkFromNeighCost OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The cost to use this link to reach this node from
                 the neighboring node."
    ::= {nlspLinkEntry 5}

nlspLinkMaxPacketSize OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The maximum size, in bytes, of a packet that may be sent
                 over this link."
    ::= {nlspLinkEntry 6}

nlspLinkThroughput OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The link's maximum throughput, in bits per second."
    ::= {nlspLinkEntry 7}

nlspLinkDelay OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The delay, in microseconds, on this link."
    ::= {nlspLinkEntry 8}

nlspLinkMediaType OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(2))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The media type of this link."
    ::= {nlspLinkEntry 9}

nlspLinkToNeighCost OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The cost to use this link to reach the neighbor from
                 this node."
    ::= {nlspLinkEntry 10}
```

```
-- Path Table
--   This table allows the path(s) that a packet may take to reach a
--   destination to be reconstructed.  The entries in this table
--   represent those links that are one hop closer to the source and
--   would be used for the minimum cost path(s) to reach the
--   destination.

nlspPathTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF NLSPPathEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The path table."
  ::= {nlspGraph 3}

nlspPathEntry OBJECT-TYPE
  SYNTAX     NLSPPathEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each row in this table represents a link to a node that
          is one hop closer to the source and would be used for
          the minimum cost path(s) to reach the destination."
  INDEX      {
          nlspPathSysInstance,
          nlspPathDestNLSPID,
          nlspPathLinkIndex
          }
  ::= {nlspPathTable 1}

NLSPPathEntry ::= SEQUENCE {
                  nlspPathSysInstance
                    INTEGER,
                  nlspPathDestNLSPID
                    NLSPID,
                  nlspPathLinkIndex
                    INTEGER
                 }

nlspPathSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP and IPX
          (via ipxSysInstance) to which this row corresponds."
  ::= {nlspPathEntry 1}

nlspPathDestNLSPID OBJECT-TYPE
  SYNTAX     NLSPID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The NLSP ID (6-octet system ID plus 1-octet pseudo-node
          ID) of this destination."
  ::= {nlspPathEntry 2}

nlspPathLinkIndex OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique value identifying this link within the
          destination node."
  ::= {nlspPathEntry 3}


-- Graph XRoutes Table
--   This table contains information about all of the XRoutes provided by
--   a node in the graph.

nlspGraphXRouteTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF NLSPGraphXRouteEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "This table contains the information about the XRoutes
          associated with a node in the graph."
  ::= {nlspGraph 4}

nlspGraphXRouteEntry OBJECT-TYPE
  SYNTAX     NLSPGraphXRouteEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry in the table contains the information for one
          XRoute associated with the node."
  INDEX      {
          nlspGraphXRouteSysInstance,
          nlspGraphXRouteNLSPID,
          nlspGraphXRouteNetNum
          }
  ::= {nlspGraphXRouteTable 1}

NLSPGraphXRouteEntry ::= SEQUENCE {
                  nlspGraphXRouteSysInstance
                    INTEGER,
                  nlspGraphXRouteNLSPID
                    NLSPID,
                  nlspGraphXRouteNetNum
                    NetNumber,
                  nlspGraphXRouteCost
                    INTEGER,
                  nlspGraphXRouteHopCount
                    INTEGER
                 }

nlspGraphXRouteSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP and IPX
          (via ipxSysInstance) to which this entry corresponds."
  ::= {nlspGraphXRouteEntry 1}

nlspGraphXRouteNLSPID OBJECT-TYPE
  SYNTAX     NLSPID
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The NLSP ID of the node."
  ::= {nlspGraphXRouteEntry 2}

nlspGraphXRouteNetNum OBJECT-TYPE
  SYNTAX     NetNumber
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The IPX network number of the XRoute's destination."
  ::= {nlspGraphXRouteEntry 3}

nlspGraphXRouteCost OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The cost to reach the XRoute's destination."
  ::= {nlspGraphXRouteEntry 4}

nlspGraphXRouteHopCount OBJECT-TYPE
```

```
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The number of hops necessary to reach the XRoute's
                destination."
    ::= {nlspGraphXRouteEntry 5}


-- Graph Services Table
--   This table contains information about all of the services provided by
--   a node in the graph.

nlspGraphServTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NLSPGraphServEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "This table contains the information about the services
                associated with a node in the graph."
    ::= {nlspGraph 5}

nlspGraphServEntry OBJECT-TYPE
    SYNTAX      NLSPGraphServEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Each entry in the table contains the information for one
                service associated with the node."
    INDEX       {
                nlspGraphServSysInstance,
                nlspGraphServNLSPID,
                nlspGraphServName,
                nlspGraphServTypeValue
                }
    ::= {nlspGraphServTable 1}

NLSPGraphServEntry ::= SEQUENCE {
                    nlspGraphServSysInstance
                      INTEGER,
                    nlspGraphServNLSPID
                      NLSPID,
                    nlspGraphServName
                      OCTET STRING,
                    nlspGraphServTypeValue
                      OCTET STRING,
                    nlspGraphServType
                      INTEGER,
                    nlspGraphServNetNum
                      NetNumber,
                    nlspGraphServNode
                      OCTET STRING,
                    nlspGraphServSocket
                      OCTET STRING
                    }

nlspGraphServSysInstance OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The unique identifier of the instance of NLSP and IPX
                (via ipxSysInstance) to which this entry corresponds."
    ::= {nlspGraphServEntry 1}

nlspGraphServNLSPID OBJECT-TYPE
    SYNTAX      NLSPID
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The NLSP ID of the node."
    ::= {nlspGraphServEntry 2}

nlspGraphServName OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(1..48))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The service name."
    ::= {nlspGraphServEntry 3}

nlspGraphServTypeValue OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(2))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The service type's hexadecimal value."
    ::= {nlspGraphServEntry 4}

nlspGraphServType OBJECT-TYPE
    SYNTAX      INTEGER {
                 unknown(1)
                }
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The service type."
    ::= {nlspGraphServEntry 5}

nlspGraphServNetNum OBJECT-TYPE
    SYNTAX      NetNumber
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The IPX network number portion of the IPX address of
                the service."
    ::= {nlspGraphServEntry 6}

nlspGraphServNode OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(6))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The node portion of the IPX address of the service."
    ::= {nlspGraphServEntry 7}

nlspGraphServSocket OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(2))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The socket portion of the IPX address of the service."
    ::= {nlspGraphServEntry 8}


-- LSP Group
--   The LSP group provides a representation of NLSP's LSP database.  This
--   group is optional.

-- LSP Header Table
--    The LSP header table contains summary information about each LSP
--    in the database as well as an OCTET STRING containing the entire
--    LSP header.

nlspLSPTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NLSPLSPEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The LSP header table."
    ::= {nlspLSP 1}
```

```
nlspLSPEntry OBJECT-TYPE
  SYNTAX     NLSPLSPEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one LSP's header."
  INDEX      {
             nlspLSPSysInstance,
             nlspLSPID
             }
  ::= {nlspLSPTable 1}

NLSPLSPEntry ::= SEQUENCE {
                  nlspLSPSysInstance
                    INTEGER,
                  nlspLSPID
                    OCTET STRING,
                  nlspLSPLifetime
                    INTEGER,
                  nlspLSPSeqNum
                    INTEGER,
                  nlspLSPChecksum
                    INTEGER,
                  nlspLSPRouterType
                    INTEGER,
                  nlspLSPOverload
                    INTEGER,
                  nlspLSPHeader
                    OCTET STRING
                 }

nlspLSPSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier for the instance of NLSP and IPX
           (via ipxSysInstance) to which this entry corresponds."
  ::= {nlspLSPEntry 1}

nlspLSPID OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(8))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The value that uniquely identifies this LSP."
  ::= {nlspLSPEntry 2}

nlspLSPLifetime OBJECT-TYPE
  SYNTAX     INTEGER (0..65535)
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The number of seconds prior to the expiration of the
           LSP."
  ::= {nlspLSPEntry 3}

nlspLSPSeqNum OBJECT-TYPE
  SYNTAX     INTEGER (0..255)
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The sequence number of the LSP."
  ::= {nlspLSPEntry 4}

nlspLSPChecksum OBJECT-TYPE
  SYNTAX     INTEGER (0..65535)
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The checksum value of the LSP."
  ::= {nlspLSPEntry 5}

nlspLSPRouterType OBJECT-TYPE
  SYNTAX     INTEGER {
                unknown(1),
                nlspLevel1Router(2)
                }
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The type of the router that sent the LSP."
  ::= {nlspLSPEntry 6}

nlspLSPOverload OBJECT-TYPE
  SYNTAX     INTEGER {
                no(1),
                yes(2)
                }
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "Indicates whether the sending router's LSP database is
           overloaded."
  ::= {nlspLSPEntry 7}

nlspLSPHeader OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(27))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The complete LSP header."
  ::= {nlspLSPEntry 8}


-- LSP Options Table
--   The LSP options table is used to obtain each option contained in
--   an LSP.

nlspLSPOptTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF NLSPLSPOptEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "The LSP Options table."
  ::= {nlspLSP 2}

nlspLSPOptEntry OBJECT-TYPE
  SYNTAX     NLSPLSPOptEntry
  ACCESS     not-accessible
  STATUS     mandatory
  DESCRIPTION "Each entry corresponds to one option from an LSP."
  INDEX      {
             nlspLSPOptSysInstance,
             nlspLSPOptLSPID,
             nlspLSPOptIndex
             }
  ::= {nlspLSPOptTable 1}

NLSPLSPOptEntry ::= SEQUENCE {
                     nlspLSPOptSysInstance
                       INTEGER,
                     nlspLSPOptLSPID
                       OCTET STRING,
                     nlspLSPOptIndex
                       INTEGER,
                     nlspLSPOptCode
                       INTEGER,
                     nlspLSPOptLength
                       INTEGER,
```

```
                    nlspLSPOptValue
                      OCTET STRING
                    }

nlspLSPOptSysInstance OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The unique identifier of the instance of NLSP and IPX
           (via ipxSysInstance) to which this entry corresponds."
  ::= {nlspLSPOptEntry 1}

nlspLSPOptLSPID OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(8))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The value that uniquely identifies the LSP."
  ::= {nlspLSPOptEntry 2}

nlspLSPOptIndex OBJECT-TYPE
  SYNTAX     INTEGER
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The value that uniquely identifies this option within the
           LSP."
  ::= {nlspLSPOptEntry 3}

nlspLSPOptCode OBJECT-TYPE
  SYNTAX     INTEGER (0..255)
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The code that identifies the type of the option."
  ::= {nlspLSPOptEntry 4}

nlspLSPOptLength OBJECT-TYPE
  SYNTAX     INTEGER (0..255)
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The length of the option's value field."
  ::= {nlspLSPOptEntry 5}

nlspLSPOptValue OBJECT-TYPE
  SYNTAX     OCTET STRING (SIZE(0..255))
  ACCESS     read-only
  STATUS     mandatory
  DESCRIPTION "The option's value field."
  ::= {nlspLSPOptEntry 6}

END
```