



## Troubleshooting Banyan VINES

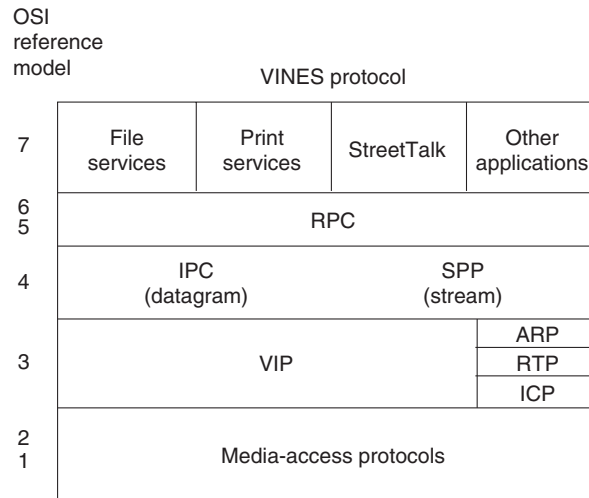
---

Banyan Virtual Integrated Network Service (VINES) implements a distributed network operating system based on a proprietary protocol family derived from Xerox Corporation's Xerox Network Systems (XNS) protocols (see Chapter 14, "Troubleshooting XNS"). VINES uses a client/server architecture in which clients request certain services, such as file and printer access, from servers. Along with Novell's NetWare, IBM's LAN Server, and Microsoft's LAN Manager, VINES is one of the best-known distributed system environments for microcomputer-based networks.

### VINES Technology Basics

The VINES protocol stack is shown in Figure 13-1.

Figure 13-1 The VINES Protocol Stack



## VINES Media Access

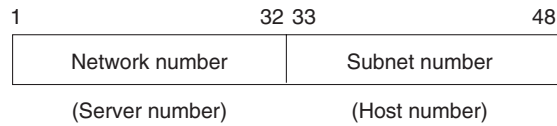
The two lower layers of the VINES stack are implemented with a variety of well-known media-access mechanisms, including High-Level Data Link Control (HDLC), Synchronous Data Link Control (SDLC) and derivatives, X.25, Ethernet, and Token Ring.

## The 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.

## VIP

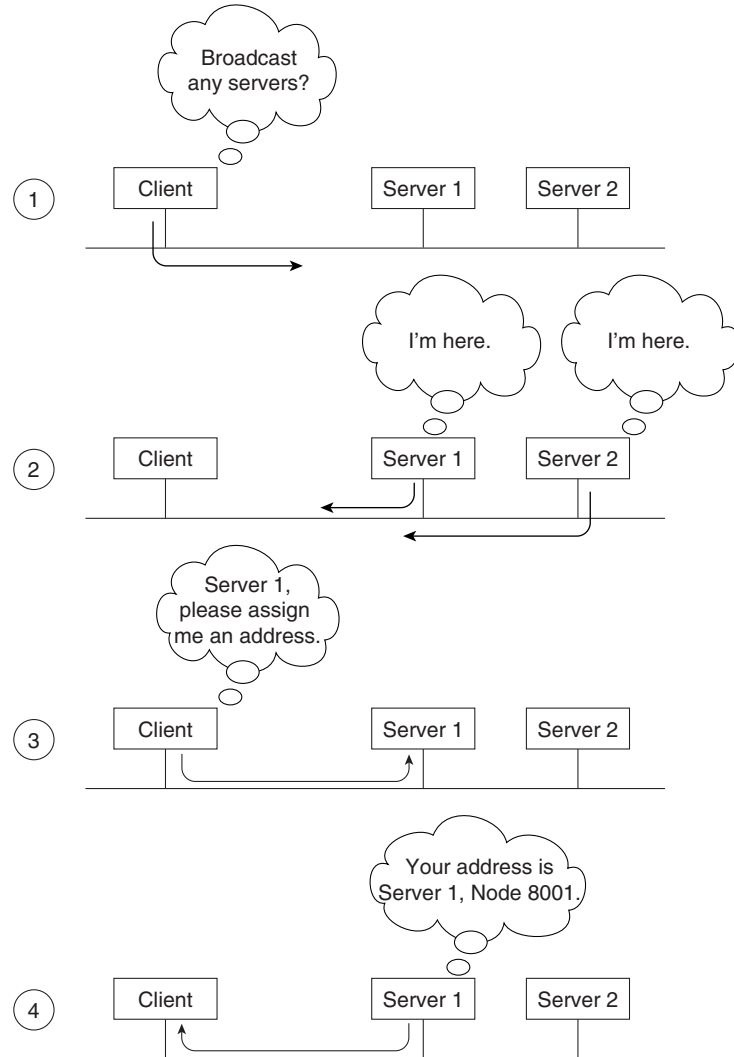
VINES network-layer addresses are 48-bit entities subdivided into network (32 bits) and subnetwork (16 bits) portions. The network number is better described as a server number because it is derived directly from the server's key (a hardware module that identifies a unique number and the software options 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 13-2 illustrates the VINES address format.

**Figure 13-2 The VINES Address Format**

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. 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 are typically assigned sequentially, starting with 8001H. Server subnetwork addresses are always 1. The VINES address selection process is shown in Figure 13-3.

Figure 13-3 The VINES Address Selection Process



Dynamic address assignment is not unique in the industry (AppleTalk also uses this process), but it is certainly 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 uniqueness of the hardware key), there is very little chance of a duplicate address (a potentially devastating problem on Internet Protocol [IP] and other networks).

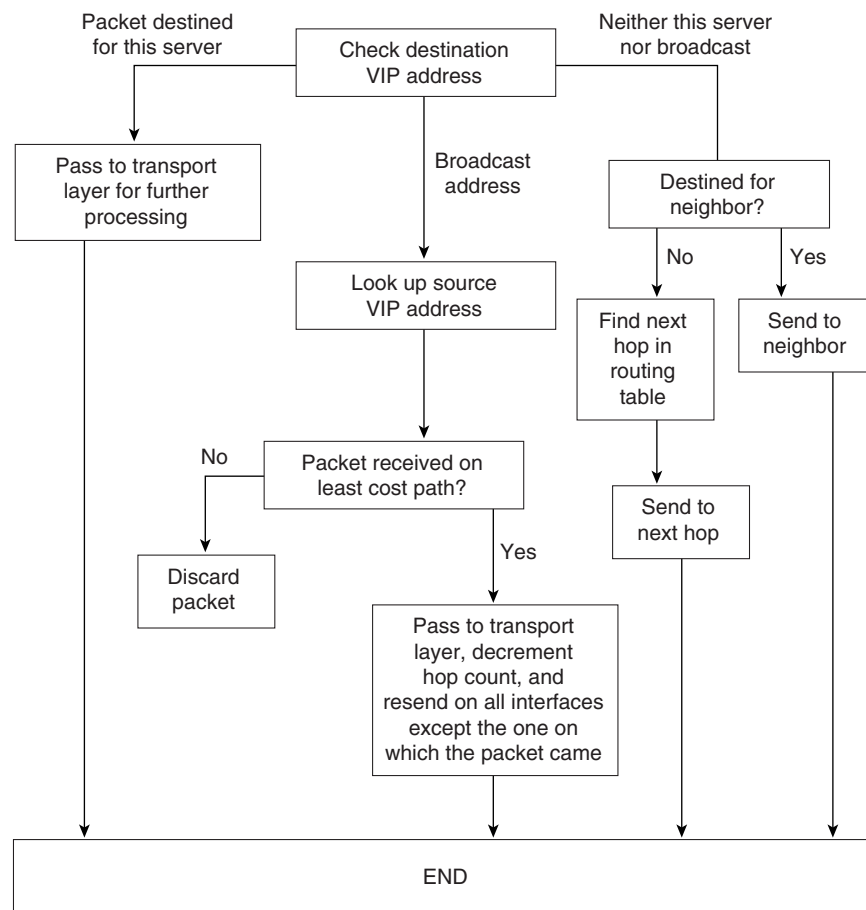
In the VINES network scheme, all servers with multiple interfaces are essentially routers. A client always chooses its own server as a first-hop router, even if another server on the same cable provides a better route to the ultimate destination. A client can learn about other routers by receiving redirect messages from its 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 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 Media Access Control (MAC)-layer addresses with the equivalent of a hello packet. Hello packets indicate that the client is still operating and network ready. The servers themselves send routing updates to other servers periodically. Routing updates alert other routers to changes in node addresses and network topology.

When a VINES server receives a packet, it checks whether the packet is destined for another server or if it's 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 whether the packet came from the least-cost path. If it did not, the packet is discarded. If it did, 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. The VINES routing algorithm is shown in Figure 13-4.

**Figure 13-4 The VINES Routing Algorithm**



The VIP packet format is shown in Figure 13-5.

Figure 13-5 The VIP Packet Format

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

The fields of a VIP packet are as follows:

- **Checksum**—Used to detect packet corruption.
- **Packet length**—Indicates the length of the entire VIP packet.
- **Transport control**—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 having to do with the type of node and the type of link the node is on. 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 1 by a transport entity when it needs to 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).
- **Protocol type**—Indicates the network- or transport-layer protocol for which the metric or exception notification packet is destined.
- **Destination network number, Destination subnetwork number, Source network number, and Source subnetwork number**—Provide VIP address information.

## 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 indicate whether the node is a client or a service node. Service nodes also 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**—Issued periodically to notify neighbors of an entity's existence.
- **Routing request**—Exchanged by entities when they need to learn the network's topology quickly.
- **Routing response**—Contains topological information and is used by service nodes to respond to routing request packets.
- **Routing redirect**—Provides better path information to nodes using inefficient paths.

Each RTP packet has a 4-byte header consisting of the following 1-byte fields:

- **Operation type**—Indicates the packet type.
- **Node type**—Indicates whether the packet came from a service node or a nonservice node.
- **Controller type**—Indicates whether the controller in the node transmitting the RTP packet has a multibuffer controller.
- **Machine type**—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.

## ARP

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

An ARP packet has an 8-byte header consisting of a 2-byte packet type, a 4-byte network number, and a 2-byte subnetwork number. There are four packet types: 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 containing the assigned internetwork address.

## ICP

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.

# The Transport Layer

VINES provides three transport-layer services:

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

- **Reliable message service**—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.

## Troubleshooting Banyan VINES

This section presents protocol-related troubleshooting information for connectivity problems related to Banyan VINES. It describes specific VINES symptoms, the problems that are likely to cause each symptom, and the solutions to those problems.

The following sections describe the most common errors experienced in Banyan VINES networks:

- VINES: Clients Cannot Communicate with Servers over Router
- VINES: Client Cannot Connect to Server over PSN
- VINES: Client on Serverless Network Cannot Connect to Server over PSN

### VINES: Clients Cannot Communicate with Servers over Router

**Symptom:** Clients cannot connect to VINES servers over one or more routers. Clients might or might not be able to connect to servers on their directly connected networks.



Table 13-1 outlines the problems that might cause this symptom and describes solutions to those problems.

**Table 13-1 VINES: Clients Cannot Communicate with Servers over Router**

Possible Problem	Solution
Router interface is down	<ol style="list-style-type: none"> <li>1. Use the <b>show interfaces</b> exec command to check the status of the router interfaces.</li> <li>2. If the status line indicates that an interface that should be up is “administratively down,” use the <b>no shutdown</b> interface configuration command on the interface.</li> </ol> <p>Refer to the troubleshooting chapter that covers the media type used in your network.</p>
Hardware or media problem	For information on troubleshooting hardware problems, refer to the troubleshooting chapter that covers the media type used in your network.
Addressing problem	<ol style="list-style-type: none"> <li>1. On a serverless segment, use the <b>show vines route</b> exec command to make sure the router is seeing server network layer addresses.</li> <li>2. If the router is not seeing server addresses, make sure that the server and router addresses are correct. To change the address, use the following syntax:</li> </ol> <p><b>vines routing</b> [<i>address</i>   <i>recompute</i>]</p>
Addressing problem (continued)	<p>Syntax Description:</p> <ul style="list-style-type: none"> <li>• <i>address</i> (Optional)—Network address of the router. You should specify an address on a router that does not have any Ethernet or FDDI<sup>1</sup> interfaces. You can also specify an address in the unlikely event that two routers map themselves to the same address.</li> <li>• <i>recompute</i> (Optional)—Dynamically redetermines the router’s network address.</li> </ul>
VINES metric value is not specified	<ol style="list-style-type: none"> <li>1. Use the <b>show vines</b> interface exec command to check the status of VINES interfaces on the router. Make sure all VINES interfaces have the <b>vines metric</b> interface configuration command configured. The <b>metric</b> command enables VINES processing on the interface.</li> <li>2. If the <b>vines-metric</b> interface configuration command is not configured on the interface, specify the command for the interface.</li> </ol> <p>Configure the <b>vines metric</b> based on whether the interface is LAN or WAN connected. Suggested metrics for LAN and WAN connections follow:</p> <ul style="list-style-type: none"> <li>• Ethernet and 16-Mbps Token Ring: <b>vines-metric 2</b></li> <li>• 4-Mbps Token Ring: <b>vines-metric 4</b></li> <li>• T1 line: <b>vines-metric 35</b></li> <li>• Other WAN link: <b>vines-metric 45</b></li> </ul>

Table 13-1 VINES: Clients Cannot Communicate with Servers over Router

Possible Problem	Solution
Missing <b>vines serverless</b> or <b>vines arp-enable</b> commands	<p>A network that does not have an attached server must be configured with the <b>vines serverless broadcast</b> and <b>vines arp-enable</b> router configuration commands.</p> <p><b>Note:</b> These commands are enabled by default in Cisco IOS Release 10.3 and later.</p> <ol style="list-style-type: none"> <li>1. Use the <b>show running-config</b> privileged exec command on routers attached to networks with no VINES servers attached. Look for <b>vines serverless</b> and <b>vines arp-enable</b> router configuration commands entries.</li> <li>2. If both the <b>vines serverless</b> and the <b>vines arp-enable</b> commands are not present, specify the commands for router interfaces in serverless networks.</li> </ol>
Missing <b>vines serverless</b> or <b>vines arp-enable</b> commands (continued)	<p>Syntax:</p> <p>The following syntax is required to enable <b>vines serverless</b>:</p> <pre><b>vines serverless</b> [dynamic   broadcast]</pre> <p>Syntax Description:</p> <ul style="list-style-type: none"> <li>• <i>dynamic</i> (Optional)—Forward broadcasts toward one server only if there are no servers present on this interface.</li> <li>• <i>broadcast</i> (Optional)—Always flood broadcasts out all other router interfaces to reach all servers.</li> </ul> <p><b>Syntax:</b></p> <p>The following syntax is required to enable <b>vines arp-enable</b>:</p> <pre><b>vines arp-enable</b> [dynamic]</pre> <p>Syntax Description:</p> <ul style="list-style-type: none"> <li>• <i>dynamic</i> (Optional)—Responds to ARP<sup>2</sup> and SARP<sup>3</sup> requests on this interface only if there are no other VINES servers present.</li> </ul>
Misconfigured access list	<ol style="list-style-type: none"> <li>1. Use the <b>show vines access-list</b> privileged exec command on routers in the path from source to destination. This command shows whether there are access lists configured on the router.</li> <li>2. Disable all access lists configured on the router using <b>no vines access-group</b> commands.</li> <li>3. Test the connection from the client to the server to see whether connections are now possible. If the connection is successful, an access list is blocking traffic.</li> <li>4. To isolate the problem access list, apply one access list statement at a time until you can no longer create connections.</li> </ol>

continues

**Table 13-1 VINES: Clients Cannot Communicate with Servers over Router**

Possible Problem	Solution
Misconfigured access list (continued)	<ol style="list-style-type: none"> <li>5. When the problem list is identified, alter it so that necessary traffic is allowed to pass. On a serverless segment, make sure that well-known ports 0x06 (VINES file service) and 0x0F (StreetTalk) are not filtered. Configure explicit <b>permit</b> statements for traffic you want the router to forward.</li> <li>6. If problems persist, continue testing for problem access lists on all routers in the path from source to destination.</li> </ol>

1. FDDI = Fiber Distributed Data Interface
2. ARP = Address Resolution Protocol
3. SARP = Sequence Address Resolution Protocol

## VINES: Client Cannot Connect to Server over PSN

**Symptom:** Clients cannot connect to VINES servers across a packet-switched network (PSN). Clients can connect to local VINES servers.

Table 13-2 outlines the problems that might cause this symptom and describes solutions to those problems.

**Table 13-2 VINES: Client Cannot Connect to Server over PSN**

Possible Problem	Solution
Address mapping error	<ol style="list-style-type: none"> <li>1. Use the <b>show running-config</b> privileged exec command to view the configuration of the router.</li> <li>2. For X.25 environments, make sure that LAN protocol-to-X.121 address mapping specified in <b>x25 map vines</b> interface configuration command entries use the VINES addresses and X.121 addresses of the destination routers. Confirm that the destination addresses used in the command entries are correct.</li> <li>3. For Frame Relay environments, make sure that the LAN protocol-to-DLCI<sup>1</sup> address mapping specified in <b>frame-relay map</b> command entries use the VINES address of the destination router and the DLCI of the local interface. Confirm that the destination address and the local DLCI used in the command entries are correct.</li> </ol>
PVC <sup>2</sup> is not set up	<ol style="list-style-type: none"> <li>1. Use the <b>show running-config</b> privileged exec command to view the configuration of the local and remote routers. Make sure there is an <b>x25 pvc n vines address</b> interface configuration command specified on the local and remote routers. This command sets up a PVC between the two routers.</li> <li>2. If the command is not present, add it to the router configuration.</li> </ol>

1. DLCI = Data Link Connection Identifier

2. PVC = permanent virtual circuit

## VINES: Client on Serverless Network Cannot Connect to Server over PSN

**Symptom:** Clients on a serverless network (that is, a network segment that has no attached VINES servers) cannot open a connection to a VINES server over a PSN.

Table 13-3 outlines the problems that might cause this symptom and describes solutions to those problems.

**Table 13-3 VINES: Client on Serverless Network Cannot Connect to Server over PSN**

Possible Problem	Solution
Address mapping error	<ol style="list-style-type: none"> <li>1. Use the <b>show running-config</b> privileged exec command to view the configuration of the router.</li> <li>2. For X.25 environments, make sure that LAN protocol-to-X.121 address mapping specified in the <b>x25 map vines</b> interface configuration command entries use the VINES addresses and X.121 addresses of the destination routers. Confirm that the destination addresses used in the command entries are correct.</li> </ol> <p>Syntax:</p> <pre><b>x25 map</b> protocol address [protocol2 address2[...[protocol9 address9]]] x121-address [option] <b>no x25 map</b> protocol address x121-address</pre>

Table 13-3 VINES: Client on Serverless Network Cannot Connect to Server over PSN (continued)

Possible Problem	Solution
Address mapping error (continued)	<p>Syntax:</p> <ul style="list-style-type: none"> <li><i>protocol</i>—Protocol type, entered by keyword. As many as nine <i>protocol</i> and <i>address</i> pairs can be specified in one command line.</li> <li><i>address</i>—Protocol address.</li> <li><i>x121-address</i>—X.121 address of the remote host.</li> <li><i>option</i>—(Optional) Additional functionality that can be specified for originated calls.</li> </ul> <p>3. For Frame Relay environments, make sure the LAN protocol-to-DLCI<sup>1</sup> address mapping specified in <b>frame-relay map</b> command entries use the VINES address of the destination router and the DLCI of the local interface. Confirm that the destination address and the local DLCI used in the command entries are correct.</p> <p>Syntax:</p> <pre>frame-relay map protocol protocol-address dlcI [broadcast] [ietf   cisco] [payload-compress {packet-by-packet   frf9 stac [hardware-options]]] no frame-relay map protocol protocol-address</pre> <p>Syntax Description:</p> <ul style="list-style-type: none"> <li><i>protocol</i>—Supported protocol, bridging, or logical link control keywords: <b>appletalk</b>, <b>decnet</b>, <b>dlsW</b>, <b>ip</b>, <b>ipx</b>, <b>llc2</b>, <b>rsrb</b>, <b>vines</b>, and <b>xns</b>.</li> <li><i>protocol-address</i>—Destination protocol address.</li> <li><i>dlcI</i>—DLCI number used to connect to the specified protocol address on the interface.</li> <li><i>broadcast</i>—(Optional) IETF<sup>2</sup> form of Frame Relay encapsulation. Used when the router or access server is connected to another vendor's equipment across a Frame Relay network.</li> <li><i>cisco</i>—(Optional) Cisco encapsulation method.</li> </ul>
Address mapping error (continued)	<ul style="list-style-type: none"> <li><i>payload-compress packet-by-packet</i>—(Optional) Enables FRF.9 compression using the <b>Stacker</b> method.</li> <li><i>compress frf9 stac</i>—If the CSA is not available, compression is performed in the software installed on the VIP2 (distributed compression). If the router contains a CSA,<sup>3</sup> compression is performed in the CSA hardware (hardware compression). If the VIP2 is not available, compression is performed in the router's main processor (software compression).</li> <li><i>hardware-options</i>—(Optional) One of the following keywords: <b>distributed</b>, <b>software</b>, or <b>csa</b>.</li> </ul>

**Table 13-3 VINES: Client on Serverless Network Cannot Connect to Server over PSN (continued)**

Possible Problem	Solution
PVC <sup>4</sup> is not set up	<ol style="list-style-type: none"> <li>1. Use the <b>show running-config</b> privileged exec command to view the configuration of the router. Make sure a PVC is set up between the routers on each side of the PSN using the <b>x25 pvc n vines address</b> interface configuration command.</li> <li>2. If the command is not present, add it to the configuration.</li> </ol>
VINES broadcasts are not forwarded across the PSN	<ol style="list-style-type: none"> <li>1. Use the <b>show running-config</b> command to examine the configuration of the router. Make sure the <b>vines propagate</b> interface configuration command is configured on the serial interface of the router that provides the serverless packet-switched node service.</li> <li>2. If the command is not present, add it to the configuration.</li> </ol>
VINES broadcasts not forwarded to all router interfaces	<ol style="list-style-type: none"> <li>1. Use the <b>show running-config</b> privileged exec command to view the router configuration. Check whether the <b>vines serverless broadcast</b> interface configuration command is configured on the router.</li> <li>2. If the command is not present, configure the router using the <b>vines serverless broadcast</b> command. This command configures the router to always flood VINES broadcasts on all interfaces.</li> </ol> <p><b>Note:</b> The <b>vines serverless broadcast</b> command is enabled by default in Cisco IOS Release 10.3 and later.</p>

1. DLCI = Data Link Connection Identifier
2. IETF = Internet Engineering Task Force
3. CSA = compression service adapter
4. PVC = permanent virtual circuit

