



Configuring Banyan VINES

This chapter describes how to configure Banyan VINES and provides configuration examples. For a complete description of the VINES commands in this chapter, refer to the “Banyan VINES Commands” chapter in the *Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS, and XNS Command Reference* publication. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the [“Identifying Supported Platforms”](#) section in the “Using Cisco IOS Software” chapter.



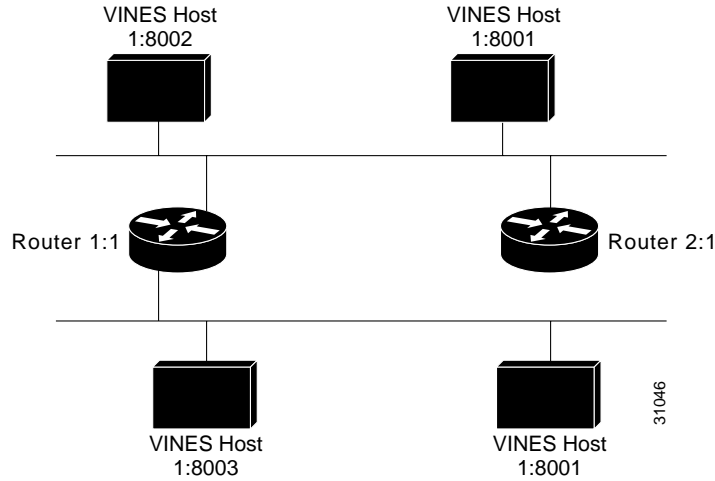
Note

Not all Cisco access servers support Banyan VINES. For more information, refer to the release notes for the current Cisco IOS release.

VINES Addresses

VINES network-layer addresses are 48-bit addresses that consist of a network number (better described as a server number) and a subnetwork number (better described as a host number). In this document, VINES addresses are expressed in the format *network:host*.

The network number identifies a VINES logical network, which consists of a single server and a group of client nodes. The network number is 32 bits (4 bytes) long, and is the serial number of the service node. [Figure 3](#) shows two logical networks: network 1 and network 2.

Figure 3 VINES Logical Network

The subnetwork number is 16 bits (2 bytes) long. For service nodes, the subnetwork number is always 1. For client nodes, it can have a value from 0x8001 through 0xFFFE.

The following is an example of a VINES network address:

```
3000577A:0001
```

In this address, the server number (or more specifically, the serial number of the service node) is 3000577A and the host number is 0001, indicating that this is a service node. Both portions of the address are expressed in hexadecimal format.

VINES Configuration Task List

To configure VINES routing, perform the tasks in the following sections:

- [Configuring VINES Routing](#) (Required)
- [Controlling Access to the VINES Network](#) (Optional)
- [Configuring VINES Network Parameters](#) (Optional)
- [Configuring VINES over WANs](#) (Optional)
- [Configuring VINES Routing Between Virtual LANs](#) (Optional)
- [Monitoring and Maintaining the VINES Network](#) (Optional)

See the “[VINES Configuration Examples](#)” section at the end of this chapter for configuration examples.

Configuring VINES Routing

To configure VINES routing, perform the tasks in the following sections. Response to Address Resolution Protocol (ARP) requests and serverless support are enabled by default for VINES serverless networks.

- [Enabling VINES Routing on the Router](#) (Required)
- [Enabling VINES Routing on an Interface](#) (Required)

- [Enabling Concurrent Routing and Bridging](#) (Optional)
- [Enabling VINES on Serverless Networks](#) (Optional)

Enabling VINES Routing on the Router

To enable VINES routing on the router, use the following command in global configuration mode:

Command	Purpose
Router(config)# vines routing [address]	Enables VINES Routing Table Protocol (RTP) routing on the router.

Enabling VINES routing on the router or access server starts the VINES RTP by default.

To enable Sequenced Routing Update Protocol (SRTP), use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# vines routing [address]	Enables VINES RTP routing on the router.
Step 2	Router(config)# vines srtp-enabled	Enables VINES SRTP routing on the router.

For an example of how to enable VINES routing, see the “[Typical VINES Network Configuration Example](#)” section at the end of this chapter.

Enabling VINES Routing on an Interface

After you have enabled VINES on the router, enable it on each interface that will handle VINES traffic. When you enable VINES processing on a specified interface, you can optionally set the metric for that interface. The metric sets the distance to another router or client accessible through that interface. The routing table uses metrics to determine which interface provides the best routing path. If you do not specify a metric, the system automatically chooses a reasonable value that is based on the interface type. The metrics are chosen to match as closely as possible the numbers that a Banyan server would choose for the same type and speed of interface.

To enable VINES routing on an interface other than a serial interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# vines metric [whole [fractional]]	Enables VINES routing on an interface.

To enable VINES routing on a serial interface, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Router# show interface	Determines the bandwidth of the interface.
Step 2	Router(config-if)# vines metric [whole [fractional]]	Enables VINES routing, explicitly setting the metric.

For a list of metric values, refer to the “Banyan VINES Commands” chapter in the *Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS, and XNS Command Reference* publication.

Enabling Concurrent Routing and Bridging

You can route VINES on some interfaces and transparently bridge it on other interfaces simultaneously. To enable this type of routing, you must enable concurrent routing and bridging. To enable concurrent routing and bridging for the router, use the following command in global configuration mode:

Command	Purpose
Router(config)# bridge crb	Enables concurrent routing and bridging for the router.

Enabling VINES on Serverless Networks

No special configuration is necessary for serverless Banyan VINES networks such as separate networks of clients and servers connected by routers. On serverless networks, Cisco IOS software provides special processing for certain broadcast packets and certain packets directed at the router. This type of processing allows clients on the serverless network to find the services that are provided by a server on another network. This special processing is especially important when two networks, one with a server and one without a server, are connected to the same router.

Client systems on VINES networks are assigned network addresses dynamically. When a VINES client boots, it has no knowledge of its address or preferred server. Immediately after it initializes its hardware interface, the client sends a broadcast request asking a server to provide it with a network-layer address. One of our routers will respond to this broadcast request if there are no VINES servers on the physical network segment. Cisco IOS software then assigns an address to the network client. (In previous releases, the software would not respond by default.) The software generates a unique network number for the client based on its own VINES address. If the software assigns an address to a client, the router or access server then acts as a network communication service provider for that client. A VINES file server must still be present somewhere on the network in order for the client to connect to all other network services.

For an example of how to configure VINES routing for various network topologies that include serverless networks, see the “[Serverless Network Configuration Examples](#)” section at the end of this chapter.

Controlling Access to the VINES Network

To control access to VINES networks, you create access lists and then apply them to filters on individual interfaces. An *access list* is a list of VINES network numbers that is maintained by the router. The list controls access to or from a particular interface. Access lists are useful for providing network security.

To filter routed traffic, you can use the following two types of VINES access lists:

- Standard access list—Restricts traffic based on the protocol, source address and mask, and destination address and mask. You can further restrict traffic by specifying a source and a destination port. Standard VINES access lists have numbers from 1 to 100.
- Extended access list—Restricts traffic in the same way as the standard access list, except that you can also specify masks for the source and destination ports. Extended VINES access lists have numbers from 101 to 200.

VINES has a third type of access list, called a *simple access list*, that restricts traffic based on source address and source address mask. This type of access list is used to decide from which stations to accept time and routing updates, not to filter traffic. Simple access lists have numbers from 201 to 300.

You can define the following three types of filters on VINES networks:

- Filters on packet protocol, source and destination addresses, address masks, and explicit port numbers
- Filters on packet protocol, source and destination addresses, address masks, port numbers, and port masks
- Filters on routing updates

Remember these points when you configure VINES network access control:

- You can assign only one access list to an interface.
- The conditions in the access list are applied to all outgoing packets not sourced by the router.
- Access list entries are scanned in the order you enter them. The first matching entry is used.
- An implicit *deny everything* entry is defined at the end of an access list unless you include an explicit *permit everything* entry at the end of the list.
- All new entries to an existing list are placed at the end of the list. You cannot add an entry to the middle of a list. Consequently, if you have previously included an explicit *permit everything* entry, new entries will never be scanned. The solution is to delete the access list and retype it with the new entries.

To control access to the VINES network, first create an access list and then apply an access list to an interface.

To create a VINES access list, use one or more of the following commands in global configuration mode:

Command	Purpose
Router(config)# vines access-list <i>access-list-number</i> {deny permit} <i>protocol source-address source-mask</i> [<i>source-port</i>] <i>destination-address destination-mask</i> [<i>destination-port</i>]	Creates a standard access list.
Router(config)# vines access-list <i>access-list-number</i> {deny permit} <i>protocol source-address source-mask</i> [<i>source-port source-port-mask</i>] <i>destination-address</i> <i>destination-mask</i> [<i>destination-port destination-port-mask</i>]	Creates an extended access list.
Router(config)# vines access-list <i>access-list-number</i> {deny permit} <i>source-address source-mask</i>	Creates a simple access list.

To apply a standard or extended access list to an interface, use the following command in interface configuration mode. Remember that you can apply only one access list to each interface.

Command	Purpose
Router(config-if)# vines access-group <i>access-list-number</i>	Applies a VINES access list to an interface.

For an example of how to create a VINES access list, see the “[Access List Example](#)” section at the end of this chapter.

Configuring VINES Network Parameters

To configure VINES network parameters, perform one or more of the tasks in the following sections:

- [Selecting an Encapsulation Type](#) (Optional)
- [Controlling the Display of Host Addresses](#) (Optional)
- [Controlling the Base of Host Addresses](#) (Optional)
- [Controlling RTP Routing Updates](#) (Optional)
- [Controlling RTP and SRTP Routing Updates](#) (Optional)
- [Disabling Fast Switching](#) (Optional)
- [Setting the Time](#) (Optional)
- [Enabling VINES Single Route on the Router](#) (Optional)
- [Configuring Static Routes](#) (Optional)
- [Configuring Static Paths](#) (Optional)
- [Controlling the Forwarding of Broadcast Packets](#) (Optional)

Selecting an Encapsulation Type

You can choose a MAC-level encapsulation type for each Ethernet, Token Ring, and IEEE 802.2 interface. This controls the type of encapsulation used by Cisco IOS software when sending broadcast packets.

To select an encapsulation type, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# vines encapsulation [arpa snap vines-tr]	Sets the MAC-level encapsulation type.



Note

You should not use the **vines encapsulation** command with the current versions of VINES software. This command is provided for future interoperability when Banyan begins using encapsulation types other than the current default ones.

Controlling the Display of Host Addresses

By default, you enter VINES addresses as numerical values. Also, addresses are displayed numerically in the output of the **show**, **ping**, and **trace** commands. You can assign a host name to each VINES address. Names are easier to remember and type. Assigning a host name allows you to enter the name instead of the address, and it means that the name instead of the numeric address is displayed in output.

To assign a host name to a VINES network address, use the following command in global configuration mode:

Command	Purpose
Router(config)# vines host name address	Assigns a host name to an address.

Controlling the Base of Host Addresses

By default, VINES addresses are represented as hexadecimal numbers. This applies to both the input of addresses and the representation of addresses in output from the router. You can configure Cisco IOS software to display addresses in decimal for consistency with Banyan network management displays.

Names are always preferred when printing addresses. If a name is not available, the address will be printed as a number in the base specified.

To display VINES addresses as decimal numbers, use the following command in global configuration mode:

Command	Purpose
Router(config)# vines decimal	Interprets VINES addresses in decimal.

Controlling RTP Routing Updates

You can control the routing updates sent by Cisco IOS software in the following ways:

- Control the interval at which the software sends RTP routing updates. The default interval is 90 seconds. The routing update interval should be the same on all VINES-speaking entities on the same physical network.



Note The **vines update interval** command does not apply to the SRTP routing protocol.

- Modify the way that routing information is propagated across the network. On LAN media, using the **vines update deltas** interface configuration command causes the software to stop sending and expecting periodic full routing updates. Instead, the software sends and expects a periodic empty routing update, also known as a hello message. On WAN media, using the **vines update deltas** interface configuration command causes the software to send three normally spaced full routing updates, and then cease transmission. The software does *not* send periodic hello messages.
- Disable split horizon. Normally, the software sends RTP updates that list only routes that it learned via other interfaces. In this situation, information is eliminated that is normally redundant and will be ignored by all routers receiving the update. When split horizon is disabled, routing updates sent out on a given interface will include all routers known by the router. Sending routing updates in this way is useful on X.25 and Frame Relay networks on which there is not a full-mesh topology.

To control routing update frequency and propagation, use one or both of the following commands in interface configuration mode:

Command	Purpose
Router(config-if)# vines update interval <i>seconds</i>	Changes the frequency of sending routing updates.
Router(config-if)# vines update deltas	Changes how routing information is propagated.



Note The **vines update deltas** command does not apply to the SRTP routing protocol.

To control the content of sent or received routing updates, or to control the source address of received routing updates, use one or more of the following commands in interface configuration mode:

Command	Purpose
Router(config-if)# vines input-router-filter <i>access-list-number</i>	Controls the source address of received routing information.
Router(config-if)# vines input-network-filter <i>access-list-number</i>	Filters the content of received routing information.
Router(config-if)# vines output-network-filter <i>access-list-number</i>	Filters the content of sent routing information.

To disable split horizon when generating regular periodic routing updates and to disable flash updates to indicate topology change for a changed route, use the following command in global configuration mode:

Command	Purpose
Router(config)# no vines enhancements	Disables split horizon and flash updates.

To disable split horizon on networks that are not fully connected mesh interfaces such as X.25 and Frame Relay, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# no vines split-horizon	Disables split horizon when sending routing updates.

**Note**

For routing updates only, when **vines enhancements** is enabled in global configuration mode by default, **vines split-horizon** is also enabled on the interface by default. In this case, if required, you can disable **vines split-horizon** on an interface like Frame Relay and X.25.

When **vines enhancements** is disabled in global configuration mode, **vines split-horizon** for RTP routing updates is disabled on all interfaces; however, one may still see **vines split-horizon** as enabled on the VINES interface when **show vines interface interface** command is used. Split horizon remains enabled because **vines split-horizon** on individual VINES interfaces, in addition to controlling RTP updates, also controls whether retransmission of broadcasts is permitted on the receiving interface.

**Note**

SRTP updates do not use split horizon.

Controlling RTP and SRTP Routing Updates

The VINES RTP sends several types of messages, including *redirect* messages. If Cisco IOS software detects that a suboptimal path between two nodes is being used, it sends redirect messages to the nodes to indicate the better path.

To control the frequency of redirect messages on a specified interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# vines redirect <i>[seconds]</i>	Sets the frequency of RTP and SRTP redirect messages.

Disabling Fast Switching

Fast switching allows higher throughput by switching packets using a cache created by previous packets. Fast switching also provides load sharing on a per-packet basis. Fast switching is enabled by default on all interfaces on which it is supported.

Packet transfer performance is generally better when fast switching is enabled. However, you might want to disable fast switching in order to save memory space on interface cards and to help avoid congestion when high-bandwidth interfaces are writing large amounts of information to low-bandwidth interfaces.

To disable fast switching on an interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# no vines route-cache	Disables fast switching.

Setting the Time

Banyan VINES servers synchronize time across the entire network by sending zero-hop and two-hop broadcast messages. Cisco IOS software can process and generate time-synchronization messages. It can also retrieve the local time and place it into the VINES time system (which is most useful when running Network Time Protocol (NTP) locally) and can use the VINES time system to set a local clock. It is also possible to provide the software with a list of up to 20 destinations for time messages.

To set the VINES network time, use one or more of the following commands in global configuration mode:

Command	Purpose
Router(config)# vines time participate	Enables the sending of time-synchronization messages across a VINES network.
Router(config)# vines time set-system	Periodically synchronizes the time of the router with the VINES network time.
Router(config)# vines time services	Provides time services for VINES clients and enables participation in the synchronization of time across a VINES network.
Router(config)# vines time use-system	Periodically synchronizes the VINES network time with the time of the router.
Router(config)# vines time access-group <i>access-list-number</i>	Accepts time updates from the stations permitted by the specified simple access list.
Router(config)# vines time destination <i>address</i>	Sends time updates only to the specified station.

For an example of how to set VINES time, see the “[Time of Day Service Example](#)” section at the end of this chapter.

Enabling VINES Single Route on the Router

VINES single route maintains a single route to the server. VINES single route can be enabled at any time after VINES routing has been enabled. If a VINES connection experiences slow performances due to low window size or cannot handle out-of-sequence packets, enable VINES single route. To set VINES single route on the router, use the following command in global configuration mode:

Command	Purpose
Router(config)# vines single-route	Enables VINES single route to maintain a single route to the server.

Configuring Static Routes

VINES uses the RTP to determine the best path when several paths to a destination exist. RTP then dynamically updates the routing table. However, you might want to add static routes to the routing table to explicitly specify paths to certain destinations.

The decision to use a static route or a dynamic route is always determined by the relative metric numbers. Be careful when assigning static routes. If a static route is assigned with a better metric than the dynamic routes, and the links associated with the static routes are lost, traffic may stop being forwarded, even though an alternative route might be available.

To add a static route to the routing table, use the following command in global configuration mode:

Command	Purpose
Router(config)# vines route <i>number address [whole [fractional]]</i>	Adds a static route to the routing table.

You can configure static routes that can be overridden by dynamically learned routes. These types of static routes are referred to as *floating static routes*. You can use a floating static route to create a path of last resort that is used only when no dynamic routing information is available.

To avoid the possibility of a routing loop occurring, by default, floating static routes are not redistributed into other dynamic protocols. Floating static routes must not be advertised on interfaces that are paths to the destination. To configure a floating static route, assign a metric to the static route that is worse (higher) than all dynamic routes.

To add a floating static route to the routing table, use the following command in global configuration mode:

Command	Purpose
Router(config)# vines route <i>number address [whole [fractional]]</i>	Adds a floating static route to the routing table.

Configuring Static Paths

You can specify static paths to neighbor stations on the network. Specifying static paths in this way is useful for testing VINES networks with test equipment that does not generate hello packets.

To add a static path to a neighbor station, use the following command in interface configuration mode:

Command	Purpose
<code>Router(config-if)# vines neighbor address mac-address encapsulation [whole [fractional]]</code>	Adds a static path to the neighbor station.

Controlling the Forwarding of Broadcast Packets

Normally, Cisco IOS software decides whether to forward a broadcast packet on an interface based on the presence of local servers and on the settings of both the “hop count” and “class” fields of the VINES IP header. If there are any local servers present, the software follows the normal rules of VINES IP and forwards the broadcast after examining both the “hop count” and “class” fields. If no local servers are present, then the “class” field is ignored when making the forwarding decision. You can override this default behavior in either of two ways. The first override is to have the software always ignore the “class” field and make the broadcast forwarding decision solely based on hop count. The second override is to have the software never ignore the “class” field and always make the broadcast forwarding decision based upon both the “class” and “hop count” fields.

To have the software modify how it forwards broadcast packets, use one of the following commands in interface configuration mode:

Command	Purpose
<code>Router(config-if)# no vines propagate</code>	Ensures that the software never ignores the “class” field when forwarding broadcast packets.
<code>Router(config-if)# vines propagate</code>	Ensures that the software always ignores the “class” field when forwarding broadcast packets.

Configuring VINES over WANs

You can configure VINES over X.25, Frame Relay, and SMDS networks by configuring the address mappings as described in the appropriate chapter of the *Cisco IOS Wide-Area Networking Configuration Guide*. You can also configure VINES over HDLC and PPP; address maps are not necessary for these two protocols. You can fastswitch VINES over serial interfaces configured for HDLC, Frame Relay, PPP, Switched Multimegabit Data Service (SMDS), and ATM.

Configuring VINES Routing Between Virtual LANs

Banyan VINES can be routed over virtual LAN (VLAN) subinterfaces using the Inter-Switch Link (ISL) encapsulation protocol. Full-feature Cisco IOS software is supported on a per-VLAN basis, allowing standard Banyan VINES capabilities to be configured on VLANs. Refer to the *Cisco IOS Switching Services Configuration Guide* for detailed information about configuring Banyan VINES over ISL between virtual LANs.

Monitoring and Maintaining the VINES Network

To monitor and maintain a VINES network, use one or more of the following commands in EXEC mode:

Command	Purpose
Router> clear vines cache [<i>interface interface</i> <i>neighbor address</i> <i>server network</i> <i>counters</i>]	Deletes entries from the VINES fast-switching cache table.
Router> clear vines ipc <i>number</i>	Deletes VINES IPC connection blocks from the router.
Router> clear vines neighbor { <i>address</i> *}	Deletes entries from the neighbor table.
Router> clear vines route { <i>network</i> *}	Deletes network addresses from the routing table.
Router> clear vines traffic	Clears the VINES-related traffic statistics displayed by the show vines traffic command.
Router> ping vines [<i>address</i>]	Sends datagrams to a host to determine network connectivity.
Router> show vines access [<i>access-list-number</i>]	Displays the VINES access lists currently defined.
Router> show vines cache [<i>address</i> <i>interface type number</i> <i>neighbor address</i> <i>server network</i>]	Displays the contents of the VINES fast-switching cache table.
Router> show vines host [<i>name</i>]	Displays the entries in the VINES host name table.
Router> show vines interface [<i>type number</i>]	Displays VINES-related interface settings.
Router> show vines ipc	Displays information about any currently active IPC connections.
Router> show vines neighbor [<i>address</i> <i>interface type number</i> <i>server number</i>]	Displays the contents of the VINES neighbor table.
Router> show vines route [<i>number</i> <i>neighbor address</i> <i>metric</i>]	Displays the contents of the VINES routing table.
Router> show vines service [<i>fs</i> <i>nsm</i> <i>ss</i> <i>vs</i>]	Displays information about the application layer support of the router.
Router> show vines traffic [<i>type number</i>]	Displays the statistics about VINES protocol traffic.
Router> trace [<i>vines</i> <i>oldvines</i>] [<i>address</i>]	Determines the path a packet takes when traversing a VINES network.

If you find that two routers have the same VINES network address, you can have the routers dynamically recompute their addresses. To recompute the addresses on each of the two routers, use the following command in global configuration mode:

Command	Purpose
Router(config)# vines routing recompute <i>address</i>	Dynamically redetermines the address of the router.

VINES Configuration Examples

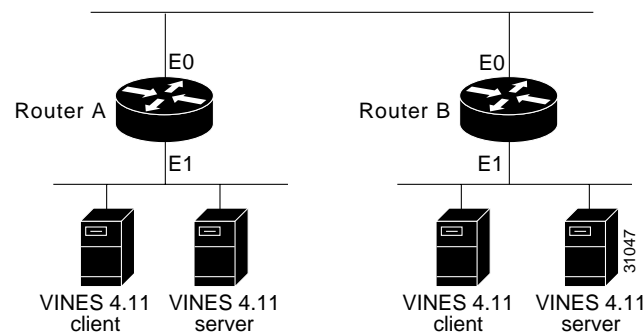
For help in configuring VINES routing on your network, use the configuration examples in the following sections:

- [Typical VINES Network Configuration Example](#)
- [Serverless Network Configuration Examples](#)
- [Access List Example](#)
- [Time of Day Service Example](#)

Typical VINES Network Configuration Example

Figure 4 illustrates how to configure a simple VINES network.

Figure 4 VINES Simple Configuration

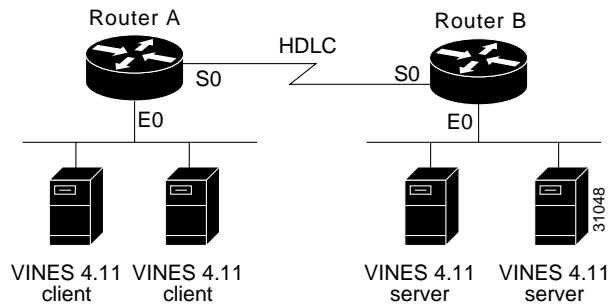


The following example shows how to configure Router A and Router B:

```
vines routing
!
interface ethernet 0
vines metric 2
!
interface ethernet 1
vines metric 2
```

Serverless Network Configuration Examples

The following examples show how to configure VINES routing for various network topologies that include serverless networks. The first example illustrates how to configure a simple serverless network (see Figure 5). Note that this configuration is no longer different from the configuration of a network that has servers.

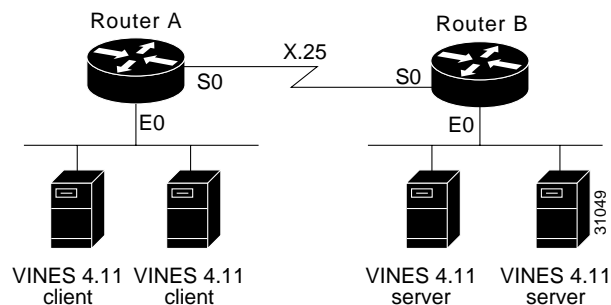
Figure 5 VINES Serverless Configuration**Configuration for Router A**

```
vines routing
!
interface ethernet 0
  vines metric 2
!
interface serial 0
  vines metric 45
```

Configuration for Router B

```
vines routing
!
interface ethernet 0
  vines metric 2
!
interface serial 0
  vines metric 45
```

The configuration in [Figure 6](#) has an X.25 interface instead of an HDLC serial line, and it also has multiple versions of VINES software running simultaneously. Again, note that there is no longer any difference between this configuration and that of a network that has servers.

Figure 6 VINES Serverless X.25 Configuration**Configuration for Router A**

```
vines routing
!
interface ethernet 0
  vines metric 2
!
interface serial 0
```

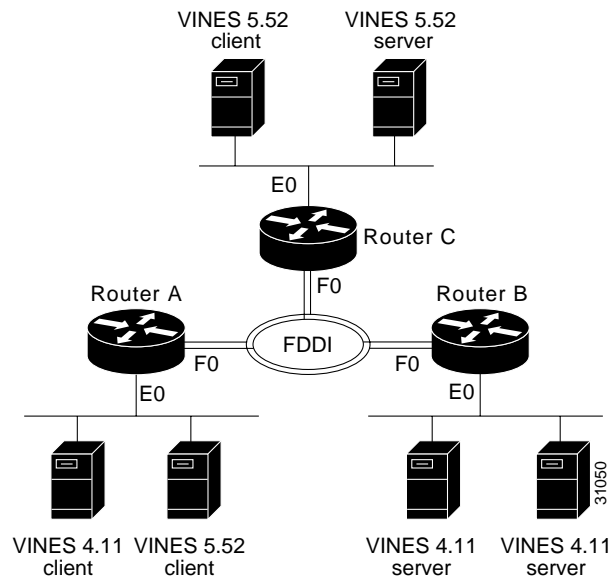
```
vines metric 55
```

Configuration for Router B

```
vines routing
!
interface ethernet 0
  vines metric 2
!
interface serial 0
  vines metric 55
```

The configuration in Figure 7 has a FDDI interface instead of a serial line. It also has the servers for the different VINES versions on different physical networks and has a requirement that the clients be able to run any VINES version.

Figure 7 VINES Complex Serverless Configuration



The best way to configure this topology is as follows:

Configuration for Router A

```
vines routing
!
interface ethernet 0
  vines metric 2
  vines serverless broadcast
!
interface fddi 0
  vines metric 1
```

Configuration for Router B and Router C

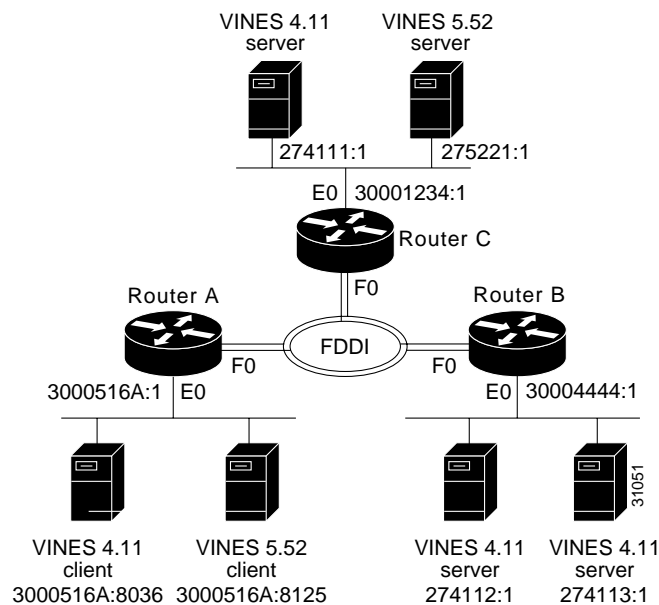
```
vines routing
!
interface ethernet 0
  vines metric 2
!
interface fddi 0
  vines metric 1
```

The **broadcast** keyword on the **vines serverless** command on server A causes it to forward packets onto the FDDI ring as broadcasts, instead of sending them to either Router B or Router C. Forwarding packets onto the FDDI ring as broadcasts allows the **default serverless processing** on both routers to forward the frame from the FDDI ring to the Ethernet network.

Access List Example

Figure 8 illustrates how to configure an access list that filters all packets between two VINES servers. For this example, the servers in the upper-left and lower-right corners are configured.

Figure 8 VINES Access List Configuration



The following example shows how to configure Router B:

```
vines routing
vines access-list 1 deny IP 274113:1 0:0 274111:1 0:0
vines access-list 1 permit IP 0:0 FFFFFFFF:FFFF 0:0 FFFFFFFF:FFFF
!
interface ethernet 0
vines metric 2
vines access-group 1
!
interface fddi 0
vines metric 1
```

The first line in the access list prohibits any communication between the two servers, and the second line allows all other communication to pass through the router.

The following example shows how to allow only mail traffic between these two servers. Port 4 is the VINES Mail port.

```
vines routing
vines access-list 101 permit IPC 274113:1 0:0 0 FFFF 274111:1 0:0 4 0
vines access-list 101 permit IPC 274111:1 0:0 4 0 274113:1 0:0 0 FFFF
vines access-list 101 deny IP 274111:1 0:0 274113:1 0:0
```

```

vines access-list 101 permit IP 0:0 FFFFFFFF:FFFF 0:0 FFFFFFFF:FFFF
!
interface ethernet 0
  vines metric 2
  vines access-group 101
!
interface fddi 0
  vines metric 1

```

The first line in the access list allows mail messages being sent from the server in the lower right to the server in the upper left. The second line allows mail messages in the other direction. The third line prohibits all other communication between these two servers. The last line allows all other communication to pass through the router.

Time of Day Service Example

The following example, using the configuration shown in [Figure 8](#), shows how to configure the “time of day” support in a VINES network. Router C is configured as an NTP server and will provide time to the VINES network.

Configuration for Router A and Router B

```

vines routing
!
interface ethernet 0
  vines metric 2
!
interface fddi 0
  vines metric 1
!
vines access-list 201 permit 30001234:1 0:0
vines access-list 201 deny 0:0 FFFFFFFF:FFFF
vines time access-group 201
vines time participate
vines time set-system

```

Configuration for Router C

```

vines routing
!
interface ethernet 0
  vines metric 2
!
interface fddi 0
  vines metric 1
!
ntp peer 128.9.2.129
vines time participate
vines time use-system

```

The access list on Router A and Router B is not necessary. This access list prevents the routers from learning the time from anyone other than Router C. Learning the time from only Router C is not very important because each time message from Router C will override any time that has been previously learned (due to the **vines time use-system** command).

