2. Basic Concepts

The goal of routing is to convey data traffic from the source end node to the intended destination. Routing methods are implemented in special network nodes called *routers*, which are connected to two or more network segments of like or unlike kind. The router forwards traffic from segment to segment. A given packet might have to be forwarded several times. To make good decisions, routers exchange information with each other about the topology of the internetwork. Several approaches to the information exchange have been used in the industry. The NLSP™ specification uses the*link state* approach. With this approach, every router keeps a road map representing the states of the links and routers in a routing area.

2.1. Link State Databases and Algorithms

The method is called link state because routers keep track of the state of every communication link in a routing area. The router keeps a record of all the routers in the area, the links connecting them, the operational status of the routers and links, and related parameters. NLSP operates within a *routing area* of links and routers. Figure 2-1 illustrates a simple example. Keep the example in mind as the discussion turns to databases, protocols, and algorithms.

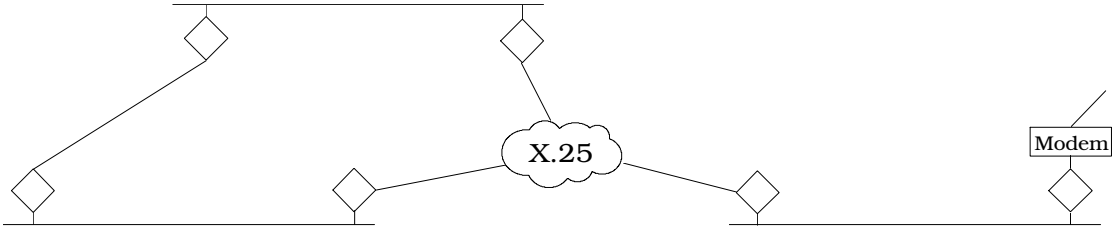


Figure 2-1: Example Internetwork

Figure 2-2 illustrates the data exchanges among routers, the information databases each router maintains, and the processing steps to accomplish routing functions.

In the figure, a horizontal line represents a LAN, an oblique line represents a point-to-point link, and the cloud in the center represents a packet data network using X.25. The diamond-shaped symbols represent routers. End nodes, such as workstations, personal computers, and nonrouting servers, are not shown in the diagram.

Each router maintains a database describing the entire area. For each point-to-point link, the database records the end point routers and the state of the link. For each LAN, the database records the routers connected to the LAN. The X.25 network is modeled as a set of point-to-point links that go up and down as calls are established and cleared. Likewise for the dial-up modem link.

2.1.1. Maintaining Adjacencies between Routers and their Neighbors

The first database, "Adjacency," keeps track of the router's immediate neighbors and the operational status of the directly attached links. Regular exchanges of *Hello* messages maintain this information with each neighbor. When a circuit is enabled, the router begins a periodic transmission of Hello messages and listens for the messages from its neighbors. When the router detects (through these exchanges) that the link is reliably operational, it enters a record of the neighbor (or neighbors) in the Adjacency database. If the exchange is interrupted, the router updates the Adjacency database to reflect this fact. This way, it keeps track of its immediate neighborhood.

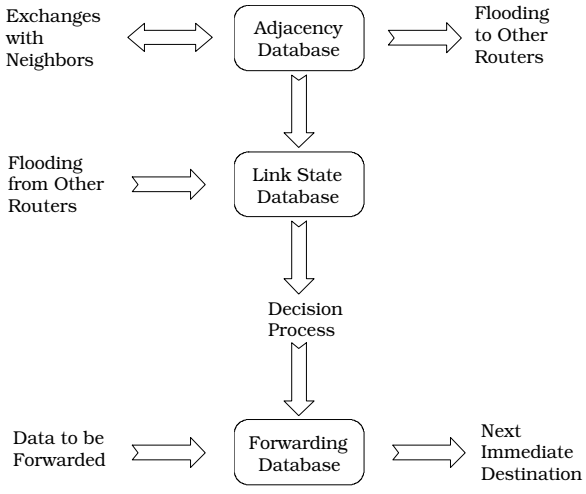


Figure 2-2: NLSP Overview

2.1.2. Synchronizing Link State Information Among Routers

Imagine aggregating the immediate neighborhood information from all routers into one database; it would represent the connectivity of an entire routing area. This is the Link State database. It is helpful to think of the Adjacency database as a subset of the Link State database—the portion describing the directly attached links. (The way things actually work, portions of the Adjacency database are combined with local information to become the part of Link State database describing a router's immediate neighborhood.) The NLSP design aims for every router in a routing area to have a copy of the Link State database locally. Moreover, they synchronize their views of the database among themselves. To a great extent, the work of NLSP is synchronization of a replicated database—keeping the copies of the Link State database consistent among the routers. When a topology change occurs, there is a short transitional period when the copies differ. Operation of NLSP makes them converge again.

The Link State database represents routers and links. It does not represent end nodes; that is, it does not represent client workstations or personal computers.

When all the copies of the Link State database are identical and conform with the actual connectivity of the internetwork, forwarding decisions made by successive routers in a

packet's path are consistent with each other. The packet progresses from its source to the intended destination successfully and efficiently. If the copies become different from each other, or if they fail to reflect the connectivity of the internetwork accurately, things can go wrong. A packet can find itself in an endless forwarding loop among routers and eventually be discarded. A packet can be routed into a *black hole*, a router that has no way to progress the packet toward the intended destination. This is why consistency of the Link State database is so important.

When a link comes up or goes down, it takes time for knowledge of that event to percolate throughout the internetwork. Meanwhile, there is a temporary inconsistency. This might wreak havoc for a Network-Layer protocol that guarantees delivery. The IPX™ protocol does not—it operates on a best-effort basis, meaning that occasional data loss is tolerable. Still, any disruption must be minimized by making the Link State databases converge to a consistent view as fast as possible. Experience in the industry indicates that Link State algorithms achieve synchronization faster than other designs for large-scale internetworks.

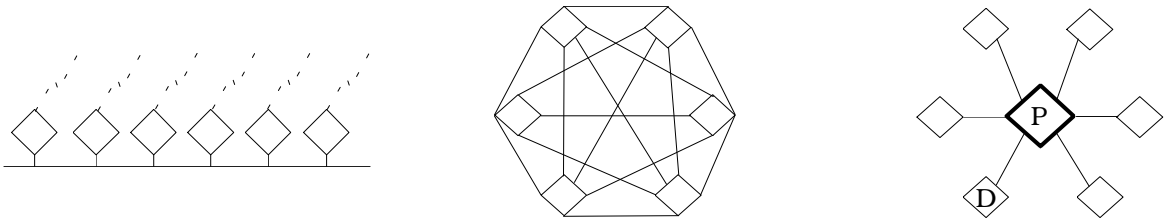
Flooding is the means used to achieve synchronization. Each router sends information from its Adjacency database to each of its neighbors. The information takes the form of *Link State Packets*, or *LSP*s. A numbering scheme detects when the same LSP arrives more than once. When a new LSP arrives—one not seen before—two things happen. First, it is retransmitted on all links except those on which that particular LSP was received. Second, it is merged into the receiving router's Link State database. The database is the collection of LSP packets of two kinds: LSP packets derived from the router's own Adjacency database and LSP packets received from other routers by flooding.

If an adjacency is lost, each router detecting this event floods an LSP indicating that the link is down. This forces routers receiving the LSP to tag the link as down. Records of unreachable destinations are not removed from the Link State database at this point.

Each LSP includes a Remaining Lifetime field, initialized to the value maxAge seconds by the originating router. Each router holding a copy of the LSP counts down the Remaining Lifetime, reinitializing it if a new copy of the LSP arrives. If the LSP times out, it is purged from the Link State database. At the same time, the router doing the purging floods the expired LSP. The purging mechanism cleans up stale information that has little prospect of becoming useful.

2.1.3. Determining a Designated Router and Constructing Pseudonodes

To keep the size of the Link State database reasonable, LANs need special measures. Consider Figure 2-3, which shows many routers connected to the same LAN. Part (a) shows each router able to relay between the LAN and a particular WAN. In terms of physical connectivity, each router can reach all the others in one hop. There is full mesh adjacency, as shown in part (b) (the WAN links are not shown there). If there are n routers, this means $n \times (n-1) \div 2$ links.



(a) Physical Topology *(b) Actual Adjacencies* *(c) Adjacency Model*

Figure 2-3: Designated Router and Pseudonode

It would be unduly burdensome to convey so much information to all the other routers in the routing domain, and to have them all process the information to make routing decisions. Instead, the LAN adjacencies among the routers are modeled conceptually, as shown in part (c). A fictitious *pseudonode* ("P" in the diagram) represents the LAN as a whole in the Link State database. Each of the routers represents itself as being directly connected to P. There are *n* "links." End nodes are thought of as being directly connected to P, although they are not represented in LSPs. It's unnecessary to represent them—they are located by having the same network number as the pseudonode.

Because P is fictitious, some actual router is chosen to represent P for Link State protocol exchanges. This is the *Designated Router* ("D" in the diagram). A regular protocol exchange occurs among routers of a LAN by which one (and only one) Designated Router is chosen and a replacement takes over in case the original one goes out of service (or becomes ineligible for some other reason). Periodically, every router multicasts a Hello packet on the LAN. The one with the highest priority (a configurable parameter) wins. In case of a tie, the IEEE MAC addresses are compared numerically, and the higher value wins. If a router is attached to several LANs, it might be Designated for some but not for others—the Designated Router election is independent on different LANs, and is decided by priority and address numbers.

The Designated Router originates LSPs on behalf of the pseudonode. If it resigns because a new router has a higher priority, it floods the area with a null LSP indicating that the old pseudonode is no longer valid.

2.1.4. Incorporating Switched Circuits into the Design

Certain wide-area media use switched, connection-oriented data-link media. To reach a remote destination, you initiate a call. If the call is established successfully, you can then transmit data over the connection. Finally, when you are finished, the call can be cleared. Because IPX is a connectionless Network-Layer protocol, there is no necessary association between an application session and a data-link connection.

One type of switched network is a packet-switched network using the X.25 standard. These networks support multiple concurrent connections (called *virtual circuits*) using a single network attachment. In NLSP, each connection is treated as a distinct adjacency, and hence as a distinct link in the Link State database. If a router is connected to an X.25 network and has five connected IPX virtual circuits at the current time, it's as if there were five point-to-point links. X.25 is not modeled as a fully connected cloud, but as a collection of point-to-

point links that come up and down as virtual circuits are established and cleared. Operation of NLSP does not by itself cause virtual circuits to be established or cleared.

Another type of switched network is an asynchronous dial-up circuit using a modem. Each modem supports one connection at a time, at most. A modem link is treated as a point-to-point link that can be up (if connected) or down (if not connected). Operation of NLSP does not by itself cause calls to be established or cleared.

2.1.5. The Circuit Concept in Link State Design

NLSP has the concept of a *circuit* both for LANs and WANs. With LANs, a circuit is a point of attachment to a network segment, through which the router can reach many other systems. If the router has two points of attachment to the same network segment, it is treated as two circuits. With a two-party WAN (either a dedicated or a dialed link), each point of attachment is treated as a circuit. The situation is different for multipoint WANs like X.25 and Frame Relay. For these WAN media, a single point of attachment provides access to many other systems using a connection-oriented Data-link layer. NLSP treats each X.25 or Frame Relay *virtual circuit* (switched or permanent) as a separate circuit for operation of the routing protocol, even though virtual circuits can share the same network point of attachment.

2.1.6. Example of Link State Database

Figure 2-4 illustrates the same example as Figure 2-1, but with the parts labeled. Each router is labeled with a letter toward the end of the alphabet. Each LAN is labeled with a letter toward the beginning of the alphabet. A pair of letters denotes a link. For example, RT denotes the point-to-point link connecting the router R with the router T. Likewise, RA denotes the (conceptual) link between R and the pseudonode that represents the LAN A. The designated routers (R, U, and W) are shown with heavy borders.

For the sake of discussion, suppose there are two active X.25 virtual circuits: SV and UV. Also, suppose the modem attached to W is not currently connected to a remote party.

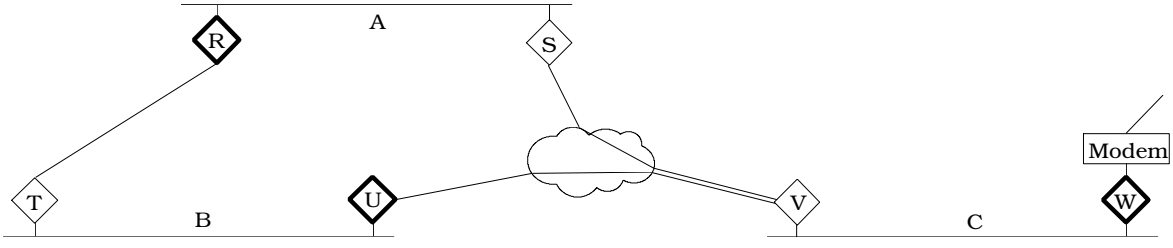


Figure 2-4: Labeled Example

Figure 2-5 shows a simplified view of the Link State database describing the example. There is one table for nodes and a second one for links. Notice that each link is represented twice: once as reported by the router at one end of the link and again as reported by the router at the other end of the link. Cost is discussed in more detail later; for now, simply consider a link to have a non-negative measurement assigned to it. Notice also that a pseudonode reports zero cost to reach the real nodes of the LAN.

Node ID	*Type*
A	Pseudonode
B	Pseudonode
C	Pseudonode
R	Router
S	Router
T	Router
U	Router
V	Router
W	Router

Link	*Cost*	*Link*	*Cost*
AR	0	RA	10
AS	0	SA	10
RT	19	TR	19
BT	0	TB	11
BU	0	UB	11
SV	40	VS	40
UV	40	VU	40
CV	0	VC	10
CW	0	WC	10

Figure 2-5: Link State Database

2.1.7. Decision Process and Forwarding

The typical routing decision is, “given that you have a packet to a particular final destination, what should the next hop be from here to get it to the destination with least cost”? (It is also possible to split the load among several next-hop links.)

The cost table in Figure 2-5 contains the information from which you can deduce routing decisions, but its form is not conducive to deciding the next hop. The *Decision Process* is a computation that puts the cost-table into a new form, called the *Forwarding database*. Unlike the Link State database, each router has a unique version of the Forwarding database, expressed from its own vantage point. The Forwarding database can be thought of as a table that maps a network number into the next hop. Figure 2-6 illustrates a Forwarding database from the vantage point of router V.

Network #	*Next Hop*
0xCCC47689	S
0xCCC47666	C
0xCCC47600	U
0x0082392C	U
0x845FAC11	S
0x05551212	W
0xC3141592	V

Figure 2-6: Forwarding Database Example

The first column in Figure 2-6 contains all the network numbers in the area. For each network number, the next hop in that row indicates which outgoing link—and destination on that link—to use for traffic destined to that network. When the router transmits a data frame, the table determines how to forward it. To take one new concept, 0xCCC47666 is the network number for LAN C; end nodes on C are reached by recourse to that table row. As another example, an NLSP router has an *internal network number*, and 0xC3141592 is the one for V. Packets addressed to that network are not forwarded by V. Of course, the internal representation of the Forwarding database is not typically a linear array, but a structure designed for fast lookup.

The Decision Process rebuilds the Forwarding database whenever the router detects a change in the Link State database. There is a hold-down timer, however. This is a minimum interval between reruns of the Decision Process. Topology changes can occur clustered in time. The hold-down timer allows the cluster of changes to be absorbed into the Link State database before running the Decision Process. Otherwise, it would run for each change of the cluster, wasting processing resources.

The Decision Process is called Dijkstra's algorithm, after Edsger W. Dijkstra, who devised the underlying method. It treats each router or pseudonode as a node in a graph, and each link as an arc connecting two nodes. Each link has a cost—actually, two costs, one in each direction, as reported by the router (or pseudonode) at each end. For example, the graph for Figure 2-4 and 2-5 looks like Figure 2-7.

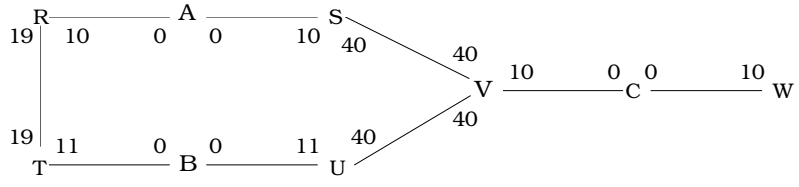


Figure 2-7: Graph with Costs

The object is to find the shortest distance from the local node (that is, from the router where the computation is running) to every other node of the graph. Here are the essential elements of Dijkstra's algorithm:

- Maintain a set of nodes whose shortest distance is already known. Call this the *Known Set*. Initially, it contains only the local node, with zero cost.
- The algorithm operates iteratively. At each iteration, add one node to the Known Set.
- How do you decide which to add? Look at the links from nodes in the Known Set to those outside it. This gives you a subset of links to consider for this step of the iteration. (The subset is equivalent to the *tentative list* used by some texts in describing Dijkstra's algorithm.) Each link in the subset has a *near node*, in the Known Set, and a *far node*, outside the Known Set. From the subset, choose the one whose far node has the lowest total cost from the local node. The node at the other end of that link is the one chosen. Add it to the Known Set.
- After n iterations, all the reachable nodes are in the Known Set (n is the number of reachable nodes in the graph).

How does this method build a Forwarding database? Each iteration identifies three things: a chosen far node, a link connecting it to a previous member of the Known Set, and the previous near node itself. For the Forwarding database, the next hop to the chosen node must be determined. If the near node is the local node itself, the next hop is the link chosen in this iteration. Otherwise, the next hop is the same as the near node's next hop.

Figure 2-8 illustrates the iterations of Dijkstra's algorithm for the graph shown in Figure 2-7. It shows the computation performed by node U.

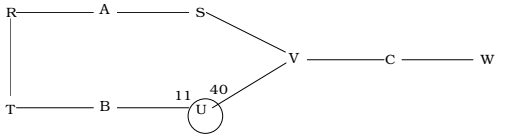
Step 6 is interesting because it involves choosing between two routes to the same node. The algorithm selects the better path, rejecting the suboptimal one. In either case, the same node (S in the example) is added to the Known Set, but it is important that the better route to it be the one recorded as the result of the step.

The computational burden to run Dijkstra's algorithm is proportional to $n \times k \times \log n$, where n is the number of routers, and k is the number of neighbors per router. Several optimizations are possible. Memory requirements are proportional to $n \times k$. For a detailed discussion, see Reference [Per92], pages 2-24 through 2-28.

Step 1

Known Set

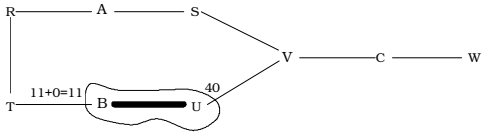
Node	Cost	Next Hop
U	0	–



Step 2

Known Set

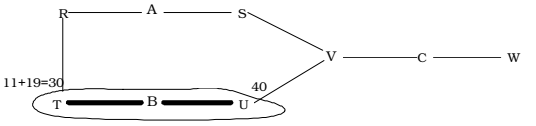
Node	Cost	Next Hop
U	0	–
B	11	B



Step 3

Known Set

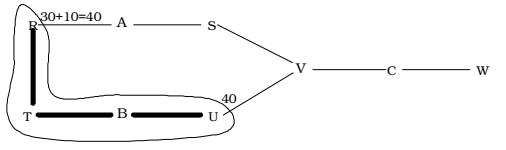
Node	Cost	Next Hop
U	0	–
B	11	B
T	11	B



Step 4

Known Set

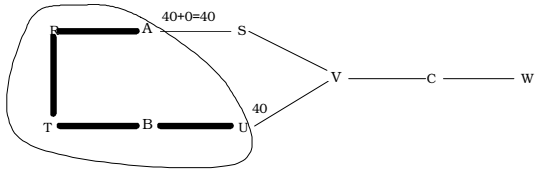
Node	Cost	Next Hop
U	0	–
B	11	B
T	11	B
R	30	B



Step 5

Known Set

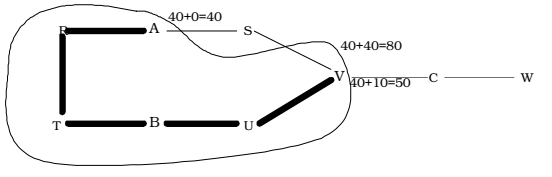
Node	Cost	Next Hop
U	0	–
B	11	B
T	11	B
R	30	B
A	40	B



Step 6

Known Set

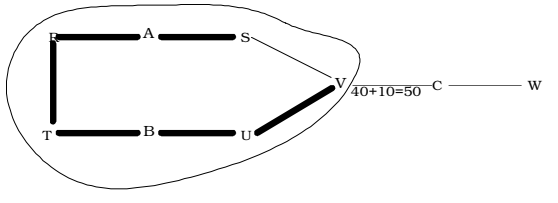
Node	Cost	Next Hop
U	0	–
B	11	B
T	11	B
R	30	B
A	40	B
V	40	V



Step 7

Known Set

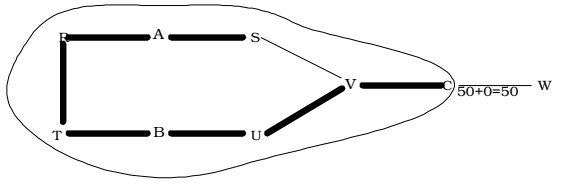
Node	Cost	Next Hop
U	0	–
B	11	B
T	11	B
R	30	B
A	40	B
V	40	V
S	40	B



Step 8

Known Set

Node	Cost	Next Hop
U	0	–
B	11	B
T	11	B
R	30	B
A	40	B
V	40	V
S	40	B
C	50	V



Step 9

Known Set

Node	Cost	Next Hop
U	0	–
B	11	B
T	11	B
R	30	B
A	40	B
V	40	V
S	40	B
C	50	V
W	50	V

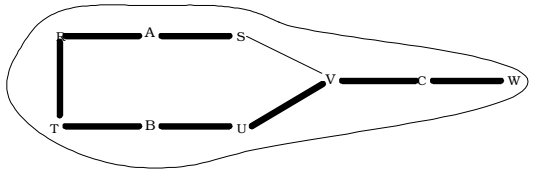


Figure 2-8: Decision Process Steps

Each iteration of Dijkstra's algorithm identifies a node and a link. The chosen links, taken together, form a subset of the graph called a *single source shortest path spanning tree*. The tree is shown in the diagram with a heavy line. Section 6 presents an alternate description of Dijkstra's algorithm. It is more akin to pseudocode.

2.1.8. Load-Splitting

NLSP supports load-splitting. That is, if there are equal cost paths, the traffic can be divided among them to make fuller use of the internetwork.

There is a *maximal splitting degree* defined. This is a number, MSD, assigned by the network administrator. It can be different for different routers in a routing area. If there are equal cost paths, MSD is an upper limit on the number used for routing. If MSD=1, the router does not do load-splitting. Load-splitting focuses on the step in Dijkstra's algorithm where you choose which node to add to the known set.

In case of a tie to the same far node, the Forwarding database has more than one entry added to it, instead of just one. How many? If the tie is among *m* links, the number of entries added is the smaller of *m* and MSD. If *m* is greater than MSD, the choice of which to add is based on a prescribed ordered list of criteria.

2.2. Reliability Features of NLSP

A number of NLSP features specifically ensure reliable operation.

2.2.1. Disseminating Updates Reliably

LAN and WAN links often do not provide guaranteed delivery. They operate on an unreliable best-effort basis. The LSP exchange protocol includes reliability features to ensure that the Link State database replicas in the routers become synchronized.

As mentioned earlier, each router sends LSP packets to its neighbors. Some LSPs describe the contents of the sender's Adjacency database; others are forwarded from other neighbors. To ensure that the Link State databases throughout the routing area are synchronized, a reliability feature is applied to these transmissions. There are two methods: one for point-to-point links and the other for broadcast links such as LANs.

For point-to-point links, a router sets a *Send Routing Message* (SRM) flag locally on an LSP before transferring it to a neighbor. The flag remains set until an acknowledgment is received. Periodically, LSPs with the SRM flag are sent.

A router replies each time it receives an LSP on a point-to-point link. Suppose A sends an LSP to B. In the typical case, the LSP is equal to the one B already has, or newer. In this case, B replies with a Partial Sequence Number Packet (PSNP), with the LSP source's identity, sequence number of the LSP, remaining lifetime of the entry, and other information. This is the form of the acknowledgment. A router can acknowledge more than one LSP in a PSNP.

On the other hand, if the received LSP is older than the one already in the Link State database, B sends the newer LSP.

Reliability works differently for broadcast networks. The Designated Router multicasts a Complete Sequence Number Packet (CSNP) series periodically. This does not contain the entire Link State database, but just enough information about each LSP so that each of the other routers can determine whether it is synchronized with the Designated Router. If the receiver of a CSNP determines that the Designated Router is out of date, it multicasts the newer information. On the other hand, if the CSNP receiver itself is out of date, it multicasts a PSNP identifying the LSPs for which it requests an update. The Designated Router responds with the missing information.

2.2.2. Operating with Database Overload

A router might find that it has insufficient resources to process an incoming LSP. The problem might be insufficient memory, misconfiguration, or certain transitory conditions. When this happens, the router cannot make correct decisions, so other routers must be warned. A bit in the LSP designates database overload, and the router experiencing overload floods an LSP with the bit set. Traffic is routed around the overloaded router until the situation returns to normal. A router must always preallocate space for the database overload LSP, so memory shortage does not prevent sending it.

2.2.3. Calculating and Testing Checksums

A checksum is used on each LSP sent, to detect corruption of data in transit. The computation is the same as the one used in ISO 8473 (Reference [ISO88]). If the receiver detects a checksum error, the packet is dropped and an error is logged.

2.2.4. Coping with System Bugs

In rare situations, a hardware or software bug can cause a memory location in a computer to be overwritten erroneously. If the computer is acting as a router, and if the corrupted data is part of the Link State database, the erroneous information causes incorrect routing decisions.

NLSP includes a provision to recover from situations like this automatically. Periodically, every router performs a checksum computation on the Link State database. The checksum result is compared with the checksum values originally received with each LSP. Detecting a mismatch is cause for drastic action—the entire LSP database is discarded and reacquired from scratch.

In the typical case, where everything checks out correctly, the Decision Process is run immediately after the checksum calculation. This protects against undetected corruption of the Forwarding database.

2.2.5. Implementing Fail-Stop Operation

Routers must be implemented with high reliability of hardware and software in mind. In particular, there should be a very low probability of corrupting data. Although NLSP does contain fault-tolerance features to recover from corrupted data and resource overload, such misadventures do take their toll on smooth operation of the network.

Routers should be implemented in a fail-stop manner. If the system detects that its routing operation might be compromised, the routing function should be deactivated, and then reactivated with an empty routing database. The routing database is a transitory one, without an archival requirement. If one router's database is lost, the information can be reacquired

readily (a) from other routers, and (b) by probing the network environment anew. Newly acquired routing information is more useful than old information. Plunging ahead with a possibly corrupted database risks spreading the failure's impact widely to other routers. Consequently, it is better to restart and gather the information anew.

2.3. The Criterion Used in Determining Routes

When making routing decisions, a router tries to determine the best path for packets to follow from source to destination. But "best" according to what criterion? To address this question, a *cost* is assigned to each link. The cost is a non-negative integer. The Decision Process operates to minimize the total cost for packets to travel from source to destination.

Each type of media has a predefined default cost defined later in this specification. This cost is an indication of the throughput capacity of a given link. The end user can override the cost assigned to a given link. This allows the user to customize routing behavior. Each side of a link has a separate view of the cost. It is recommended (but not required) that the views indicate the same cost. If they don't, routing can be asymmetric. That is, the route from A to B can traverse different links than the route from B to A. Asymmetric routing still works, but problems can be more difficult to diagnose than with symmetric routes.

2.4. Information Useful to Higher Layers

NLSP conveys information about the characteristics of the links throughout the routing area: the connectivity, the costs, the media types, and so forth. This information is used within the IPX network layer, to do a good job of routing. Because NLSP includes all the mechanisms to convey such information reliably throughout the area, those mechanisms are exploited to convey information not used for routing, but which is useful to other software. There are three items currently defined, with room to add more in the future. The three are identified as follows:

MTU size. This is the maximum packet size (in bytes of IPX user data) that can be transmitted over a link in a given direction. Knowing this value for each link allows a router to determine the MTU size end-to-end from an originator to a final recipient. This allows the originator to send data packets of just the right size, making best use of the internetwork's capacity.

Delay. This is a measurement of the time (in microseconds) to send a byte of data (excluding protocol headers) across the link. Aggregating this measurement over the links in an end-to-end path, a router can calculate end-to-end delay with fine granularity. This information allows a transport protocol to fine-tune the timeout interval to use waiting for an acknowledgment. The better the fine-tuning, the better the performance achieved.

Throughput. This is the amount of data (in bits per second) that can flow through the interface. This information helps the Transport layer decide the maximum useful unacknowledged data to have in its output queue. It allows fine-tuning of the offered window size of a transport connection.

These values can be accessed using network management. In future protocol revisions there might be additional, more specialized protocols to access the values.

2.5. Routing Information Protocol

The Routing Information Protocol (RIP) has been used with IPX from the early days, and its use continues. RIP is a distance vector routing protocol modeled after the Xerox* XNS* protocol of the same name (Reference [Xer81]). Routers exchange routing information with their neighbors through periodic broadcasts. They consolidate the information and pass the summarized data along to other RIP routers and to nonrouting end nodes.

NLSP is a successor to RIP for communicating among routers. For router/end node communication, however, RIP continues to be the one method used.

Reference [Nov92] contains a complete specification of RIP.

RIP uses a value called *RIP Link Delay* as its measurement criterion. It is measured in *ticks* (18.21 ticks per second). RIP Link Delay is the time to deliver a 576-byte packet to a destination across the link (it must be at least one). Conceptually, this value contains components of latency and throughput. NLSP deals separately with latency (called *NLSP Delay* in the protocol) and throughput (*NLSP Throughput*), but combines them for the benefit of RIP compatibility.

2.5.1. RIP for Communicating with End Nodes

This part of RIP support must be operational unconditionally. It provides the way for end nodes to learn (a) the network number of their directly connected network, and (b) which router provides the best next hop to an intended destination. It does not include periodic broadcast.

RIP is a request/reponse protocol. The request includes space to ask about routes to specific network numbers, or the requester can send a general request for routes to all networks. Every router on the directly connected segment responds with a measurement of the distance through it to the networks. Based on the reponse, an end node decides which router provides the best path to a given network. This is purpose (b).

When an end node begins operation, it does not know the network numbers of the segments to which it is attached. It broadcasts any RIP or SAP request on its directly connected LAN segments, using zero as the source and destination network number in the IPX header. A router (or server) receiving such a request makes the appropriate reply, and includes in the reply's IPX header the segment's actual network number. From a response's IPX header, the end node learns the segment network number. This is purpose (a). If there are no routers operating, the end node can still communicate with peers on the same network using network number zero.

When a router replies to a RIP request, it supplies two measurements for each destination network:

- Number of hops (the number of routers that must be traversed to reach the destination network)
- Number of ticks

NLSP routers determine the number of hops from the tree constructed by the Decision Process. The number of ticks is calculated from the delay and throughput measurements described in Section 2.4, “Information Useful to Higher Layers.”

2.5.2. RIP for Backward Compatibility

Internetworks can contain both RIP and NLSP routers. NLSP routing includes the method to connect the two worlds into a unified whole. When activated, this compatibility feature involves (a) generating RIP broadcasts so that RIP routers learn how to send data traffic to NLSP routers, and (b) heeding RIP broadcasts, so that NLSP routers learn how to send data traffic to RIP routers.

This part of RIP support is operational conditionally, on a per-link basis.

RIP routes are absorbed into the Link State database as *external routes*, like those to other routing areas. This means, in particular, that NLSP routes are preferred over RIP routes if both are available to the same destination. The RIP route for a network is attached to the node that detects a RIP broadcast on a directly connected network. For a LAN, it is attached to the pseudonode. The two RIP measurements (hop count and tick count) are included. When LSPs are flooded, the RIP information is carried along.

When composing a RIP broadcast, an NLSP router uses the information from (a) the external routes, and (b) the tree calculated by the Decision Process. First, consider an NLSP route; that is, a network number accessible by traversing NLSP routers only—no RIP routers. The router determines the number of hops from the tree constructed by the Decision Process. The number of ticks is calculated from the delay and throughput measurements described in Section 2.4, “Information Useful to Higher Layers.”

Next, consider a network number reachable by a concatenation of an NLSP path and a RIP route. The router determines the number of hops using the RIP portion of the path alone. What matters is the best values of hops and ticks reported by any RIP router, detected by any NLSP router. The portion of the path within the NLSP routing area between NLSP routers is not counted when composing a RIP broadcast.

RIP filtering is implemented when sending and receiving RIP broadcasts on a per-link basis. The user can configure network numbers (or number patterns) to be included or excluded from consideration. However, the RIP routes carried along in LSPs flooded among NLSP routers are not filtered.

Routers allow the RIP broadcast frequency to be configurable on each link individually. The default is once per minute.

RIP routes are purged if they are not refreshed by receipt of a new RIP broadcast. This timeout interval is configurable on each link individually. The default is four minutes.

2.6. Service Advertising Protocol

The Service Advertising Protocol (SAP) has been used with IPX from the early days, and its use continues. SAP uses a broadcast method similar to RIP for disseminating information, but instead of routing information it conveys information about network services and their

addresses. Servers advertise their services and addresses, and routers gather the information to share with other routers and with end nodes.

With the newer NetWare® Directory Services™ (NDS™) design, the use of SAP can be considerably reduced. Workstations locate services of interest by consulting an NDS server. The information about services on the NDS server is disseminated by direct, unicast-based protocols. It is not disseminated by broadcast-based SAP. NDS is part of NetWare 4. However, even in a network of all NetWare 4 nodes, some uses of SAP remain. For example, SAP locates the nearest NDS server on startup. SAP usage can be considerably reduced with NDS, but is not eliminated at this point. Moreover, other IPX systems have a customer presence and are still being deployed: NetWare 3, NetWare 2, and products from many parties that add value to those environments. For all these reasons, NLSP routers must support SAP.

Incoming SAP broadcasts are processed in a way similar to incoming RIP broadcasts. The received SAP information is included in the receiving router's Link State database, and from there is flooded to other routers. For a LAN, the SAP information is attached to the pseudonode. Information can come from a RIP router, a nonrouting end node, or a software module running a network service on the router system itself. It is all treated similarly.

An NLSP router responds to SAP resource-location requests using the SAP information gleaned from SAP broadcasts and from other NLSP routers. It also generates periodic SAP broadcasts based on the same information. SAP support is configurable on a per-link basis.

SAP filtering is implemented on a per-link basis when sending and receiving SAP broadcasts. The user can configure names (or name patterns) to be included or excluded from consideration. However, the SAP carried along in LSPs flooded among NLSP routers is not filtered.

Routers allow the SAP broadcast frequency to be configurable individually on each link. The default is once per minute.

SAP information is purged if it is not refreshed by receipt of a new SAP broadcast. This timeout interval is configurable individually on each link. The default is four minutes.

2.7. NetBIOS and Packet Type 20

For certain protocol implementations to function correctly, NLSP routers propagate certain broadcast packets. The protocol most affected is IPX-based NetBIOS. The IPX packet type 20 is used specifically for this purpose. The special handling applies only to broadcast packets—those with destination node number all-ones in the IPX header. Details are in Section 5 of Reference [Nov92].

End users require the ability to set boundaries on the propagation of these packets. Routers implement a user-configurable, per-link filter governing whether packets of type 20 are sent and heeded on that link.

2.8. Enhancing NLSP to Hierarchical Routing

This document specifies routing within a routing area—Level 1 routing. As NLSP develops in the future, there will be facilities for linking areas together into routing domains (Level 2), and routing domains into a global internetwork (Level 3). Figure 2-9 illustrates the hierarchy of

domains and areas. Figure 2-9 shows the internals of one of the areas for clarity, and one of the domains. Imagine each area as containing LANs and WANs such as in Figure 2-1. Areas are connected to each other by Level 2 routers; domains are connected to each other by Level 3 routers.

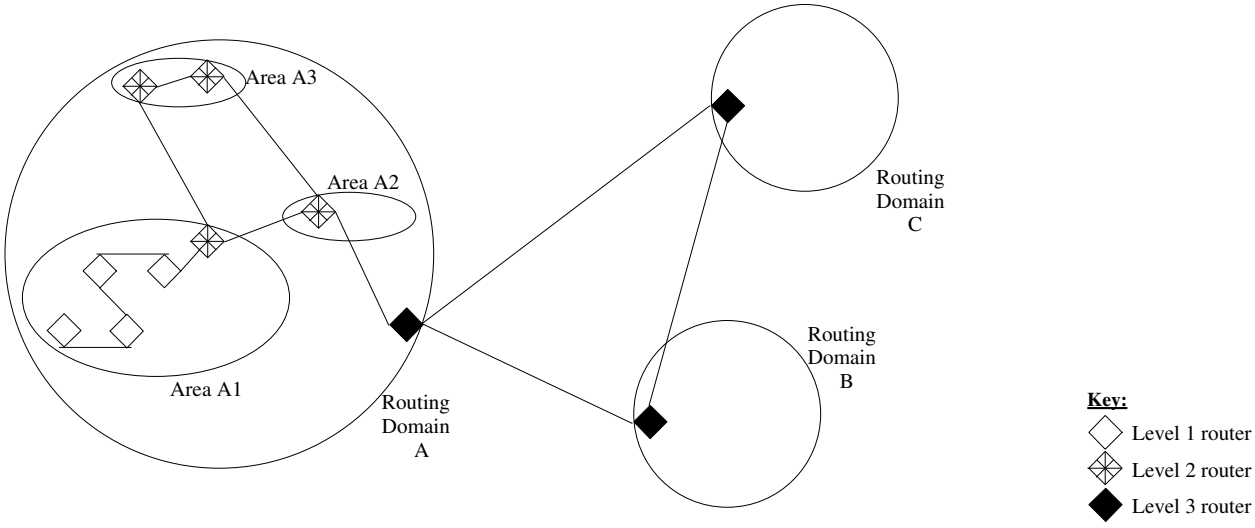


Figure 2-9: Hierarchical Routing

Hierarchical routing intends that each independent organization is a separate routing domain. This could be a company, a university, or an agency. It could be a public carrier connecting organizations with each other. Within a routing domain, each local campus, or other suborganization having its own administrative staff, constitutes a routing area.

Level 2 routing aims to enhance scalability by reducing the amount of information that every router must store and process to route throughout a domain. Instead of keeping the link states throughout the routing domain, it keeps this level of detail only for its own area. To get to other areas, it need only maintain information on a per-area basis, and be able to find the Level 2 router for each area other than its own. This information is a much coarser level of detail. Hierarchical routing wins information compression and economy of computation.

Level 3 routing aims first to provide another level of scalability and information compression, and then to imagine different routing domains belonging to different organizations entirely—companies or agencies with an arms-length relationship. A public network service provider that connects different organizations together might have a domain of its own. To support administrative separation, interdomain routing is usually considered with imposing policy criteria on forwarding decisions. For example, imagine that each of the three domains in Figure 2-9 is a separate company's internal internetwork. Domain B might want to forbid transit traffic from Domain A bound for Domain C.

A Level 2 router also acts as a Level 1 router within its own area; likewise, a Level 3 router also acts as a Level 2 router within its own domain.

2.9. IPX Addressing and its Relationship to Routing

The IPX Network-Layer address contains three parts:

- A four-byte network number
- A six-byte node number
- A two-byte socket number

A network number is assigned to each LAN segment. Also, NetWare 3 and 4 servers have *internal network numbers*. Network numbers receive the most attention in this document, because they guide routing decisions.

The node number identifies a specific system attached to the network. It is taken from the MAC address space administered by IEEE. This makes it globally unique. (In some situations, it must really be unique only within a network. This fact is sometimes exploited, bypassing global uniqueness.)

The socket number identifies a higher-layer entity within a system.

Two 32-bit quantities identify each routing area: a network address and a mask. The mask contains *n* leading "one" bits and *32–n* trailing "zero" bits. The pair of numbers is called an *area address*. The term *Category n* area refers to an area where the mask has *n* leading "one" bits. The larger *n* is, the smaller the area. An example is

```
01234500
FFFFFF00
```

The first number identifies the area: address `01234500`. The second number, the mask, indicates how much of the area number identifies the area itself, and how much identifies an individual network within the area. In the example, the first 24 bits of the address (`012345`) are the routing area. (Twenty-four is six hex F's times four bits per hex digit.) When writing the area address as a number/mask pair as above, write zeros after `012345` to pad it to 32 bits.

There is a relationship between an area address and the IPX network numbers in the area. If the mask part of an area address has *n* leading one bits, the first *n* bits of an network number must equal those of the area's address part. When this happens, the network number *matches* the area address, and the network assigned that number is in the area. In the example of the preceding paragraph, every network number in the area starts with `012345`. The remaining eight bits identify individual network numbers in the routing area; for example, `012345AB` and `01234500` are network numbers in the area. The example is a Category 24 area.

A routing area can have as many as three area addresses. NLSP treats the three as synonymous identifiers of the same routing area. To be in the area, an IPX network number matches any one of the three. The three can be of different categories. In fact, no numerical relationship is prescribed between the addresses. Having more than one address allows the routing areas to be reorganized without interrupting operation. For example, to split a routing area in half, you introduce the new area address into half the routers, one by one. Then, remove the old address from those routers, one by one. As long as one of the routers has both addresses, there is one routing area. As soon as the routers become two disjoint groups, there are two areas.

Figure 2-10 provides a simple example, with four routers (R, S, T, U) in a linear sequence. All four start in one area, and conclude with two areas of two routers each.

Figure 2-10: Splitting a Routing Area

To accomplish the split, the following steps can be followed:

Step 1	This is the starting point, one area address.			
Router	R	S	T	U
Level	1	1	1	1
Area Addresses	56789AB0 FFFFFFF0	56789AB0 FFFFFFF0	56789AB0 FFFFFFF0	56789AB0 FFFFFFF0

Step 2	Add the second area address to R.			
Router	R	S	T	U
Level	1	1	1	1
Area Addresses	56789AB0 11111110 FFFFFFF0 FFFFFFF0	56789AB0 FFFFFFF0	56789AB0 FFFFFFF0	56789AB0 FFFFFFF0

Step 3	Add it to S. Remove the old area address from R. It's still all one area, though.			
Router	R	S	T	U
Level	1	1	1	1
Area Addresses	11111110 FFFFFFF0	56789AB0 11111110 FFFFFFF0 FFFFFFF0	56789AB0 FFFFFFF0	56789AB0 FFFFFFF0

Step 4	Promote S and T from Level 1 to Level 2 Routers. It's still one area.			
Router	R	S	T	U
Level	1	2	2	1
Area Addresses	11111110 FFFFFFF0	56789AB0 11111110 FFFFFFF0 FFFFFFF0	56789AB0 FFFFFFF0	56789AB0 FFFFFFF0

Step 5	Remove S's original address. The routers now see two nonoverlapping areas.			
Router	R	S	T	U
Level	1	2	2	1
Area Addresses	11111110 FFFFFFF0	11111110 FFFFFFF0	56789AB0 FFFFFFF0	56789AB0 FFFFFFF0

Now go from routing areas to the next level of aggregation: routing domains. There is no numerical relationship required among area addresses within a domain, except that no two areas can contain the same area address. When Level 3 routers convey traffic between domains, they share a description of which addresses belong in which domain. This is a list of all the area addresses contained in each domain.

Area addresses in a routing domain do not have to share a common prefix for NLSP to work correctly. Any area addresses can be combined in a domain. But if common prefixes are, in fact, used, routing can operate more efficiently. First, the data shared among domains is more compact, because the data can be expressed using the common prefix instead of by listing all the area addresses. Second, as a consequence, routers making routing decisions have less data from other domains to process, so they have less work to do. This saves compute cycles. Finally, confidentiality can play a role: a company or carrier might not want to reveal its internal network topology. Consequently, those planning address assignments should consider assigning them hierarchically, even though such an assignment is not required for correct operation. For example, if addresses in the Figure 2-9 internetwork were assigned hierarchically, the Level 3 router in Domain A might have the following address list:

Domain A		Domain B		Domain C	
Area A1	CCCC1500	All areas	CCCC2000	All areas	CCCC3000
	FFFFFF00		FFFFF000		FFFFF000
Area A2	CCCC1520				
	FFFFFFF0				
Area A3	CCCC1600				
	FFFFFFF0				

Note that the information about domains B and C have collapsed to a single entry each. Likewise, information about Domain A collapses to a single entry from the point of view of B or C. This works because they all differ in the fifth hexadecimal digit: the digit is "1" for domain A; "2" for B, and "3" for C. Within Domain A, only one entry need be circulated to the Level 1 and Level 2 routers to describe Domains B and C. If address assignment is not hierarchical, there are many entries. This consumes network bandwidth to convey to all routers, memory to record the entries, and computational capacity to process them.

2.10. Network Management

NLSP routers are managed using the SNMP (Simple Network Management Protocol) standard. SNMP runs over IP and IPX.

Specifying NLSP network management amounts to defining an SNMP MIB. The MIB is a set of values that can be read and changed remotely by action of the SNMP protocol, and a set of traps that can be generated by the router to raise alarms about exceptional conditions. The NLSP MIB is divided into groups, as follows:

System Group	Global information about the IPX protocol entity
Circuit Group	Information about each interface on the system

Forwarding Group	Forwarding database used to route data traffic
Services Group	Known Services that advertise themselves using SAP
Neighbors Group	Information about neighboring NLSP routers and the adjacencies with them
LSP Group	Tables representing the LSP database
Translation Group	Various mappings (for example, System ID to Server Name)
Graph Group	Representation of the network topology
Traps Group	Notifications of significant events

There are actually three MIBs, as follows:

- IPX MIB includes basic information about IPX Network-Layer operation. It is applicable to all IPX-speaking systems, both routers and end nodes. It includes information in the following groups: System, Circuit, Forwarding, Services, and Traps.
- RIP/SAP MIB augments IPX MIB with information specific to the RIP and SAP protocols. It extends the System and Circuit groups.
- NLSP MIB augments IPX MIB with information specific to the NLSP routing protocol. It extends the System, Circuit, and Forwarding groups, and adds the Neighbors, Translation, Graph, and LSP groups.

2.11. Capabilities Assumed in the System Environment

Certain characteristics are required of a system for it to be suitable to act as an NLSP router. Certain characteristics of the underlying data links are also required for support of NLSP.

2.11.1. Configuration of Parameters by End Users

Certain operational values and characteristics are configurable by the end user. To accommodate this requirement, a router needs a method for user interaction. The method is an implementation choice. For example, it could be one of the following:

a) Terminal attached directly to the router

b) Remote terminal-emulation facility

c) File edited by the user and read by the router

d) An implementation-dependent protocol

e) The NLSP MIB defined in this specification

f) Different MIB for a different network management protocol

g) Combination of methods

2.11.2. Event Handling

There are various places in the specification where it is convenient to describe behavior of the router in terms of events. The term *event* is meant in a generic sense. Implementation of event handling is system-specific.

Event handling can include notifying a system manager of abnormal conditions, making an entry in an error log, and incrementing a counter.

2.11.3. Characteristics of Links to Support NLSP

The links connecting routers with each other must support frames of 576 bytes. This number is the data portion of the frame, including the Network-Layer (IPX) header, but not data-link headers. If larger frames are supported, performance and scalability can improve.

Links can operate on a best-effort datagram basis, providing a high probability of delivering individual frames. There is no requirement for the links to provide confirmed or guaranteed delivery. There is no requirement for the links to guarantee that the order of delivery matches the order of transmission. (However, if frames are reordered often, performance can be affected.) NLSP includes reliability features that allow it to operate over unreliable links. NLSP operates over all sorts of links: broadcast, nonbroadcast, local area, metropolitan area, wide area, connectionless, connection-oriented, point-to-point, multipoint, switched, dedicated, and high and low data rates.

The router needs a way to detect when a link becomes active or inactive, and when failures or degraded conditions prevail.

Certain kinds of multipoint networks have the ability to address multiple stations with the same data-link frame. Addressing all stations attached to a network segment is called *broadcast*; when a subset is addressed, the function is called *multicast*. NLSP uses multicast addressing (on those links where it is available) to address all the NLSP routers on the network segment, without burdening other systems. Where broadcast is available but not multicast, the same frames (which would have been multicast) are sent using the broadcast mechanism.

2.11.4. Imposing Jitter on Timed Operations

When packets are transmitted as a result of timer expiration, there is a danger that timers of individual systems might become synchronized. This would result in a traffic distribution containing peaks of intense activity. Where there are large numbers of synchronized systems, the peaks can overload both the transmission medium and the receiving systems. To prevent this from occurring, periodic timers that cause packet transmissions have *jitter* introduced. When jitter is applied, the actual time interval varies randomly between 75% and 100% of the specified value.

The procedure `DefineJitteredTimer` below indicates how jitter operates in NLSP. In the description, `BaseTimeValueInSeconds` is the nominal timeout value for a specific action, and `ExpirationAction` is a procedure that can be called to perform that action. For example, `DefineJitteredTimer (10, SendHelloPacket)` causes the action *SendHelloPacket* to be performed at random intervals of between 7.5 and 10 seconds.

```
DefineJitteredTimer (BaseTimeValueInSeconds, ExpirationAction)

  CONSTANTS:

       Jitter = 25           -- Percentage jitter defined in this specification.

       Resolution = 100      -- Timer resolution in milliseconds for a
                             -- particular router.

  VARIABLES FOR THE PROCEDURE AS A WHOLE:

       BaseTimeValue         -- BaseTimeValueInSeconds, converted to timer
                             -- intervals.

       MaximumTimeModifier   -- Largest possible jitter.

       Running               -- Starts as TRUE; set to FALSE to halt the
                             -- activity.

  VARIABLES REEVALUATED IN THE INNER LOOP:

       WaitTime              -- Time interval for this loop's timeout.

       NextExpiration        -- next clock time at which the ExpirationAction occurs.

  METHOD:

       BaseTimeValue ← BaseTimeValueInSeconds × 1000 / Resolution

       MaximumTimeModifier ← BaseTimeValue × Jitter / 100

       WHILE Running

             BEGIN           -- Running loop

             WaitTime ← BaseTimeValue - Random (MaximumTimeModifier)

             NextExpiration  ← CurrentTime () + WaitTime

             ExpirationAction ()

             WaitUntil (NextExpiration)

             END          -- Running loop
```

The description assumes availability of the following functions in the environment:

```
 Random (max)      -- Returns a uniformly distributed random integer with
                   -- 0 < Random(max) < max.

 CurrentTime ()    -- Returns the current time in milliseconds.

 WaitUntil (time)  -- This procedure waits until the current time is "time,"
                   -- then returns.
```

The essential point of the algorithm is that the expiration time is randomized within the inner loop. Note that the new expiration time is set immediately on expiration of the last interval, rather than when `ExpirationAction` has completed.

2.12. IPX Protocol

IPX is a connectionless datagram Network-layer protocol, adapted from the Xerox Network Systems design, Reference [Xer81]. The IPX packet structure is defined on page 2-28.

Packets whose processing is specified in this document use the IPX header values listed in Figure 2-11.

Protocol	Socket	Packet Type	Where Specified
IPX Ping	0x9086	0x04	Page 2-26
RIP	0x0453	0x01	Section 7
SAP	0x0452	0x04	Section 7
IW2	0x9004	0x04	Section 3
Propagated Packet	Varies	0x20	Chapter 5 of Reference [Nov 92]
NLSP	0x9001	0x00	Sections 4 and 5

Figure 2-11: Socket Values and Packet Types Pertinent to This Document

2.12.1. General Processing of Incoming IPX Packets

When a router receives a valid IPX packet, it proceeds in the following order:

a) If the final destination of the packet is on the same system as the router, it is processed by the software entity identified by the Destination Socket field. This includes packets addressed directly to this system, as well as packets sent by a data-link multicast or broadcast on a directly connected network segment. Exit after completing step a) for such a packet—the remaining steps do not apply. If there is no software entity on the system prepared to receive the packet, the packet is discarded.

   The following general acceptance tests apply to incoming packets destined for the routing software entity on this system:

   1. Size consistency tests apply to NLSP packets. Suppose

      D is the data-link user data size reported by the Data-Link layer,

      I is the Packet Length reported in the IPX header,

H is the IPX header size (30 bytes), and

N is the Length Indicator in an NLSP packet.

Every valid packet has

$D \geq I$

$I = N + H$

If a received packet fails these size consistency tests, a packetRxSmall event is generated and the packet is discarded.

The "$D \geq I$" test also applies to all arriving IPX packets.

2. The IPX Packet Type field determines how to process the packet. For the IPX Ping, IW2, and NLSP packets specified in Sections 2 through 5 of this document, check that the IPX Destination Socket field is compatible with the Packet Type. Valid combinations are in Figure 2-11. Discard packets having an invalid combination. Do not perform this check for RIP and SAP packets.

3. If the Major Version field of an incoming NLSP packet is not one, a mismatchedVersion event is generated and the packet is discarded. (The Minor Version field is not checked.)

4. If an NLSP packet arrives on a circuit for which the NLSP protocol is not active, the packet is discarded.

b) If Packet Type = 20, process it as specified in Chapter 5 of Reference [Nov92].

c) If the Transport Control field is maxHops or greater, the packet is discarded. Traditionally, the value of maxHops is 16. It is configurable. When a packet is discarded, an ipxInTooManyHops event is logged.

d) All other packets are data packets to be relayed by operation of the router's traffic forwarding role. If the next hop is impossible because the packet is too large for the medium's supported data-link frame size, the packet is discarded. When the router forwards a packet, it increments the Transport Control field by one.

Note: If either the Source or Destination Network Number field is zero, it should be filled in with the actual Network Number before being passed to the appropriate software entity, or before being forwarded. When forwarding a packet, the only two modifications a router can make to the IPX header or data are (a) incrementing the Transport Control field and (b) filling in the actual Network Number.

It is valid to forward a packet even if the Destination Node is all-ones, indicating a broadcast. If the router is attached directly to such a packet's Destination Network, the packet is forwarded as a data-link broadcast on that circuit.

2.12.2. IPX Checksum

A checksum can optionally be included by the originator of the IPX packet. The checksum is end-to-end, from originator to final destination. A Checksum field containing all-ones (0xFFFF) signifies that the originator did not compute the checksum for that packet. Typically, upper-layer protocols negotiate the use or non-use of checksums for a session.

Note: Use of the Checksum field is a recent addition to IPX. In the past, the Checksum field of IPX packets was set unconditionally to all ones. Many installed systems still do not calculate the checksum.

The IPX checksum is generated over the range of bytes starting with the Checksum field itself, and ending with the last byte of the IPX Data. For the purpose of checksum calculation, the Checksum field and the Transport Control field are both considered to contain zero. Because Transport Control reflects hop count, this provision means that a router does not update the Checksum of forwarded packets.

The range of bytes is treated as an array of 16-bit words. If there would be an odd number of bytes, the range is considered to be padded with an extra zero byte for the purpose of the calculation. The first byte of each word is considered the most significant numerically.

The checksum calculation uses a variable S containing the partial sum. Denote by w[i] the i-th word from the array of 16-bit words. The checksum calculation proceeds as follows.

a. Initialize S to zero.

b. Iterate over index values i covering the array of 16-bit words:

   b.1 $S \leftarrow S + w[i]$; this is a twos-complement arithmetic sum.

   b.2 If Step b.1 resulted in a carry, $S \leftarrow S + 1$.

c. Replace S with its bit-wise logical "not," inverting each bit.

Because Step b includes adding in the carry bit, 32-bit processors can employ 32-bit operations (rather than 16-bit operations) to calculate the checksum efficiently. The final 32-bit sum after Step b would be split into two 16-bit words and added (with the carry added in as in step b.2) before Step c.

The final value of S is stored in the Checksum field, with the numerically most-significant byte appearing first.

Note: The calculated checksum cannot be all-ones, based on the following line of reasoning.

- Step c is a bit-wise logical "not," so it is enough to show that the last iteration of Step b cannot result in zero.
- Each iteration of Step b involves summing two terms, then adding in the carry out of that sum. The only way this result can be zero is if both of the terms are zero. To see why, consider three cases.
  - If Step b.1 produces all ones, there cannot be a carry, so b.2 produces all ones, not zero.
  - If Step b.1 produces zero and at least one term is nonzero, there is a carry, so Step b.2 produces one, not zero.
  - If Step b.1 produces anything other than zero or all ones, Step b.2 cannot produce zero.
- Consequently, as soon as Step b encounters nonzero w[i], S becomes nonzero and remains nonzero for the subsequent iterations.

- An IPX packet cannot be all zeros. For example, the Packet Length field is always at least 30.
- Combining the last two points shows that the calculated checksum cannot be all-ones.

If a packet reaches its final destination with an IPX checksum error, the packet is discarded.

Note: One of the LAN encapsulations used with IPX is 802.3 (without an 802.2 header). A packet having a calculated checksum must not be transmitted over a link using this encapsulation.

2.12.3. IPX Ping Protocol

The IPX Ping Protocol (Version 1) provides a simple, end-to-end Network layer reachability test. It uses the packet format described on page 2-30. The Ping packet is transmitted as the data portion of an IPX packet.

Note: The Ping Protocol is a recent innovation. Many installed systems do not support it.

When sending a Ping Request

- The Destination Socket in the IPX header is 0x9086, denoting the Ping protocol.
- The Source Socket in the IPX header is arbitrary.
- The Type field of the Ping packet is 0x00, denoting a Ping Request.
- The Data field of the Ping packet is arbitrary.
- The other IPX Ping fields are as described on page 2-30.

When a Ping Request destined for this system arrives, perform the following steps:

a) If the packet is too small (less than 40 bytes, including the IPX header), discard it.

b) If the Signature field is incorrect, discard the packet.

Note: The Version field is not tested.

c) If the above tests succeed, transmit a Ping Response to the request's originator, with the following field values:

- Copy the Request's IPX header Source Socket into the Response's Destination Socket.
- The Source Socket in the IPX header is 0x9086.
- The Type field of the Ping packet is 0x01, denoting a Ping Response.
- Copy the Data and Ping ID fields from the Request.
- Store in the Result field one of the following two values:
  0x00 if the Ping Request's Data field was empty,
  Or
  0x01 if the Ping Request's Data field contained one or more bytes.

Note: In future versions of IPX Ping, the responder might perform additional steps based on a Request's Version, Reserved, and Data fields. If the Request is version 1, the Data field will still be ignored and returned unchanged.

- The other IPX Ping fields are as described on page 2-30.

An arriving Ping Reply destined for this system is recognized by its IPX Destination Socket equaling a previously transmitted Ping Request's Source Socket, and by the value 0x01 in the

Ping packet's Type field. When a Ping Reply destined for this system arrives, perform the following steps:

d) If the packet is too small (less than 40 bytes, including the IPX header), discard it.

e) If the Signature field is incorrect, discard the packet.

Note: The Version field is not tested.

f) If the Result field is neither 0x00 nor 0x01, discard the packet.

g) If the packet does not originate from a system to which a Ping Request was sent previously, discard the packet.

h) If the Ping ID field matches that of a Ping Request previously transmitted to the Response's sender, process that Ping Request as completed.

2.13. Packet Structures

All the packets defined in this specification are carried as the data portion of IPX Network layer packets. In the packet, they immediately follow the IPX header specified on page 2-28.

The IPX Network layer packets, in turn, are carried as the data portion of data-link frames. They immediately follow the data-link header, and immediately precede the data-link trailer (if any). Data-link headers are media-dependent, and are not defined in this specification. This is illustrated in Figure 2-12. The data-link headers are specified in Section 2 for WANs and Section 10 for standard LANs.

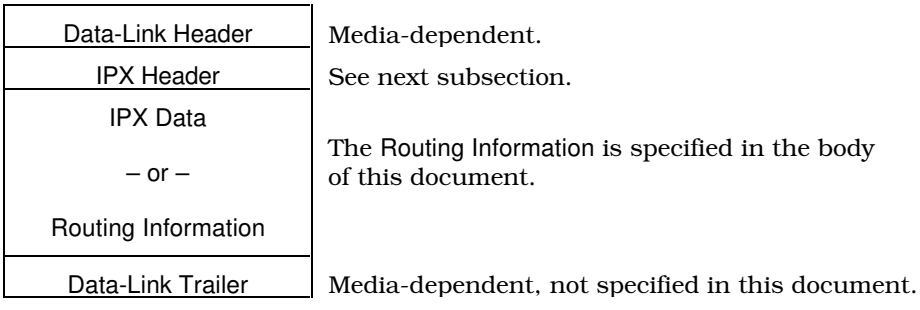


Figure 2-12: Data-Link Packet

All multibyte fields are transmitted with the most significant byte first.

When NLSP packets are sent by multicast to all NLSP routers on the directly connected segment, the following destination MAC addresses are used in the data-link header:

- IEEE 802.3 — `0x09001BFFFFFF`
- IEEE 802.5 — `0xC00010000000`
- FDDI — `0x09001BFFFFFF`

The two structures specified in this section are the IPX header and the IPX Ping packet.

2.13.1. IPX Header

Checksum: The contents of this field are described on page 2-25.

Packet Length: The number of bytes in the IPX packet, including the IPX header and the subsequent IPX data.

Transport Control: The number of routers a packet has traversed on its way to its destination. Sending nodes always set this field to zero when originating an IPX packet.

IPX Header	Number of Bytes
Checksum	2
Packet Length	2
Transport Control	1
Packet Type	1
Destination Address	12
Source Address	12

Packet Type: The type of service offered or required by the packet. Values pertinent to this specification are in Figure 2-11.

> **Note:** For some versions of existing products, the Packet Type field is not a reliable indicator of the type of packet encapsulated. The source and destination Socket fields should be used to determine the packet type when this determination is required. Propagated packets are an exception; the Packet Type should be checked for the value 20 to determine whether the packet is to be propagated.

Destination Address: The IPX address identifying the final destination of the packet. Each IPX address has three subfields, as follows:

IPX Address	Number of Bytes
Network Number	4
Node Number	6
Socket	2

- Network number: Identifies the physical network segment to which a system is attached. If a system is attached to more than one segment, it has a corresponding address for each. Certain systems (for example, NLSP routers and NetWare 3 servers) also have an *internal network number* that does not correspond to an actual network segment, but is thought of as a virtual network residing inside the system. There is no broadcast network number. Network Numbers must be unique throughout an internetwork.

  Network number 0 (zero) has a special meaning: "The directly attached network segment onto which this packet is being transmitted." Systems that are starting up, and have not yet discovered the network number of a segment, use this value. Routers do not forward packets with Destination Network Number 0.

  Network number 0xFFFFFFFF (all ones) is not valid.

- Node number: Identifies a particular system on the network segment. The value is media-dependent. When the Network Number refers to an IEEE 802.3, 802.4, or 802.5 LAN, the value is the six-byte MAC address. If a medium has a smaller address (for example, Omninet), it is right-justified and zero-padded. The Node Number of a router or server on its internal network is one, by current convention. The broadcast value 0xFFFFFFFFFFFF means "all systems on the directly attached segment." Node numbers must be unique within a Network Number.

- Socket: Identifies a software entity within a networked system. Values pertinent to this specification are in Figure 2-11.

Source Address: The IPX address identifying the originator of the packet. It has the same three-part IPX Address structure as the destination address. Broadcast addresses are not allowed in the Source Node Number.

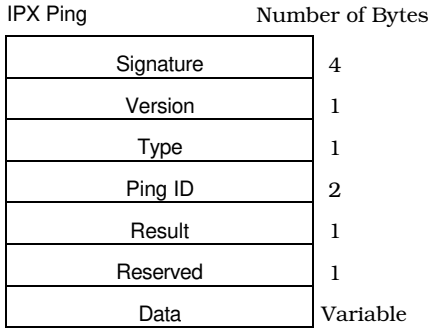
2.13.2. IPX Ping

Signature: 0x50696E67 (ASCII for “Ping”).

Version: 1, ignored on receipt.

Type: 0 = “Ping Request”
1 = “Ping Response”

Ping ID: A unique identifier assigned by the requester and echoed in the response. It is used to match responses with requests.



Result: In a Ping Request, this field is zero. In a Ping Response, it reports the responder’s actions or observations.

Reserved: 0, ignored on receipt.

Data: Zero or more arbitrary bytes chosen by the requester and echoed in the response.