

RIVERSTONE NETWORKS SWITCH ROUTER QUICK START GUIDE

1 USING THE CLI

This section provides information about the Riverstone Networks Switch Router Command Line Interface (CLI). The CLI is the primary interface through which Riverstone switch routers are configured. The CLI provides a text-based environment, accessed through either the local console or through a Telnet session.

**Note**

For a full description of all Riverstone Networks switch route CLI commands and configuring your switch router, refer to the “*Riverstone Networks Switch Router Command Line Interface Reference Manual*” and the “*Riverstone Networks Switch Router User Guide*.”

1.1 Command Modes

The CLI provides access to four different command modes: User, Enable, Config, and Boot PROM. Each command mode provides a group of related commands. This section describes how to access and list the commands available in each command mode.

User Mode

User mode is the initial console mode when first logging into the switch router. The User commands available are a subset of those available in the Enable mode. In general, the User commands allow you to display basic information and use basic utilities such as ping.

The User mode command prompt consists of the switch router name, followed by the angle bracket (>), as shown below:

```
rs>
```

The default name is **rs** for Riverstone Networks Switch Routers or **es** for Riverstone Networks Layer-3 Switches, such as the ES 10170. Refer to [Section 3, "Setting the Basic System Information."](#) for the procedures for changing the system name.

Enable Mode

Enable mode provides more facilities than User mode. Enable mode allows you to display critical features, including router configuration, access control lists, and SNMP statistics. To enter Enable mode from the User mode, enter the command **enable** (or **en**), then supply the password when prompted.

The Enable mode command prompt consists of the switch router name followed by the pound sign(#):

```
rs#
```

To exit Enable mode and return to User mode, either type **exit** and press Return, or press Ctrl+Z.

Configure Mode

Configure mode provides the capabilities to configure all features and functions on the switch router. To enter Configure mode, enter the command **config** from Enable mode.

The Configure mode command prompt consists of the switch router name followed by (**config**) and a pound sign (#):

```
rs(config)#
```

To exit Configure mode and return to Enable mode, either type **exit** and press Return, or press Ctrl+Z.

Boot PROM Mode

If your RS does not find a valid system image on the external PC flash, the system might enter programmable read-only memory (PROM) mode. If this occurs, reboot the switch router by entering the **reboot** command at the boot PROM prompt. If the system fails to reboot successfully, please call Riverstone Networks Technical Support to resolve the problem.

For information on how to upgrade the boot PROM software and boot using the upgraded image, see the “*Riverstone Switch Router Getting Started Guide*.”

1.2 Line Editing Commands

The Riverstone Networks switch router CLI provides line editing capabilities that are similar to Emacs, a Unix text editor. For example, you can use certain line editing keystrokes to move forward or backward on a line, delete or transpose characters, and delete portions of a line. To use the line editing commands, use a VT-100 terminal or terminal emulator.

[Table 1](#) details the line editing commands that can be used with CLI.

Table 1-1 CLI line editing commands

Command	Resulting Action
Ctrl-a	Move to beginning of line
Ctrl-b	Move back one character
Ctrl-c	Abort current line
Ctrl-d	Delete character under cursor
Ctrl-e	Move to end of line
Ctrl-f	Move forward one character
Ctrl-g	Abort current line
Ctrl-h	Delete character just prior to the cursor
Ctrl-i	Insert one space (tab substitution)
Ctrl-j	Carriage return (executes command)
Ctrl-k	Kill line from cursor to end of line
Ctrl-l	Refresh current line
Ctrl-m	Carriage return (executes command)
Ctrl-n	Next command from history buffer
Ctrl-o	None
Ctrl-p	Previous command from history buffer
Ctrl-q	None
Ctrl-r	Refresh current line
Ctrl-s	None
Ctrl-t	Transpose character under cursor with the character just prior to the cursor
Ctrl-u	Delete line from the beginning of line to cursor
Ctrl-v	None
Ctrl-w	None
Ctrl-x	Move forward one word
Ctrl-y	Paste back what was deleted by the previous Ctrl-k or Ctrl-w command. Text is pasted back at the cursor location
Ctrl-z	If inside a subsystem, it exits back to the top level. If in Enable mode, it exits back to User mode. If in Configure mode, it exits back to Enable mode.
ESC-b	Move backward one word
ESC-d	Kill word from cursor's current location until the first white space.

Table 1-1 CLI line editing commands (Continued)

Command	Resulting Action
ESC-f	Move forward one word
ESC-BackSpace	Delete backwards from cursor to the previous space (essentially a delete-word-backward command)
SPACE	Attempts to complete command keyword. If word is not expected to be a keyword, the space character is inserted.
!*	Show all commands currently stored in the history buffer.
!#	Recall a specific history command. '#' is the number of the history command to be recalled as shown via the '!' command.
"<string>"	Opaque strings may be specified using double quotes. This prevents interpretation of otherwise special CLI characters.

1.3 <port-list> Syntax

The <port-list> parameter of the port set commands is a comma-separated list of ports to be configured. Wildcard or range sequences may be used as the last field of a port name. For example, when specifying a channel, the channel number may be a wildcard. However, when specifying a virtual circuit, the virtual circuit can be a wildcard or range, but a channel number must be explicitly specified.



Note Where appropriate, the keyword **all-ports** may also be used, if a command is to be applied to all the relevant ports.

The syntax for each <port-list> element is:

- For channelized ports:
 <port-type>.<slot>.<port>[:<channel-number>][.<vc>]
- For other ports, including unchannelized T1/E1:
 <port-type>.<slot>.<port>[.<vc>]
- For SRP ports:
 <port-type>.<slot>.<port>[.<side>]

<port-type>

The type of interface being configured, which can be one of the types shown in [Table 1](#).

Table 1-1 Port Types

Port Type	Description
at	An Asynchronous Transfer Mode (ATM) interface.
e1	A G.703 European level-1 interface, with optional G.704/G.706 framing (Channelized E1).
e3	A G.703 European level-3 interface, with optional G.751 framing. (Channelized E3 or Clear Channel E3).
et	An Ethernet interface.
gi	A Gigabit Ethernet interface.
hs	A HSSI interface.
se	A Serial interface.
so	A Packet Over SONET (POS) interface.
sr	A Spatial Reuse Protocol (SRP) port. An SRP interface contains two sides (A and B). Some commands allow the user to affect only one side so command syntax includes the use of .a and .b suffixes to identify a specific side. Example: sr.5.1.b . To address an SRP interface as a single entity, use an identifier such as sr.5.1 (no .a or .b suffix).
t1	A DS-1 or DSX-1 interface, as specified in ANSI T1.403/T1.408, and so on (Channelized T1).
t3	A DS-3 interface, as specified in ANSI-T1.404 (Channelized T3 or Clear Channel T3).

You can create pseudo devices, with port types shown in [Table 1](#).

Table 1-2 Pseudo Device Port Types

Port Type	Description
mp	A Multilink PPP Bundle.
st	A SmartTRUNK.

2 CONFIGURATION CHANGES AND SAVING THE CONFIGURATION FILE

The switch router uses three special configuration files:

Table 2-1 Configuration file contents

File	Descriptions
Scratchpad	The configuration commands you have entered during a management session. These commands do not become active until you explicitly activate them. Because some commands depend on other commands for successful execution, the switch router scratchpad simplifies system configuration by allowing you to enter configuration commands in any order, even when dependencies exist. When you activate the commands in the scratchpad, the switch router sorts out the dependencies and executes the commands in their proper sequence.
Active	The commands from the Startup configuration file and any configuration commands that you have made active from the scratchpad.
Startup	The configuration file that the switch router loads into the Active configuration when the system is powered on.

**Caution**

The active configuration remains in effect only during the current power cycle. If you power off or reboot the router without saving the active configuration changes to the Startup configuration file, the changes are lost.

2.1 Activating the Configuration Commands in the Scratchpad

Use the following procedure to activate the configuration commands in the scratchpad.

1. Ensure that you are in Enable mode by entering the **enable** command in the CLI.
2. Ensure that you are in Configure mode by entering the **configure** command in the CLI.
3. Enter the following command:

```
save active
```

The CLI displays the following message:

```
Do you want to make the changes Active? [y]
```

4. Enter **y** to activate the changes.



Note If you exit the Configure mode (by entering the **exit** command or by pressing Ctrl+z), the CLI will ask you whether you want to make active the changes in the scratchpad. If you do not make the changes in the scratchpad active, the changes will be lost when you log out.

2.2 Viewing the Current Configuration

To view the current, active configuration:

1. Ensure that you are in Enable mode by entering the **enable** command.
2. Enter the following command to display the status of each command line:

```
rs# system show active-config
```



Note Remember that the Active configuration contains both the Startup configuration and any configuration changes that you've made active in the current configuration session.

The CLI displays the Active configuration file with the following possible annotations:

- Commands without errors are displayed without any annotation.
- Commands with errors are annotated with an “**E:**.”
- If a particular command has been applied such that it can be expanded on additional interfaces/line cards, it is annotated with a “**P:**.” For example, if you enable STP on all ports on the switch router, but the switch router contains only one line card, the configuration lines that enable STP will be applied to all ports on all other line cards as they are added to the system.

A command like **stp enable port et.*.*** would be displayed as follows:

```
P: stp enable port et.*.*
```

If you update the configuration file to state specifically which Ethernet ports STP is enabled on, the “**P:**” annotation in the above command line would disappear.

2.3 Saving the Active Configuration to the Startup Configuration File

Use the following procedure to save Active configuration changes into the Startup configuration file so that the switch router remembers and uses the changes when you reboot the software.

1. Enter the following command from Configure mode:

```
rs(config)# save startup
```

2. When the CLI displays the following message, enter **y** to save the changes:

```
Are you sure you want to overwrite the Startup configuration [no]? y  
%CONFIG-I-SAVED, configuration saved to Startup configuration.  
rs(config)#
```

Alternately, to save the Active configuration to the Startup configuration from Enable mode, perform the following steps:

1. Ensure that you are in Enable mode by entering the **enable** command in the CLI.
2. Enter the following command to copy the Active configuration to the Startup configuration:

```
rs# copy active to startup
```

3. When the CLI displays the following message, enter **yes** to save the changes.

```
Are you sure you want to overwrite the Startup configuration [no]? y  
%CONFIG-I-WRITTEN, File copied successfully
```

The new configuration changes are added to the Startup configuration file located in the Control Module's boot flash.

To view the Startup configuration:

1. Ensure that you are in Enable mode by entering the **enable** command.
2. Enter the following command to display the Startup configuration:

```
rs# system show startup-config
```

3 SETTING THE BASIC SYSTEM INFORMATION

Follow the procedures in this section to set the following system information:

- System time and date
- System name, location, and contact name (the person to contact regarding this router)
- Login banner
- IP address for the management port on the Control Module
- Default Gateway



Note Some of the commands in this procedure accept a string value. String values can be up to a maximum of 255 characters in length including blank spaces. Surround strings that contain blanks with quotation marks (for example: "**string with internal blanks**").

1. Enter the **enable** command to reach Enable mode in the CLI.
2. Enter the following commands to set the system time and date and to verify your settings.

```
system set date year <number> month <month-name> day <day> hour <hour> minute <minute>
second <second>

system show date
```

Here is an example:

```
rs# system set date year 2003 month march day 27 hour 11 minute 54
second 0
Time changed to: Mon Mar 27 11:54:00 2003
rs# system show date
Current time: Mon Mar 27 11:54:04 2003
```

3. Enter the **configure** command to get to Configure mode in the CLI. The following commands can be entered only from Configure mode.
4. Enter the following commands to set the system name, location, and contact information:

```
system set name <string>
system set location <string>
system set contact <string>
```

Here is an example:

```
rs(config)# system set name rs
rs(config)# system set location "Houston, TX"
rs(config)# system set contact "John Smith"
```

5. Use the **system set login-banner** command to set the login banner on the switch router. The login banner is displayed whenever a new user connects to the system.

Here is an example:

```
rs(config)# system set login-banner "Welcome to the switch router"
```

6. Use the **interface add ip** command to set the IP address and netmask for the en0 Ethernet interface. The en0 Ethernet interface is used by the management port on the Control Module.

Here is an example:

```
rs(config)# interface add ip en0 address-netmask 16.50.11.22/16
```

If your management workstation is not on the same subnet as the RS it is managing, you will need to configure a static route as described in step 7.

7. Use the **ip add route** command to add static routes to the RS.

For example, if you wanted to create a static route for the en0 Ethernet interface configured in Step 6 to a management workstation that isn't on the same subnet, you would have to add a static route to the subnet where your management workstation is and specify a gateway router on your subnet. The following command creates a static route between the subnet where your management workstation is located (16.50.6.0) and a gateway that resides on the same subnet as the RS (16.50.11.35).

```
rs(config)# ip add route 16.50.6.0/16 gateway 16.50.11.35
```

Other ports on the RS can use a default gateway for routing to other subnets as shown in the next step.

8. Use the **ip add route default gateway** command to set the default gateway for the RS.

Here is an example:

```
rs(config)# ip add route default gateway 10.4.1.1
```

**Note**

The en0 interface (reserved for the management port on the Control Module) cannot be configured to use the default gateway. To access the management port from a remote location, you must configure a static route between the RS and the subnet where your management workstation is located.

9. To activate the system commands entered in the previous steps, use the following command:

```
save active
```

The CLI displays the following message:

```
Do you want to make the changes Active? [y] y
```

10. Enter “y” to activate the changes.

11. To display the Active configuration, exit the Configuration mode, then enter the following command.

```
system show active-config
```

Alternately, you can view the Active configuration within Configure mode by entering the **show** command.

Here is an example:

```
rs(config)# show
Running system configuration:
    !
    ! Last modified from Console on Mon Jan 25 11:55:35 2001
    !
  1 : interface add ip en0 address-netmask 10.0.0.1/24
    !
  2 : system set name "rs"
  3 : system set location "Houston, TX"
  4 : system set contact "John Smith"
  5 : system set login-banner "Welcome to the switch router"
```

12. Save the Active configuration to the Startup configuration file using the following command from Enable mode:

```
copy active to startup
```

13. When the CLI displays the following message, enter **y** to save the changes to the Startup configuration file:

```
Are you sure you want to overwrite the Startup configuration [no]? y
%CONFIG-I-WRITTEN, file copied successfully
rs#
```

4 SETTING UP PASSWORDS

You can password protect CLI access to the switch router by setting up passwords for User mode access, Enable mode access, and Diag mode access. Users who have a User password but not an Enable password can use only the commands available in User mode. Users with an Enable password can use commands available in the Enable and Configure modes, as well as the commands in User mode.

In addition, you can set up the switch router for TACACS, TACACS+, and/or RADIUS authentication by a TACACS or RADIUS server. Procedures for configuring the router for TACACS and RADIUS can be found in the [Riverstone RS Switch Router User Guide](#).

To add password protection to the CLI, use the following procedure.

1. Ensure that you are in Enable mode by entering the **enable** command in the CLI.

2. Ensure that you are in Configure mode by entering the **configure** command in the CLI.
3. Type the following command for each password you want to set:

```
system set password login|enable|diag <string>|none
```

4. Use the **show** command to examine the commands you just entered.
5. Use the **save active** command to activate the commands.
6. Use the **show** command within Configure mode to verify the active changes.

Here is an example:

```
rs(config)# show
Running system configuration:
!
! Last modified from Console on Mon Jan 25 11:55:35 2001
!
1 : interface add ip en0 address-netmask 10.0.0.1/24
!
2 : system set name "rs"
3 : system set location "Houston, TX"
4 : system set contact "John Smith"
5 : system set login-banner "Welcome to the switch router"
6 : system set hashed-password login jNIssH c976b667e681d03ccd5fc527f219351a
7 : system set hashed-password enable zcGzbO 5d1f73d2d478ceaa062a0b5e0168f46a
8 : system set hashed-password diag jdfbyp 67e681d3d2d478cf21935a0b5e016f2193
```

Notice that the passwords are shown in the Active configuration in an encrypted format. Passwords also appear this way in the Startup configuration. To keep your passwords secure, the router does not have a command for displaying passwords in an unencrypted format.



Caution Test all new passwords before saving the active configuration to the Startup configuration file.

If you forget your passwords, consult the *Riverstone Switch Router Getting Started Guide* and follow the procedures to regain access to your switch router.

5 SETTING UP SNMP

To use SNMP to manage the RS, you need to set up an SNMP community and specify the IP address of the target host for SNMP traps. Otherwise, the RS's SNMP agent runs in local trap process mode, unless disabled using the **snmp stop** command.

Use the following procedure to add the SNMP community string, specify the target host for traps, and the trap interface.

1. Ensure that you are in Enable mode by entering the **enable** command in the CLI.
2. Ensure that you are in Configure mode by entering the **configure** command in the CLI.
3. Use the following commands to add an SNMP community string and set a target host IP address for the traps:

```
rs(config)#snmp set community <community-name> privilege read|read-write
rs(config)#snmp set target <IP-addr> community <community-name> status enable|disable
```

**Note**

If the IP address of the trap target is more than one hop away from the RS, configure the RS with a static route to the target. If the RS is rebooted, the static route allows a cold start trap to be sent to the trap target. Without a static route, the cold-start trap is lost while the routing protocols are converging.

4. Use the **save startup** command to activate the commands entered in the previous steps.

Here is an example of the commands and output for configuring SNMP and saving the changes.

```
rs# config
rs(config)# snmp set community public privilege read-write
rs(config)# snmp set target 16.50.11.12 community public status enable
rs(config)# save startup
Are you sure you want to overwrite the Startup configuration [no]? yes

There are non-committed configuration changes. Do you want to make
these changes active and then save everything to Startup [yes]? yes

%CONFIG-I-MAKED, 2001-09-02 21:53:54 %GATED-I-RECONFIGDONE, Routing
configuration changes completed (pid 0x809eab20).
configuration saved to Startup configuration.
rs(config)#
```

By default, SNMP information is sent and received on the Control Module's en0 Ethernet port. If you want SNMP to use a different port on the RS, use the following command.

```
snmp set trap-source <interface>|<IPaddr>
```

Here is an example:

```
rs(config)# snmp set trap-source 134.152.78.192
```

SNMP will now use the port with IP address 134.152.78.192. Remember, to make this change permanent, enter the **save startup** command.

6 RIVERSTONE-CISCO OSPF CONNECTION

To set up a basic OSPF connection between a Cisco and a Riverstone router, you need to perform the following tasks:

1. Configure a loopback address. Physical interfaces may or may not be in the Up state. Unlike physical interfaces, the loopback interface is a virtual interface that is always in the Up state when the switch router is running. Using the loopback interface, instead of physical interfaces, in configuring routing protocols increases routing stability.
2. Configure the physical connections between the two routers that you would like to connect.
3. Configure OSPF on each router and set OSPF to communicate via the loopback interface.
4. (Optional) Configure any redistribution that you want to occur into/out of OSPF.

The following example configures a Riverstone router to route with a Cisco router via OSPF. [Figure 6-1](#) illustrates this topology. The complete Riverstone and relevant partial Cisco configurations follow. Configurations specifically applicable to this example are annotated and highlighted.

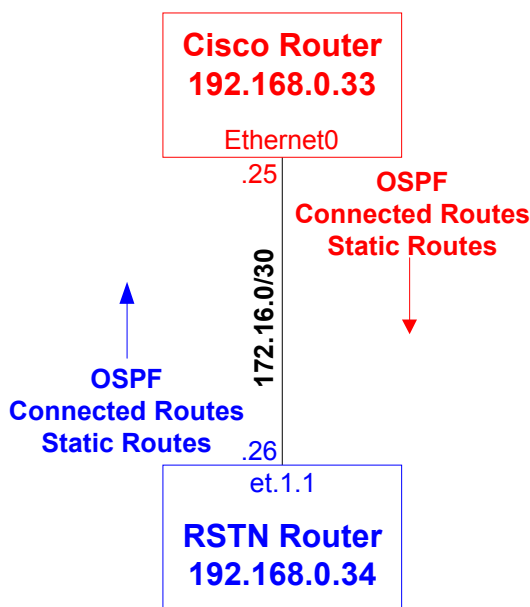


Figure 6-1 Cisco-Riverstone OSPF connect

<The relevant partial Cisco configuration follows...>

```
!  
// Configure the Loopback0 interface with an IP address  
interface Loopback0  
  ip address 192.168.0.33 255.255.255.255  
!  
// Configure the Ethernet0 interface with an IP address  
interface Ethernet0  
  description ethernet port  
  ip address 172.16.0.25 255.255.255.252  
!  
// Configure an OSPF routing process  
router ospf 1  
// Redistribute all directly-connected routes into OSPF. Since the loopback interface  
// is considered to be directly-connected, this command also redistributes the loopback  
// address into OSPF. The 'subnets' keyword is necessary to allow non-classful addresses  
// to be redistributed.  
  redistribute connected subnets  
// Redistribute all static routes into OSPF.  
  redistribute static subnets  
// Add the 172.16.0.25/30 network to the OSPF backbone area. Since the Ethernet0  
// interface is in this network, OSPF begins running on this interface.  
network 172.16.0.25 0.0.0.3 area 0.0.0.0
```

```

rs(config)# show
Running system configuration:
    !
    ! Last modified from Console on Mon Jan 25 11:55:35 2001
    !
// Configure the et.1.1 port with an IP address
1 : interface create ip ToCisco address-netmask 172.16.0.26/30 port et.1.1

// Assign the loopback address for use in OSPF routing
2 : interface add ip lo0 address-netmask 192.168.0.34/32
3 : interface add ip en0 address-netmask 10.0.0.1/24
    !
// Redistribute all directly-connected routes into OSPF. For more information on
// redistribution, see the Riverstone RS Switch Router User Guide
4 : ip-router policy redistribute from-protocol direct to-protocol ospf
// Redistribute all static routes into OSPF.
5 : ip-router policy redistribute from-protocol static to-protocol ospf
    !
// Configure a basic backbone OSPF connection to the Cisco router
6 : ospf create area backbone
// Add the loopback interface to the backbone area for use in OSPF routing
7 : ospf add stub-host 192.168.0.34 to-area backbone cost 1
// Add the physical interface that connects the Cisco Router to the backbone area
8 : ospf add interface ToCisco to-area backbone
// Start OSPF
9 : ospf start
    !
10 : system set name "rs"
11 : system set location "Houston, TX"
12 : system set contact "John Smith"
13 : system set login-banner "Welcome to the switch router"
14 : system set hashed-password login jNIssH c976b667e681d03ccd5fc527f219351a
15 : system set hashed-password enable zcGzbO 5dlf73d2d478ceaa062a0b5e0168f46a
16 : system set hashed-password diag jdfbyp 67e681d3d2d478cf21935a0b5e016f2193

```

7 WORKING WITH VLANs

Virtual LANs (VLANs) are a means of dividing a physical network into several logical (virtual) LANs. The division can be done on the basis of various criteria, giving rise to different types of VLANs. For example, the simplest type of VLAN is the port-based VLAN. Port-based VLANs divide a network into a number of VLANs by assigning a VLAN to each port of a switching device. Then, any traffic received on a given port of a switch *belongs* to the VLAN associated with that port.



Note This document is intended only to introduce one to the basic concepts of VLANs. For more information about VLANs and their capabilities see the *Riverstone Switch Router User Guide*.

VLANs are primarily used for broadcast containment. A layer-2 (L2) broadcast frame is normally transmitted all over a bridged network. By dividing the network into VLANs, the *range* of a broadcast is limited, i.e., the broadcast frame is transmitted only to the VLAN to which it belongs. This reduces the broadcast traffic on a network by an appreciable factor.

The type of VLAN depends upon one criterion: how a received frame is classified as belonging to a particular VLAN. Riverstone switch routers support the following types of VLANs:

- Port-based VLANs
- Protocol-based VLANs

7.1 Port-based VLANs

Ports of L2 devices (switches, bridges) are assigned to VLANs. Any traffic received by a port is classified as belonging to the VLAN to which the port belongs. For example, if ports 1, 2, and 3 belong to the VLAN named “Marketing”, then a broadcast frame received by port 1 is transmitted on ports 2 and 3. It is not transmitted on any other port (see [Figure 7-2](#)).

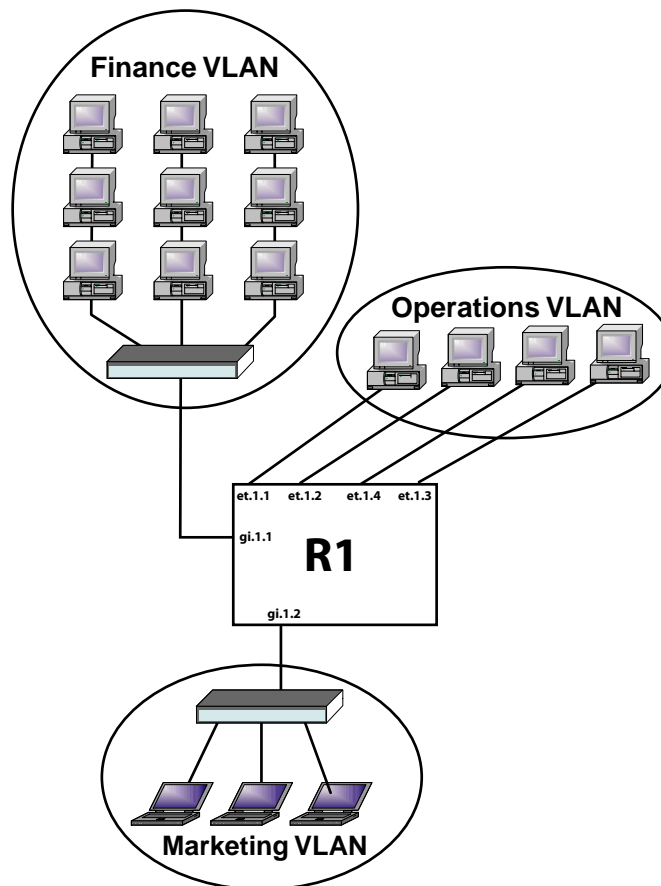


Figure 7-2 Basic port-based VLANs

7.2 Protocol-based VLANs

Protocol-based VLANs divide the physical network into logical VLANs based on protocol. When a frame is received at a port, its VLAN is determined by the protocol of the packet. For example, there could be separate VLANs for IP, IPX and AppleTalk. An IP broadcast frame will only be sent to all ports in the IP VLAN, and so on.

7.3 RS VLAN Support

When using the RS as an L2 bridge/switch, use the port-based and protocol-based VLAN types. When creating protocol-based VLANs, you can create VLANs for AppleTalk, DECnet, SNA, and IPv6 traffic as well as for IP and IPX traffic. You can create a VLAN for handling traffic for a single protocol, such as a DECnet VLAN or you can create a VLAN that supports several specific protocols, such as SNA and IP traffic.

7.4 Ports, VLANs, and L3 Interfaces

The term *port* refers to a physical connector on the RS, such as an Ethernet port. Each port must belong to at least one VLAN. When the RS is not configured, each port belongs to a VLAN called the “default VLAN.” By creating VLANs and adding ports to the created VLANs, the ports are moved from the default VLAN to the newly created VLANs.

Unlike traditional routers, the Riverstone switch router has the concept of logical interfaces rather than physical interfaces. An L3 interface is a logical entity created by the administrator. It can contain more than one physical port. When an L3 interface contains exactly one physical port, it is equivalent to an interface on a traditional router. When an L3 interface contains several ports, it is equivalent to an interface of a traditional router which is connected to a layer-2 device such as a switch or bridge.

7.5 Configuring VLANs

To create a port or protocol based VLAN, perform the following steps in the Configure mode.

1. Create a port or protocol based VLAN.
2. Add physical ports to a VLAN.

VLANs are used to associate physical ports on the RS with connected hosts that may be physically separated but need to participate in the same broadcast domain. To associate ports to a VLAN, you must first create a VLAN and then assign ports to the VLAN. For example the following is the configuration for the VLANs shown in [Figure 7-2](#).

```
rs(config)# vlan create Finance port-based id 100
rs(config)# vlan create Operations port-based id 200
rs(config)# vlan create Marketing port-based id 300
rs(config)# vlan add ports gi.1.1 to Finance
rs(config)# vlan add ports et.1.1-4 to Operations
rs(config)# vlan add ports gi.1.2 to Marketing
```

VLANs and IP Interfaces

VLANs on a Riverstone switch router cannot communicate with each other. In the example in [Figure 7-2](#), the VLANs Finance, Operations, and Marketing are oblivious of each other's existence.

However, when interfaces are applied to VLANs on the same Riverstone switch router, routing occurs automatically between the VLANs. For example, [Figure 7-3](#) shows two Riverstone switch routers (R1 and R2). R2 has three VLANs configured on it, each within its own subnet and each with an IP interface. As [Figure 7-3](#) shows, routing between the three VLANs occurs automatically. In contrast, R1 contains a single VLAN. If R1's VLAN is to communicate with R2's VLANs, a number of additional interfaces and static routes must be configured.

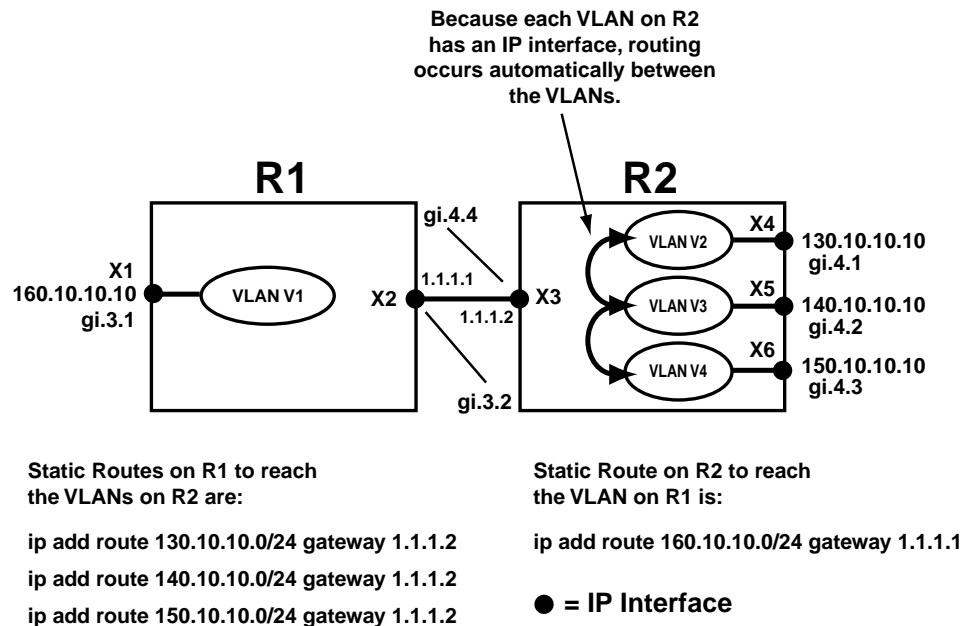


Figure 7-3 VLAN and interface interaction

The following is the configurations for R1 as shown in [Figure 7-3](#).

```
rs(config)# vlan create V1 ip id 100
rs(config)# vlan add ports gi.3.1 to V1
rs(config)# interface create ip X1 address-netmask 160.10.10.10/24 vlan V1
rs(config)# interface create ip X2 address-netmask 1.1.1.1/24 port gi.3.2
rs(config)# ip add route 130.10.10.0/24 gateway 1.1.1.2
rs(config)# ip add route 140.10.10.0/24 gateway 1.1.1.2
rs(config)# ip add route 150.10.10.0/24 gateway 1.1.1.2
```

The following is the configurations for R2 as shown in [Figure 7-3](#).

```
rs(config)# vlan create V2 ip id 200
rs(config)# vlan create V3 ip id 300
rs(config)# vlan create V4 ip id 400
rs(config)# vlan add ports gi.4.1 to V2
rs(config)# vlan add ports gi.4.2 to V3
rs(config)# vlan add ports gi.4.3 to V4
rs(config)# interface create ip X3 address-netmask 1.1.1.2/24 port gi.4.4
rs(config)# interface create ip X4 address-netmask 130.10.10.10/24 vlan V2
rs(config)# interface create ip X5 address-netmask 140.10.10.10/24 vlan V3
rs(config)# interface create ip X6 address-netmask 150.10.10.10/24 vlan V4
rs(config)# ip add route 160.10.10.0/24 gateway 1.1.1.1
```

Notice that in order for VLAN V1 to communicate with VLANs V2, V3, and V4, a number of additional interfaces and static routes had to be added to R1's configuration. In contrast, traffic is routed automatically between VLANs V2, V3, and V4 on R2 and an additional interface and static route had to be added to R2's configuration so that it can communicate with V1 on R1.

Trunk Ports and 802.1Q VLAN Tagging

Riverstone switch routers support trunk ports, which allow multiple VLANs to carry traffic across the same physical port – while each VLAN is unaware of the other VLAN's existence. This ability for a port to carry more than one VLAN is accomplished through the use of 802.1Q VLAN tagging. Essentially, the ingress trunk port is feed by several ports, each with its own VLAN. As the VLANs enter the trunk port, their frames are kept separate using the VLAN's id number. At the other end of the trunk (the egress port), VLAN traffic is separated and sent to the appropriate ports.

Trunk ports and 802.1Q tagging provides a way to support VPN-like topologies, where two or more parts of the same VLAN do not need to be geographically located in the same place.

Figure 7-4 Shows a topology where parts of each VLAN exists in two different geographical locations through the use of trunk ports. This distance, however, and the existence of the other VLANs is transparent to any of the VLAN users.

Notice that three ports on R1 are fed into the trunk port and are distributed back out to three separate ports on R3. Notice also that while R2 acts primarily as a transport, the VLANs must also exist on R2, as well.

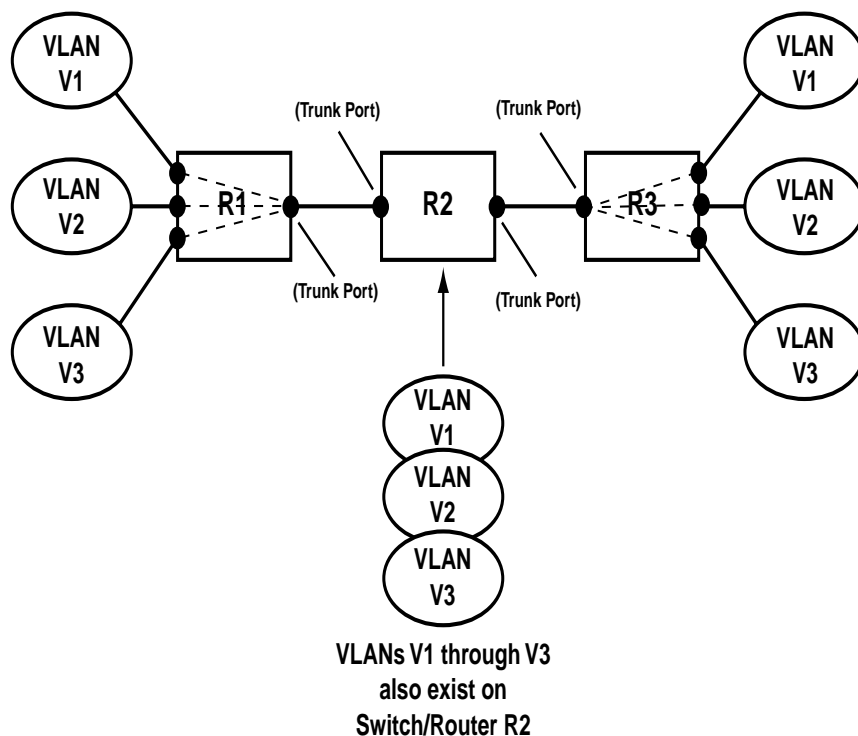


Figure 7-4 Trunk ports and VPN-like behavior

The following is the VLAN and trunk port configurations for Riverstone switch routers R1, R2, and R3.

Configuration for R1:

```
1 : vlan make trunk-port et.1.1
2 : vlan create v1 port-based id 100
3 : vlan create v2 port-based id 120
4 : vlan create v3 port-based id 130
5 : vlan add ports et.2.1 to v1
6 : vlan add ports et.2.2 to v2
7 : vlan add ports et.2.3 to v3
8 : vlan add ports et.1.1 to v1
9 : vlan add ports et.1.1 to v2
10 : vlan add ports et.1.1 to v3
```

Notice in the configuration for R1 that all three VLANs are added to the same port (et.1.1), which is the trunk port.

Configuration for R2:

```
1 : vlan make trunk-port et.1.1
2 : vlan make trunk-port et.1.2
3 : vlan create v1 port-based id 100
4 : vlan create v2 port-based id 120
5 : vlan create v3 port-based id 130
6 : vlan add ports et.1.1 to v1
7 : vlan add ports et.1.1 to v2
8 : vlan add ports et.1.1 to v3
9 : vlan add ports et.1.2 to v1
10 : vlan add ports et.1.2 to v2
11 : vlan add ports et.1.2 to v3
```

Notice in R2's configuration that two trunk ports are created and that the three VLANs must be added to both trunk ports.

Configuration for R3:

```
1 : vlan make trunk-port et.9.1
2 : vlan create v1 port-based id 100
3 : vlan create v2 port-based id 120
4 : vlan create v3 port-based id 130
5 : vlan add ports et.9.1 to v1
6 : vlan add ports et.9.1 to v2
7 : vlan add ports et.9.1 to v3
8 : vlan add ports et.9.2 to v1
9 : vlan add ports et.9.3 to v2
10 : vlan add ports et.9.4 to v3
```

Notice in R3's configuration that the entry port (et.9.1) must be a trunk port, and that the VLANs are ultimately distributed out to individual ports (et.9.2 through et.9.4).

8 WORKING WITH T1 WAN CONNECTIONS

T1 is a channelized WAN protocol that uses Time Division Multiplexing (TDM) to divide the T1 line into 24 equal time slots. Each time slot has a bandwidth of 64 Kilobits, for a total bandwidth of approximately 1.544 Megabits. When creating T1 connection, a port and channel must be specified; where the channel contains the specified time slots. This allows for a flexible use of a T1 line's bandwidth. For example, the same T1 port can use several channels, where, for instance, channel 1 contains time slots 1-10, while channel 2 on the same port contains time slots 11-20, and so on.

Figure 8-5 shows the components of the command line for creating a T1 connection using the `port set` command.

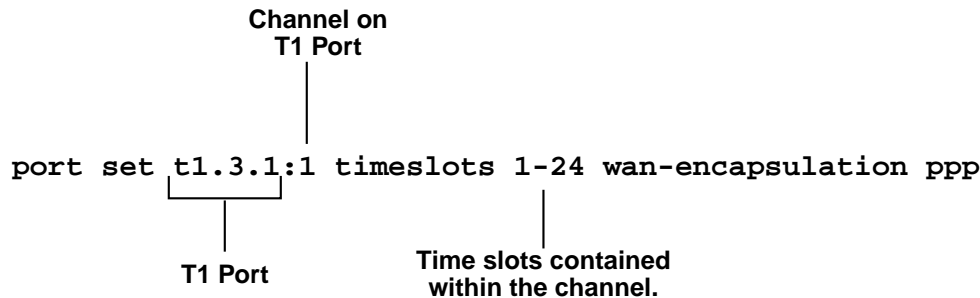


Figure 8-5 Port set command on a T1 port

Notice in Figure 8-5 that the port `t1.3.1` has a single channel (1) and that all 24 time slots are assigned to channel 1.



Note In Figure 8-5 wan-encapsulation is set to the PPP protocol. There are other encapsulation types that can be used such as Frame Relay, however, for this document discussion is confined to PPP.

8.1 Multiple Channels on a T1 Connection

In most cases, the RS treats each port and channel combination as a separate entity. For instance, `t1.7.1:1` and `t1.7.1:2` are considered as two separate connections. Because of this, most CLI commands that affect T1 ports require that both the port number and the channel number are specified.

The following example creates two different port/channel pairs on the same T1 port, then assigns each port/channel pair to its own IP interface:

```
rs(config)# port set t1.7.1:1 timeslots 1-10 wan-encapsulation ppp
rs(config)# port set t1.7.1:2 timeslots 11-23 wan-encapsulation ppp

rs(config)# interface create ip Wan1 address-netmask 100.10.10.10/24 port t1.7.1:1
rs(config)# interface create ip Wan2 address-netmask 200.10.10.10/24 port t1.7.1:2
```

Notice in the example above that when creating each interface that the port and channel needed to be specified for the `port` parameter. Notice, also, that while interfaces `Wan1` and `Wan2` are in separate IP subnets, both interfaces exist on the same physical port (`t1.7.1`).

Use the **port show port-status** command to see how the bandwidth of T1 line is divided among the two connections.

```
rs156# port show port-status t1.7.1:1-2
```

Flags: M - Mirroring enabled B - MLP Bundle S - SmartTRUNK port P - Configured as 802.1p

Port	Port Type	Duplex	Speed	Negotiation	IFG Value	Link State	Admin State	Flags
t1.7.1:1	T1	Full	640000	n/a		Up	Up	
t1.7.1:2	T1	Full	896000	n/a		Up	Up	



Note For more information about WAN port and channel configurations, see the “*WAN Configuration*” chapter in the “*RiverStone Switch Router User Guide.*”

8.2 Injecting a Default Route into OSPF

This section explains how to inject a default route into an OSPF Autonomous System (AS) by way of the AS' Border Router. Consider [Figure 8-6](#). Router R1 is the border router for the backbone area that contains routers R3, R4, and R5. R1 is connected by a T1 connection to router R2.

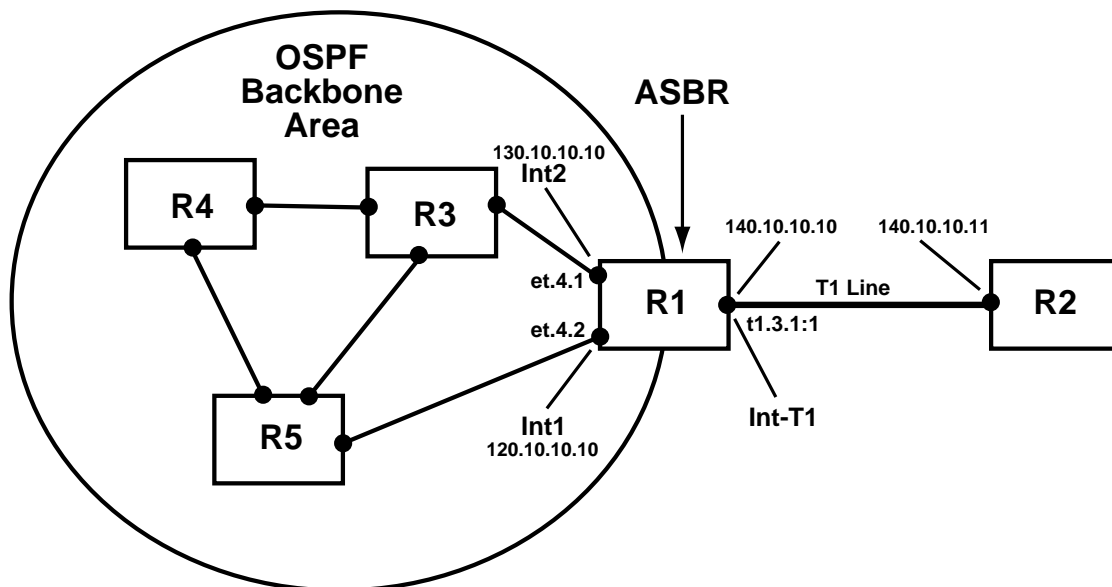


Figure 8-6 T1 connected to OSPF AS

The object of this example is to inject a route from the T1 connection into the backbone AS through the ASBR so that all of the routers in the OSPF backbone area can reach other parts of the network through R2.

Router Configurations

The following example configuration shows the necessary CLI commands that configure router T1 so that it can inject the route on R2 into the OSPF backbone, where it becomes available to R3, R4, and R5.

```
1 : port set t1.7.1:1 timeslots 1-24 wan-encapsulation ppp
!
2 : interface create ip Int-T1 address-netmask 140.10.10.10/24 port t1.7.1:1
3 : interface create ip Int1 address-netmask 120.10.10.10/24 port et.4.1
4 : interface create ip Int2 address-netmask 130.10.10.10/24 port et.4.2
!
5 : ip add route default gateway 140.10.10.11
!
6 : ip-router policy redistribute from-proto direct to-proto ospf network
140.10.10.0/24
7 : ip-router policy redistribute from-proto static to-proto ospf network default
!
8 : ospf create area backbone
9 : ospf add interface Int1 to-area backbone
10 : ospf add interface Int2 to-area backbone
11 : ospf start
```

The following is an explanation of each line shown in the example configuration above:

1. Create the WAN connection on ASBR R1 and assign to it all 24 time slots.
2. Create an interface on the T1 port/channel pair that connects to R2 (140.10.10.10).
3. Create an interface for R1's Ethernet port et.4.1
4. Create an interface for R1's Ethernet port et.4.2
5. Create a static route that identifies R2's IP address (140.10.10.11) as the default gateway. This is the T1 interface on R2 that connects to R1's T1 interface.
6. Create a route policy that injects the 140.10.10.0 subnet into OSPF.
7. Create a route policy that specifies the default gateway (line 5) as part of OSPF
8. Create the OSPF backbone area.
9. Add R1's Int-1 Ethernet interface to the OSPF backbone area.
10. Add R1's Int-2 Ethernet interface to the OSPF backbone area.
11. Start OSPF

Routers R3, R4, and R5 (which also belong to the OSPF backbone) now use R2's T1 interface (140.10.10.11) as their default gateway out of the OSPF backbone area and onto the 140.10.10.0 subnet.



Note OSPF must be running on routers R3, R4, and R5; and their interfaces must be within the OSPF backbone area for the above example to work.



Note For more information about WAN connections, router policies, and OSPF, see the *"Riverstone Networks Switch Router User Guide."*