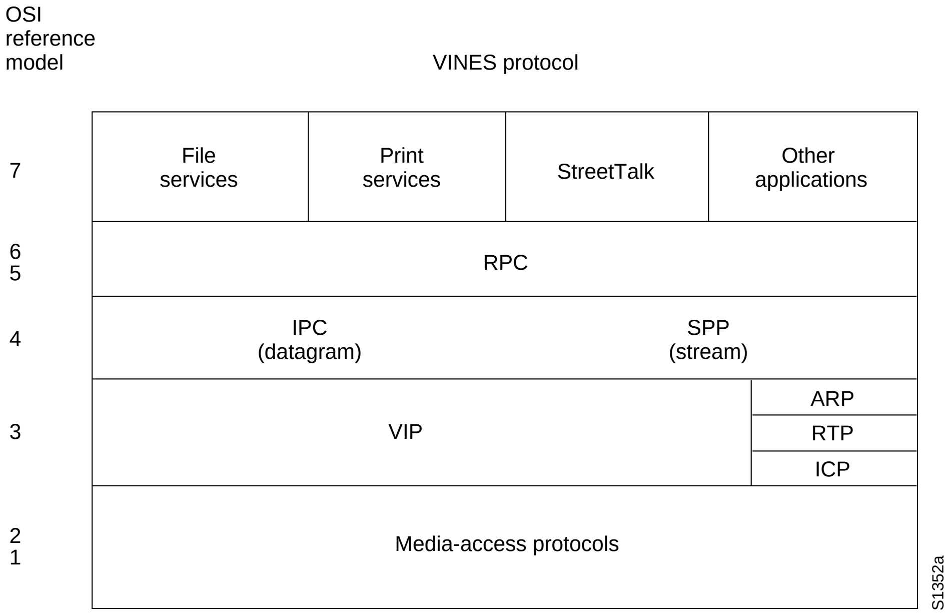
Banyan VINES

Background

Banyan Virtual Integrated Network Service (VINES) implements a distributed network-operating system based on a proprietary protocol family derived from the Xerox Corporation's *Xerox Network Systems* (XNS) protocols. VINES uses a client-server architecture in which *clients* request certain services, such as file and printer access, from *servers*. This chapter provides a summary of VINES communications protocols. The VINES protocol stack is illustrated in Figure 33-1.

Figure 33-1 The VINES protocol stack consists of five separate levels.



Media Access

The lower two layers of the VINES stack are implemented with a variety of well-known media-access mechanisms, including *High-Level Data Link Control* (HDLC), *X*.25, Ethernet, and *Token* Ring.

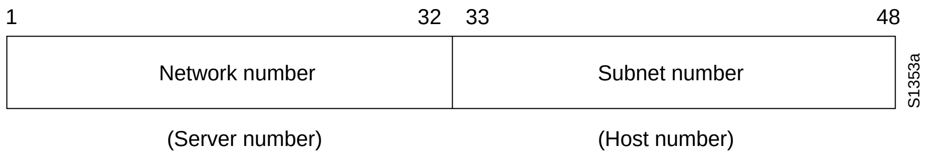
Network Layer

VINES uses the *VINES Internetwork Protocol* (VIP) to perform Layer 3 activities (including internetwork routing). VINES also supports its own *Address- Resolution Protocol* (ARP), its own version of the *Routing Information Protocol* (RIP)—called the *Routing Table Protocol* (RTP)—, and the *Internet Control Protocol* (ICP), which provides exception handling and special routing cost information. ARP, ICP, and RTP packets are encapsulated in a VIP header.

VINES Internetwork Protocol (VIP)

VINES network-layer addresses are 48-bit entities subdivided into network (32 bits) and subnetwork (16 bits) portions. The network number is better described as a server number because it is derived directly from the server's *key* (a hardware module that identifies a unique number and the software options for that server). The subnetwork portion of a VINES address is better described as a host number because it is used to identify hosts on VINES networks. Figure 33-2 illustrates the VINES address format.

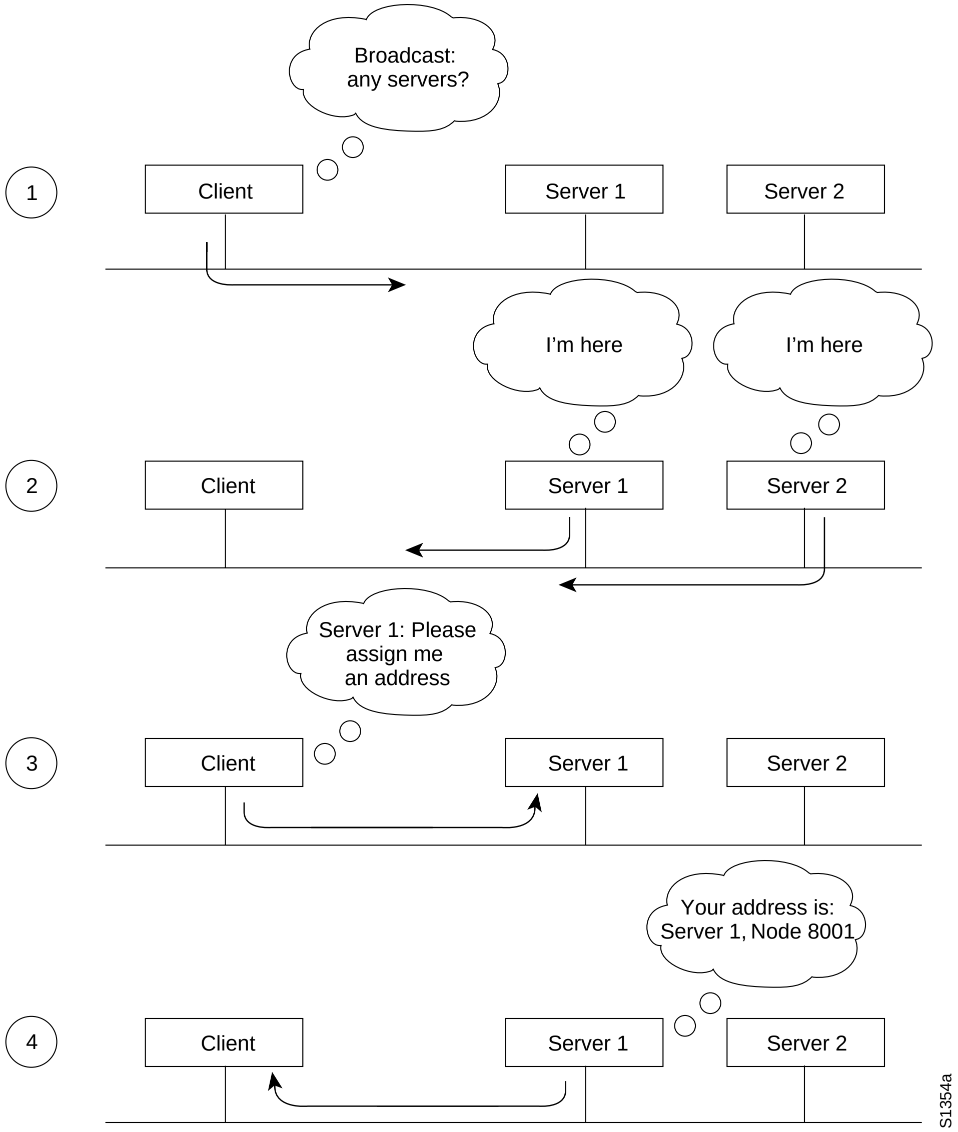
Figure 33-2 A VINES address consists of a network number and a subnet number.



The network number identifies a VINES logical network, which is represented as a two-level tree with the root at a *service node*. Service nodes, which are usually servers, provide address resolution and routing services to *clients*, which represent the leaves of the tree. The service node assigns VIP addresses to clients.

When a client is powered on, it broadcasts a request for servers, and all servers that hear the request respond. The client chooses the first response and requests a subnetwork (host) address from that server. The server responds with an address consisting of its own network address (derived from its key), concatenated with a subnetwork (host) address of its own choosing. Client subnetwork addresses typically are assigned sequentially, starting with 8001H. Server subnetwork addresses are always 1. Figure 33-3 illustrates the VINES address-selection process.

Figure 33-3 VINES moves through four steps in selecting an address.



Dynamic address assignment is not unique in the industry (AppleTalk also uses this process), but it certainly is not as common as static address assignment. Because addresses are chosen exclusively by a particular server (whose address is unique as a result of the hardware key), very little chance exists for duplicating an address. This is fortunate, because duplicate addresses could cause potentially devastating problems for *Internet Protocol* (IP) and other networks.

In the VINES network scheme, all servers with multiple interfaces are essentially routers. Clients always choose their own server as a first-hop router, even if another server on the same cable provides a better route to the ultimate destination. Clients can learn about other routers by receiving redirect messages from their own server. Because clients rely on their servers for first-hop routing, VINES servers maintain routing tables to help them find remote nodes.

VINES routing tables consist of host/cost pairs, where the host corresponds to a network node that can be reached, and cost corresponds to a delay (expressed in milliseconds), to get to that node. RTP helps VINES servers find neighboring clients, servers, and routers.

Periodically, all clients advertise both their network-layer and their MAC-layer addresses with the equivalent of a *hello* packet, which indicates that the client is still operating and network-ready. The servers themselves send routing updates to other servers periodically to alert other routers to changes in node addresses and network topology.

When a VINES server receives a packet, it checks to see whether the packet is destined for another server or whether it is a broadcast. If the current server is the destination, the server handles the request appropriately. If another server is the destination, the current server either forwards the packet directly (if the server is a neighbor) or routes it to the next server in line. If the packet is a broadcast, the current server checks to see whether the packet came from the least-cost path. If not, the packet is discarded. If so, the packet is forwarded on all interfaces except the one on which it was received. This approach helps diminish the number of broadcast storms, a common problem in other network environments. Figure 33-4 illustrates the VINES routing algorithm.

Figure 33-4 The VINES routing algorithm determines the appropriate path to a destination.

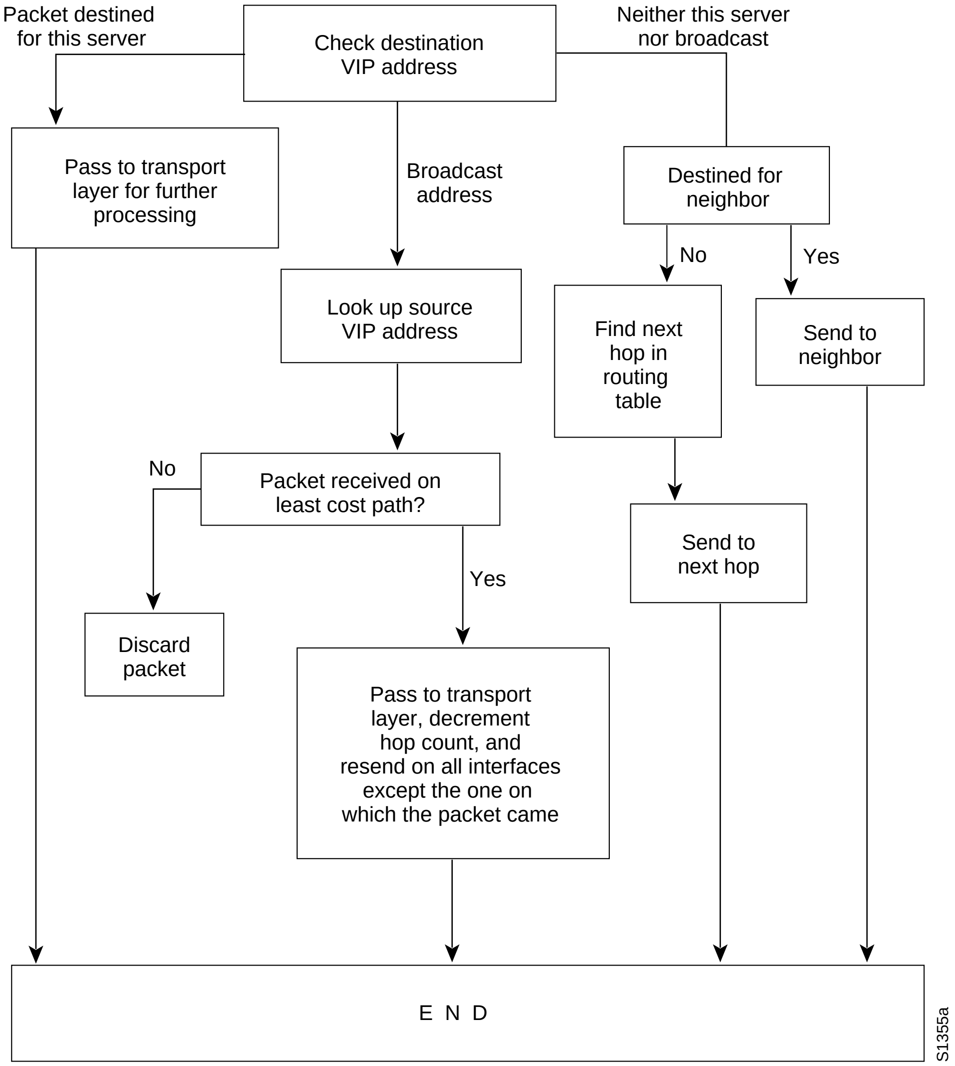

Packet destined for this server
Check destination VIP address
Neither this server nor broadcast
Pass to transport layer for further processing
Broadcast address
Destined for neighbor
No
Yes
Look up source VIP address
Find next hop in routing table
Send to neighbor
No
Packet received on least cost path?
Send to next hop
Yes
Discard packet
Pass to transport layer, decrement hop count, and resend on all interfaces except the one on which the packet came
E N D
S1355a

Figure 33-5 illustrates the VIP packet format.

Figure 33-5 A VIP packet consists of nine individual fields.

Field length, in bytes	2	2	1	1	4	2	4	2	Variable
	Checksum	Packet length	Transport control	Protocol type	Destination network number	Destination subnetwork number	Source network number	Source subnetwork number	Data

S1356a

The fields of a VIP packet include the *checksum, packet length, transport control, protocol type,* d*estination network number, destination subnetwork number, source network number,* and *source subnetwork number.*

- *The checksum* field is used to detect packet corruption. The packet-length field indicates the length of the entire VIP packet.
- *The transport-control field* consists of several subfields. If the packet is a broadcast packet, two subfields are provided: *class* (bits 1 through 3) and *hop count* (bits 4 through 7). If the packet is not a broadcast packet, four subfields are provided: *error*, *metric*, *redirect*, and *hop count*. The class subfield specifies the type of node that should receive the broadcast. For this purpose, nodes are broken into various categories according to the type of node and the type of link on which the node is found. By specifying the type of nodes to receive broadcasts, the class subfield reduces the disruption caused by broadcasts. The hop-count subfield represents the number of hops (router traversals) the packet has been through. The error subfield specifies whether the ICP protocol should send an exception- notification packet to the packet's source if a packet turns out to be unroutable. The metric subfield is set to one by a transport entity when it must learn the routing cost of moving packets between a service node and a neighbor. The redirect subfield specifies whether the router should generate a redirect, when appropriate.
- The *protocol-type field indicates* the network- or transport-layer protocol for which the metric or exception-notification packet is destined.
- Finally*, destination network number, destination subnetwork number, source network number,* and *source subnetwork number* all provide VIP address information.

Routing-Table Protocol (RTP)

RTP distributes network-topology information. Routing-update packets are broadcast periodically by both client and service nodes. These packets inform neighbors of a node's existence and also indicate whether the node is a client or a service node. In each routing-update packet, service nodes include, in each routing update packet, a list of all known networks and the cost factors associated with reaching those networks.

Two routing tables are maintained: a *table of all known networks* and a *table of neighbors*. For service nodes, the table of all known networks contains an entry for each known network except the service node's own network. Each entry contains a network number, a routing metric, and a pointer to the entry for the next hop to the network in the table of neighbors. The table of neighbors contains an entry for each neighbor service node and client node. Entries include a network number, a subnetwork number, the media-access protocol (for example, Ethernet) used to reach that node, a local area network (LAN) address (if the medium connecting the neighbor is a LAN), and a neighbor metric.

RTP specifies four packet types: *routing update, routing request, routing response, and routing redirect.* Routing update is issued periodically to notify neighbors of an entity's existence. Routing requests are exchanged by entities when they must learn the network's topology quickly. Routing responses contain topological information and are used by service nodes to respond to routing-request packets. A *routing-redirect* packet provides better path information to nodes using inefficient paths.

RTP packets have a 4-byte header that consists of the following 1-byte fields: o*peration type, which* indicates the packet type; *node type, which* indicates whether the packet came from a service node or a nonservice node; c*ontroller type, which indicates* whether the controller in the node transmitting the RTP packet has a multibuffer controller; and *machine type*, which indicates whether the processor in the RTP sender is fast or slow.

Both the controller-type and the machine-type fields are used for pacing.

Address Resolution Protocol (ARP)

Address-Resolution Protocol (ARP) entities are classified as either *address-resolution clients* or *address-resolution services*. *Address-resolution clients* usually are implemented in client nodes, whereas *address*-resolution services typically are provided by service nodes.

ARP packets have an 8-byte header that consists of a 2-byte *packet type*, a 4-byte *network number*, and a 2-byte *subnetwork number.* Four packet types exist: a *query request*, which is a request for an ARP service; a *service response*, which is a response to a query request; an *assignment request*, which is sent to an ARP service to request a VINES internetwork address; and an *assignment response*, which is sent by the ARP service as a response to the assignment request. The network-number and subnet-number fields have meaning only in an assignment- response packet.

ARP clients and services implement the following algorithm when a client starts up. First, the client broadcasts query- request packets. Then, each service that is a neighbor of the client responds with a service- response packet. The client then issues an assignment- request packet to the first service that responded to its query-request packet. The service responds with an assignment-response packet that contains the assigned internetwork address.

Internet Control Protocol (ICP)

The Internet Control Protocol (ICP) defines *exception-notification* and *metric-notification* packets. *Exception-notification* packets provide information about network-layer exceptions; *metric*-notification packets contain information about the final transmission used to reach a client node.

Exception notifications are sent when a VIP packet cannot be routed properly, and the error subfield in the VIP header’s transport control field is enabled. These packets also contain a field identifying the particular exception by its error code.

ICP entities in service nodes generate metric-notification messages when the metric subfield in the VIP header’s transport-control field is enabled, and the destination address in the service node’s packet specifies one of the service node’s neighbors.

Transport Layer

VINES provides three transport-layer services: unreliable-datagram service, reliable-message service, and data-stream service:

- *Unreliable-datagram service* sends packets that are routed on a best-effort basis but not acknowledged at the destination.

- *Reliable- message service* is a virtual-circuit service that provides reliable sequenced and acknowledged delivery of messages between network nodes. A reliable message can be transmitted in a maximum of four VIP packets.
- *Data-stream service* supports the controlled flow of data between two processes. The data-stream service is an acknowledged virtual -circuit service that supports the transmission of messages of unlimited size.

Upper-Layer Protocols

As a distributed network, VINES uses the *remote procedure call* (RPC) model for communication between clients and servers. RPC is the foundation of distributed-service environments. The *NetRPC* protocol (Layers 5 and 6) provides a high-level programming language that allows access to remote services in a manner transparent to both the user and the application.

At Layer 7, VINES offers file-service and print-service applications, as well as *StreetTalk*, which provides a globally consistent name service for an entire internetwork.

VINES also provides an integrated applications-development environment under several operating systems, including DOS and UNIX. This development environment allows third parties to develop both clients and services that run in the VINES environment.