Avaya Secure Router 1000 Series Configuration Guide

9.4
NN47262-501, 02.01
December 2010

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya generally makes available to users of its products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked Web sites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on its Hardware and Software (“Product(s)”). Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this Product while under warranty is available to Avaya customers and other parties through the Avaya Support Web site: http://support.avaya.com. Please note that if you acquired the Product(s) from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by said Avaya reseller and not by Avaya.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO/ ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AUTHORIZED AVAYA RESELLER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AUTHORIZED AVAYA RESELLER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA AUTHORIZED RESELLER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, or Hardware provided by Avaya. All content on this site, the documentation and the Product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Third-party components

Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements (“Third Party Components”), which may contain terms that expand or limit rights to use certain portions of the Product (“Third Party Terms”). Information regarding distributed Linux OS source code (for those Products that have distributed the Linux OS source code), and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply to them is available on the Avaya Support Web site: http://support.avaya.com/Copyright.

Trademarks

The trademarks, logos and service marks (“Marks”) displayed in this site, the Documentation and Product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation and Product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners, and “Linux” is a registered trademark of Linus Torvalds.

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support Web site: http://support.avaya.com.

Contact Avaya Support

Avaya provides a telephone number for you to use to report problems or to ask questions about your Product. The support telephone number is 1-800-242-2121 in the United States. For additional support telephone numbers, see the Avaya Web site: http://support.avaya.com.

Contents

Chapter 1: New in this release

Feature content from existing Release Notes and Readmes from release 9.2 to 9.4 is now incorporated into this document. For more information, see:

Chapter 2: Preface

This guide describes Avaya Secure Router 1000 Series Secure Router's implementation and command usage of BGP4, OSPF, RIP, and other routing protocols by providing typical configurations for key protocols, as well as Security, VLANs, VPN, WAN, and other key topics relevant to the configuration and operation of the Secure Router 1000 Series products.

The Avaya Secure Router 1000 series includes the Secure Router 1004, Secure Router 1002, Secure Router 1001, and Secure Router 1001s models. In certain areas of this Configuration Guide when discussing features, the term SR1000 is utilized to refer to any of these models. Please refer to the *SR1000 Series Installation Guide* for complete details on each model and interface support.

Organization

Each chapter describes how to configure a specific feature of the Secure Router. There is no inherent order in the chapter arrangement although related topics are grouped together to make it easier to use.

Documentation

Avaya user guides, which are provided in portable document format (PDF), are included on the Avaya Secure Router Documentation CD-ROM that ships with the Secure Router 1000 Series. The PDF files are also available on the Avaya website: http://www.avaya.com

To view PDF files, Adobe Acrobat® Reader® 4.0, or newer, must be installed on your workstation. If you do not have the Adobe Acrobat Reader installed on your system, you can obtain it free from the Adobe website: http://www.adobe.com

About the Avaya Secure Router Documentation CD

This product ships with a CD that includes the following documentation:

- *Avaya Secure Router 1000 Series Quick Start Guide*
- *Avaya Secure Router 1000 Series Installation Guide*

- *Avaya Secure Router 1000 Series Command Reference Guide*
- *Avaya Secure Router 1000 Series Routing Guide*
- *Avaya Secure Router 1000 Series Configuration Guide*
- *Avaya Secure Router 1000 Series Web UI User Guide*
- Supported standard and enterprise MIBs
- Feature summaries
- SNMP trap descriptions with default configurations

Navigation

Upon inserting the *Avaya Secure Router Documentation* CD into your CD-ROM drive. Click a link to open a PDF version of the target document. If you do not have Adobe Acrobat (version 4.0, or later) or Acrobat Reader installed on your PC, click the Adobe button on the navigation screen to go to the Adobe website, where you can download a free copy of the Acrobat Reader application.

If a browser session is not opened, click "Start\Run," enter the drive letter of your CD-ROM drive in the "Open" entry box, and click "OK."

Printing Documents

To print any PDF document on the CD, follow this procedure.

1. Open the desired document by clicking the document link in the CD navigation window.
2. Click the "Printer" icon on the Adobe Acrobat tool bar.
3. In the "Windows Print" dialog box, select a local default printer in the "Printers" drop down selection box.
4. Click "OK."

The following list includes other available and related documentation.

- *Release Notes*

  Printed release notes provide the latest information. Follow the instructions contained within the release notes provided with your product instead of those provided in other documentation.

- *Secure Router 1000 Series Quick Start Guide*

This guide is designed for advanced users who need minimal installation, configuration, and operation information.

- *Secure Router 1000 Series Installation Guide*

  This detailed guide is designed for network managers and technicians who are responsible for the installation of networking equipment in Telco and service provider network environments.

- *Secure Router 1000 Series Command Line Reference*

  This detailed guide provides a complete listing of all commands including descriptions, syntax, examples, and applicable systems.

- *Secure Router 1000 Series Routing User Guide*

  This guide explains how each feature is used.

- *Secure Router 1000 Series WebUI User Guide*

  This guide explains how to configure the Secure Router 1000 Series using the WebUI.

To view PDF files, Adobe Acrobat® Reader® 4.0 (or later) must be installed on your PC. If you do not have the Adobe Acrobat Reader installed on your system, you can obtain it free from the Adobe website: http://www.adobe.com.

Customer service

Visit the Avaya Web site to access the complete range of services and support that Avaya provides. Go to www.avaya.com or go to one of the pages listed in the following sections.

Navigation

- Getting technical documentation on page 19
- Getting product training on page 19
- Getting help from a distributor or reseller on page 20
- Getting technical support from the Avaya Web site on page 20

Getting technical documentation

To download and print selected technical publications and release notes directly from the Internet, go to www.avaya.com/support.

Getting product training

Ongoing product training is available. For more information or to register, you can access the Web site at www.avaya.com/support. From this Web site, you can locate the Training contacts link on the left-hand navigation pane.

Getting help from a distributor or reseller

If you purchased a service contract for your Avaya product from a distributor or authorized reseller, contact the technical support staff for that distributor or reseller for assistance.

Getting technical support from the Avaya Web site

The easiest and most effective way to get technical support for Avaya products is from the Avaya Technical Support Web site at www.avaya.com/support.

Chapter 3: Secure Router Basics

Default Login Parameters

By default, the Secure Router ships with the following login parameters for all management methods:

```
Username: admin

Password: setup
```

Note:
Login information is case sensitive.

Default settings

The default settings are as follows:

- WebUI is disabled
- SNMP is disabled
- Telnet server is disabled
- Telnet client is enabled
- TFTP server is disabled
- FTP server is disabled

Use the CLI to change the default settings.

Enable Telnet Server

After upgrading the Secure Router, telnet server is disabled by default. To enable the telnet server, use the following command:

```
SR/config> telnet_server
```

Enable Web User Interface

To enable the Web User Interface, use the following command:

```
SR/config> gui enable
```

Applying licenses

While the Secure Router can be purchased with up to 4 ports activated, the Secure Router 1002 and 1004 base models each ship with one active WAN port. Activating additional ports requires only a simple change to the router configuration with a software key that may be purchased to activate up to three additional WAN ports. This key is different than the software upgrade key.

To obtain a port upgrade key, contact your reseller or Avaya. You will be asked to provide the serial number, model number, and the number of ports that are currently active on your router.

The following procedure describes how to activate additional WAN ports in the SR1002 and SR1004.

1. From the command line interface, issue the following command:

   ```
   Avaya>configure terminal
   ```

2. Issue the following command:

   ```
   Avaya>system licenses < option >
   ```

   The option parameters are:

   - enable_1_port
   - enable_2_ports
   - enable_3_ports
   - enable_4_ports

   Note:
   The total number of active ports is equal to the sum of existing active ports and the type of license purchased (1, 2, or 3 port) to a maximum of 2 ports on the Secure Router 1002 and 4 ports on the Secure Router 1004

   You will be prompted for the port upgrade license key.

3. Enter the license key provided.

The license key is case sensitive.

Note:
It is important that you do not enter any extra spaces at the end of the license key, as this may produce an error.

4. Reboot the router.

Daylight Saving Time support

Daylight Saving Time is now supported on the Secure Router for time zones for in US, Canada, and Australia.

To enable Daylight Saving Time, use the following procedure.

1. To enter the configuration mode, enter:

   ```
   configure terminal
   ```

2. To enable daylight saving time, enter:

   ```
   dst enable
   ```

3. To display the daylight savings time configuration, enter:

   ```
   show dst
   ```

Multiple SNTP Server support

The Secure Router 1000 Series and 3120 provide support for the Multiple Simple Network Time Protocol (SNTP) Server feature. SNTP is a simple form of the Network Time Protocol (NTP), which is an internet protocol used for synchronization of computer clocks.

The Multiple SNTP Server feature provides support for up to 10 SNTP servers. Multiple servers provide redundant backup for synchronizing time on the Secure Router. During configuration, servers can be specified by hostname or IP address, and a timeout value must be set for the query. The Multiple SNTP Server features operates by having the SNTP service query configured SNTP servers on a round robin basis. If any SNTP server is queried and fails to respond, the router will send a request to the next configured SNTP server. The sntp server support is not active until the service is enabled. While the service is enabled the configuration can not be changed.

The **`show sntp`** command has been modified to display the current state of SNTP, the server it is contacting to receive the current time, as well as all configured servers. When specifying a

server by domain name, note that DNS entries need to be configured before SNTP will function properly.

Configuring multiple SNTP servers

Use the following procedure to configure multiple SNTP servers.

Procedure steps

1. To configure multiple SNTP servers, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Since DNS entries must be configured for SNTP to function properly, configure primary and secondary DNS servers.

   ```
   ip pname_server <A.B.C.D>
   ```

   ```
   ip name_server <A.B.C.D>
   ```

3. To configure an SNTP server, enter the sntp sub-tree.

   ```
   sntp
   ```

4. Configure the source address of the SNTP client.

   ```
   source-address <A.B.C.D>
   ```

5. Configure the number of retries per SNTP server.

   ```
   retries <count>
   ```

6. Configure an NTP server.

   ```
   server <server> [timeout]
   ```

7. To add up to 10 SNTP servers, repeat step 6.
8. Enable the SNTP client.

   ```
   enable
   ```

Table 1: Variable definitions

Variable	Value
<A.B.C.D>	An IP address.
<count>	The number of retries the NTP server performs, in the range 1 to 5. Default is 3.
<server>	The NTP server to use for updates.
<timeout>	The maximum response time, in the range 10 to 7200. Default is 1024.

Multiple Syslog Server support

The Secure Router 1000 Series and 3120 provide support for multiple Syslog servers. A Syslog Server monitors incoming Syslog messages on UDP ports and decodes them for logging purposes. In addition, several network devices are now able to be configured to generate Syslog messages. In the past, the Secure Router 1000 Series and 3120 only provided support for logging on a single Syslog Server, but this enhancement allows for the configuration of up to 5 Syslog Servers. Since they are logged simultaneously, all Syslog servers will contain the same Syslog records.

To achieve backward compatibility with previous Syslog implementation, the provision of a port number during configuration of the host IP address remains optional. If a user does not specify a port during CLI configuration, UDP port 514 is used by default. In addition, the enabling of message logging remains unchanged.

As a limitation, all enable or disable functions will apply to all configured servers. Configuration of Syslog message logging on selected servers is not supported.

Note that when viewing Syslog Server information, the SNMP interface can only display information for one server at a time.

Configuring multiple Syslog servers

Use the following procedure to configure multiple Syslog servers.

Procedure steps

1. To configure multiple Syslog servers, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Enter the system logging sub-tree.

   ```
   system logging
   ```

3. Access the Syslog command tree.

   ```
   syslog
   ```

4. Specify a host IP address and UDP port. If a port number is not specified, port 514 will be used by default.

   ```
   host_ipaddr <A.B.C.D> [port]
   ```

5. To add another Syslog server address, repeat step 4 until up to 5 Syslog servers are added.
6. Enable Syslog.

   ```
   enable
   ```

Table 2: Variable definitions

Variable	Value
<A.B.C.D>	The host IP address.
[port]	Optionally, the UDP port. If not specified, port 514 is used by default.

Top command

The **top** command replaces the **pop** command to exit to the top of the configuration tree. It now can be executed either interactively or through a configuration file read locally or over the network.

Reading system.cfg from an alternate drive at startup

When rebooting the router, if you boot the router from an alternate drive (/cf0 or /usb0) and a system.cfg resides on the same drive, the router executes the system.cfg file.

banner.txt file

Banner.txt file is now supported on all platforms. The banner.txt file is displayed logging into the router through telnet or SSH.

Chapter 4: Source IP Enhancements

The Secure Router 1000 Series and 3120 provide support for adding source address information to existing services. The services modified to accept a source address are:

- File Transfer
- QoS Historical Statistics
- RADIUS
- SNMP
- SNTP
- Syslog
- TACACS

The source address parameter is configurable on a global basis, where all the above services are configured with the same source address. The exception to this is when the source address is configured separately for the service, in which case the service configuration takes precedence. The source address can be configured using the IP address or the interface name.

To accommodate this feature, all router output displays that contain a `source address` field will display the source IP address and the interface name associated with it. If the feature is configured by IP address, but has no associated interface specified, the interface will show as `not configured`. Likewise, if the feature is configured by interface name, with no IP address specified, the IP address will show as `not configured`. Global source address information can be found using the **`show system configuration`** command.

The command **`source-address`** is available to enable this feature. In the case of Radius and SNMP, the previous commands (**`src_address`** and **`snmp-source`** respectively) have been deprecated in lieu of this command.

Since file transfer commands are not stored in a configuration it will use the global source address if configured. Each of the file transfer commands accepts a source-address parameter to override the global source address.

Warning:

When a source address is configured for a service which is valid (IP address and interface associated with it) and the source-address interface is down the service may fail to work if it is bi-directional. By using a loopback interface for the source address which is always up it will insure that the above problem does not occur.

Configuring global source address

Use the following procedure to configure source addresses on services.

Procedure steps

1. To configure source addresses for a service, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Configure the global source address.

   ```
   system source-address {<A.B.C.D> | <interface-name>}
   ```

Table 3: Variable definitions

Variable	Value
<A.B.C.D>	Specify source address by IP address.
<interface-name>	Specify source address by interface name.

Configuring Radius or TACACS source address

Use the following procedure to configure Radius or TACACS server source address for all services.

Procedure steps

1. To configure source addresses for a service, enter Configuration Mode.

   ```
   configure terminal
   ```

2. To configure Radius or TACACS source addresses, enter the aaa command sub-tree.

   ```
   aaa
   ```

3. Configure the source address.

   ```
   source-address {<A.B.C.D> | <interface-name>}
   ```

Table 4: Variable definitions

Variable	Value
<A.B.C.D>	Specify source address by IP address.
<interface-name>	Specify source address by interface name.

Configuring SNMP source address

Use the following procedure to configure SNMP server source address for all services. Note that the SNMP server must be disabled prior to setting the source address.

Procedure steps

1. To configure source addresses for a service, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Enter the snmp-server subtree.

   ```
   snmp-server
   ```

3. Disable snmp server.

```
no snmp-enable
```

4. Configure the source address.

```
source-address {<A.B.C.D> | <interface-name>}
```

5. Enable snmp server.

```
snmp-enable
```

Table 5: Variable definitions

Variable	Value
<A.B.C.D>	Specify source address by IP address.
<interface-name>	Specify source address by interface name.

Configuring SNTP source address

Use the following procedure to configure SNTP server source address for all services.

Procedure steps

1. To configure source addresses for a service, enter Configuration Mode.

```
configure terminal
```

2. Enter the sntp subtree

```
sntp
```

3. Configure the source address.

```
source-address {<A.B.C.D> | <interface-name>}
```

Table 6: Variable definitions

Variable	Value
<A.B.C.D>	Specify source address by IP address.
<interface-name>	Specify source address by interface name.

Configuring Syslog source address

Use the following procedure to configure Syslog server source address for all services.

Procedure steps

1. To configure source addresses for a service, enter Configuration Mode.

```
configure terminal
```

2. Enter the system logging subtree.

```
system logging
```

3. Enter the syslog subtree.

```
syslog
```

4. Configure the source address.

```
source-address {<A.B.C.D> | <interface-name>}
```

Table 7: Variable definitions

Variable	Value
<A.B.C.D>	Specify source address by IP address.
<interface-name>	Specify source address by interface name.

Configuring QoS Historical Statistics source address

Use the following procedure to configure QoS Historical Stats server source address for all services.

Procedure steps

1. To configure source addresses for a service, enter Configuration Mode.

```
configure terminal
```

2. Enter the qos subtree.

```
qos
```

3. Enter the historical-stats subtree.

```
historical-stats
```

4. Configure the source address.

```
source-address {<A.B.C.D> | <interface-name>}
```

Table 8: Variable definitions

Variable	Value
<A.B.C.D>	Specify source address by IP address.
<interface-name>	Specify source address by interface name.

Chapter 5: Multiple IP Helper Addresses on VLAN

The Secure Router 1000 Series and 3120 provide support for Multiple IP Helper. The Multiple IP Helper feature assists in broadcasting network traffic between client machines and servers residing on different subnets. There are situations in which a user may want to control which broadcast packets and protocols should be forwarded by the router. The Multiple IP Helper feature provides this functionality.

Multiple IP Helper is useful when UDP broadcasts are sent to a DNS server by a network host. If the network host happens to reside on a segment without a DNS server, the UDP broadcast will fail. When this occurs, a helper address is configured and a protocol assigned to an interface. The exceptions to this are DHCP and BOOTP broadcasts, which are handled by DHCP Relay.

The Multiple IP Helper feature has been implemented on primary ethernet interfaces and VLAN-enabled ethernet subinterfaces, with a maximum of 6 helper addresses able to be configured per interface.

Configuring multiple IP Helper addresses

Use the following procedure to configure IP Helper addresses.

Procedure steps

1. To configure IP Helper addresses, enter Configuration Mode.

   ```
   configure terminal
   ```

2. To configure an interface, enter Interface Mode.

   ```
   interface <interface>
   ```

3. Specify an IP address.

   ```
   ip address <A.B.C.D> <subnet mask>
   ```

4. Specify a Helper address.

   ```
   ip helper-address <A.B.C.D>
   ```

5. To configure a subinterface, exit back a level.

   ```
   exit
   ```

6. Specify the subinterface.

   ```
   interface <subinterface>
   ```

7. Specify vlan ID.

   ```
   encapsulation dot1q <vlan-id>
   ```

8. Specify an IP address.

```
ip address <A.B.C.D> <subnet mask>
```

9. Specify a Helper address (by default all ip helper services are enabled).

   ```
   ip helper-address <A.B.C.D>
   ```

10. If desired, specify a Helper address for a service.

    ```
    ip helper-address <A.B.C.D> service <service>
    ```

11. If desired, specify a Helper address for a protocol or port.

    ```
    ip helper-address <A.B.C.D> protocol <protocol> port <port>
    ```

Table 9: Variable definitions

Variable	Value
<A.B.C.D>	The IP address.
<interface>	The interface to work with.
<port>	The port number in the range 1 to 65535..
<protocol>	The protocol to be used. Options available are: UDP - to a specific UDP port.
<service>	Service name to specify IP helper for a service. Options available are: • dns -- Domain Name Service • netbios-dgm -- NetBIOS datagram service • netbios-ns -- NetBIOS name service • netbios-ss -- NetBIOS session service • tftp -- Trivial File Transfer Protocol • time -- Time
<subinterface>	The subinterface IP address.
<type>	The type of encapsulation to apply.

Chapter 6: TCP MSS Clamping

The TCP MSS feature enables the configuration of the maximum segment size (MSS) for transient packets that traverse the router. The **ip tcp-mss** command under the interface tree specifies the MSS value on the intermediate router of the TCP SYN packets to avoid truncation. When a TCP SYN packet traverses the router the MSS option is lowered to the specified value in the TCP packet.

The ability to set the TCP MSS value is supported on Ethernet interfaces (including subinterfaces), bundles, GRE/IPIP Tunnels, and firewall policies. MSS clamping at the firewall policy level provides greater granularity by allowing the clamping to be performed only for certain hosts.

When setting the TCP MSS value it is recommended that the MSS value be at least 40 bytes less than the MTU of the interface. The TCP header takes up 20 bytes of data (or more if options are used); the IP header also uses 20 or more bytes. This means that between them a minimum of 40 bytes are needed for headers, all of which is non-data overhead.

Configuring TCP MSS on a GRE/IPIP Tunnel Interface

1. To enter the configuration mode, enter:

   ```
   configure terminal
   ```

2. To select the tunnel, enter:

   ```
   interface tunnel <tunnel-name>
   ```

3. To specify the IP address for the tunnel, enter:

   ```
   ip address <A.B.C.D> <subnet-mask>
   ```

4. To specify the source address of the tunnel, enter:

   ```
   tunnel source <A.B.C.D>
   ```

5. To specify the destination address of the tunnel, enter:

   ```
   tunnel destination <A.B.C.D>
   ```

6. To specify the tcp-mtu of the tunnel, enter:

   ```
   ip tcp-mss <value>
   ```

7. To exit the tunnel configuration mode, enter:

   ```
   exit
   ```

Configuring Ethernet Interface

1. To enter the configuration mode, enter:

   ```
   configure terminal
   ```

2. To select the Ethernet interface, enter:

```
interface ethernet <port number>
```

3. To specify the tcp-mtu of the Ethernet interface, enter:

```
ip tcp-mss <value>
```

4. To exit the Ethernet configuration mode, enter:

```
exit
```

Configuring TCP MSS on Existing PPP Bundle Interface

1. To enter the configuration mode, enter:

```
configure terminal
```

2. To select the bundle interface, enter:

```
interface bundle <name>
```

3. To specify the tcp-mtu of the interface, enter:

```
ip tcp-mss <value>
```

4. To exit the bundle configuration mode, enter:

```
exit
```

Configuring TCP MSS on an Existing Frame Relay Bundle PVC Interface

1. To enter the configuration mode, enter:

```
configure terminal
```

2. To select the bundle interface, enter:

```
interface bundle <name>
```

3. To specify Frame Relay configuration, enter:

```
fr
```

4. To select the PVC interface, enter:

```
pvc <pvc-number>
```

5. To specify the tcp-mtu of the interface, enter:

```
ip tcp-mss <value>
```

6. To exit the Frame Relay PVC configuration mode, enter:

```
exit
```

7. To exit the Frame Relay configuration mode, enter:

```
exit
```

8. To exit the bundle configuration mode, enter:

```
exit
```

Configuring TCP MSS on Firewall Policy

1. To enter the configuration mode, enter:

   ```
   configure terminal
   ```

2. To select the firewall map, enter:

   ```
   firewall <zone>
   ```

3. To specify the tcp-mtu of the map, enter:

   ```
   policy <priority> <direction values {in|out}>
   ```

4. To specify the tcp-mtu for the policy, enter:

   ```
   ip tcp-mss <value>
   ```

5. To exit the firewall policy, enter:

   ```
   exit
   ```

6. To exit the firewall configuration mode, enter:

   ```
   exit
   ```

Chapter 7: IP MULTIPLEXING

IP Unnumbered Auto-Configuration

Auto-configuration simplifies the deployment of Secure Routers in IP multiplexing applications by reducing the routing information that must be manually configured. Auto-configuration can be utilized in applications with Secure Routers on each side of the WAN connection.

An IP multiplexing example is shown below; split subnet IP addressing is used in the example, with the WAN bundles running IP unnumbered.

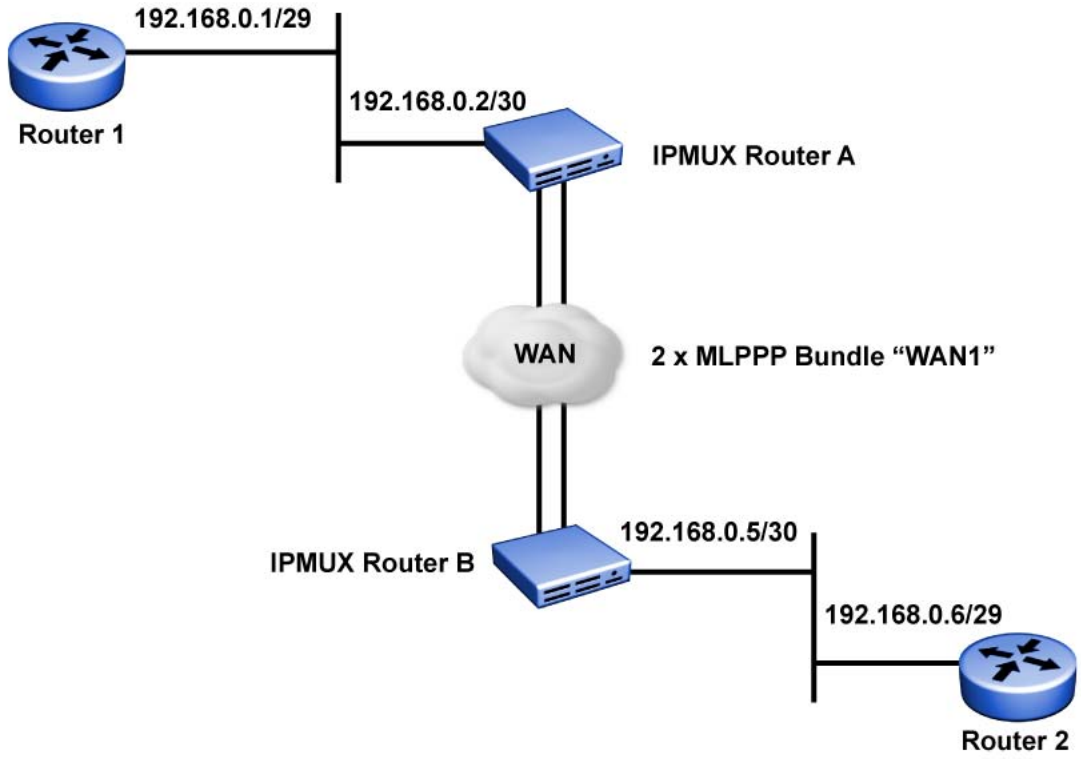


Figure 1: IP Multiplexing Auto-Configuration

The effect of auto-configuration is that the two Secure Routers exchange Ethernet IP addressing and subnet information as well as configured source forwarding destinations. Based on this information, Secure Router A would install the following routes:

- A 32-bit host route to the Secure Router B Ethernet address: 192.168.0.5 255.255.255.255 wan 1
- A 30-bit network route to the Secure Router B Ethernet subnet: 192.168.0.4 255.255.255.252 wan 1
- A 32-bit host route to router 2: 192.168.0.6 255.255.255.255 wan 1

The two Secure Routers do not exchange information on manually configured IP routes, nor do they exchange information on any routes learned through auto-configuration with other Secure Routers.

The installation of the network route is based on that route being different from the Secure Router A interface route. Consider a single subnet addressing approach with:

- Router 1: 192.168.0.1/29
- Router 2: 192.168.0.4/29
- Secure Router A: 192.168.0.2/29
- Secure Router B: 192.168.0.3/29

Secure Router A would not install a network route to the remote Ethernet subnet since it would duplicate the Secure Router A Ethernet interface route.

Configure the Secure Router 1000 Series at Site A

```
SR> configure term
SR/configure> interface ethernet 0
SR/configure/interface/ethernet> ip addr 192.168.0.2
255.255.255.252
SR/configure/interface/ethernet> exit

SR/configure> interface bundle wan1
SR/configure/interface/bundle> link t1 1-2
SR/configure/interface/bundle> encap ppp
SR/configure/interface/bundle> exit

SR/configure> autoconf
SR/configure> exit
```

Configure the Secure Router 1000 Series at Site B

```
SR> configure term
SR/configure> interface ethernet 0
SR/configure/interface/ethernet> ip addr 192.168.0.5
255.255.255.252
SR/configure/interface/ethernet> exit

SR/configure> interface bundle wan1
SR/configure/interface/bundle> link t1 1-2
SR/configure/interface/bundle> encap ppp
SR/configure/interface/bundle> exit

SR/configure> autoconf
SR/configure> exit
```

Chapter 8: DHCP Relay

This application describes the functionality of the DHCP relay feature and includes CLI command examples.

Feature Overview

The DHCP relay feature eliminates the need for a DHCP server on every LAN, because DHCP requests can be relayed up to 4 DHCP servers on each ethernet interface including subinterfaces. Avaya 's implementation of DHCP relay is based on RFC 1532. BOOTP/ DHCP messages are relayed (vs. forwarded) between the server and client.

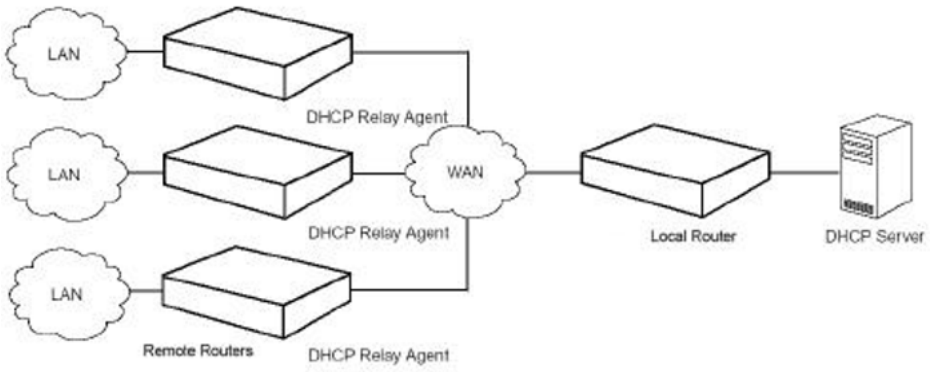


Figure 2: DHCP Relay Overview

Functionality

The DHCP relay feature uses BOOTP requests and replies to negotiate packet delivery between the DHCP client and server.

BOOTP Requests

BOOTP requests are messages from client to server. Request messages include DHCP DISCOVER, DHCP REQUEST, DHCP RELEASE, etc. The relay agent modifies the packet

header by adding relay information to the DHCP gateway address (giaddr) field. The server replies to the gateway address specified in the packet's giaddr field.

Figure 3: BOOTP Requests

BOOTP Replies

BOOTP replies are messages from the server to the client. Reply messages include DHCP OFFER, DHCP ACK, DHCP NAK, etc. The relay agent looks up the MAC address and either sends the packet to the client or broadcasts it on the LAN.

Figure 4: BOOTP Replies

Using DHCP Relay with NAT

When NAT is enabled, the DHCP server may discard packets because the giaddr does not match the source of the packet. Additionally, it may not know how to route the packet back to the client. See Figure 4. The solution is that the gateway address (giaddr) field needs to have IP address 192.168.20.1 (in this example). The DHCP server configuration should be able to give 10.1.1.x addresses for packets from 192.168.20.1. However, there may be a limitation that the DHCP server does not allow configuration using IP addresses from a different subnet, although this is mentioned in the RFC.

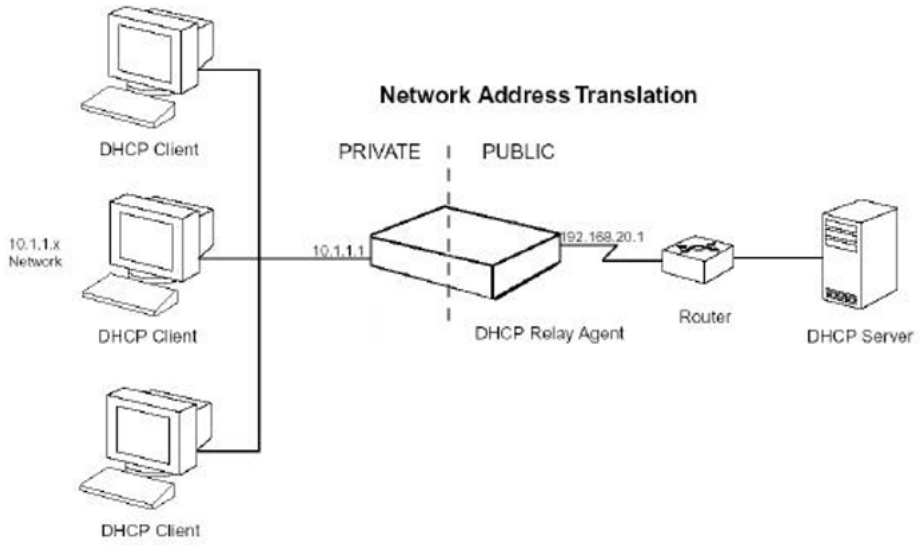


Figure 5: A Typical Scenario

Command Line Interface

The following are examples of command strings relevant to DHCP relay:

`dhcp-relay` is only a command to configure a DHCP relay. For this command, a DHCP server and a gateway address are mandatory in case NAT enabled is optional parameter.

Enabling DHCP Relay

The following example shows configuring 4 DHCP server addresses. Here, 20.1.1.1, 20.1.1.2, 20.1.1.3, and 20.1.1.4 are DHCP server IP addresses. You can configure a maximum of 4 DHCP server addresses for an interface.

```
SR> configure terminal
Router/configure> interface ethernet 0
Router/configure/interface/ethernet 0> dhcp-relay 20.1.1.1
Router/configure/interface/ethernet 0> dhcp-relay 20.1.1.2
Router/configure/interface/ethernet 0> dhcp-relay 20.1.1.3
Router/configure/interface/ethernet 0> dhcp-relay 20.1.1.4
```

Disabling DHCP Relay

```
Router/configure/interface/ethernet 0> no dhcp-relay 20.1.1.1
Router/configure/interface/ethernet0> no dhcp-relay 20.1.1.2
Router/configure/interface/ethernet 0> no dhcp-relay 20.1.1.3
Router/configure/interface/ethernet 0> no dhcp-relay 20.1.1.4
```

Configuring the Gateway Address field when NAT is enabled

The following example shows configuring 4 DHCP server addresses with NAT enabled.

```
Router/configure/interface/ethernet 0> dhcp-relay 20.1.1.1 192.168.20.1
Router/configure/interface/ethernet 0> dhcp-relay 20.1.1.2 192.168.20.1
Router/configure/interface/ethernet 0> dhcp-relay 20.1.1.3 192.168.20.1
Router/configure/interface/ethernet 0> dhcp-relay 20.1.1.4 192.168.20.1
```

Displaying DHCP Configuration

The following screen captures show the displayed results of issuing show commands relevant to DHCP relay, with and without gateway addresses configured.

```
> show dhcp_relay

DHCP RELAY CONFIGURATION
-------------------------
Ethernet 0: Disabled
Ethernet 1: Enabled: DHCP Server 10.1.1.1

SR>
```

Figure 6: show dhcp_relay Command

```
> show dhcp_relay

DHCP RELAY CONFIGURATION
-------------------------
Ethernet 0/2: Disabled
Ethernet 0/2: Enabled: DHCP Server 10.1.1.1 (Gateway Address:
192.168.20.1)

SR>
```

Figure 7: show dhcp_relay Command

Displaying Statistics

```
> show interface ethernet 1


ethernet 1
ipaddr       192.168.120.1
netmask      255.255.255.0
description  -
status       down, operationally down
configured   auto
  speed       -
  mode        -
actual
  speed      100
  mode       half_duplex
mtu          1500

ethernet1 (unit number 1)
Type: ETHERNET (802.3)
Flags: (0x807c203)  UP, MULTICAST-ROUTE
Internet Address: 192.168.120.1
Internet Netmask: 255.255.255.0
Internet Broadcast: 192.168.120.255
Maximum Transfer Unit: 1500 bytes
Mac Address:  00:00:23:00:60:01

port counters since last boot/clear
  Bytes Rx                          0  Bytes Tx                          0

  Packets Rx                        0  Packets Tx                        0

  Runts Rx                          0  Collisions                        0

  Babbels Rx                        0  Late Collisions                   0

  Err Packets Rx                    0  Up/Down States (Phys)             0

  Up/Down States (Admin)            2

port counters for the last five minutes
  Bytes Rx                          0  Bytes Tx                          0

  Packets Rx                        0  Packets Tx                        0

  Runts Rx                          0  Collisions                        0

  Babbels Rx                        0  Late Collisions                   0

  Err Packets Rx                    0  Up/Down States (Phys)             0

  Up/Down States (Admin)            0
SR>
```

Figure 8: Displaying Ethernet Interface Statistics

DHCP request display

Counters are available in the Ethernet display to show the number of DHCP requests dropped due to being the VRRP Backup Router or having no route to the DHCP Server.

DHCP Limitations

There are limitations when using DHCP relay on a Secure Router. DHCP can be enabled only on Ethernet interfaces (not on bundles). And last, DHCP can be enabled in IP routing (static and dynamic) mode, but not in IP Mux mode.

Chapter 9: DHCP Client on Ethernet interfaces

The Secure Router 1000 Series and 3120 provide support for Dynamic Host Configuration Protocol (DHCP) for IPv4 clients on Ethernet interfaces. A DHCP client obtains configuration parameters such as an IP address.

Using DHCP, a client can contact a central DHCP server that is responsible for maintaining a list of IP addresses available to be assigned on one or more subnets. The DHCP client requests an address from the pool and uses it temporarily to communicate on a network. In addition to this, the DHCP protocol is capable of supplying a client with important details about the network to which it is attached. This is important since a client may require these parameters during boot or normal run time.

The DHCP protocol client implementation allows the client to obtain an IP address and, if configured, a default gateway from the DHCP server. An interface specified as a DHCP client cannot be specified as a DHCP server. Likewise, an interface specified as a DHCP client cannot be specified as a relay agent.

As limitations, DHCP clients are not supported on subinterfaces, and only work after the system has booted.

Configuring DHCP Client on Ethernet interfaces

Use the following procedure to configure a DHCP client on an Ethernet interface.

Procedure steps

1. To configure a DHCP client, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Enter Interface Mode.

   ```
   interface <interface>
   ```

3. Specify a DHCP client lease.

   ```
   dhcp-client lease <duration>
   ```

4. Specify a DHCP client hostname.

   ```
   dhcp-client hostname <hostname>
   ```

5. Configure the default router IP source to be the server.

   ```
   dhcp-client request-default-router
   ```

6. Specify the retry interval.

   ```
   dhcp-client retry-interval <interval>
   ```

7. Enable the DHCP client on the interface.

```
dhcp-client enable
```

Variable	Value
<duration>	The duration of the lease in the range 30 to 4294967.
<hostname>	The hostname of the DHCP client.
<interface>	The interface to work with.
<interval>	The timeout interval, in seconds, for the DHCPv4 client negotiation process.

Chapter 10: DHCP Server Configuration

For small network environments which require a DHCP Server, the DHCPS feature on the Secure Router 1000 is recommended. The steps below show a typical configuration for DHCP deployment in a network.

To configure the DHCP Server:

1. Associate the physical interface of the SR1000 with the DHCP Server.

   ```
   config/term> ip dhcps
   config/term/ip/dhcps> interface ethernet0
   ```

   Note:
   The interface must be active before it can be associated with the DHCP server. To ensure this you must configure an IP address on Ethernet 0.

2. Enable the DHCP server.

   ```
   config/term/ip/dhcps> enable
   ```

   Note:
   In order to add or remove interfaces from the DHCP server, the server must be disabled

   ```
   config/term/ip/dhcps> no enable
   ```

3. Configure the DHCP pools and assign the desired parameters. In this example we configure a pool for the "floor3” subnet:

   a. Create the pool.

      ```
      config/term> ip dhcps
      config/term> ip dhcps pool floor3
      ```

   b. Configure the network and mask for the subnet.

      ```
      config/term/ip/dhcps> pool floor3 network 20.20.20.0 24
      ```

   c. Configure the default router for the pool.

      ```
      config/term/ip/dhcps> pool floor3 default_router 20.20.20.1
      ```

   d. Configure the DNS server address for the pool.

      ```
      config/term/ip/dhcps> pool floor3 dnsserver 10.19.25.130
      ```

   e. If applicable, exclude an address range or multiple address ranges within the address pool.

```
config/term/ip/dhcps> pool floor3 exclude-range 10.10.10.1
10.10.10.10

config/term/ip/dhcps> pool floor3 exclude-range 10.10.10.200
10.10.10.255
```

f. If applicable, change the lease timer. The default is 3600 seconds; the range is from 0 (doesn't expire) to 4294967 seconds.

```
config/term/ip/dhcps> pool floor3 lease [duration in seconds]
```

g. Once the pool is configured, issue the **commit** command to activate the pool.

```
config/term/ip/dhcps> pool floor3 commit
```

Note:
Once the pool is committed no changes can be made unless it is uncommitted:

```
config/term/ip/dhcps> pool pool3 no commit
```

h. Configure relay agent(s). In this example the relay agent is 20.20.20.1 and the network is 20.20.20.0.

```
config/term/ip/dhcps> relay 20.20.20.1 20.20.20.0
```

Note:
There is no CLI command to link a given relay agent to a given address pool; the linking is done automatically through the IP address/network association.

i. To display DHCP server pool configurations, address bindings, statistics, etc. use the following commands respectively:

```
SR1000/config> show ip dhcps config

SR1000/config> show ip dhcps address pools

SR1000/config> show ip dhcps bindings

SR1000/config> show ip dhcps statistics
```

Note:
If the SR1000 fails/recovers, it does not keep track of the DHCP bindings which were assigned before the router failed.

Below is another example of DHCP server configuration.

Configuring the DHCP Server

To configure the DHCP server, enter:

```
SR/configure> ip dhcps
SR/configure/ip/dhcps> pool DynIP
SR/configure/ip/dhcps/pool DynIP> domain Avaya.net
SR/configure/ip/dhcps/pool DynIP> network 10.1.1.0 255.255.255.0
SR/configure/ip/dhcps/pool DynIP> default_router 10.1.1.1
SR/configure/ip/dhcps/pool DynIP> dnsserver 10.1.1.2
SR/configure/ip/dhcps/pool DynIP> exclude-range 10.1.1.2 10.1.1.10
SR/configure/ip/dhcps/pool DynIP> exclude-range 10.1.1.250 10.1.1.254
SR/configure/ip/dhcps/pool DynIP> commit
SR/configure/ip/dhcps/pool DynIP> exit pool
SR/configure/ip/dhcps> interface ethernet0
SR/configure/ip/dhcps> enable
SR/configure/ip/dhcps> exit 2
SR/configure>
```

IP Phone Support for Full mode with DHCP Server

On the Secure Router, the DHCP server can understand Avaya-specific DHCP options used to configure Avaya IP Phones in full mode. When the IP phones are configured in full mode, they initiate a DHCP discover broadcast on the network to which they are attached. The Secure Router matches the IP Phone to the corresponding DHCP pool and returns all the DHCP options configured for that DHCP pool. All the Avaya-specific DHCP options are defined under the **`ip dhcps pool`** subtree.

The DHCP options 66 and 150 are configured by setting the **`tftpserver`** option under **`dhcp pool`**. Option 66 returns the primary tftp server IP address (first entry) as a text field. DHCP option 150 returns multiple TFTP server IP addresses as a length encoded binary field where each address is 4 bytes.

The DHCP option 150 is defined by Cisco for the use of SIP phones so that they can have redundant backup for downloading the images on the SIP phones.

Use the following procedure to configure a DHCP pool with IP Phone options.

1. Enter configuration mode:

   ```
   configure terminal
   ```

2. Specify DHCP server configuration:

   ```
   ip dhcps
   ```

3. Specify the DHCP pool to configure:

```
pool <name>
```

4. To configure the alternate VLAN ID for IP Phones, enter:

```
altvlan <vlanid>
```

5. To specify the call server for IP Phones, enter:

```
callserver <ip1> port <port_val> appserver <ip2> svpserver
<ip3>
```

6. To specify a wireless server name for IP Phones, enter:

```
wireless <wireless-server>
```

Note:
This command cannot be present with any of the other IP Phone options.

7. To specify a TFTP server for IP Phones, enter:

```
tftpserver <tftp-server>
```

Name	Description
altvlan <vlanid>	Specifies an alternate VLAN ID for IP Phones. Valid range for <vlan id> is an integer of value 0 - 65535. This command configures DHCP option 191, which configures the alternate VLAN id that the IP phone is to use. This command configures a dummy DHCP option 128 so that the IP phone accepts this option. Example: `R1/configure/ip/dhcps/pool x # altvlan 100`
callserver <ip1> port <port_val> appserver <ip2> svpserver <ip3>	Specifies the call server for IP Phones. • <ip1>: Specifies the IP address of the call server port • port <port_val>: Specifies the port number on which the call server is listening, in the range 1024 – 65535 (default 4100) • appserver <ip2>: Specifies the IP address of the XAS application server • svpserver <ip3>: Specifies the IP address of the SpectraLink Voice Priority (SVP) server • This command configures DHCP option 128. There can be up to 2 call servers in a DHCP pool. The first call server entered is the primary call server. The svpserver option configures dhcp option 151. Example: `R1/configure/ip/dhcps/pool x # callserver 10.10.10.10 port 4200 appserver 20.20.20.20 svpserver 30.30.30.30`

wireless <wireless-server>	Specifies the IP address of a wireless server. This command cannot be present with any of the other IP Phone options. The maximum number of wireless servers is 3. This parameter configures DHCP option 43. Example: `R1/configure/ip/dhcps/pool x # wireless 10.10.10.20`
tftpserver <tftp-server>	Specifies the IP address of the TFTP server. The maximum number of TFTP servers is 8. This parameter configures DHCP option 66 and option 150 (multiple TFTP severs). Example: `R1/configure/ip/dhcps/pool x # tftpserver 10.10.10.30`

Chapter 11: Proxy DNS

The Secure Router 1000 Series and 3120 provide support for Proxy DNS. Proxy DNS receives a request from a host, resolves the domain name through communication with the DNS server, and sends the response to the host. Proxy DNS is disabled by default.

Without Proxy DNS, if a master link connected to an ISP-based DNS server went down, DNS queries could not be resolved. The solution to this issue would have been to change the DNS server IP address to the address of a backup link. Even though a Windows-based PC host can be configured with up to 10 DNS server entries, it is often not feasible to configure this many DNS servers on every available host. With the addition of Proxy DNS, the solution becomes much more simple.

Proxy DNS functions in such a way that it receives a request from a client and sends a response back. The DNS server is specified as the interface address connecting the PC to the router. Using Proxy DNS, clients do not need to worry about an ISP link or an exact DNS server, as the Proxy DNS feature handles these. In the case of a host, all that is required is configuration of the interface address of the router as the DNS server address.

The Proxy DNS feature supports multiple static (2) or dynamic (4) DNS server entries, of which any static entries have higher precedence. Dynamic entries can be added to the list of DNS servers by DHCP & PPPoE modules during registration of the module and can be removed when unregistered. When a client makes a request to Proxy DNS for the address of a particular domain name, Proxy DNS contacts a list of DNS servers in succession to resolve the domain name. When the domain has been resolved to an IP address, the entry is added to the cache and also sent to the requesting client. When a DNS response is received from the DNS server it is stored in the cache for the length of time specified by the TTL received for the particular name. The cache supports up to 80 entries. If a client queries for a previously cached domain, Proxy DNS responds with the cached entry. Removing the need to contact the DNS server for this entry reduces traffic. When the cache table reaches its 80 entry capacity older dynamic cache entries are removed to accommodate the new entries.

The DNS client will remain functioning as it did previously, as long as a primary and secondary name server exists.

Configuring Proxy DNS

Use the following procedure to manually configure the proxy DNS feature to cache an address.

Procedure steps

1. 1 To configure proxy DNS, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Enter the ip sub-tree

   ```
   ip
   ```

3. Ensure a DNS server has been configured.

```
pname_server <A.B.C.D>
```

4. Optionally, add a second DNS server.

   ```
   name_server <A.B.C.D>
   ```

5. Enter the proxy-dns sub-tree.

   ```
   proxy-dns
   ```

6. Enable Proxy DNS.

   ```
   enable
   ```

7. Add a DNS cache entry via the CLI.

   ```
   add-cache <domain>
   ```

Table 10: Variable definitions

Variable	Value
<A.B.C.D>	The primary name server address.
<domain>	The domain to add to the proxy cache.

Chapter 12: CONFIGURING AUTHENTICATION

Configuring Authentication

Users can configure a RADIUS profile on the SR1000 to authenticate users centrally using a RADIUS server. A sample topology outlining Secure Router 1000 Series authentication:

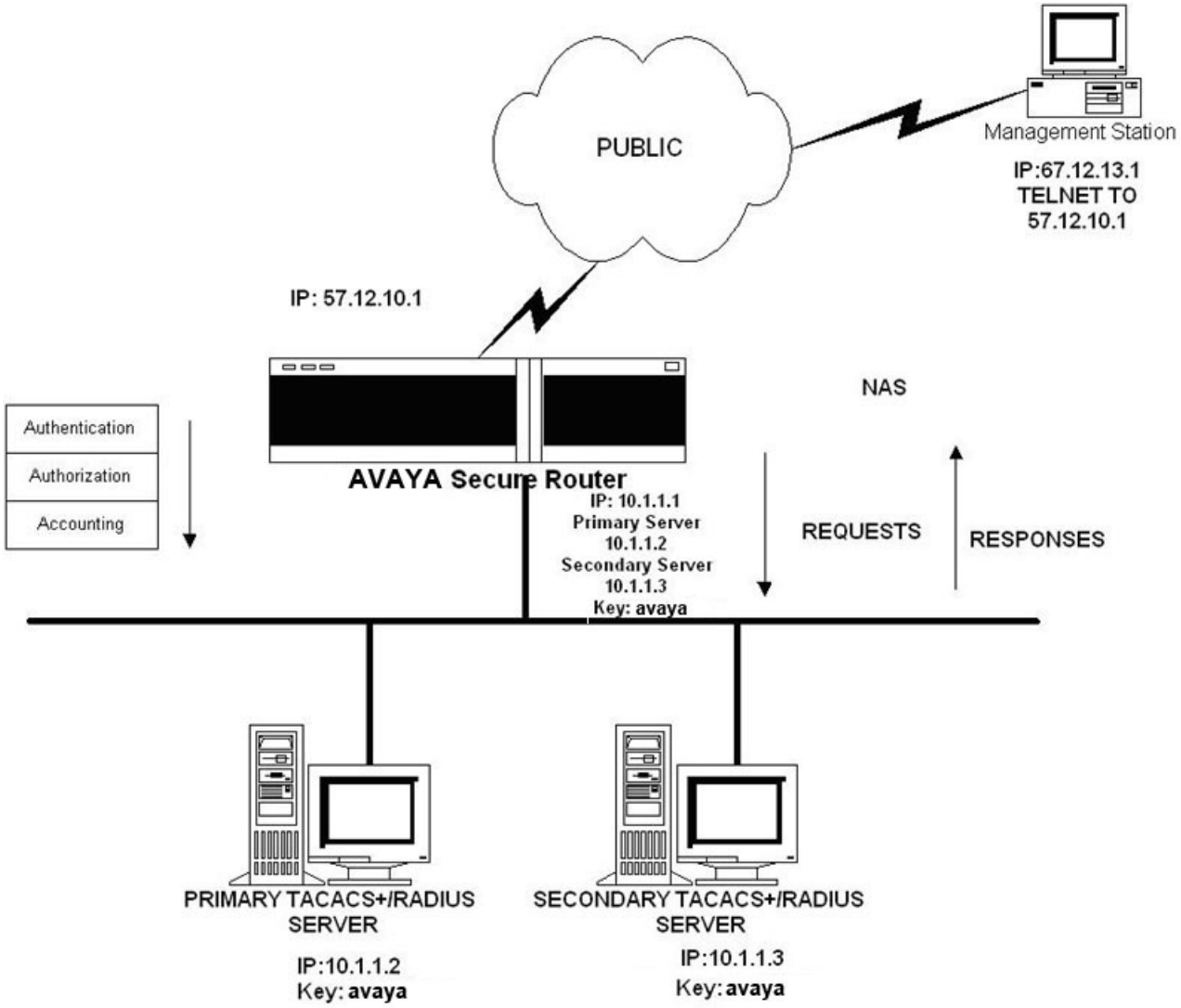


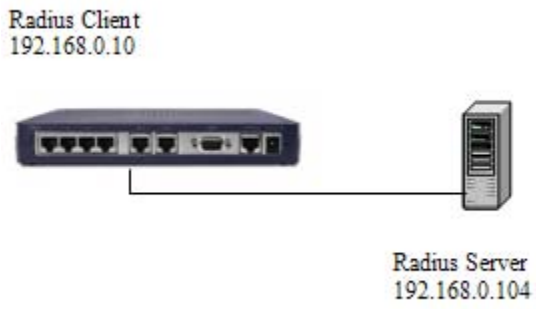
Figure 9: Configuring Authentication

Configure Secure Router 1000 Series authentication

```
Avaya/configure/aaa > enable
Avaya/configure/aaa > radius
Avaya/configure/aaa/radius > primary_server 10.1.1.2
Primary Radius server configured.
Avaya/configure/aaa/radius > secondary_server 10.1.1.3
Secondary Radius server configured.
Avaya/configure/aaa/radius > src_address 10.1.1.1
Radius Client Source Address configured.
Avaya/configure/aaa/radius > shared_key avaya
```

```
Shared secret changed successfully.
Avaya/configure/aaa/radius > show aaa radius

RADIUS CLIENT CONFIGURATION
---------------------------
Source address      : 10.1.1.1
Primary server      : 10.1.1.2
Secondary server     : 10.1.1.3
Authentication port    : 1812
Timeout in seconds   : 8
Maximum retries    : 3

Avaya/configure/aaa/radius > exit

Avaya/configure/aaa > tacacs
Avaya/configure/aaa/tacacs > primary_server 10.1.1.2
Avaya/configure/aaa/tacacs > secondary_server 10.1.1.3
Avaya/configure/aaa/tacacs > shared_key avaya
Avaya/configure/aaa/tacacs > exit 2

Avaya/configure > show aaa tacacs

TACACS+ CLIENT CONFIGURATION
----------------------------
Primary server      : 10.1.1.2
Secondary server     : 10.1.1.3
Server port        : 49
Timeout in seconds    : 5
Maximum retries     : 2

Avaya/configure > aaa
Avaya/configure/aaa > authentication login default
tacacs/radius/local
Avaya/configure/aaa > authentication protocols default
pap/chap/ascii
Avaya/configure/aaa > authorization commands default
tacacs/local
Avaya/configure/aaa > show aaa status
```

Below is an example of configuring Radius authentication with FreeRadius Server and establishing user levels.




```
conf t
aaa
authentication login default radius/local
authentication protocols default ascii
enable
```

```
radius
primary_server 192.168.0.104
src_address 192.168.0.10
shared_key avayanet
exit radius
exit aaa
interface ethernet 0
ip address 192.168.0.10 255.255.255.0
exit ethernet
```

There is a need to modify 4 files on the Radius server:

- Hosts file
  - (/etc/hosts)
  - Need to add in client
  - 192.168.0.10 SR1004
- Client.conf file
  - (/usr/local/etc/raddb/clients.conf)
  - Need to add client and shared key
  - }

    client 192.168.0.10 {

    secret = avayanet

    shortname = SR1004 }
- Users file

  (/usr/local/etc/raddb/user)

  Need to add the user

  kirk Auth-Type := Local, User-Password = “jamest”

  Service-Type = Admin-User

  spock Auth-Type := Local, User-Password ="vulcan"

  Service-Type = Level2-User

  mccoy Auth-Type := Local, User-Password ="bones"

  Service-Type = Level3-User

  Sulu Auth-Type := Local, User-Password ="helm"

  Service-Type = Level4-User
- Dictionary file

  (/usr/local/share/freeradius/dictionary)

  Need to add in the different user levels

  #Avaya Dictionary:

VALUE Service-Type Admin-User 1

VALUE Service-Type Level2-User 2

VALUE Service-Type Level3-User 3

VALUE Service-Type Level4-User 4

Support for Vendor Specific Attribute (VSA) on RADIUS clients

During the user authentication process, in response to the access-request, the RADIUS server sends the privilege level as part of the vendor-specific attribute (26) in the access-accept packet.

The RADIUS client on the Secure Router detects vendor-specific attribute data, gets the privilege level, and maps it to the appropriate access levels on the Secure Router.

The following table shows the mapping of the VSA value to router user privilege level. Note that, if both the Service-Type attribute and the Vendor-Specific attribute are configured for a user, access privilege level for that user is based on the Service-Type.

Radius Vendor Specific Attribute Value	Secure Router Privilege Level
1 Login	4
2 Framed	2
3 Callback Login	3
4 Callback Framed	4
5 Outbound	4
6 Administrative	1
7 NAS Prompt	4
8 Authenticate Only	4
9 Callback NAS Prompt	4
10 Call Check	4
11 Callback Administrative	4

Chapter 13: Accounting under TACACS support

The Secure Router 1000 Series and 3120 provide support for Terminal Access Controller Access Control System (TACACS) accounting. This feature allows an administrator to audit user activity on a router at any date or time. TACACS accounting details what commands were issued by a particular user.

The TACACS accounting system tracks and stores Attribute Value data on a TACACS accounting server. This accounting data includes details such as user name, the user's IP address, a timestamp and the activity - perhaps a Login or execution of a particular command. The data can then be analyzed for user activity on a router at any date or time. For example, when a user connects to an interface remotely via Telnet or SSH using the correct username and password, a log will be written and can be viewed on the TACACS server.

All accounting methods must be defined through Authentication Authorization Accounting (AAA). Much like AAA, TACACS accounting is configured through the definition of a named list of accounting commands with specific methods, then applying this list to one or more interfaces.

There are two main TACACS accounting commands:

- network - If applied to an interface, enables accounting for users login and logout.
- commands - If applied to an interface, enables accounting for all commands executed by a user.

There are three methods of TACACS accounting:

- stop-only - If specified, sends a notice to stop record accounting at the end of the specified activity.
- start-stop - If specified, sends a notice to start record accounting when a process begins and sends a notice to stop record accounting at the end of the specified activity. This allows the requested user process to begin even if the start accounting record was not acknowledged by the accounting server.
- wait-start - If specified, sends a notice to start and stop accounting to the accounting server. In this scenario, the user service does not begin until the start accounting record is acknowledged.

Note:

If you create an accounting method list with a list name of "default", all interfaces will use this list without applying in on an interface. You can override this "default" list only when you create an explicit method list and apply it to the interface.

Configuring TACACS accounting

Use the following procedure to configure TACACS accounting.

Procedure steps

1. To configure TACACS accounting, enter Configuration Mode.

```
configure terminal
```

2. Enter the aaa command sub-tree.

```
aaa
```

3. Configure an access-list for commands.

```
accounting commands <listname | [default]> <start_stop|stop_only|
wait-start>
```

4. Configure an access-list for a network.

```
accounting network <listname | [default]> <start_stop|stop_only|wait-
start>
```

5. Exit back a level.

```
exit
```

6. Enter interface mode.

```
interface <interface>
```

7. Apply accounting to the interface.

```
aaa accounting <commands|network> <list>
```

Table 11: Variable definitions

Variable	Value
<commands|network>	The type of accounting to apply to the interface.
<interface>	The interface to work with.
<list>	The list to apply to the interface.
<listname>	The name of the accounting list. If list name is specified as "default", all interfaces use this list without further configuration.
<start_stop|stop_only|wait-start>	• start_stop - Start and Stop records are sent. • stop_only - Only Stop records are sent. • wait-start - Start and Stop records are sent, but service starts after acknowledgement.

Chapter 14: Compressed RTP

Compressed RTP (cRTP) is a QoS feature that is required for bandwidth-limited wide area connections. Bandwidth limited connections are wide area links with less than T1 (1.54Mb/s) speeds. Compressed RTP reduces the number of times the complete VoIP header need to be send, thereby reducing the volume of traffic on the network.

The cRTP compression method sends a complete IP/UDP/RTP header over the PPP connection at the beginning of every flow to provide all the header information for the egress end of the PPP connection. This is followed by sending only the information that changes in the header encoded into 2-4 bytes. Since there is a probability that packets could get lost over the PPP connection where header compression is performed, therefore a complete uncompressed IP/UDP/RTP header is sent once every few seconds to re-synchronize ingress compressor with the egress decompressor.

cRTP can be used on MLPPP bundles as well as PPP bundles. Currently, cRTP is supported only on IPv4. The interface should be configured with an IPv4 address before cRTP can be enabled on the interface.

Configuring cRTP on the Secure Router 100x

The following is a basic summary of cRTP configuration:

1. Configure the bundle
2. Encapsulate the bundle with PPP protocol
3. Assign an IP address
4. Enable RTP compression

This example shows how to configure cRTP on the Secure Router 100x.

```
SR1004/configure> interface bundle wan

SR1004/configure/bundle wan> link t1 1

SR1004/configure/bundle wan> encapsulation ppp

SR1004/configure/bundle wan> ip address 5.5.5.1 24

SR1004/configure/bundle wan> rtp
```

cRTP Considerations:

- TCP header compression is not supported.
- Packets with IP options are not compressed.
- Max-header size that can be compressed or decompressed is 40 bytes. This value is not configurable.
- Only RTP packets will be compressed.
- If the Secure Router using cRTP is in networks with existing Secure Routers running code earlier than 9.2 for 3120 and 9.2 for SR1000, use the CLI command **`no negotiation`** for the backward compatibility mode.

Configuring cRTP timeout

cRTP timeout is used to configure the timeout (flushing time) for the context table entries for both compressor and decompressor engine on the same router.

Note:
Flushing for any particular context entry will happen only if for this (timeout) duration no packets are received with that particular context-id.

Syntax:

```
Router/configure/interface/bundle name> rtp

Router/configure/interface/bundle name/rtp> [no] timeout [seconds]
```

Example:

```
Router/configure/interface/bundle WAN1/rtp > timeout 120
```

Troubleshooting cRTP Common Problems

To check if compression is enabled use the **`show interface bundle [bundle-name]`** command. If compression is enabled, “Compressor is ON” will be displayed in the output, otherwise “Compressor is OFF” will be displayed.

If RTP is not compressed, check for the following possible causes:

- All the contexts on that bundle/interface are occupied by other rtp-streams.
- The UDP destination port number is not even or it’s not greater than 1024.
- RTP version is not equal to 2.

Configuring interoperability with the Cisco 2800

This example shows you how to configure the Secure Router 100x to interoperate with the Cisco 2800.

In this example, you must have a Secure Router on which the RTP and cRTP options are already configured.

```
Cisco> enable

Cisco> configure terminal

Cisco(config)> interface serial0/1/0:0

Cisco(config-if)> encapsulation ppp

Cisco(config-if)> ip address 5.5.5.2 255.255.255.0

Cisco(config-if)> ip rtp header-compression ietf-format

Router(config-if)> ip rtp compression-connections 150
```

Chapter 15: DTE-to-DTE Multilink Frame Relay

Multilink Frame Relay (MFR) is actually composed of two standards: FRF.15 and FRF.16. The latter is more common and defines UNI/NNI interfaces for implementing MFR. FRF.16 is used for multiplexing dedicated T1s in the local loop and requires compatible equipment at the carrier POP. FRF.15, or DTE-to-DTE MFR is used for multiplexing frame relay T1s between end points without impacting POP equipment. As a result, FRF.15 can be implemented across multiple frame relay carriers to provide additional redundancy. This application discusses considerations for using this standard. All Secure Routers support FRF.15 and FRF.16.

Features:

- Low cost way of providing added bandwidth to private networks
- Can be done without the knowledge of the Frame Relay provider
- Scalable bandwidth using MFR for customers based on T1 access
- Traffic can be routed or switched using Frame Relay at the end points

A customer desiring to implement DTE-to-DTE MFR can use the architecture illustrated in Figure 1. The normal ordering process can be used to obtain the fame relay T1s. From the perspective of the CPE, the SR1001s combine those different frame relay PVCs into a consolidated, larger pipe.

FRF.15 uses an aggregated virtual circuit (AVC) for the combined interface. The AVC is composed of constituent virtual circuits (CVC) that represent the frame relay T1s ordered from the carrier(s). In this example, the SR1001s are configured with DTE LMI; the carrier frame switches are DCE.

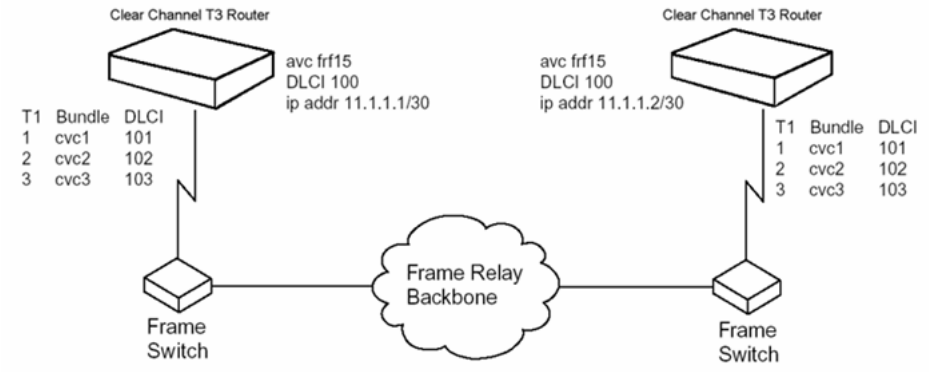


Figure 10: MFR Using FRF.15

In Secure Routers, CVCs are configured using the bundle construct normally used for a non-ethernet interface. After the CVCs are configured, they can be assigned to the AVC. The AVC, frf15 in this case, is assigned a DLCI of 100 on both ends and an IP address in the 11.1.1.0/30 subnet. The AVC names and DLCI numbers can be different on each end if necessary. The frame switches are configured for DLCIs

101, 102, and 103 on the respective T1s. In this example, the Secure Router 1000 Series configurations are almost identical. The primary difference is the IP address assigned to the AVC. The configuration for the left Secure Router 1000 Series is shown below.

```
> Configure Ethernet interface
SR/configure> interface ethernet 0
SR/configure/ethernet0> ip addr 192.168.1.1
255.255.255.0
SR/configure/ethernet0> exit
```

```
> Configure CVC1
SR/configure> interface bundle cvc1
SR/configure/interface/bundlecvc1> link t1 1
SR/configure/interface/bundle cvc1> encapsulation
frelay
SR/configure/interface/bundle cvc1> fr
SR/configure/interface/bundle cvc1/fr> intf_type dte
SR/configure/interface/bundle cvc1/fr> pvc 101
SR/configure/interface/bundle cvc1/fr> exit 3
```

```
> Configure CVC2
SR/configure> interface bundle cvc2
SR/configure/interface/bundle cvc2> link t1 2
SR/configure/interface/bundle cvc2> encapsulation
frelay
SR/configure/interface/bundle cvc2> fr
SR/configure/interface/bundle cvc2/fr> intf_type dte
SR/configure/interface/bundle cvc2/fr> pvc 102
SR/configure/interface/bundle cvc2/fr> exit 3
```

```
> Configure CVC3
SR/configure> interface bundle cvc3
SR/configure/interface/bundle cvc3> link t1 2
SR/configure/interface/bundle cvc3> encapsulation
frelay
SR/configure/interface/bundle cvc3> fr
SR/configure/interface/bundle cvc3/fr> intf_type dte
SR/configure/interface/bundle cvc3/fr> pvc 103
SR/configure/interface/bundle cvc3/fr> exit 3
```

```
> Configure AVC
SR/configure> interface avc frf15 100
SR/configure/interface/avc frf15 100> cvc 101 cvc1
SR/configure/interface/avc frf15 100> cvc 102 cvc2
SR/configure/interface/avc frf15 100> cvc 103 cvc 3
SR/configure/interface/avc frf15 100> ip address
11.1.1.1 255.255.255.252
SR/configure/interface/avc frf15 100> exit
SR> configure
```

The above configuration does not include statements for policing and traffic shaping, so all PVCs are given the full CIR for the interface. Once the AVC is configured, the Secure Routers can be configured for transparent IP multiplexing or for static routing. These details are omitted.

The primary advantage of FRF.15 is that no support is required on the POP side of the network. That fact makes this standard ideal for companies that need increased bandwidth for their frame relay based private network. FRF.15 is more susceptible to differential delay because the multiplexed links extend well beyond the local loop. Fortunately, differential delay encountered within the United States is typically small enough to have little to no impact on actual traffic flow.

Chapter 16: IGMP CONFIGURATION GUIDE

Internet Group Management Protocol (IGMP)

Internet Group Management Protocol (IGMP) is enabled on hosts and routers that want to receive multicast traffic. IGMP informs locally-attached routers of their multicast group memberships. Hosts inform routers of the groups of which they are members by multicasting IGMP Group Membership Reports. When multicast routers listen for these reports, they can exchange group membership information with other multicast routers. This reporting system allows distribution trees to be formed to deliver multicast datagrams. The original version of IGMP was defined in RFC 1112, Host Extensions for IP Multicasting. Extensions to IGMP, known as IGMP version 2.

IGMPv2 improves performance and supports the following message types:

- IGMP Query: IGMP Query is sent by the router to know which groups have members on the attached network.
- IGMP Reports: IGMP reports are sent as a response to the query by hosts to announce their group membership. Reports can be sent "unsolicited" when the hosts come up.
- IGMP Leaves: IGMP Leaves are sent by the host when it relinquishes membership of a group.

The latest extension to the IGMP standard is Version 3, which includes interoperability with version 2 and version 1 hosts, also provides support for source filtering. Source filtering enables a multicast receiver host to signal to a router which groups it wants to receive multicast traffic from, and from which source(s) this traffic is expected. This membership information enables the router to forward traffic only from those sources from which receivers requested the traffic.

IGMPv3 supports applications that explicitly signal sources from which they want to receive traffic. With IGMPv3, receivers signal membership to a multicast host group in the following two modes:

Example

- INCLUDE mode: In this mode, the receiver announces membership to a host group and provides a list of IP addresses (the INCLUDE list) from which it wants to receive traffic.
- EXCLUDE mode: In this mode, the receiver announces membership to a host group and provides a list of IP addresses (the EXCLUDE list) from which it does not want to receive traffic. This indicates that the host wants to receive traffic only from other sources whose IP addresses are not listed in the EXCLUDE list. To receive traffic from

all sources, like in the case of the Internet Standard Multicast (ISM) service model, a host expresses EXCLUDE mode membership with an empty EXCLUDE list.

IGMPv3 is used by the hosts to express their desire to be a part of the source-specific multicast (SSM) which is an emerging standard used by routers to direct multicast traffic to the host only if its is from a specific source.

IGMP Commands

The IGMP commands supported are:

```
ip igmp
ignore-v1-messages
ignore-v2-messages
last-member-query-count
last-member-query-interval
query-interval
query-response-interval
require-router-alert
robustness
send-router-alert
startup-query-count
startup-query-interval
group filter
version
debug ip igmp
debug ip igmp state
debug ip igmp normal
debug ip igmp packet query
debug ip igmp packet report
debug ip igmp packet leave
show ip igmp groups
show ip igmp interface
```

```
clear ip igmp groups
```

IGMP Configuration Examples

Use the examples shown in this section to use IGMP in multicast configurations.

Example 1

The following example enables IGMP.

```
Router/configure> ip igmp
```

Example 2

With the command line still in Interface Configuration Mode, the following example disables IGMP.

```
Router/configure> no ip igmp
```

Example 3

In the following example, the **ignore-v1-messages** command is used to disable processing of IGMPv1 messages on interface ethernet 0.

```
Router/configure/ip/igmp/interface ethernet0> ignore-v1-messages
Router/configure/ip/igmp/interface ethernet0> exit 3
Router/configure>
```

Example 4

In the following example, the **ignore-v2-messages** command disables processing of IGMPv1 messages on interface ethernet 0.

```
Router/configure/ip/igmp/interface ethernet0> ip igmp
ignore-v2-messages
Router/configure/ip/igmp/interface ethernet0> exit 3
Router/configure>
```

Example 5

The following example configures the Last Member Query Count to be 4 on ethernet 0.

```
Router/configure/ip/igmp/interface ethernet0> last-member-query-count 4
```

Example 6

In the following example for interface ethernet 0, the Robustness is configured to be 3. The Last Member Query count is configured to be 5.

```
Router/configure/ip/igmp/interface ethernet0> robustness 3
Router/configure/ip/igmp/interface ethernet0> last-member-query-count 5
Router/configure/ip/igmp/interface ethernet0> exit 3
Router/configure>
```

Example 7

The following example configures ethernet 0 with the default Last Member Query Interval of 2000 milliseconds (20 seconds).

```
Router/configure/ip/igmp/interface ethernet0>
last-member-query-interval 2000
```

Example 8

The following example configures ethernet 0 with the default Query Interval to be 200 seconds.

```
Router/configure/ip/igmp/interface ethernet0> query-interval 200
```

Example 9

The following example configures the default Query Response Interval to be 10 seconds (or 100 deciseconds) for ethernet 0.

```
Router/configure/ip/igmp/interface ethernet0> query-response-time 100
```

Example 10

The following example turns require-router-alert on for interface ethernet 0.

```
Router/configure/ip/igmp/interface ethernet0> require-router-alert
```

Example 11

The following example configures the default Robustness to be 3 for interface ethernet 0.

```
Router/configure/ip/igmp/interface ethernet0> ip igmp robustness 3
```

Example 12

The following example turns the send-router-alert option off for interface ethernet 1.

```
Router/configure/ip/igmp/interface ethernet1> no send-router-alert
```

Example 13

The following example configures IGMP version 2 to run on interface ethernet 0.

```
Router/configure/ip/igmp/interface ethernet0> version 2
Router/configure/ip/igmp/interface ethernet0> exit 3
Router/configure>
```

IGMP Snooping

IGMP snooping allows a Secure Router to read (snoop) IGMP packets transferred between IP multicast routers and IP multicast hosts to learn the IP Multicast group membership. Without IGMP Snooping, the Secure Router handles IP multicast traffic in the same manner as network broadcast traffic and forward frames received on one interface to all other interfaces. This creates excessive traffic on the network and affects network performance. IGMP Snooping allows routers to monitor network traffic and determine hosts that want to receive multicast traffic.

IGMP snooping currently supports the following configurations:

- Global enabling or disabling of IGMP snooping. When disabled, IGMP snooping does not process any IGMP related packets. When enabled, the packets are processed only on the VLANs on which IGMP snooping is enabled
- Enabling or disabling of IGMP snooping on a specified VLAN. By default, IGMP snooping is disabled on all VLANs.
- Configuring the Secure Router to specify the interface and VLAN on which a layer 3 multicast router is configured
- Enabling or disabling of querier on a VLAN. On enabling the querier, the Secure Router will send periodic query messages on the VLAN
- Enabling or disabling of fast leave. If enabled, the VLAN leaves the group immediately after receiving a membership leave message. If disabled, the router sends query messages 3 times to verify whether any host is still interested in receiving the multicast stream on this VLAN.
- Configuring IGMP version on a VLAN. This parameter specifies the version of IGMP message to be used on a VLAN. Version 1 and 2 are supported.
- Configuring query interval on a VLAN. This parameter specifies the query interval in milliseconds for query messages to be sent on a VLAN (default is 125 000 milliseconds).
- Configuring last member query interval. This parameter specifies the interval in millisecond of the query message to be sent upon receiving a membership leave message (default is 1000 milliseconds)
- Configuring maximum response time in centi-seconds (default 100 centi-seconds or 1000 milliseconds). This value is used to calculate the membership expiry timer using the following formula:

  Membership expiry = 2 x query interval + maximum response time in seconds

CLI configuration commands

This section describes the CLI commands used to configure IGMP Snooping.

To enable IGMP Snooping globally:

```
Host/configure #igs
Host/configure/igs# snooping-enable
```

To disable IGMP Snooping globally:

```
Host/configure #igs
Host/configure/igs# no snooping-enable
```

To enable IGMP Snooping on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs /vlan 10 # snooping-enable
```

To disable IGMP Snooping on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs /vlan 10 # no snooping-enable
```

To configure a multicast router port for a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs /vlan 10 # mrouter wan1
```

To enable querier on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs/vlan 10 # querier-enable
```

To disable querier on a VLAN

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs/vlan 10 # no querier-enable
```

To enable fast leave on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs/vlan 10 # fast-leave-enable
```

To disable fast leave on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs/vlan 10 # no fast-leave-enable
```

To configure the IGMP version on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs/vlan 10 #version 1
```

To configure the query interval on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs/vlan 10 #query-interval 150000
```

To configure the last member query interval on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs/vlan 10 # last-member-query-interval 1500
```

To configure the max response time on a VLAN:

```
Host/configure #igs
Host/configure/igs#vlan 10
Host/configure/igs/vlan 10 # max-response-time 150
```

CLI Display commands

This section describes the CLI commands used to display the IGMP Snooping configuration.

To display configuration details:

```
Host # show igs config
```

Sample output:

```
Config
IGMP Snooping: ENABLED
Vid Snooping Fast Querier IGMP Query Last Max
Leave Ver Interval Query Resp
Interval Time
10 ENABLED ENABLED ENABLED 1 150000 1500 150
```

To display multicast groups learned by IGMP snooping on a particular interface:

```
Host # show igs groups interface wan2
```

Sample output:

```
Groups:
Vid GroupIPAddress Interface
10 227.1.1.1 wan2
10 227.1.1.10 wan2
```

To display the interfaces on which a particular multicast IP address is learned by IGMP snooping:

```
Host # show igs groups ip 227.1.1.1
```

Sample output:

```
Groups:
Vid GroupIPAddress Interface
10 227.1.1.1 wan2
10 227.1.1.1 ethernet0/
```

To display multicast groups learned by IGMP snooping on a particular VLAN:

```
Host # show igs groups vlan 10
```

Sample output:

```
Groups:
Vid GroupIPAddress Interface
10 227.1.1.1 wan2
10 227.1.1.1 ethernet0/1
10 227.1.1.10 wan2
```

To display all groups learned by IGMP snooping:

```
Host # show igs groups all
```

Sample output:

```
Groups:
Vid GroupIPAddress Interface
10 227.1.1.1 wan2
10 227.1.1.1 ethernet0/1
10 227.1.1.10 wan2
```

To display multicast routers learned or configured for IGMP snooping:

```
Host # show igs mrouters
```

Sample output:

```
Mrouters:
Vid Interface
10 wan1
```

To display IGMP snooping packet statistics:

```
Host # show igs statistics
```

Sample output:

```
Statistics:
RXCNT:
Interface Join Leave Query Invalid
wan1 15 2 25 0
wan2 2 0 0 0
TXCNT:
Interface Join Leave Query Invalid
wan1 15 1 25 0
wan2 0 0 25 0
```

To display IGMP snooping in detail:

```
Host # show igs detail
```

Sample output:

```
Config
IGMP Snooping: ENABLED
Vid Snooping Fast Querier IGMP Query Last Max
Leave Ver Interval Query Resp
Interval Time
10 ENABLED ENABLED ENABLED 1 150000 1500 150
Groups:
Vid GroupIPAddress Interface
10 227.1.1.1 wan2
10 227.1.1.1 ethernet0/1
10 227.1.1.10 wan2
Mrouters:
Vid Interface
10 wan1
Statistics:
RXCNT:
Interface Join Leave Query Invalid
wan1 15 2 25 0
wan2 2 0 0 0
TXCNT:
Interface Join Leave Query Invalid
wan1 15 1 25 0
wan2 0 0 25 0
```

CLI Debug commands

To redirect debug messages to “/flash1/IgsDbg.txt” and to disable console printing of debug messages:

```
Host # debug igs file-logging
```

To disable file logging and enable console printing of debug messages:

```
Host # no debug igs file-logging
```

To print configuration related debug messages:

```
Host # debug igs configurations
```

To disable configuration related debug messages:

```
Host # no debug igs configurations
```

To enable error/failure related debug messages:

```
Host # debug igs errors
```

To disable error/failure related debug messages:

```
Host # no debug igs errors
```

To enable events related debug messages:

```
Host # debug igs events
```

To disable events related debug messages:

```
Host # no debug igs events
```

To enable interface related debug messages:

```
Host # debug igs interface
```

To disable interface related debug messages:

```
Host # no debug igs interface
```

To enable memory related debug messages:

```
Host # debug igs memory
```

To disable memory related debug messages:

```
Host # no debug igs memory
```

To enable packets related debug messages:

```
Host # debug igs packets
```

To disable packets related debug messages:

```
Host # no debug igs packets
```

To enable timer related debug messages:

```
Host # debug igs timer
```

To disable timer related debug messages:

```
Host # no debug igs timer
```

To enable all debug messages:

```
Host # debug igs all
```

To disable all debug messages:

```
Host # no debug igs all
```

Chapter 17: IP MULTIPLEXING OVERVIEW

Theory and Application

IP Multiplexing is a method for the transparent forwarding of IP packets between LAN and WAN interfaces. LAN to WAN forwarding is accomplished through a Proxy ARP process. A Secure Router maps a unique MAC address to each WAN link then responds with this MAC address when a device on the LAN broadcasts an ARP request for a remote device. These MAC addresses serve as "tags" for forwarding packets received on the LAN. WAN to LAN and WAN to WAN forwarding is based on configured forwarding entries.

IP Multiplexing differs from bridging and switching in that it does not flood traffic or perform address learning. IP Multiplexing devices differ from routers in that they do not appear as a router hop, and they cannot be specified as a default router/gateway on a LAN.

Packet Forwarding Modes

There are two modes for WAN to LAN and WAN to WAN packet forwarding

- IP Routes – Forwarding based on routing statements, both specific and default.
- Source Forwarding – Forwards all traffic arriving on a specified WAN bundle to a specified device on the LAN.

The following table provides information about applications and a suggested forwarding mode for each.

Table 12: Applications and Suggested Forwarding Modes

Application	Suggested Forwarding Mode
Forwarding traffic from different WAN links to separate routers on the LAN	Source Forwarding
Forwarding all WAN traffic to a single router on the LAN	Default IPMux Routes
Forwarding to both LAN and WAN router	Specific IPMux Routes

Proxy ARP and Packet Forwarding

In the simple network example below, router 1, router 2, and both Secure Router Ethernets are on a single 29-bit IP subnet. Consider the sequence that occurs when router 1 pings router 2.

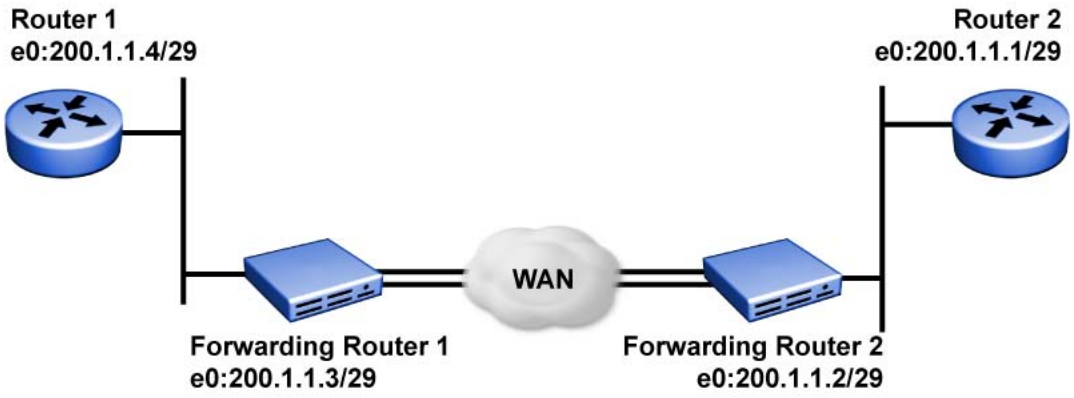


Figure 11: Proxy ARP and Packet Forwarding

1. Router 1 broadcasts an ARP request for 200.1.1.1.
2. Forwarding Router 1 recognizes that router 200.1.1.1 is reachable through its WAN interface, based on a configured IP route.
3. Forwarding Router 1 Proxy ARPs, responding with the MAC address mapped to bundle WAN1.
4. Router 1 unicasts the ping echo request to that MAC address.
5. Forwarding Router 1 forwards the echo request for 200.1.1.1 through the WAN1 bundle.
6. Forwarding Router 2 receives a packet on WAN2 and forwards it to directly connected router 2.
7. The echo reply from router 2 to router 1 is returned in the same manner.

Addressing in IP Multiplexing Networks

IP addressing in an IP Multiplexing design must take into account the fact that the router on the LAN must see the remote router as residing on the same LAN or IP network. There are a number of addressing schemes that can fulfill this requirement, including:

- Single subnet
- Split subnet
- Secondary addressing

Consider the following network, consisting of three remote sites. Two remote sites utilize Avaya equipment, while the third is a simple router/DSU combination. Five IP addressing schemes are provided below, all refer to the following network.

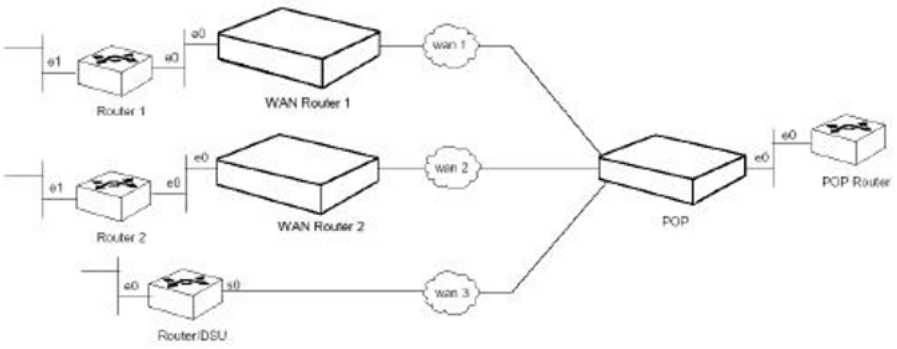


Figure 12: Addressing in IP Multiplexing Networks

Single Subnet

The emphasis in the single subnet approach is that all seven devices have interfaces in a single 28-bit subnet 192.1.1.0 / 28. The WAN addressing utilizes reserved address space.

Table 13: Single Subnet Addressing

POP Router	e0:	192.1.1.1/28
POP Avaya	e0: wan1: wan2: wan3:	192.1.1.2/28 10.1.1.1/30 10.1.1.5/30 10.1.1.9/30
Avaya 1	e0: wan1:	192.1.1.3/28 10.1.1.2/30
Router 1	e0:	192.1.1.4/28
Avaya 2	e0: wan1:	192.1.1.5/28 10.1.1.6/30
Router 2	e0:	192.1.1.6/28
Router/DSU	s0:	192.1.1.7/28

Split Subnet

This is similar to the single subnet scheme in that all four routers are in the same 28-bit subnet, but the Avaya products are on smaller, 30-bit subnets.

Table 14: Split Subnet Addressing

POP Router	e0:	192.1.1.1/28
POP Avaya	e0: wan1: wan2: wan3:	192.1.1.2/30 10.1.1.1/30 10.1.1.5/30 10.1.1.9/30
Avaya 1	e0: wan1:	192.1.1.5/30 10.1.1.2/30
Router 1	e0:	192.1.1.6/28
Avaya 2	e0: wan1:	192.1.1.9/30 10.1.1.6/30
Router 2	e0:	192.1.1.10/28
Router/DSU	s0:	192.1.1.14/28

Secondary Addressing: POP Only

Secondary addressing approaches rely on configuring the POP router with a secondary Ethernet address for each remote site. The POP-only approach uses secondary addresses at the POP while the remote router utilizes only a primary address.

Table 15: POP Only Secondary Addressing

POP Router	e0:	200.1.1.1/30 primary 199.1.1.1/29 secondary 199.1.1.9/29 secondary 199.1.1.17/29 secondary
POP Avaya	e0: wan1: wan2: wan3:	200.1.1.2/30 10.1.1.1/24 10.1.2.1/24 10.1.3.1/24
Avaya 1	e0: wan1:	199.1.1.2/29 10.1.1.2/24
Router 1	e0:	199.1.1.3/29
Avaya 2	e0: wan1:	199.1.1.10/29 10.1.2.2/24
Router 2	e0:	199.1.1.11/29
Router/DSU	s0:	199.1.1.18/29

Secondary Addressing: 30 Bit

This approach relies on configuring the POP router with a secondary Ethernet address for each remote site. The remote router is also configured with a secondary address in that same subnet. The 30-bit approach uses reserved addresses for bundle addressing. The router primary and the directly connected Secure Router reside in a different 30-bit subnet.

Table 16: 30-Bit Secondary Addressing

POP Router	e0:	200.1.1.1/30 primary 199.1.1.1/30 secondary 199.1.1.5/30 secondary 199.1.1.9/30 secondary
POP Avaya	e0: wan1: wan2: wan3:	200.1.1.1/30 10.1.1.1/30 10.1.1.5/30 10.1.1.9/30
Avaya 1	e0: wan1:	201.1.1.2/30 10.1.1.2/30
Router 1	e0:	201.1.1.1/30 primary 199.1.1.2/30 secondary
Avaya 2	e0: wan1:	202.1.1.2/30 10.1.1.6/30
Router 2	e0:	202.1.1.1/30 primary 199.1.1.6/30 secondary
Router/DSU	s0:	199.1.1.10/30

Secondary Addressing: 29 Bit

This approach utilizes a 29-bit subnet for each remote connection. Within each 29-bit subnet is the POP router secondary, the Secure Router WAN addressing, and the remote router secondary.

Table 17: 29-Bit Secondary Addressing

POP Router	e0:	200.1.1.1/30 primary 199.1.1.1/29 secondary 199.1.1.9/29 secondary 199.1.1.17/29 secondary
POP Avaya	e0: wan1: wan2: wan3:	200.1.1.2/30 199.1.1.2/29 199.1.1.10/29 199.1.1.18/29
Avaya 1	e0: wan	201.1.1.2/30 199.1.1.3/29
Router 1	e0:	201.1.1.1/30 primary 199.1.1.4/29 secondary
Avaya 2	e0: wan1:	202.1.1.2/30 199.1.1.11/29
Router 2	e0:	202.1.1.1/30 primary 199.1.1.12/29 secondary

Router/DSU	s0:	199.1.1.19/29

Pros and Cons of Different IP Addressing Schemes

The following table provides information about addressing scheme pros and cons.

Table 18: Addressing Schemes: Pros and Cons

Approach	Pros	Cons
Single Subnet	Minimizes consumption of IP address space	POP Avaya requires two route statements per remote connection.
Split Subnet	Less routes required in Avaya	Consumes 29-bit subnet per remote site.
Secondary Addressing	Easily Scalable	Consumes 29- or 30-bit subnet per remote. Not transparent to certain routing protocols.

Routing Considerations for IP Multiplexing

- RIP/RIP2/IGRP – Turn off split horizons to enable routing updates through secondary addresses, if used.
- EIGRP – Updates are sourced only from primary addresses, although routers will listen to updates arriving on primary and secondary.
- OSPF – For Cisco-compatible and other routers, routing updates are sourced and detected only on primary addresses, therefore secondary addressing schemes are not usable.
- BGP4 – Routing updates are fully functional over primary and secondary addresses.

Chapter 18: PPP, MLPPP, and HDLC

Layer Two Configurations:

Secure Routers may be configured for a variety of Layer 2 protocols. This document outlines HDLC, PPP, and Multilink PPP (MLPPP) configurations. Other Secure Router documents outline Frame Relay and Multilink Frame Relay configuration.

SR3120s are often used at POPs to aggregate data for WAN transmission. The following figure details PPP and multilink PPP connections from two CPE sites to a main site.

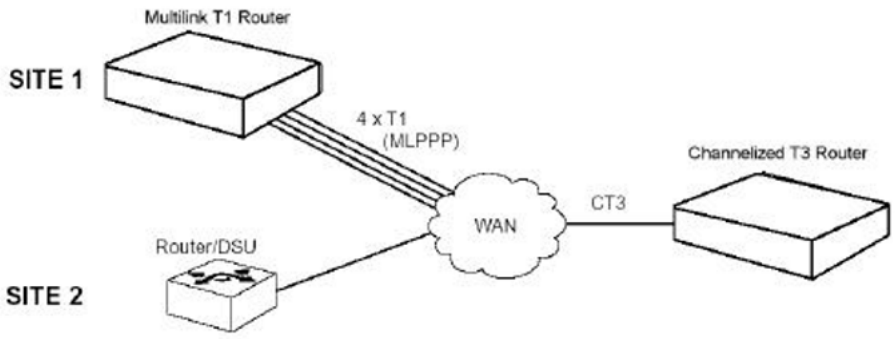


Figure 13: PPP/MLPPP Application

Site 1 uses a Multilink T1 Router to establish a 6 Mb/s MLPPP connection (four T1 lines) to the main site. In this example, MLPPP segmentation is configured lower than the default setting of 512 bytes, and the differential delay tolerance is tighter than the default 128 milliseconds.

Site 2 connects to the main site over a single T1 link with PPP encapsulation. The Channelized T3 Router PPP parameters (for example,, the maximum transmit and receive byte sizes) are adjusted to comply with the Site 1 router configuration.

MLPPP Configuration

Configure the SR1004 at Site 1

```
SR> configure term
SR/configure> interface bundle ToMain
SR/configure/interface/bundle> link t1 1-4
SR/configure/interface/bundle> encap ppp
SR/configure/interface/bundle> mlppp seg_threshold 1400
differential_delay 50
SR/configure/interface/bundle> ip addr 192.168.1.2 24
SR/configure/interface/bundle> exit
```

PPP and MLPPP Configuration

Configure the SR3120 at the Main Site

```
SR>/configure> interface bundle ToSite1
SR>/configure/interface/bundle> link ct3 1/1/5-8
SR>/configure/interface/bundle> encap ppp
SR>/configure/interface/bundle> mlppp seg_threshold 1400
differential_delay 50
SR>/configure/interface/bundle> ip addr 192.168.1.1
255.255.255.0
SR>/configure/interface/bundle> exit
```

```
SR>/configure> interface bundle ToSite2
SR>/configure/interface/bundle> link ct3 1/1/9
SR>/configure/interface/bundle> encap ppp
SR>/configure/interface/bundle> ppp mtu 100-250-1000 mru
100-250-1000
SR>/configure/interface/bundle> ip addr 192.168.2.1
255.255.255.0
SR>/configure/interface/bundle> exit
```

HDLC Configuration

HDLC encapsulation may be substituted for PPP between the main site and site 2

Configure the SR3120 at the Main Site

```
SR>/configure> interface bundle ToSite2
SR>/configure/interface/bundle> link ct3 1/1/9
SR>/configure/interface/bundle> encap hdlc
SR>/configure/interface/bundle> hdlc keepalive 20
SR>/configure/interface/bundle> ip addr 192.168.2.1
255.255.255.0
SR>/configure/interface/bundle> exit
```

Note:

In the above command sequence, the HDLC keepalive time interval was changed from its default setting of 10 seconds to 20 seconds

HDLC Errors

The way a Secure Router responds to excessive HDLC errors on a link in a bundle has changed in releases 8.0.1 and higher.

Previously, excessive HDLC errors on a specific bundle would cause that bundle and all bundles on the router to go up and down. When this happened, the only possible indication was an event that excessive errors had occurred on a certain freedom channel. This event would occur if there were 8000 HDLC errors in a row. This information had little value since there are no displays showing the association between freedom channels and bundle links. Also the router could get into this condition of all the bundles going up and down prior to the threshold of 8000 HDLC errors being reached. The problem can happen on T1, E1 and unframed E1 bundles whether single or multilink PPP.

When this link was bad and the router was flooded with HDLC errors the router would continually try to bring that link back up immediately. If the error rate was high enough and the link was brought down it would have the additional problem of not being able to reactivate the link after the errors stopped.

The additional problem is that when a link has excessive HDLC errors the only reliable way to stop the problem is to remove the link from the bundle.

In release 8.0.1 and higher, the router a SNMP bundle link down trap is sent when the link receives excessive HDLC errors. The following example shows a multilink PPP bundle Secure Router 1000 Series where bundle link E1 10 is having excessive HLDC errors.

```
snmpTrapOID.0                              BUNDLE-MIB|linkDownTrap
[1]linkNum (Integer):                      10<-------------------------- E1 Number
[2]linkType (Inte                          e1
[3]linkCt3Num (Iner):                      0
[4]linkDownCause(Integer):                 l1-failures<----------------- Drop cause
is Layer 1 failure
```

```
[5]linkBundleName (DisplayString):         SR1001 <-------------------- Bundle Name
[6]linkCircuitId (DisplayString):
[7]linkContactInfo (DisplayString):
[8]linkNameInfo (DisplayString):
[9]linkDescrInfo (DisplayString):
[10] snmpTrapEnterprise.0 (Object ID):    bundleTraps
```

If the box continues to receive excessive HDLC errors the router will continue to send bundle down link SNMP traps. The current default threshold is 1000 HLDC errors in a row. The threshold is now tunable and can range from 100 to 12,000. The **hdlc_error** command under the system command subtree sets the threshold for all links on the router. The syntax of the command is:

```
host/configure> system hdlc_error ?
NAME
hdlc_error - Setting the threshold for hdlc errors
(default: 1000)
SYNTAX
hdlc_error [ error_limit ]
DESCRIPTION
error_limit -- Number of continous hldc errors on a
link (default: 1000)
Valid Range(s) : 100 - 12000
```

After the excessive HLDC error threshold is exceeded there are two methods to stop the errors on the link. The first method is to shutdown the bundle. The **shutdown** and **no shutdown** commands are changed to deactivate or activate all the links in the bundle. Deactivating the link stops the router from receiving any errors on that link.

The second method is to configure the router to deactivate the link when it has exceeded the excessive HDLC error threshold. If it is a multilink bundle, the non-errored link can still pass traffic and the customer is still up. The new command for configuring whether to deactivate links after excessive HLDC errors is **hdlc_link_deactivate** under the system subtree. The default value is to not deactivate the link when it has reached the excessive HDLC error threshold.

When you save the configuration on the router with the command **save local** the HDLC error threshold and whether to automatically deactivate the link is saved. It is then read in on subsequent reboots.

The **show system configuration** command is updated to show the current settings for both the **hdlc_error** and **hdlc_link_deactivate** command. The output is displayed below for an Secure Router 1000 Series router:

```
show system configuration
```

```
System Configuration:
-----------------------------------------------
===============================================
Hardware Status:

DRAM quantity:           256MB
DRAM type:               DDR
Model Number:            1001
Serial Number:           1001
Processor ID:            NEC VR7701
HW Assembly Revision:    A
PCB Revision:            A
MB FPGA Revision:        0x11

Level 2 Cache:           256KB
Level 3 Cache:           0KB

===============================================
BOOT Device:                          FLASH
Downloadable FLASH Bootcode Version: hurr_111706
Physical EPROM Bootcode Version:     HB_hm062505

===============================================

----------------------------------------
VPN Accelerator card is present
-----------------------------------------------
WAN Interface ports -
         SERIAL - 2 ports available
            CT3 - 1 port available
-----------------------------------------------
-----------------------------------------------

Motherboard information:
Board Type: MBOARD
Serial Number: 00560CF3D5B10001
PCB Fab Revision: B
PCB Fab Number: 700-00085-01
Assembly  Revision: B
Assembly Number: 300-00085-01
Manufacturing Date: 110805

Linecard 1 information:
Board Type: DSERIAL
Serial Number: 00590AF3D5A10016
PCB Fab Revision: 3.0
PCB Fab Number: 700-00089-01
Assembly  Revision: 6.0
Assembly Number: 300-00089-01
Manufacturing Date: 102605

Linecard 2 information:
Board Type: CT3
Serial Number: N/A
PCB Fab Revision: N/A
PCB Fab Number: N/A
Assembly  Revision: N/A
Assembly Number: N/A
Manufacturing Date: N/A
-----------------------------------------------
Software Status:

Application Image Version:           r9.2
BOOT Device:                         FLASH
Downloadable FLASH Bootcode Version: hurr_111706
Physical EPROM Bootcode Version:     HB_hm062505
Mode:                                Routing
-----------------------------------------------
```

```
------------------------------------------------
Memory Status:

TOTAL DRAM: 0x10000000 bytes
 status   bytes    blocks   avg block  max block
 ------ --------- -------- ---------- ----------
current
   free  120131504       58    2071232 120068896
  alloc  100974688    73709       1369         -
cumulative
  alloc  933237552  16073214         58         -
------------------------------------------------
Flash Status:/flash1 ---------
Total Memory    Free Memory in flash
66453504                36765696
------------------------------------------------
System Diagnostics Results:
     DRAM Test:                  PASSED
     Flash Memory Test:          PASSED
     Temperature Test:           PASSED

---------------------------------------
HDLC Error Handling
Deactivate channel after 1000 hdlc errors
Channel inactive for 20 seconds
Log events after 900 hdlc errors
---------------------------------------
Hardware Watchdog Timer Status: enabled
------------------------------------------------
```

A new command is added at the bundle subtree to reactivate all the links in that bundle. The new command is **hdlc_link_activate** and has no parameters. If a link has automatically deactivated due to excessive HDLC errors this allows the link to be brought up without having to shutdown the bundle. The link could also be reactivated by issuing a **shutdown** and **no shutdown** command on the bundle.

The problem where PPP bundle link would not come back up when there were no errors after that link had been down due to excessive HDLC errors is fixed. The display of the bundle will now show both that there is a loss of frame along with the excessive HDLC errors under the link that has the problem. In the example below the PPP multilink bundle other has a link t1 13 which is down due to excessive HLDC errors and loss of frame.

If the router has the **hdlc_link_deactivate** command set then when the link that has excessive HDLC errors on it is clean of errors, it will recover frame but the link will remain down. The display of the bundle will still show the link has excessive HDLC errors on it. The link is still deactivated and to reactivate it issue the **hdlc_link_active** command on the bundle or the bundle that is shutdown and brought back up.

If the router has the **hdlc_link_deactivate** command not set, then when the link that has excessive HDLC errors on it is clean of errors it will recover frame and the link will come up. To unset **hldc_link_deactivate** enter the command **no system hdlc_link_deactivate**.

Chapter 19: Dial Backup via External Modem

The Secure Router 1000 Series and 3120 provide support for Dial Backup, which enables redundancy for routes. Backup routes using PPP bundles created over a dialup connection will become active when a primary route goes down.

The Secure Router connects to an external modem via the Aux port and will establish a dialup connection to a phone number specified in the backup PPP configuration using a feature called Dial-on-Demand Routing (DDR). There are two types of Dial-on-Demand Routing:

- Dial-on-Demand Routing:

  Dials when traffic needs to traverse a link

- Backup Dial-on-Demand Routing

  Dials when a designated primary interface goes down. You can configure a Backup Dial-on-Demand Routing interface by including the appropriate backup commands to a normal DDR interface configuration.

The IP address for the bundle is specified in the bundle configuration.

The Backup DDR mechanism

The Secure Routers use the Floating Static Route mechanism to automatically dialup to backup another route. To accomplish this, a secondary route is specified in addition to the primary route, with an administrative distance greater than the primary route. When the primary interface is functional it is used to route traffic. If the primary interface goes down packets are automatically sent to the backup interface where they trigger commands to dial a connection. A keepalive time is specified by the user during bundle configuration so that commands are automatically sent to disconnect a connection when there is no traffic for the allowed keepalive time period.

To allow this feature to function properly, the following Hayes AT commands will be supported via the CLI:

Table 19: Supported Hayes AT commands

S2	escape character
S3	carriage return character
S4	line feed character
S37	line connection speed
V1	result code will be sent in work form X1 send OK, CONNECT, RING, NO CARRIER, ERROR, NO ANSWER and CONNECT SPEED

Table 20: Programmed modem default settings

S0	rings to auto answer
S1	ring counter
S7	wait for carrier after dialing
S9	carrier detect response time
S10	lost carrier hang up delay

Table 21: Operation commands

A	cause modem to go off hook, works with ring detection
D	dial digit
E0	echo off
H0	on hook
H1	off hook
N1	enable auto mode
+++	mode change between data or command mode

Users have the option of creating multiple PPP backup bundles containing different configuration criteria and specifying them by order of priority. At this time, the Secure Routers contain only one Aux port, however the design of the feature is easily scalable should the option of multiple Aux ports become available.

The modems currently supported by this feature include Creative Blaster V9.2, Diamond Supra Max V9.2 and Best Data 56 K v9.2/v4.4.

Configuring dial backup via external modem

Use the following procedure to configure dial backup via external modem.

Procedure steps

1. To configure dial backup, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Create a dialer.

   ```
   dialer <name>
   ```

3. Configure async parameters.

   ```
   async
   ```

4. Configure the async port.

   ```
   port <port>
   ```

5. Configure the baud rate.

```
rate <baudrate>
```

6. Configure data bits.

```
databits <databits>
```

7. Configure the parity settings.

```
parity <setting>
```

8. Configure stop bits.

```
stopbits <stopbits>
```

9. Exit back a level.

```
exit
```

10. Begin configuring the modem.

```
modem
```

11. Configure the phone number to be called by the modem.

```
phonenum <number>
```

12. Configure the number of rings before answering.

```
answer <rings>
```

13. Configure the number of rings to wait during call setup.

```
call-setup-timeout <rings>
```

14. Configure the "lost carrier" hang up delay.

```
hangup <delay>
```

15. Configure the carrier detect response time.

```
detect <responsetime>
```

16. Configure the "wait for carrier" after dial delay.

```
wait <wait>
```

17. Configure using an AT string.

```
at <at_string>
```

18. Exit back a level.

```
exit
```

19. Configure the dialer idle-timeout interval.

```
idle-timeout <timeout>
```

20. Exit back a level.

```
exit
```

21. To attach to a bundle, create a bundle.

```
interface bundle <bundlename>
```

22. Configure the bundle to use the dialer.

```
link dialer <dialer>
```

23. Continue normal configuration of the bundle.

Table 22: Variable definitions

Variable	Value
<at_string>	The AT string used to configure the dialer.
<baudrate>	The Baud rate of the modem, default 56000.
<bundlename>	The name of the bundle.
<databits>	The number of databits, default 8.
<delay>	The lost carrier hang up delay, in the range 1 to 255. Default is 14.
<dialer>	The dialer name to link, maximum 8 characters.
<name>	The dialer name, maximum 8 characters.
<number>	The phone number, maximum length 25 characters, with or without hyphens. Prepending p or t indicates pulse or tone dialing.
<port>	The port description, maximum 10 characters.
<responsetime>	The carrier detect response time, in the range 1 to 255. Default is 6.
<rings>	The number of rings, in the range 1 to 255.
<setting>	The parity setting -none, even, or odd. Default is none.
<stopbits>	The number of stopbits -1, 2, or 3. Default is 1.
<timeout>	The idle timeout time, in the range 1 to 6000. Default is 180.
<wait>	The length of time to wait for dial delay, in the range 1 to 255. Default is 50.

Chapter 20: IP Packet Filter List

Configurations

Secure Routers can be configured for IP traffic filtering capabilities. IP traffic filtering allows creation of rule sets that selectively block TCP/IP packets on a specified interface. Filters are applied independently to all interfaces: Ethernet, serial, or WAN, as well as independently to interface direction: IN (packets coming in to the Secure Router) or OUT (packets going out of the Secure Router).

IP packet filtering capability can be used to restrict access to the Secure Router from untrusted, external networks or from specific, internal networks. An example would be a filter that prohibits external users from establishing Telnet sessions to the Secure Router, and allows only specific internal users Telnet access to the Secure Router.

- At the end of every rule list is an implied "deny all traffic" statement. Therefore, all packets not explicitly permitted by filtering rules, are denied. This effectively means that once you enter a "deny" statement in your filter list, you are implicitly denying all packets from crossing the interface. Therefore, it is important that each filter list contain at least one "permit" statement.
- The order in which you enter the filtering rules is important. As the Secure Router is evaluating each packet, the Avaya OS tests the packet against each rule statement sequentially. After a match is found, no more rule statements are checked. For example, if you create a rule statement that explicitly permits all traffic, all traffic is passed since no further rules are checked.
- The Avaya OS permits easy reordering of filter commands through **`filter_list insert`** and **`delete`** commands.

Example 1

Consider a Secure Router connected through a bundle WAN1 (WAN IP address 200.1.1.1) to an ISP, with Ethernet 0 (IP address 222.199.19.3) connected to the internal network. The network administrator wants to completely block Telnet access to the Secure Router from all external networks as well as from all internal networks except 222.199.19.0/28. All other TCP/IP traffic, such as FTP, Ping, and HTTP, is to flow unrestricted through the Secure Router.

Configure the Secure Router 1000 Series.

```
SR> configure term
SR/configure> ip
SR/configure/ip> filter_list filtera (gives the list a
name)
SR/configure/ip/filter_list> add deny tcp any
200.1.1.1 dport =23
SR/configure/ip/filter_list> add permit tcp
222.199.19.0/28 222.199.19.3 dport =23
SR/configure/ip/filter_list> add deny tcp any
222.199.19.3 dport =23
SR/configure/ip/filter_list> add permit ip any any
SR/configure/ip/filter_list> exit

SR/configure/ip> apply_filter ether0 filtera in
SR/configure/ip> apply_filter WAN1 filtera in
SR/configure/ip> exit
SR/configure> exit
SR> save local
```

Example 2

Consider the same network addressing as in example 1. The network administrator has a slightly different requirement - he wishes to permit FTP sessions from all networks to the internal FTP server (222.199.19.12), deny FTP sessions to all other addresses, and permit all other traffic to flow through the Secure Router.

Configure the Secure Router 1000 Series

```
SR> configure terminal
SR/configure> ip
SR/configure/ip> filter_list filterb (gives the list a
name)
SR/configure/ip/filter_list> add permit tcp any
222.199.19.12 dport =21
SR/configure/ip/filter_list> add deny tcp
any 222.199.19.0 dport =21
SR/configure/ip/filter_list> add permit ip any any
SR/configure/ip/filter_list> exit

SR/configure/ip> apply_filter WAN1 filterb in
SR/configure/ip> exit
SR/configure> exit
SR> save local
```

Example 3

Example 3 focuses on a filter list where the network administrator is specifically denying all traffic from a specific external network (197.100.200.0/24) access through the Secure Router.

Configure the Secure Router 1000 Series

```
SR> configure terminal
SR/configure> ip
SR/configure/ip> filter_list filterc (gives the list a
name)
SR/configure/ip/filter_list> add deny ip
197.100.200.0/24 any
SR/configure/ip/filter_list> add permit ip any any
SR/configure/ip/filter_list> exit

SR/configure/ip> apply_filter WAN1 filterc in
SR/configure/ip> exit
SR/configure> exit
SR> save local
```

IP Packet Filtering on VLAN subinterfaces

The Secure Router 1000 Series and 3120 provide support for IP packet filtering over VLAN subinterfaces. IP packet filtering involves the use of Access Control Lists (ACL) to filter network traffic by permitting or blocking packets at a router's interface.

Previously, the Secure Router 1000 Series and 3120 allowed configuration of subinterfaces, but did not contain support for attaching ACLs to the VLAN subinterfaces. Packet filters could only be attached to main interfaces. Adding this support allows for the attachment of ACLs to these VLAN subinterfaces, which enabled the use of packet filtering over such interfaces. No existing functionality has been affected.

To accommodate these enhancements, existing CLI commands have been modified for additional support of VLAN subinterfaces. The existing **`access-group`** command which was used to attach an access-list to an interface has now been modified to support a range of VLAN subinterfaces. This provides the user ease of attaching a single access-list to multiple VLAN subinterfaces, instead of issuing the command uniquely for each VLAN subinterface.

The maximum number of rule sets supported per system is 50, while the maximum number of filter rules per set is 2000. Users can configure up to 98 VLAN subinterfaces per Ethernet interface.

Some limitations of this feature include the lack of support for VLAN ID-based filtering, and the behavior to ignore any non-configured VLAN subinterface that falls within the range of the access-group.

Configuring IP packet filtering on VLAN subinterfaces

Use the following procedure to configure IP packet filtering on VLAN subinterfaces.

Procedure steps

1. To configure IP packet filtering on VLAN subinterfaces, enter Configuration Mode.

```
configure terminal
```

2. Create an access list.

```
ip access-list <listname>
```

3. Add a rule to the current filter list.

```
add <rule_action> <protocol> <source> <destination> [sport]
[dport] [icmptype] [icmpcode] [precedence] [tos] [flags]
[log] [expire]
```

4. Exit back a level.

```
exit
```

5. Attach to a subinterface.

```
ip access-group <VLAN subinterface> <listname> <direction>
```

Table 23: Variable definitions

Variable	Value
<destination>	IP destination address (a.b.c.d/a.b.c.d or a.b.c.d/0-32 or any).
<direction>	The direction of packets to filter, in or out.
<listname>	The ACL name.
<procotol>	The protocol, tcp/udp/icmp/ip or 0-255.
<rule_action>	permit rule or deny rule or reject rule (reject rule can be specified only with ICMP protocol).
<source>	IP source address (a.b.c.d/a.b.c.d or a.b.c.d/0-32 or any).
<VLAN subinterface>	The subinterface to apply the ACL.

Inserting a new rule to an already configured access list

Procedure steps

1. Enter Configuration Mode.

```
configuration terminal
```

2. Select the access-list.

```
ip access-list <listname>
```

3. Insert the rule at a specific line number in the access-list.

```
insert <rule_lineno> <rule_action> <protocol> <source>
<destination> [sport] [dport] [icmptype] [icmpcode]
[precedence] [tos] [flags] [log] [expire]
```

Table 24: Variable definitions

Variable	Value
<destination>	IP destination address (a.b.c.d/a.b.c.d or a.b.c.d/0-32 or any).
<listname>	The ACL name.
<procotol>	The protocol, tcp/udp/icmp/ip or 0-255.
<rule_action>	permit rule or deny rule or reject rule (reject rule can be specified only with ICMP protocol).
<rule_lineno>	The line number, in the range 1 to 65535.

Deleting a rule from a rule list

Procedure steps

1. Enter Configuration Mode.

   ```
   configuration terminal
   ```

2. Select the access-list.

   ```
   ip access-list <listname>
   ```

3. Delete the rule.

   ```
   delete <rule_lineno>
   ```

Table 25: Variable definitions

Variable	Value
<listname>	The ACL name.
<rule_lineno>	The line number, in the range 1 to 65535.

Displaying access lists, rule sets and statistics

Procedure steps

1. Display all access lists.

   ```
   show ip access-lists all
   ```

2. Display access list rules.

   ```
   show ip access-list-rules <all | [VLAN subinterface]>
   ```

3. Display access list statistics.

   ```
   show ip access-list-stats <VLAN subinterface>
   ```

Table 26: Variable definitions

Variable	Value
<VLAN subinterface>	A single subinterface name or a range of subinterfaces (specified as ethernet0.1-5 which implies range of subinterfaces starting from ethernet0.1 till ethernet0.5.)

Chapter 21: Multilink Frame Relay Configuration

Layer Two Configurations

Figure 1 outlines a Multilink Frame Relay (MFR) configuration with three sites. PVC 16 connects Site 1 to Site 3, while PVC 31 connects Site 2 to Site 3. The Frame Relay switching equipment is represented simply as a Frame cloud.

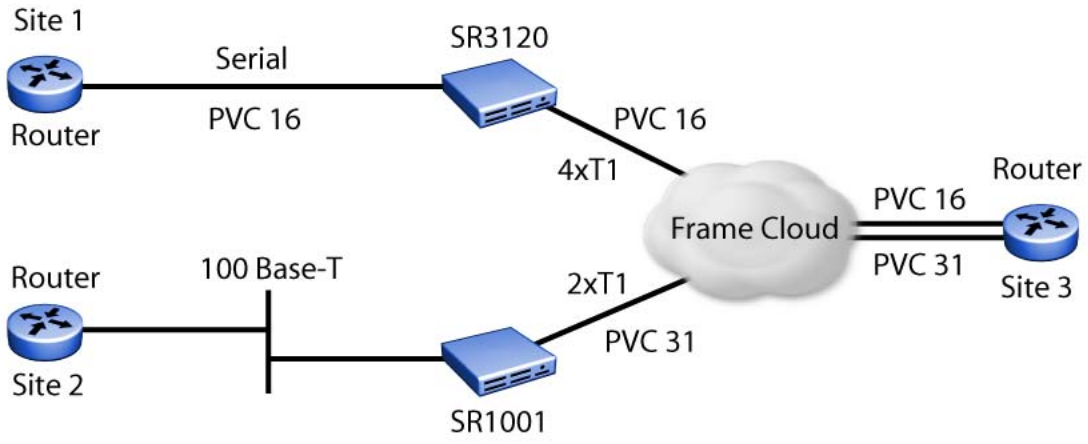


Figure 14: Multilink Frame Relay with Three Sites

Figure 17 provides greater detail, including the use of an SR3120 inside the cloud as a Frame Relay switching device, and SR3120 and Secure Router 1000 Series units at the CPE sites 1 and 2.

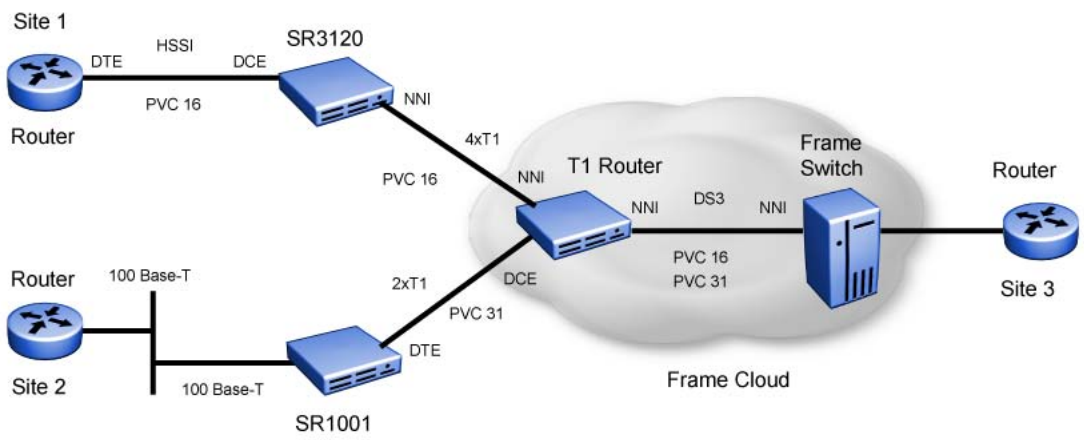


Figure 15: Multilink Frame Relay with Three Site Details

MFR Configuration

The 4 x T1 MFR bundle between the Secure Routers connects two Frame Relay switches, therefore it represents an NNI interface. The sample configuration defines the 4 x T1 bundle to be of Class C; that is, a minimum of 2 T1 links are required to be up in order to keep the bundle up. Settings for Bcmin on the MFR bundle are set to correspond with the Class C configuration; that is, the minimum anticipated bandwidth will be 2 x T1.

Configure the Secure Router 1004 Series at Site 1

```
SR/configure> int bundle wan1
SR/configure/interface/bundle> link t1 5-8
SR/configure/interface/bundle> description "6 Mbps MFR"
SR/configure/interface/bundle> encap fr
SR/configure/interface/bundle> fr
SR/configure/interface/bundle/fr> intf_type nni
SR/configure/interface/bundle/fr> mfr class C 2
/* specifies that the bundle remain up as long as two
T1s are up */
SR/configure/interface/bundle/fr> lmi ansi
SR/configure/interface/bundle/fr/lmi> keepalive 8
SR/configure/interface/bundle/fr/lmi> exit
SR/configure/interface/bundle/fr> pvc 16
/* pvc's default cir set to 6144000 bps */
SR/configure/interface/bundle/fr/pvc> shaping cir
6144000 bcmax 6144000 bcmin 3072000
```

```
/* Bcmin consistent with minimum possible bundle
bandwidth of two T1s */
SR/configure/interface/bundle/fr/pvc> switch 16 toRouter
/* switch between wan1:16 and toRouter:16 established
*/
SR/configure/interface/bundle/fr> exit
```

Configure the Secure Router 3120

```
SR/configure> int bundle toSR1004
SR/configure/interface/bundle> link ct3 1 1-4
SR/configure/interface/bundle> description "6Mbps MFR
to SR1004"
SR/configure/interface/bundle> encap fr
SR/configure/interface/bundle> fr
SR/configure/interface/bundle/fr> intf_type nni
SR/configure/interface/bundle/fr> mfr class C 2
SR/configure/interface/bundle/fr> lmi ansi
SR/configure/interface/bundle/fr/lmi> keepalive 10
SR/configure/interface/bundle/fr/lmi> exit
SR/configure/interface/bundle/fr> pvc 16
SR/configure/interface/bundle/fr/pvc> shaping cir
6144000 bcmax 6144000 bcmin 3072000
SR/configure/interface/bundle/fr/pvc> exit
SR/configure/interface/bundle/fr> exit 2
```

A Secure Router 1004 at Site 2 serves as the Frame Relay termination point, connecting the Site 2 IP network to the SR3120. This MFR bundle utilizes 2 T1 links for an approximate 3 Mb/s bandwidth. Since it is the Frame Relay terminating point and is defined as a DTE frame relay interface, an IP address is assigned to the WAN bundle.

Configure the Secure Router 1004 Series at Site 2

```
SR/configure> int bundle to1004
SR/configure/interface/bundle> link t1 1-2
SR/configure/interface/bundle> description "3 Mbps to
1004"
SR/configure/interface/bundle> encap fr
SR/configure/interface/bundle> fr
SR/configure/interface/bundle/fr> intf_type dte /* this
is default */
SR/configure/interface/bundle/fr> mfr class A /* this is
default */
SR/configure/interface/bundle/fr> lmi ansi
SR/configure/interface/bundle/fr/lmi> keepalive 10
SR/configure/interface/bundle/fr/lmi> exit
SR/configure/interface/bundle/fr> pvc 31
/* pvc’s default cir set to 3072000 bps */
SR/configure/interface/bundle/fr/pvc> ip addr 10.0.2.1
255.255.255.252
SR/configure/interface/bundle/fr/pvc> enable
SR/configure/interface/bundle/fr/pvc> exit
```

Configure the SR3120

```
SR3120/configure> int bundle to3120
SR3120/configure/interface/bundle> link ct3 1 5-6
SR3120/configure/interface/bundle> description "3Mbps
MFR to 1001"
SR3120/configure/interface/bundle> encap fr
SR3120/configure/interface/bundle> fr
SR3120/configure/interface/bundle/fr> intf_type dce
SR3120/configure/interface/bundle/fr> lmi ansi
SR3120/configure/interface/bundle/fr/lmi> keepalive 10
SR3120/configure/interface/bundle/fr/lmi> exit
SR3120/configure/interface/bundle/fr> pvc 31
SR3120/configure/interface/bundle/fr/pvc> exit 3
```

Chapter 22: Network Address Translation

Network Address Translation (RFC 1631) is commonly known as NAT. This application discusses NAT and provides a technical explanation and configuration examples.

Features:

- Dynamic Address/Port Translation
- Static Address/Port Translation
- Forward and Reverse NAT
- Non-Translated Address Pass Through

In the most common NAT application, the device (Secure Router) that connects the user LAN to the Internet will have two IP addresses:

- A private IP address on the LAN side for the RFC 1918 address range
- A public address, able to be routed over the Internet, on the WAN side

Consider a PC on the LAN sending a packet destined for some.server.com. The source IP address and port are in the packet together with the destination IP address and port. When the packet arrives at the Secure Router it will be de-encapsulated, modified, and re-encapsulated. The re-encapsulated packet sent by the Secure Router destined for the Internet contains the Secure Router's public IP address, a source port allocated from its list of available ports, and the same destination IP address and port number generated by the PC. The Secure Router also adds an entry into a table it keeps, which maps the internal address and source port number that the PC generated against the port number it allocated to this session. Therefore, when some.server.com sends a reply packet to the PC, the Secure Router can quickly determine how it needs to rewrite the packet before transmitting it back on to the LAN.

Dynamic NAT

Dynamic NAT is used when packets destined for the Internet are transported from a LAN using the public source IP address assigned to the local router. Dynamic NAT performs this task well, but it does not permit providing services to the Internet from inside a LAN. In these instances, static NAT is used.

Static NAT

Static NAT also requires a public address from the upstream service provider. Individual PCs within a LAN are assigned RFC 1918 reserved IP addresses to enable access to other PCs

within the LAN. The Secure Router is configured with static mapping, which maps the internal RFC 1918 IP addresses for each PC to the appropriate public IP address. Then when traffic is sent to the public address listed in the static mapping, the Secure Router forwards the packets to the correct PC within the LAN, according to the mapping relationship established.

Figure 16: Dynamic and Static NAT on page 108 illustrates dynamic and static NAT. The static translation between 192.168.1.6 and 100.1.1.6 automatically matches the port addresses, thus a request destined for 100.1.1.6 tcp port 25 is translated to 192.168.1.6 tcp port 25 and so on.

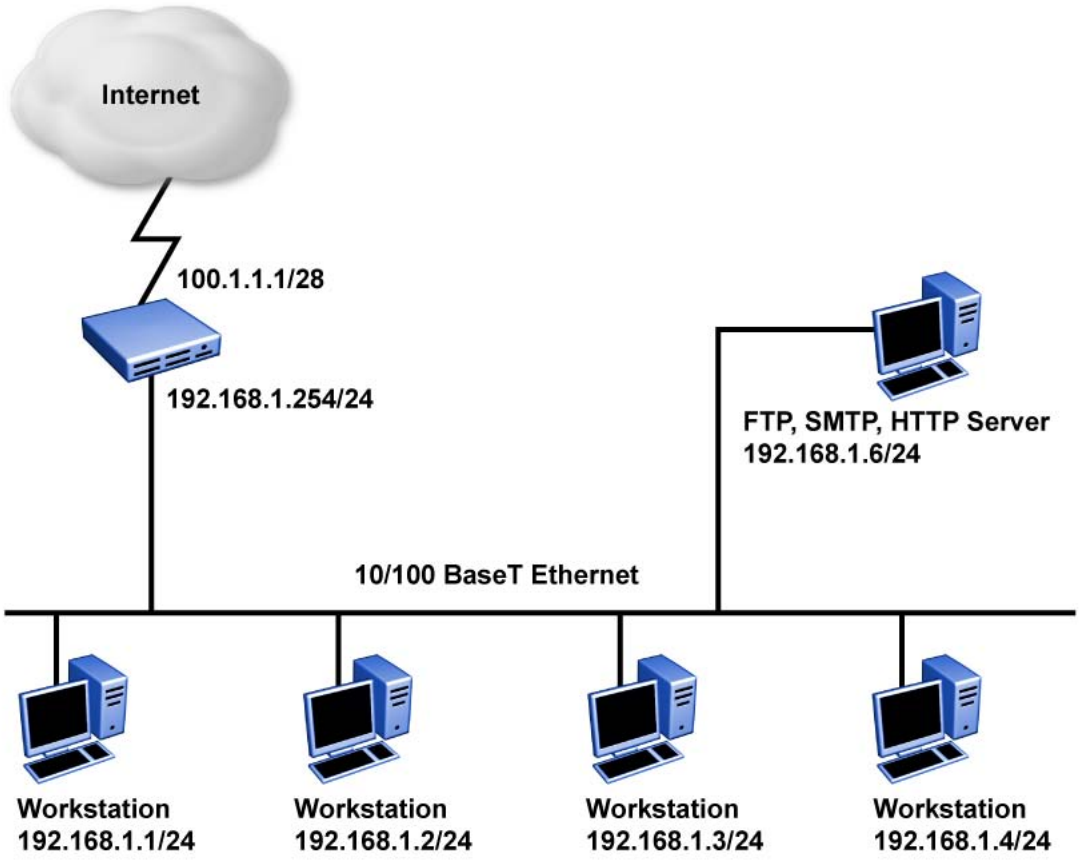


Figure 16: Dynamic and Static NAT

Configuration for Dynamic and Static NAT

```
SR> configure terminal
SR/configure> interface bundle Trenton
SR/configure/interface/bundle Trenton> nat
SR/configure/interface/bundle Trenton/nat> enable
dynamic
```

```
SR/configure/interface/bundle Trenton/nat> enable static
SR/configure/interface/bundle Trenton/nat> address
192.168.1.6 100.1.1.6
```

Figure 17: Mapping Ports on page 109 provides an example of static port mapping. TCP port 81 of the web server at private address 192.168.1.6 is mapped to the same TCP port of the public address.

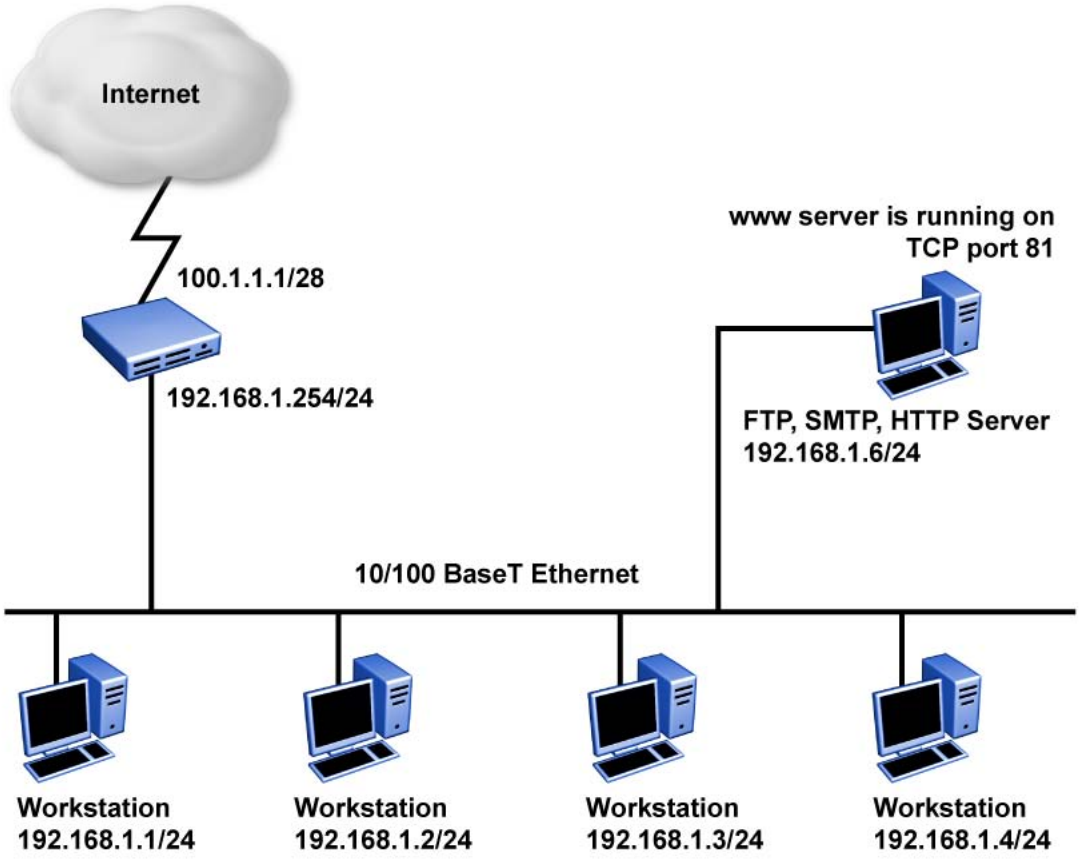


Figure 17: Mapping Ports

Configuration for Mapping Ports

```
SR> configure terminal
SR/configure> interface bundle Trenton
SR/configure/interface/bundle Trenton> nat
SR/configure/interface/bundle Trenton/nat> enable
dynamic
SR/configure/interface/bundle Trenton/nat> enable static
```

```
SR/configure/interface/bundle Trenton/nat> port tcp
192.168.1.6 81 100.1.1.6 81
```

Reverse NAT

Reverse NAT could be used in a situation where one LAN is using private RFC 1918 IP addresses and a second LAN is using real Internet IP addresses. Figure 18: Reverse NAT on page 110 illustrates how reverse NAT would be applied.

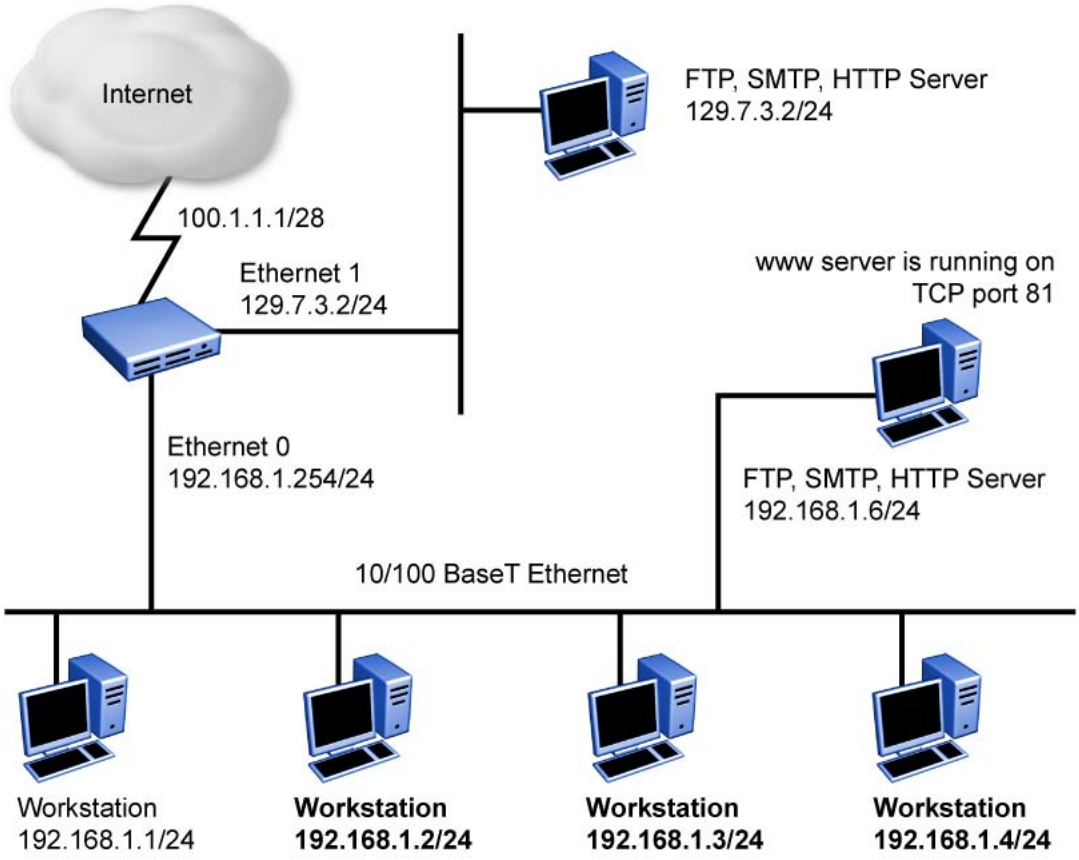


Figure 18: Reverse NAT

Configuration for Reverse NAT

```
SR> configure terminal
SR/configure> interface ethernet 0
```

```
SR/configure/interface/ethernet0> nat
SR/configure/interface/ethernet0/nat> reverse
SR/configure/interface/ethernet0/nat> ip 100.1.1.1
SR/configure/interface/ethernet0/nat> enable dynamic
SR/configure/interface/ethernet0/nat> enable static
SR/configure/interface/ethernet0/nat> port tcp 100.1.1.6
25 192.168.1.6 25
SR/configure/interface/ethernet0/nat> port tcp 100.1.1.6
81 192.168.1.6 81
SR/configure/interface/ethernet0/nat> port tcp 100.1.1.6
21 192.168.1.6 21
```

NAT-Failover for firewalls

This feature enables failover from a primary interface (T1 WAN bundle) to a backup interface (PPPoE or ISDN) when using firewall based Port Address Translation. This feature applies to firewall NAT policies which are configured with the interface name of the primary interface. The user must specify the primary and backup interface using the firewall global **`nat-failover`** command.

When the primary interface is up, packets going out through it will be translated using the IP address of the primary interface. When it goes down, the IP address of the backup interface will be used and the stale firewall connections will be flushed. Without this feature, NAT translations will continue to use the IP address of the primary interface since firewall policies do not change when an interface goes up or down. Hence traffic will be lost.

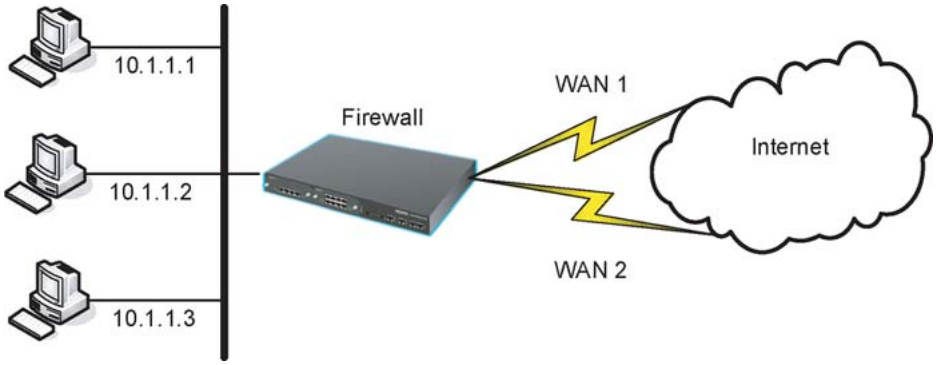


Figure 19: NAT Failover for Firewalls

Configuration for NAT Failover for Firewalls

Traffic from corporation to Internet should go through primary interface WAN1 (with PAT address 50.1.1.5). If the primary goes down, the traffic should go through backup interface WAN2 (with PAT address 60.1.1.5). The PAT address will be the interface address through which the traffic goes to Internet. PAT allows multiple hosts to share the same IP address.

In this configuration example:

- Private IP address:10.1.1.1-10.1.1.3
- PAT IP address: primary address is WAN1 IP address and backup is WAN2 IP address
- Add an outbound policy with the source IP address range and specify the NAT-IP address in the policy command (the source address can be wild carded to any).

```
Router/configure> firewall corp
policy 2 out address 10.1.1.1 10.1.1.3 any nat-ip wan1
exit 2
firewall global
nat-failover wan1 wan2
exit
Router/configure >
```

Add two routes to Internet using WAN1 (primary PAT interface) and WAN2 (secondary PAT interface) as the gateways (Ensure that the route to Internet through backup interface has a higher metric.

```
Router/configure >
ip route 0.0.0.0 0 wan1 metric 1
ip route 0.0.0.0 0 wan2 metric 2
Router/configure >
```

Chapter 23: NAT Configurations

Network Address Translation (NAT) was defined to serve two purposes:

- Allowed LAN administrators to create secure, private IP networks, unable to be routed over the Internet, behind firewalls
- Stretched the number of available IP addresses by allowing LANs to use one public (real) IP address as the gateway with a very large pool of NAT addresses behind it.

In the most common NAT application (which is to provide secure networking behind a firewall), the device (Secure Router) that connects the user LAN to the Internet will have two IP addresses:

- A private IP address on the LAN side for the RFC 1918 address range
- A public address, able to be routed over the Internet, on the WAN side

Consider a PC on the LAN sending a packet destined for some.server.com. The source IP address and port are in the packet together with the destination IP address and port. When the packet arrives at the Secure Router it will be de-encapsulated, modified, and re-encapsulated. The re-encapsulated packet sent by the Secure Router destined for the Internet contains the Secure Router's public IP address, a source port allocated from its list of available ports, and the same destination IP address and port number generated by the PC. The Secure Router also adds an entry into a table it keeps, which maps the internal address and source port number that the PC generated against the port number it allocated to this session. Therefore, when some.server.com sends a reply packet to the PC, the Secure Router can quickly determine how it needs to rewrite the packet before transmitting it back on to the LAN.

Dynamic NAT is used when packets destined for the Internet are transported from a LAN using the public source IP address assigned to the local router. Dynamic NAT performs this task well, but it does not permit providing services to the Internet from inside a LAN which requires the use of static NAT. Static NAT also requires a public address from the upstream service provider. Individual PCs within a LAN are assigned RFC 1918 reserved IP addresses to enable access to other PCs within the LAN. The Secure Router is configured with static mapping, which maps the internal RFC 1918 IP addresses for each PC to the appropriate public IP address. When traffic is sent to the public address listed in the static mapping, the Secure Router forwards the packets to the correct PC within the LAN, according to the mapping relationship established.

NAT Configuration Examples

Dynamic NAT (many to many)

In dynamic (many-to-many) NAT type, multiple source IP addresses in the corporate network will be mapped to multiple NAT IP addresses (not necessarily of equal number). For a set of local IP address from 10.1.1.1 to 10.1.1.4 there will be a set of NAT IP address from 60.1.1.1 to 60.1.1.2. In case of many-to-many NAT, only IP address translation takes place, for example, if a packet travels from 10.1.1.1 to Yahoo.com, Secure Router-Firewall only substitutes the source address in the IP header with one of the NAT IP address and the source port will be the same as the original. If traffic emanates from the same client to any other server, the same NAT IP address is assigned. The advantage is that the NAT IP addresses are utilized in a better and optimum manner dynamically.

If a NAT IP address cannot be allocated dynamically at the connection creation time, the packet would be dropped.

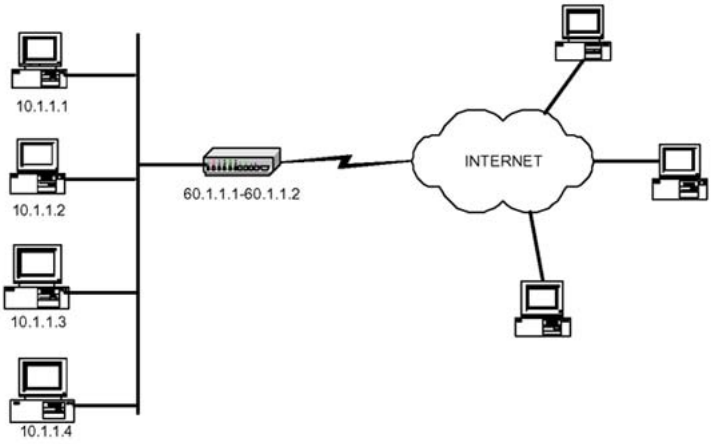


Figure 20: Dynamic NAT

The dynamic NAT configuration shown in Figure 20: Dynamic NAT on page 114 includes:

- Private network addresses: 10.1.1.1—10.1.1.4
- Public (NAT) IP address range: 60.1.1.1—60.1.1.2

To create NAT pool with type dynamic, specify the IP address and the NAT ending IP address. Then add a policy with the source IP address range, and attach the NAT pool to the policy.

```
Router/configure> firewall corp
Router/configure/firewall corp> object
Router/configure/firewall corp/object> nat-pool addresspoolDyna dynamic 60.1.1.1
60.1.1.2
```

```
Router/configure/firewall corp/object> exit
Router/configure/firewall corp> policy 8 out address 10.1.1.1 10.1.1.4 any any
Router/configure/firewall corp/policy 8 out> apply-object nat-pool addresspoolDyna
Router/configure/firewall corp/policy 8 out> exit 2
Router/configure>
```

Static NAT (one to one)

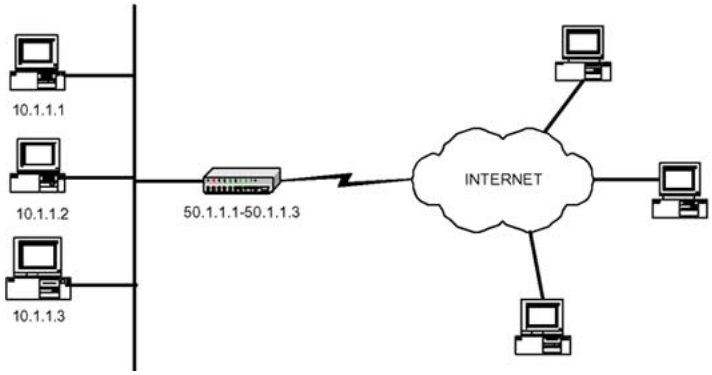


Figure 21: Static NAT

In static (one-to-one) NAT type, for each IP address in the corporate network, one NAT IP address will be used. For example, for the three IP addresses from 10.1.1.1 to 10.1.1.3, there is a set of three NAT IP address from 50.1.1.1 to 50.1.1.3. In case of one-to-one NAT, only IP address translation takes place, that is, if a packet travels from 10.1.1.1 to Yahoo.com, the Secure Router-Firewall only substitutes the source address in the IP header with the NAT IP address. The source port will be the same as the original.

The static NAT configuration shown in Figure 21: Static NAT on page 115 includes:

- Private network address: 10.1.1.1—10.1.1.3
- Public (NAT) IP address range: 50.1.1.1—50.1.1.3

To create NAT pool with type static, specify the IP address and the ending NAT IP address. Add a policy with source IP address range and attach NAT pool to the policy.

```
Router/configure> firewall corp
Router/configure/firewall corp object
Router/configure/firewall corp/object> nat-pool addresspoolStat static 50.1.1.1
50.1.1.3
Router/configure/firewall corp/object> exit
Router/configure/firewall corp> policy 7 out address 10.1.1.1 10.1.1.3 any any
Router/configure/firewall corp/policy 7 out> apply-object nat-pool addresspoolStat
Router/configure/firewall corp/policy 7 out> exit 2
Router/configure>
```

Port Address Translation (many to one)

NAT allows multiple IP addresses to be mapped to one address.

There are two methods to configure Port Address Translation (PAT) on the Secure Router gateway. In the first method, specify the IP address to the nat-ip parameter in the **policy** command. In the second method, create a pool of type PAT and then attach it to the policy.

In PAT, multiple hosts can share the same IP address.

The PAT configuration includes:

- Private network address: 10.1.1.1—10.1.1.3
- PAT address: 50.1.1.5

Method 1: Specifying NAT address with the policy command

To configure this method of PAT, add the policy with the source IP address range, then specify the nat-ip address in the **policy** command.

```
Router/configure> firewall corp

Router/configure/firewall corp> policy 2 out address 10.1.1.1 10.1.1.3 any any nat-
ip 50.1.1.5

Router/configure/firewall corp/policy 2 out> exit 2

Router/configure>
```

Method 2: Attaching NAT pool to the policy

To configure the second type of NAT, create a NAT pool with type pat and specify the IP address. Then add the policy with the source IP address range. Finally, attach the NAT pool to the policy.

```
Router/configure> firewall corp

Router/configure/firewall corp> object

Router/configure/firewall corp/object> nat-pool addresspoolPat pat 50.1.1.5

Router/configure/firewall corp/object> exit

Router/configure/firewall corp> policy 2 out address 10.1.1.1 10.1.1.3 any any

Router/configure/firewall corp/policy 2 out> apply-object nat-pool addresspoolPat

Router/configure/firewall corp/policy 2 out> exit 2

Router/configure>
```

Cone NAT

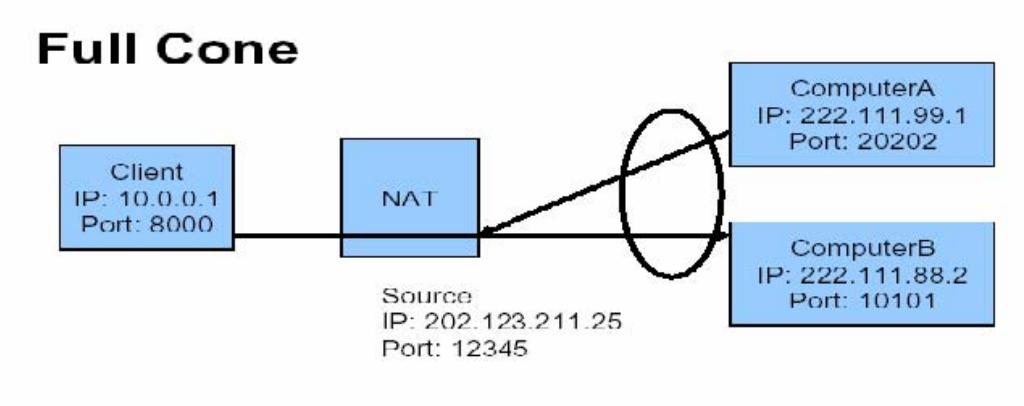
Network Address Translation (NAT) is used to map private address into public addresses through a NAT device. This is accomplished through address mangling and/or port mangling.

Since some protocols, like UNIStim are not standardized they can be difficult to interact with. To deal with this problem the behavior of the NAT device is altered so that it becomes friendly to new sessions. This change allows these protocols to work through NAT devices with no need for an Application Level Gateway (ALG), which can be difficult to maintain.

Cone NAT supports any STUN client/server. No new configuration is required specifically. Old configuration files which defined a NAT translation will now behave in the Cone NAT style. For related Cone NAT show and debug commands, refer to the *Command Reference Guide*.

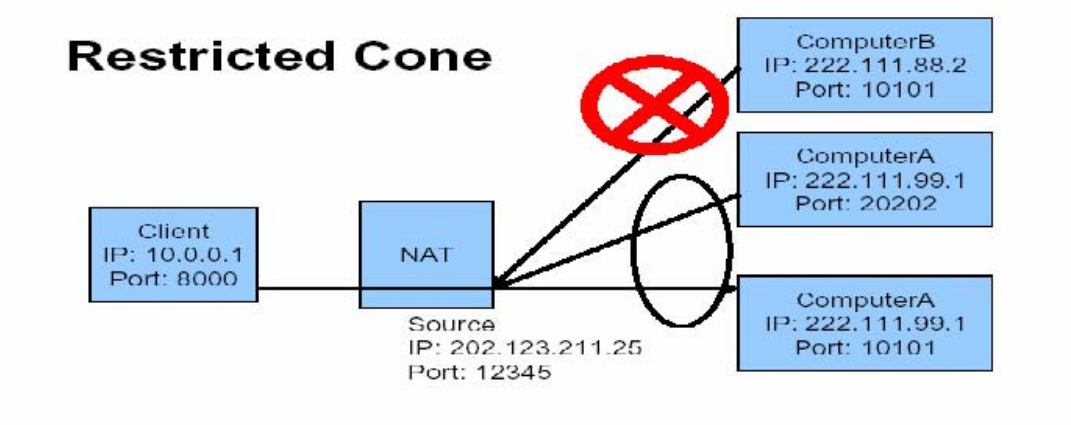
Full Cone

A full cone NAT is one where all requests from the same internal IP address and port are mapped to the same external IP address and port. Furthermore, any external host can send a packet to the internal host, by sending a packet to the mapped external address.



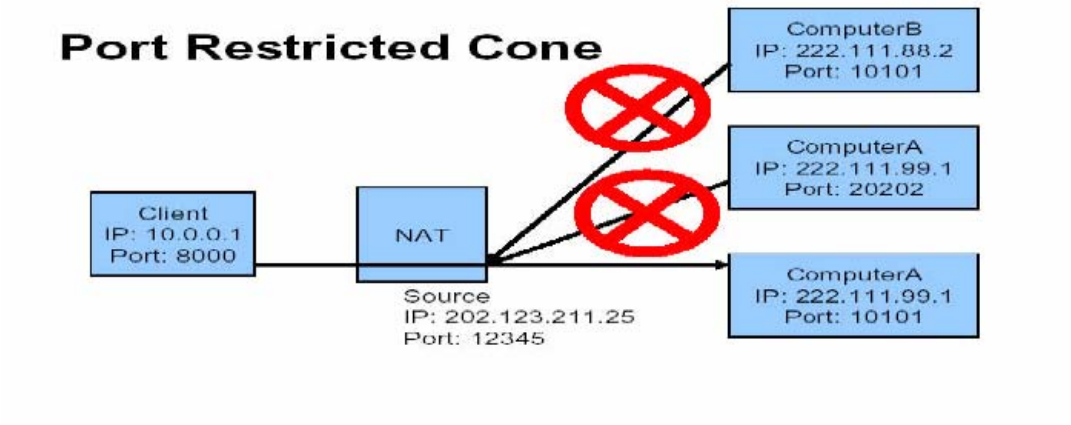
Restricted Cone

A restricted cone NAT is one where all requests from the same internal IP address and port are mapped to the same external IP address and port. Unlike a full cone NAT, an external host (with IP address X) can send a packet to the internal host only if the internal host had previously sent a packet to IP address X.



Port Restricted Cone

A port restricted cone NAT is like a restricted cone NAT, but the restriction includes port numbers. Specifically, an external host can send a packet, with source IP address X and source port P, to the internal host only if the internal host had previously sent a packet to IP address X and port P.



Troubleshooting Cone NAT Common Problems

- Phones do not register

  check network connectivity and policy between the phone and the call server.

- Public phone cannot call private phone

  Ensure the phones and call servers are using STUN or a STUN like echo server for UNIStim.

- One way audio
  - Ensure the phones and call servers are using STUN or a STUN like echo server for UNIStim.
  - Confirm the NAT port assignments using show firewall nat-translations all.
- Phones reboot or reset frequently
  - Ensure the pings between the phones and the echo servers are being allowed to traverse the firewall.
  - Ensure the ping frequency is faster than the firewall UDP connection timeout.

NAT hairpinning

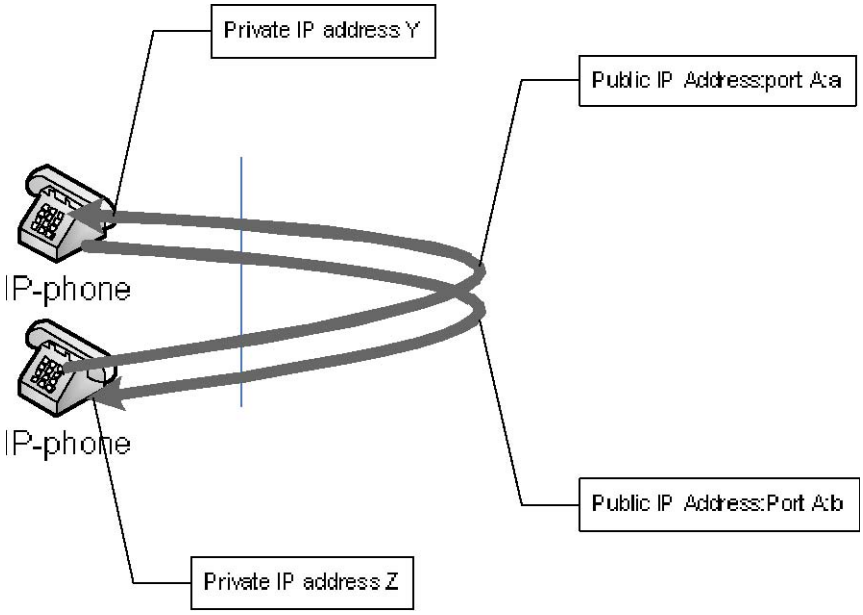
NAT hairpinning could be described as allowing endpoint devices on the internal side of a NAT to communicate when using each other's external addresses and ports. Two scenarios where hairpinning would be needed would be:

- A set of UNISTim, IP phones are deployed behind a NAT and the CS-series call server is located on the public side. Here the CS-series call server will instruct the phones to ‘talk’ to each other on their apparent public IP addresses/ports.
- Any set of RTP based applications which are STUN aware are deployed behind a NAT and a STUN server is publicly reachable. Here the STUN server will instruct the STUN aware applications to ‘talk’ to each other on their apparent public IP addresses/ports.

Hairpinning must be configured through CLI using the following commands:

```
configure/firewall global > hairpinning-SelfIp
```

```
configure/firewall global > no hairpinning-SelfIp
```



Troubleshooting Hairpinning Common Problems

- Phones do not register

  check network connectivity and policy between the phone and the call server

- One way or no audio between private phones

  Turn hairpinning on. (Default is off)

- One way/no audio between private phones and hairpinning is on

  - If using UNIStim: ensure Phones and Call Server are using STUN or echo server
  - If using SIP: endure sip-alg is on (default is on)

- For Isolation, try NAT avoidance

SIP ALG Interoperability with Avaya Call Servers

Ability to Enable/Disable Firewall ALGs

All the firewall ALGs are enabled by default when the firewall is configured. It can become necessary to selectively disable ALGs in the firewall when applications fail due to

incompatibility with the Firewall ALG. When a configuration is saved on the router if a firewall ALG is disabled the disabling of that ALG will be saved. The configuration of the ALG is under the firewall/global/ALG subtree.

The following example show how to disable the SIP ALG in the firewall and how to Display the current enabled firewall ALGs.

```
R1/configure/firewall global/algs > show firewall algs
```

Firewall Algs	Status
aim	Enabled
cuseeme	Enabled
dns	Enabled
ftp	Enabled
gatekeeper	Enabled
h323	Enabled
icq	Enabled
l2tp	Enabled
msgtcp	Enabled
msgudp	Enabled
msn	Enabled
mszone	Enabled
n2p	Enabled
n2pe	Enabled
nntp	Enabled

pcanywhere	Enabled
pptp	Enabled
rpc	Enabled
rtsp554	Enabled
rtsp7070	Enabled
sip	Enabled
smtp	Enabled
sql	Enabled
tftp	Enabled
web	Enabled

```
R1/configure/firewall global/algs > no ftp
```

Firewall FTP Alg disabled

```
R1/configure/firewall global/algs > no sip
```

Firewall SIP Alg disabled

```
R1/configure/firewall global/algs > show firewall algs
```

Firewall Algs	Status
_____	_____
aim	Enabled
cuseeme	Enabled
dns	Enabled
ftp	Disabled

gatekeeper	Enabled
h323	Enabled
icq	Enabled
l2tp	Enabled
msgtcp	Enabled
msgudp	Enabled
msn	Enabled
mszone	Enabled
n2p	Enabled
n2pe	Enabled
nntp	Enabled
pcanywhere	Enabled
pptp	Enabled
rpc	Enabled
rtsp554	Enabled
rtsp7070	Enabled
sip	Disabled
smtp	Enabled
sql	Enabled
tftp	Enabled

web	Enabled

```
R1/configure/firewall global/algs > sip
```

Firewall SIP Alg enabled

```
R1/configure/firewall global/algs > show firewall algs
```

Firewall Algs	Status
aim	Enabled
cuseeme	Enabled
dns	Enabled
ftp	Disabled
gatekeeper	Enabled
h323	Enabled
icq	Enabled
l2tp	Enabled
msgtcp	Enabled
msgudp	Enabled
msn	Enabled
mszone	Enabled
n2p	Enabled
n2pe	Enabled
nntp	Enabled

pcanywhere	Enabled
pptp	Enabled
rpc	Enabled
rtsp554	Enabled
rtsp7070	Enabled
sip	Enabled
smtp	Enabled
sql	Enabled
tftp	Enabled
web	Enabled

NAT ACL enhancements

The Secure Router 1000 Series and 3120 provide support for NAT ACL enhancements. These enhancements add flexibility in configuring a network Access Control List. Access Control Lists are used to filter packets going to the global NAT subsystem. A separate ACL is allowed for static and dynamic address modules. Access Control Lists are applied to both outbound and inbound traffic for translation.

If a packet matches a permit rule, the packet enters that NAT module. If a packet matches a deny rule, it is transmitted without being modified. In the event a packet traverses all NAT ACLs without a rule match, the packet is dropped. One single NAT ACL is allowed in the Global NAT module to control access. The Global NAT ACL may be applied selectively to any interface.

NAT ACL Packet Processing

This section contains information on Packet Translation in a forwarding scenario for both incoming and outgoing packets.

Outgoing Packet Translation

During outgoing packet translation packets sent from a private client to a host on a public network are known as outgoing packets. Nat translation is enabled on the public interface. An ACL is applied if either the inbound interface ACL is enabled on a private interface or if the

outbound interface filter is enabled on a public interface. A check is performed on the outgoing interface for NAT ability prior to the packet being sent out.

If an outgoing packet matches a static translation route the packet is translated and sent. IF ACL filters are configured for Address NAT the following actions are taken:

- Packet is translated if it matches a permit rule
- Packet is forwarded, without being translated if it matches a deny rule
- Packet is forwarded to Address NAT module if no rule is matched.
- In the case of Dynamic Address NAT, if the module is not enabled the packet is dropped.

Incoming Packet Translation

Packets returned to the private client from a host in a public network are known as Incoming Packets. When the packet is received, prior to route lookup, processing of address translation for the incoming packets takes place. All inbound packets are subjected to reverseACL to apply NAT translations; reverseACL enabled by default.

Configuring NAT ACL

Use the following procedure to manually configure a NAT ACL.

Procedure steps

1. To configure NAT ACL, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Enter IP mode.

   ```
   ip
   ```

3. Enter the nat subtree.

   ```
   nat
   ```

4. Create an access list.

   ```
   access-list <listname>
   ```

5. If applicable, specify an address or range to permit.

   ```
   add permit ip <range-start> <range-end>
   ```

6. If applicable, specify an address or range to deny.

   ```
   add deny ip <range-start> <range-end>
   ```

7. Exit the access-list configuration to finish or create another.

   ```
   exit
   ```

8. Create an address pool.

   ```
   pool <poolname>
   ```

9. Specify the address pool range. Note that you can specify more than one range using the same command syntax.

```
range <range-start> <range-end> <mask>
```

10. Configure an access group to use the address pool.

    ```
    access-group <groupname> address-pool <poolname>
    ```

11. If applicable, configure ACL access to a static NAT module.

    ```
    access-group <groupname> static
    ```

Table 27: Variable definitions

Variable	Value
<groupname>	The name given to an access group.
<listname>	The name given to the Access Control List.
<mask>	The subnet mask of a supplied address range.
<poolname>	The identifying name given to an address pool.
<range-end>	The range end address used when configuring an ACL.
<range-start>	The address to add or range-start address used when configuring an ACL.

Firewall behavior with invalid ACKs on TCP connections

In previous releases, when accessing Web pages through the firewall to some SSH servers, the Web browser would freeze occasionally while accessing a secure page. To avoid this issue, the updated default firewall behavior is to send a reset for an invalid ACK to a TCP connection coming back to the firewall .

To set the firewall to ignore invalid ACKs, you can use the **`no reset-invalid-acks`** command under the **`firewall global`** tree.

Use the following procedure to configure the **`reset-invalid-acks`** option.

1. Enter configuration mode:

   ```
   configure terminal
   ```

2. Specify global firewall configuration

   ```
   firewall global
   ```

3. To disable the reset-invalid-acks option, enter:

   ```
   no reset-invalid-acks
   ```

4. To enable the reset-invalid-acks option, enter:

   ```
   reset-invalid-acks
   ```

5. To disable the reset-invalid-acks configuration, enter:

   ```
   show firewall reset-invalid-acks
   ```

Example

The following shows a sample configuration:

```
host/configure/firewall global> reset-invalid-acks
host/configure/firewall global > show firewall reset-invalid-acks
reset-invalid-acks is enabled

host/configure/firewall global > no reset-invalid-acks
host/configure/firewall global > show firewall reset-invalid-acks
reset-invalid-acks is disabled
```

Firewall ALG behavior

This section describes firewall ALG behavior.

Default behavior of firewall ALG

With the Secure Router 1000 Series and 3120, firewall ALGs are disabled by default. To use the typical ALG set, a new cli command (**`enable-typical`**) has been added. This command enables only a specific set of ALGs as follows:

aim, aimudp , ftp , l2tp, msn, pptp, rpc, rtsp554, rtsp7070, smtp, web, ike, tftp Remaining ALGs (sip, sip-tcp, h323, gatekeeper, msnudp, dns, n2p, pcanywhere, sql, msgtcp, irc, n2pe, ils, cuseeme, mszone, ils2, nntp) are in the disabled state.

Configuring a typical ALG set

Use the following procedure to configure a typical ALG set.

Procedure steps

1. Enter Configuration Mode.

   ```
   configure terminal
   ```

2. Navigate to the firewall global sub-tree.

   ```
   firewall global
   ```

3. Disable all ALGs.

   ```
   no enable-all
   ```

4. Enable the typical ALG set.

   ```
   enable-typical
   ```

Changes to the DNS ALG

The Secure Router 1000 Series and 3120 provide support for DNS ALG. The DNS ALG is used when a DNS client on an untrusted side wants to access a DNS server behind a NAT in trusted side.

A DNS client in the untrusted side sends a DNS Standard Query to the Secure Router. The Secure Router receives the DNS query with the destination port 53. The secure router translates the IP header based on the reverse NAT policy. When the response comes from the DNS server (which is present in trusted side), the Secure Router translates the header based on the reverse NAT policy and the DNS payload is translated from private IP record to global IP record which will be taken from the DNS pool database.

A DNS client in the untrusted side sends a DNS Reverse Query to the Secure Router. The secure router translates the IP header based on the reverse NAT policy and the DNS payload is translated from global IP record to private IP record which were added through the CLI. When the response comes from the DNS server (which is present in trusted side), the secure router translates header based on the reverse NAT policy and the DNS payload is translated from private IP record to global IP record which will be taken from the DNS pool database.

Configuring DNS ALG

Procedure steps

1. Enter Configuration Mode.

   ```
   configure terminal
   ```

2. Enter the firewall global sub-tree.

   ```
   firewall global
   ```

3. Enter the algs sub-tree.

   ```
   algs
   ```

4. Enter the dns sub-tree.

   ```
   dns
   ```

5. Enable the DNS ALG.

   ```
   enable
   ```

6. Ensure the DNS pool has been configured.

   ```
   pool <pool-name> <private-ip> <global-ip>
   ```

7. Display the pool name.

   ```
   show firewall dns-alg translate-pool pool-name
   ```

8. Display all static pool names which were added.

   ```
   show firewall dns-alg translate-pool
   ```

Table 28: Variable definitions

Variable	Value
<global-ip>	The global IP address for the pool.
<pool-name>	The identifying name for the pool.
<private-ip>	The private IP address for the pool.

The following table shows the settings for the firewall ALGs if the **`enable-all`** command is used.

Firewall ALG Name	Protocol and Port Number	Default Factory Setting	Notes
sip Session initiation protocol	UDP Port 5060	Enabled	
sip-tcp Session initiation protocol	TCP Port 5060	Enabled	
msn Microsoft Networs Messenger	TCP Port 1863	Enabled	Works with MSN client to version 7.0
gatekeeper Micosoft NetMeeting H323-Gatekeep(server to server)	UDP Port 1719	Disabled	Unusual use case for H323 trunkingthroug h NAT
msgudp Microsoft Gaming Zone	UDP Port 47624	Disabled	
tftp Trivial file transfer protocol	UDP Port 69	Enabled	
rpc Remote Procedure call	UDP Port 111	Enabled	
dns Domain Name Service	UDP Port 53	Disabled	Unusual use case – DNS Server on the private side missing needed CLI configuration
n2p Net2Phone private protocol	UDP Port 6801	Disabled	Old version, n2p protocol mostly replaced by SIP even in n2p clients
pcanywhere Norton/ Symantec's pcanywhere protocol	UDP Port 5632	Disabled	Rare use case version 5.0.0

Firewall ALG Name	Protocol and Port Number	Default Factory Setting	Notes
l2tp Layer 2 Tunneling protocol	UDP Port 1701	Enabled	
sql Structured Query Language Oracle's port	UDP Port 1521	Disabled	Port not really registered wth IANA,rare use case
rtsp554 Real Time Streaming Protocol	UDP Port 554	Enabled	
Rtsp7070 Real Time Streaming Protocol Apple's quicktime port	UDP Port 7070	Enabled	
h323 H323 Protocol	UDP Port 1720	Enabled	
irc Internet Relay Chat	UDP Port 6667	Disabled	
aim AOL Instant Messager	TCP Port 5190	Disabled	Only compatible with older versions that do not encrypt
pptp point to point tunneling protocol (management session)	TCP Port 1723	Enabled	
ftp File Transfer Protocol	TCP Port 21	Enabled	
web Hyper Text Protocol	UDP Port 80	Enabled	
smtp Simple Mail Transfer Protocol	TCP Port 25	Enabled	
n2pe Net2Phone Private Protocol	TCP Port 81	Disabled	
ils Micosoft NetMeeting over LDAP Internet Location Server	TCP Port 389	Disabled	
cuseeme CU-SeeMe	TCP Port 7648	Disabled	Rare use case
mszone Microsoft Gaming Zone	TCP Port 28801	Disabled	
nntp Network New Transfer Protocol	TCP Port 119	Disabled	Proxy transport system may not be reliable or stable
netbios	TCP Port 139	Disabled	Rare use case

Firewall ALG Name	Protocol and Port Number	Default Factory Setting	Notes
aimudp AOL Instant Messager	UDP Port 5190	Enabled	
ike Internet Key Exchange Protocol	UDP Port 500	Disabled	
ils2 Microsoft NetMeeting over LDAP Internet Location Server	TCP Port 1002	Disabled	

Chapter 24: IPSec EXAMPLES

Introduction to Security

This release supports a wide range of robust industry-standard security features including:

- Virtual Private Networking
- IPSec encryption and tunneling
- Generic Routing Encapsulation
- Firewall with private network management (Network Address Translation and Port Address Translation)

This chapter explains each of these features in detail.

Enabling Security Features

Licenses control access to:

- Basic VPN Management (**`vpn_mgmt`**) — allows users to manage a remote Secure Router.
- Advanced VPN (**`advance_vpn`**) — allows users to manage remote LANs.

Note:
The VPN Management license is enabled by default. However, this is not backwards compatible with earlier 8.x releases where the VPN Management license is disabled by default.

To see the licenses available in this release, enter:

```
SR-1002/configure > system licenses ?

NAME
  licenses - Configure feature upgrade licenses

SYNTAX
  licenses license_type <cr>

DESCRIPTION
  license_type         -- Specifies the type of feature upgrade license
    The parameter may have any of the following values:
        enable_1_port -- Enable  1 port
        enable_2_ports-- Enable  2 ports
        BGP4          -- BGP4 routing
        vpn_mgmt      -- Enable VPN Mgmt License
        advance_vpn   -- Enable Advance VPN License
```

To install the advanced VPN license and use all the security features available in this release, enter:

```
/configure> system licenses advance_vpn
Enter Security Upgrade License key: 024f3bc296b4ea7265
```

Securing Remote Access Using IPSec VPN

The IPSec VPN features allow administrators to form a security tunnel to join two private networks over the Internet. The following examples show how to set up an end-to-end tunnel with a single proposal and pre-shared key authentication, with multiple proposals and pre-shared key authentication, and with an SA Bundle, and pre-shared key authentication.

The corporate network no longer has a clearly defined perimeter inside secure building and locked equipment closets. Increasingly, companies have a need to provide remote access to their corporate resources for the employees on the move.

Traditionally, remote users could access the corporate LAN through dial-up and ISDN lines which were terminated in the corporate remote access servers. However, these point-to-point connection technologies do not scale well to the growing number of remote users and the corresponding increase in the infrastructure investments and maintenance costs.

A solution to meeting the needs of increasing numbers of remote users and for controlling access costs is to provide remote access through the Internet using firewalls and a Virtual Private Network (VPN). Internet Protocol Security (IPSec) keeps the connection safe from unauthorized users.

In a typical IPSec remote access scenario, the mobile user has connectivity to Internet and an IPSec VPN client loaded on their PC. The remote user connects to the Internet through their

Internet service provider and then initiates a VPN connection to the IPSec security gateway (the VPN server) of the corporate office, which is typically an always-on Internet connection.

One of the main limitations in providing remote access is the typical remote user connects with a dynamically assigned IP address provided by the ISP. IPSec uses the IP address of users as an index to apply the Internet Key Exchange (IKE) and IPSec policies to be used for negotiation with each peer. When the VPN client has a dynamic IP address, the VPN server cannot access the policies based on the IP address of the client. Instead, the VPN server uses the identity of the VPN client to access the policies.

Access Methods

Avaya supports two types of IPSec remote access using VPNs.

Remote Access: User Group

One of the methods to achieve IPSec remote access in Avaya is the user group method. In this method, the administrator creates an IKE policy for a logical group of users such as a department in an organization. Each user in the group is identified with unique information that is uniquely configured in the IKE policy. Also, an IPSec template is attached to the user group.

Once the VPN user is authenticated using IKE, the users dynamically-assigned IP address is added to the destination address field in the IPSec template attached to the user group. The VPN user now has the required IPSec policy that allows access through the gateway to the corporate LAN.

Remote Access: Mode Configuration

The other method to achieve IPSec remote access in Avaya is the mode configuration method.

This method makes the VPN client an extension of the LAN being accessed by the VPN client. The remote client appears as a network accessing some resource behind the VPN server.

The VPN client is allocated a private IP address by the VPN server and the client uses this as the source IP address in the inner IP header in tunnel mode.

In tunnel mode, at each IKE end point, the IP traffic to be protected is completely encapsulated with another IP packet. In this, the inner IP header remains the same as seen in the original traffic to be protected. In the outer IP header, the source and destination addresses are the addresses of the tunnel end points.

Typically, for a remote user, the source address of the outer IP header is the dynamic public IP address provided by the ISP. When mode configuration is enabled, the source address of the inner IP header is the private address allocated by the VPN server to the VPN client.

As in the case of user group method, the administrator creates an IKE policy for a logical group of users such as a department in an organization. The identity information used to identify each user uniquely is configured in the IKE policy. The IKE policy is attached to a mode configuration record. The mode configuration record contains an IPSec policy template to be used for creating dynamic IPSec policy. Also, the record contains one or more pools of private IP addresses to be used for allocating the addresses to the VPN clients. Besides the private IP address, the VPN server can also provide WINS and DNS server addresses.

Upon successful IKE authentication of a VPN client, the server checks whether the IKE policy used to authenticate the VPN client is enabled for mode configuration. If so, the server allocates a private IP address from one of the IP pools in the mode configuration record to the VPN client. The destination address field in the IPSec template attached to the user group is filled in with the private IP address allocated to the VPN client and this is installed as an IPSec policy.

This guide provides information and examples on how to configure IPSec.

Installing Licenses

Licenses control access to:

Example

- Basic VPN Management (**`vpn_mgmt`**) — allows users to manage a remote Secure Router.
- Advanced VPN (**`advance_vpn`**) — allows users to manage remote LANs.

To see the licenses available in this release, enter:

```
SR-1002/configure > system licenses ?

NAME
  licenses - Configure feature upgrade licenses

SYNTAX
  licenses license_type <cr>

DESCRIPTION
  license_type         -- Specifies the type of feature upgrade license
    The parameter may have any of the following values:
        enable_1_port -- Enable  1 port
        enable_2_ports-- Enable  2 ports
        BGP4          -- BGP4 routing
        vpn_mgmt      -- Enable VPN Mgmt License
        advance_vpn   -- Enable Advance VPN License
```

To install the advanced VPN license and use all the security features available in this release, enter:

```
/configure> system licenses advance_vpn
Enter Security Upgrade License key: 024f3bc296b4ea7265
```

Example 1: Securely Managing the Secure Router 1000 Series Over an IPSec Tunnel

This example demonstrates how to manage a router through an IP security tunnel. Steps are presented for configuring the Networks1 and Networks2 routers to assist any host on the LAN side of Networks2 to manage the Networks1 router through the IP security tunnel.

The security requirements are:

Example

- Phase 1: 3DES with SHA1
- Phase 2: IPSec ESP with 128-bit AES and HMAC-SHA1

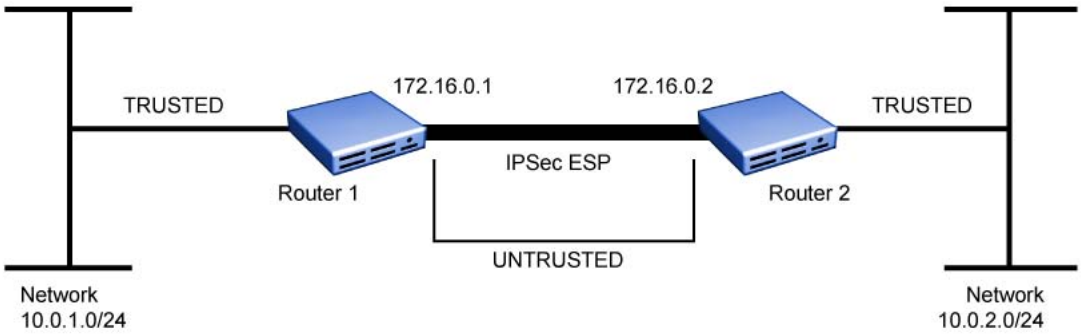


Figure 22: Tunnel Mode Between Two Security Gateways - Single Proposal

Step 1: Configure a WAN bundle of network type untrusted

```
Networks1/configure> interface bundle wan1
message: Configuring new bundle
Networks1/configure/interface/bundle wan1> link t1 1
Networks1/configure/interface/bundle wan1> encapsulation ppp
Networks1/configure/interface/bundle wan1> ip address 172.16.0.1 24
Networks1/configure/interface/bundle wan1> crypto untrusted
Networks1/configure/interface/bundle wan1> exit
```

Step 2: Configure the Ethernet interface with trusted network type

```
Networks1/configure> interface ethernet 0
message: Configuring existing Ethernet interface
Networks1/configure interface/ethernet 0> ip address 10.0.1.1 24
Networks1/configure/interface/ethernet 0> crypto trusted
Networks1/configure/interface/ethernet 0> exit
```

Step 3: Display the crypto interfaces

```
Networks1> show crypto interfaces
```

```
Interface      Network
Name           Type
---------      -------
ethernet0      trusted
wan1           untrusted
```

Step 4: Add the route to the peer LAN

```
Networks1/configure> ip route 10.0.2.0 24 wan1
```

Step 5: Configure IKE to the peer gateway

```
Networks1/configure> crypto
Networks1/configure/crypto> ike policy Networks2 172.16.0.2
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> local-
address 172.16.0.1
message: Default proposal created with priority1-des-sha1-pr_shared-g1
message: Key String has to be configured by the user
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> key secretkey
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> proposal 1
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2/proposal 1>
encryption-algorithm 3des-cbc
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2/proposal 1>
exit
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> exit
```

Step 6: Display the IKE policies

```
Networks1> show crypto ike policy all
```

Step 7: Display the IKE policies in detail

```
Networks1> show crypto ike policy all detail
```

Step 8: Configure the IPSec tunnel to the remote host

```
Networks1/configure/crypto> ipsec policy
Networks2 172.16.0.2
Networks1/configure/crypto/ipsec/policy Networks2
172.16.0.2> match address 10.0.1.0 24 10.0.2.0 24
message: Default proposal created with
priority1-esp-3des-sha1-tunnel and activated.
Networks1/configure/crypto/ipsec/policy Networks2
172.16.0.2> proposal 1
Networks1/configure/crypto/ipsec/policy Networks2
172.16.0.2/proposal 1> encryption-algorithm des-cbc
Networks1/configure/crypto/ipsec/policy Networks2
172.16.0.2/proposal 1> exit
Networks1/configure/crypto/ipsec/policy Networks2
172.16.0.2> proposal 2
message: Proposal added with
priority2-esp-3des-sha1-tunnel.
Networks1/configure/crypto/ipsec/policy Networks2
172.16.0.2/proposal 2> encryption-algorithm aes256-cbc
Networks1/configure/crypto/ipsec/policy Networks2
172.16.0.2/proposal 2> exit
Networks1/configure/crypto/ipsec/policy Networks2
172.16.0.2> exit
Networks1/configure/crypto> exit
Networks1/configure>
```

Note:

For IPSec only – when you create an outbound tunnel, an inbound tunnel is automatically created. The inbound tunnel applies the name that you provide for the outbound tunnel and adds the prefix IN to the name.

Step 9: Display the IPSec policies

```
Networks1> show crypto ipsec policy all
```

Step 10: Display IPSec policies in detail

```
Networks1> show crypto ipsec policy all detail
```

Step 11: Configure firewall policies to allow IKE negotiation through untrusted interface

```
Networks1/configure> firewall internet
Networks1/configure/firewall internet> policy 1000 in service ike self
Networks1/configure/firewall internet/policy 1000 in> exit
Networks1/configure/firewall internet> exit
```

Step 12: Configure firewall policies to allow desired services through untrusted interface to manage the router

```
Networks1/configure> firewall internet
Networks1/configure/firewall internet> policy 1001 in service snmp self
Networks1/configure/firewall internet/policy 1001 in> exit
Networks1/configure/firewall internet> policy 1002 in service telnet self
Networks1/configure/firewall internet/policy 1002 in> exit
Networks1/configure/firewall internet> policy 1003 in protocol icmp self
Networks1/configure/firewall internet/policy 1003 in> exit
Networks1/configure/firewall internet> exit
```

Step 13: Display firewall policies in the Internet map

```
Networks1> show firewall policy internet
```

Step 14: Display firewall policies in the Internet map in detail

```
Networks1> show firewall policy internet detail
```

Step 15: Enable SNMP on the Networks1 router

```
Networks1/configure/crypto/> exit
Networks1/configure> snmp
Networks1/configure/snmp> community public rw
Networks1/configure/snmp> exit
```

Step 16: Display SNMP communities

```
Networks1> show snmp communities
Community = public, privilege=rw
```

Step 17: Repeat steps 1 - 16 with suitable modifications on Networks2 prior to managing Networks1 from the Networks2 LAN side

Step 18: Test the IPSec tunnel for managing the Networks1 router from a host on the Networks2 LAN.

Step 19: When the SNMP manager starts managing Networks1 from the Networks2 LAN, display the IKE and IPSec SA tables

```
Networks1> show crypto ike sa all
```

```
Networks1> show crypto ike sa all detail
```

```
Networks1> show crypto ipsec sa all
```

```
Networks1> show crypto ipsec sa all detail
```

Example 2: Joining Two Private Networks with an IP Security Tunnel

The following example demonstrates how to form an IP security tunnel to join two private networks: 10.0.1.0/24 and 10.0.2.0/24. The security requirements are as follows:

Example

- Phase 1: 3DES with SHA1
- Phase 2: IPSec ESP with AES (256-bit) and HMAC-SHA1

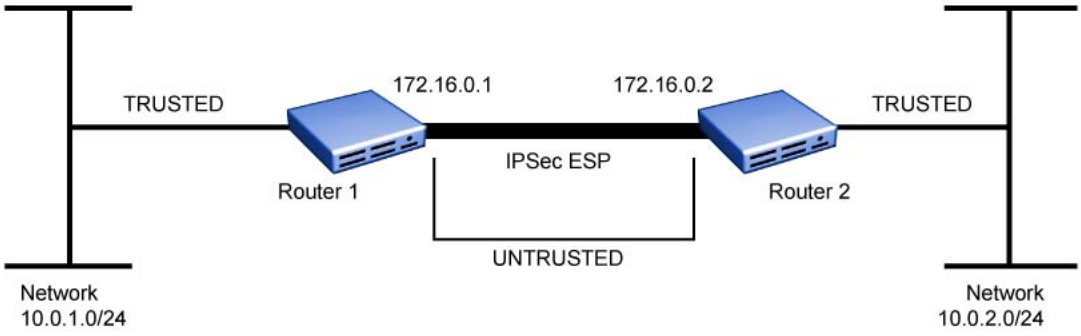


Figure 23: Tunnel Mode Between Two Security Gateways - Single Proposals

Step 1: Configure a WAN bundle of network type untrusted

```
Networks1/configure/interface/bundle wan1> link t1 1
Networks1/configure/interface/bundle wan1> encapsulation ppp
Networks1/configure/interface/bundle wan1> ip address 172.16.0.1 24
Networks1/configure/interface/bundle wan1> crypto untrusted
Networks1/configure/interface/bundle wan1> exit
```

Step 2: Configure the Ethernet interface with trusted network type

```
Networks1/configure> interface ethernet 0
message: Configuring existing Ethernet interface
Networks1/configure interface/ethernet 0> ip address
10.0.1.1 24
Networks1/configure/interface/ethernet 0> crypto trusted
Networks1/configure/interface/ethernet 0> exit
```

Step 3: Display the crypto interfaces

```
Networks1> show crypto interfaces
Interface      Network
Name           Type
---------      -------
ethernet0      trusted
wan1           untrusted
```

Step 4: Add route to peer LAN

```
Networks1/configure> ip route 10.0.2.0 24 172.16.0.2
```

Step 5: Configure IKE to the peer gateway

```
Networks1/configure> crypto
Networks1/configure/crypto> ike policy Networks2 172.16.0.2
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> local-
address 172.16.0.1
message: Default proposal created with priority1-des-sha1-pre_shared-g1
message: Key String has to be configured by the user
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> key secretkey
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> proposal 1
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2/proposal 1>
encryption-algorithm 3des-cbc
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2/proposal 1>
exit
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> exit
```

Step 6: Display the IKE policies

```
Networks1> show crypto ike policy all
```

Step 7: Display the IKE policies in detail

```
Networks1> show crypto ike policy all detail
```

Step 8: Configure IPSec tunnel to the remote host

```
Networks1/configure/crypto> ipsec policy
Networks2 172.16.0.2
Networks1/configure/crypto/ipsec policy Networks2 172.16.0.2> match
address 172.16.0.1 32 10.0.2.0 24
message: Default proposal created with priority1-esp-3des-sha1-tunnel and
activated.
Networks1/configure/crypto> ipsec policy Networks2 172.16.0.2> proposal 1
Networks1/configure/crypto> ipsec policy Networks2 172.16.0.2/proposal -
algorithm aes128-cbc
Networks1/configure/crypto> ipsec policy Networks2 172.16.0.2/proposal 1>
exit
Networks1/configure/crypto> ipsec policy Networks2 172.16.0.2> exit
```

Note:

For IPSec only – when you create an outbound tunnel, an inbound tunnel is automatically created. The inbound tunnel applies the name that you provide for the outbound tunnel and adds the prefix IN to the name.

Step 9: Display IPSec policies

```
Networks1> show crypto ipsec policy all
```

Step 10: Display IPSec policies detail

```
Networks1> show crypto ipsec policy all detail
```

Step 11: Configure firewall policies to allow IKE negotiation through untrusted interface

```
Networks1/configure> firewall internet
Networks1/configure/firewall internet> policy 1000 in service ike self
Networks1/configure/firewall internet/policy 1000 in> exit
Networks1/configure/firewall internet> exit
```

Step 12: Display firewall policies in the Internet map

```
Networks1> show firewall policy internet
```

Step 13: Display firewall policies in the Internet map in detail

```
Networks1> show firewall policy internet detail
```

Step 14: Configure firewall policies to allow transit traffic from remote LAN to the local LAN

```
Networks1/configure> firewall corp
Networks1/configure/firewall corp> policy 1000 in address 10.0.2.0 24
```

```
10.0.1.0 24
Networks1/configure/firewall corp/policy 1000 in> exit
Networks1/configure/firewall corp> exit
```

Step 15: Display firewall policies in the corp map

```
Networks1> show firewall policy corp
```

Step 16: Display firewall policies in the corp map in detail

```
Networks1> show firewall policy corp detail
```

Step 17: Repeat steps 1 -16 with suitable modifications on Networks2 prior to passing traffic

Step 18: Test the IPSec tunnel between Networks1 and Networks2 by passing traffic from the 10.0.1.0 to the 10.0.2.0 network

Step 19: After transit traffic is passed through the tunnel, display the IKE and IPSec SA tables

```
Networks1> show crypto ike sa all
```

```
Networks1> show crypto ike sa all detail
```

```
Networks1> show crypto ipsec sa all
```

```
Networks1> show crypto ipsec sa all detail
```

Example 3: Joining Two Networks with an IPSec Tunnel using Multiple IPSec Proposals

The following example demonstrates how a security gateway can use multiple IPSec (phase2) proposals to form an IP security tunnel to join two private networks: 10.0.1.0/24 and 10.0.2.0/24.

IKE Proposal offered by both Networks1 and Networks2:

Example

Phase 1: 3DES and SHA1

IPSec Proposals offered by Networks1:

Example

- Phase 2: Proposal1: IPSec ESP with DES and HMAC-SHA1
- Phase 2: Proposal2: IPSec ESP with AES (256-bit) and HMAC-SHA1

IPSec Proposal offered by Networks2:

Example

Phase 2: Proposal1: IPSec ESP with AES (256-bit) and HMAC-SHA1

In this example, the Networks1 router offers two IPSec proposals to the peer while the Networks2 router offers only one proposal. As a result of quick mode negotiation, the two routers are expected to converge on a mutually acceptable proposal, which is the proposal "IPSec ESP with AES (256-bit) and HMAC-SHA1" in this example.

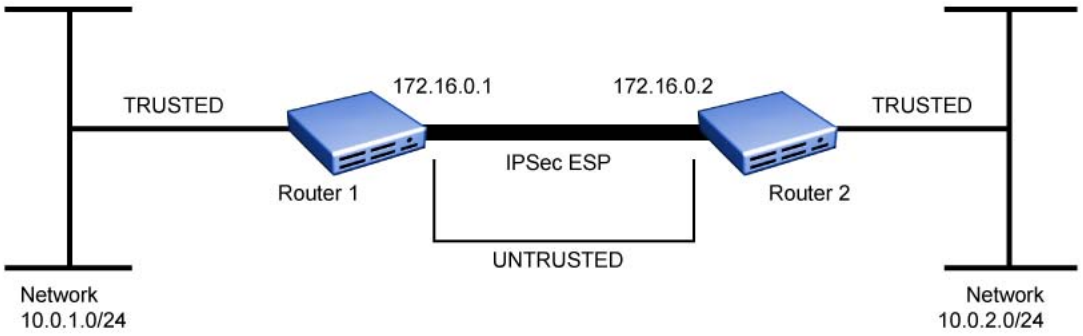


Figure 24: Tunnel Mode Between Two Security Gateways - Multiple Proposals

Step 1: Configure a WAN bundle of network type untrusted

```
Networks1/configure/interface/bundle wan1> link t1 1
Networks1/configure/interface/bundle wan1> encapsulation ppp
Networks1/configure/interface/bundle wan1> ip address 172.16.0.1 24
Networks1/configure/interface/bundle wan1> crypto untrusted
Networks1/configure/interface/bundle wan1> exit
```

Step 2: Configure the Ethernet interface with trusted network type

```
Networks1/configure> interface ethernet 0
Configuring existing Ethernet interface
Networks1/configure interface/ethernet 0> ip address 10.0.1.1 24
Networks1/configure/interface/ethernet 0> crypto trusted
Networks1/configure/interface/ethernet 0> exit
```

Step 3: Display the crypto interfaces

```
Networks1> show crypto interfaces
Interface      Network
Name           Type
```

```
---------       -------
ethernet0       trusted
wan1            untrusted
```

Step 4: Add the route to the peer LAN

```
Networks1/configure> ip route 10.0.2.0 24 wan1
```

Step 5: Configure IKE to the peer gateway

```
Networks1/configure> crypto
Networks1/configure/crypto> ike policy Networks2 172.16.0.2
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> local-
address 172.16.0.1
message: Default proposal created with priority1-des-sha1-pre_shared-g1
message: Key String has to be configured by the user
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> key secretkey
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> proposal 1
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2/proposal 1>
encryption-algorithm 3des-cbc
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2/proposal 1>
exit
Networks1/configure/crypto/ike/policy Networks2 172.16.0.2> exit
```

Step 6: Display the IKE policies

```
Networks1> show crypto ike policy all
```

Step 7: Display the IKE policies in detail

```
Networks1> show crypto ike policy all detail
```

Step 8: Configure IPSec tunnel to the remote host

```
Networks1/configure/crypto> ipsec policy Networks2 172.16.0.2
Networks1/configure/crypto/ipsec/policy Networks2 172.16.0.2> match
address 10.0.1.0 24 10.0.2.0 24

message:Default proposal created with priority1-esp-3des-sha1-tunnel and
activated.

Networks1/configure/crypto/ipsec/policy Networks2 172.16.0.2> proposal 1
Networks1/configure/crypto/ipsec/policy Networks2 172.16.0.2/proposal 1>
encryption-algorithm aes256-cbc
Networks1/configure/crypto/ipsec/policy Networks2 172.16.0.2/proposal 1>
exit
Networks1/configure/crypto/ipsec/policy Networks2 172.16.0.2> exit
```

Note:

For IPSec only – when you create an outbound tunnel, an inbound tunnel is automatically created. The inbound tunnel applies the name that you provide for the outbound tunnel and adds the prefix IN to the name.

Step 9: Display the IPSec policies

```
Networks1> show crypto ipsec policy all
```

```
Networks1> show crypto ipsec policy all detail
```

Step 10: Configure firewall policies to allow IKE negotiation through untrusted interface

```
Networks1/configure> firewall internet
Networks1/configure/firewall internet> policy 1000 in service ike self
Networks1/configure/firewall internet/policy 1000 in> exit
Networks1/configure/firewall internet> exit
```

Step 11: Display firewall policies in the Internet map

```
Networks1> show firewall policy internet
```

Step 12: Display firewall policies in the Internet map in detail

```
Networks1> show firewall policy internet detail
```

Step 13: Configure firewall policies to allow transit traffic from remote LAN to the local LAN

```
Networks1/configure> firewall corp
Networks1/configure/firewall corp> policy 1000 in address 10.0.2.0 24
10.0.1.0 24
Networks1/configure/firewall corp/policy 1000 in> exit
Networks1/configure/firewall corp> exit
```

Step 14: Display firewall policies in the corp map

```
Networks1> show firewall policy corp
```

Step 15: Display firewall policies in the corp map in detail

```
Networks1> show firewall policy corp detail
```

Step 16: Repeat steps 1 -15 with suitable modifications on Networks2 prior to passing bi-directional traffic

Step 17: Test the IPSec tunnel between Networks1 and Networks2 by passing traffic from the 10.0.1.0 network to the 10.0.2.0 network

Step 18: After traffic is passed through the tunnel, display the IKE and IPSec SA tables

```
Networks1> show crypto ike sa all
```

```
Networks1> show crypto ike sa all detail
```

```
Networks1> show crypto ipsec sa all
```

```
Networks1> show crypto ipsec sa all detail
```

Example 4: Supporting Remote User Access

The following example demonstrates how to configure a router to be an IPSec VPN server using user group method with extended authentication (XAUTH) for remote VPN clients. The client could be any standard IPSec VPN client.

In this example, the client needs to access the corporate private network 10.0.1.0/24 through the VPN tunnel. The security requirements are as follows:

Example

- Phase 1: 3DES with SHA1, Xauth (Radius PAP)
- Phase 2: IPSec ESP tunnel with AES256 and HMAC-SHA1

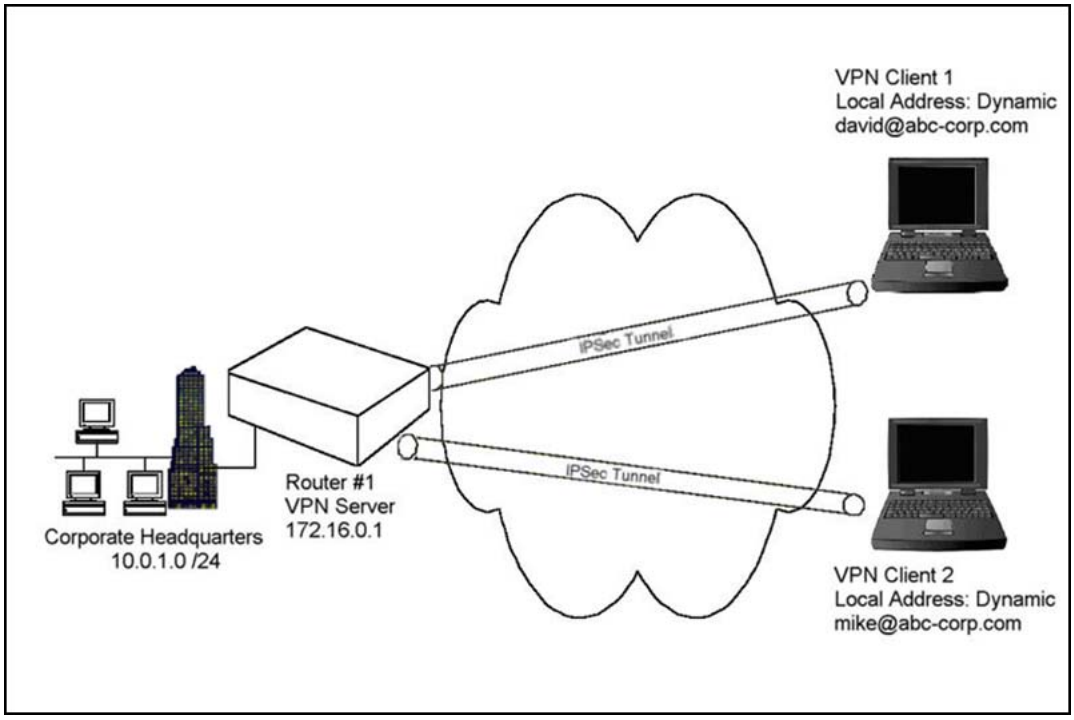


Figure 25: IPSec Tunneling Using User Group Method

Step 1: Configure a WAN bundle of network type untrusted

```
Networks1/configure> interface bundle wan1
message: Configuring new bundle
Networks1/configure/interface/bundle wan1> link t1 1
Networks1/configure/interface/bundle wan1> encapsulation ppp
Networks1/configure/interface/bundle wan1> ip address 172.16.0.1 24
Networks1/configure/interface/bundle wan1> crypto untrusted
Networks1/configure/interface/bundle wan1> exit
```

Step 2: Configure the Ethernet interface with trusted network type

```
Networks1/configure> interface ethernet 0
message: Configuring existing Ethernet interface
Networks1/configure interface/ethernet 0> ip address 10.0.1.1 24
Networks1/configure/interface/ethernet 0> crypto trusted
Networks1/configure/interface/ethernet 0> exit
```

Step 3: Display the crypto interfaces

```
Networks1> show crypto interfaces
Interface      Network
Name           Type
---------      -------
ethernet0      trusted
wan1           untrusted
```

Step 4: Configure dynamic IKE policy for a group of mobile users

```
Networks1/configure> crypto
Networks1/configure/crypto> dynamic
Networks1/configure/crypto/dynamic> ike policy sales
Networks1/configure/crypto/dynamic/ike/policy sales> local-address
172.16.0.1
Networks1/configure/crypto/dynamic/ike/policy sales>remote-id email-id
david@abc-corp.com david

New user david is added to the group sales
Default proposal created with priority1-des-sha1-pre_shared-g1
Key String has to be configured by the user
Networks1/configure/crypto/dynamic/ike/policy sales> remote-id email-id
mike@abc-corp.com mike
New user mike is added to the group sales

Networks1/configure/crypto/dynamic/ike/policy sales>
keysecretkeyforsalesusers
Networks1/configure/crypto/dynamic/ike/policy sales> proposal 1
Networks1/configure/crypto/dynamic/ike/policy sales/proposal 1>
encryption-algorithm 3des-cbc
Networks1/configure/crypto/dynamic/ike/policy sales/proposal 1> exit
Networks1/configure/crypto/dynamic/ike/policy sales> client
authentication radius pap
Networks1/configure/crypto/dynamic/ike/policy sales> exit
Networks1/configure/crypto/dynamic>
```

Step 5: Display dynamic IKE policies

```
Networks1> show crypto dynamic ike policy all
```

Step 6: Display dynamic IKE policies in detail

```
Networks1> show crypto dynamic ike policy all detail
```

Step 7: Configure dynamic IPSec policy for a group of mobile users

```
Networks1/configure/crypto/dynamic> ipsec policy sales

Networks1/configure/crypto/dynamic/ipsec/policy sales> match address
10.0.1.0 24
Default proposal created with priority1-esp-3des-sha1-tunnel and
activated.

Networks1/configure/crypto/dynamic/ipsec/policy sales> proposal 1
Networks1/configure/crypto/dynamic/ipsec/policy sales/proposal 1>
encryption-algorithm aes256-cbc
Networks1/configure/crypto/dynamic/ipsec/policy sales/proposal 1> exit
Networks1/configure/crypto/dynamic/ipsec/policy sales> exit
Networks1/configure/crypto/dynamic>
```

Step 8: Display dynamic IPSec policies

```
Networks1> show crypto dynamic ipsec policy all
```

Step 9: Display dynamic IPSec policies in detail

```
Networks1> show crypto dynamic ike policy all detail
```

Step 10: Configure radius server (applicable only if client authentication is configured in dynamic IKE policy)

```
Networks1/configure> aaa
Networks1/configure/aaa> radius
Networks1/configure/aaa/radius> primary_server 172.168.2.1
Primary Radius server configured.
Networks1/configure/aaa/radius> secondary_server 192.168.2.1
Secondary Radius server configured.
Networks1/configure/aaa/radius> exit
Networks1/configure/aaa> exit
```

Step 11: Configure firewall policies to allow IKE negotiation through untrusted interface

```
Networks1/configure> firewall internet
Networks1/configure/firewall internet> policy 1000 in service ike self
Networks1/configure/firewall internet/policy 1000 in> exit
Networks1/configure/firewall internet> exit
```

Step 12: Display firewall policies in the Internet map

```
Networks1> show firewall policy internet
```

Step 13: Display firewall policies in the Internet map in detail

```
Networks1> show firewall policy internet detail
```

Step 14: Configure firewall policies for a group of mobile users to allow access to the local LAN

```
Networks1/configure/firewall corp>
Networks1/configure/firewall corp> policy 1000 in user-group sales
address any  any 10.0.1.0 24
Networks1/configure/firewall corp/policy 1000 in> exit
Networks1/configure/firewall corp>
```

Note:

Be sure to match the user group name in the policy command with the name used in Step 4 (the dynamic IKE policy).

Step 15: Display firewall policies in the corp map

```
Networks1> show firewall policy corp
```

Step 16: Display firewall policies in the corp map in detail

```
Networks1> show firewall policy corp detail
```

Step 17: Test the IPSec tunnel between the VPN client and the server by passing traffic from the client to the 10.0.1.0 network

Step 18: After passing traffic through the tunnel, display the list of clients logged onto the VPN server and the IKE and IPSec SA tables

```
Networks1> show crypto dynamic clients
Client Address    Client Id                Policy             Advanced
--------------    ---------                ------             --------
192.168.107.105 david@abc-corp...           sales              UserGrp
```

```
Networks1> show crypto ike sa all
```

```
Networks1> show crypto ike sa all detail
```

```
Networks1> show crypto ipsec sa all
```

```
Networks1> show crypto ipsec sa all detail
```

Example 5: Configuring IPSec Remote Access to Corporate LAN with Mode-Configuration Method

The following example demonstrates how to configure a router to be an IPSec VPN server using mode-configuration method. The client could be any standard mode configuration enabled IPSec VPN client.

In this example, the client needs to access the corporate private network 10.0.1.0/24 through the VPN tunnel. The server has a pool of ip addresses from 20.1.1.100 through 20.1.1.150 to be allocated for mode configuration enabled VPN clients. The assigned IP address is used by the VPN client as the source address in the inner IP header. The outer IP header carries the dynamic IP address assigned by the Internet Service Provider as the source address. The security requirements are as follows:

Example

- Phase 1: 3DES with SHA1, Mode Configuration
- Phase 2: IPSec ESP tunnel with AES256 and HMAC-SHA1

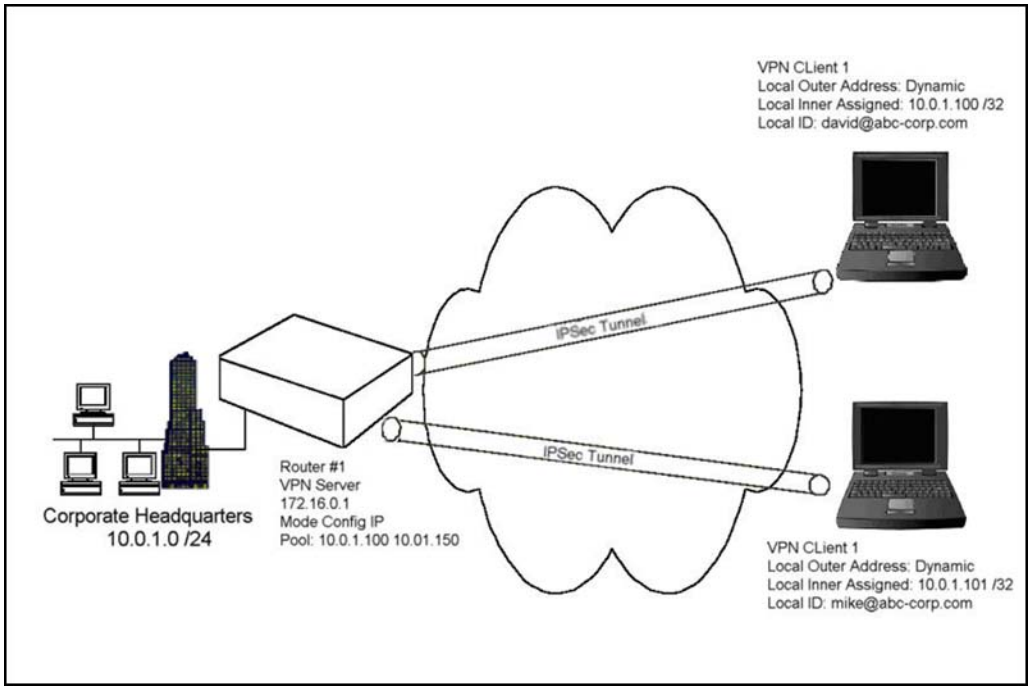


Figure 26: IPSec Tunneling Using Mode Configuration Method

Step 1: Configure a WAN bundle of network type untrusted

```
Networks1/configure> interface bundle wan1
message: Configuring new bundle
Networks1/configure/interface/bundle wan1> link t1 1
Networks1/configure/interface/bundle wan1> encapsulation ppp
Networks1/configure/interface/bundle wan1> ip address 172.16.0.1 24
Networks1/configure/interface/bundle wan1> crypto untrusted
Networks1/configure/interface/bundle wan1> exit
```

Step 2: Configure the Ethernet interface with trusted network type

```
Networks1/configure> interface ethernet 0
message: Configuring existing Ethernet interface
Networks1/configure interface/ethernet 0> ip address 10.0.1.1 24
Networks1/configure/interface/ethernet 0> crypto trusted
Networks1/configure/interface/ethernet 0> exit
```

Step 3: Display the crypto interfaces

```
Networks1> show crypto interfaces
Interface      Network
Name           Type
---------      -------
ethernet0      trusted
wan1           untrusted
```

Step 4: Configure dynamic IKE policy for a group of mobile users

```
Networks1/configure> crypto
Networks1/configure/crypto> dynamic
Networks1/configure/crypto/dynamic> ike policy sales modecfg-group
Networks1/configure/crypto/dynamic/ike/policy sales> local-address
142.168.55.52

Networks1/configure/crypto/dynamic/ike/policy sales> remote-id email
david@abc-corp.com

Default proposal created with priority1-des-sha1-pre_shared-g1
Key String has to be configured by the user
Default ipsec proposal 'sales' added with priority1-3des-sha1-tunnel

Networks1/configure/crypto/dynamic/ike/policy sales> remote-id email
mike@abc-corp.com

Networks1/configure/crypto/dynamic/ike/policy sales> key
secretkeyforsales
Networks1/configure/crypto/dynamic/ike/policy sales> proposal 1
Networks1/configure/crypto/dynamic/ike/policy sales/proposal 1>
encryption-algorithm 3des-cbc
Networks1/configure/crypto/dynamic/ike/policy sales/proposal 1> exit
Networks1/configure/crypto/dynamic/ike/policy sales> client configuration
Networks1/configure/crypto/dynamic/ike/policy sales/client/
configuration> address-pool 1 20.1.1.100 20.1.1.150
Networks1/configure/crypto/dynamic/ike/policy sales/client/
configuration> exit
Networks1/configure/crypto/dynamic/ike/policy sales> exit
Networks1/configure/crypto/dynamic> exit
```

Step 5: Display dynamic IKE policies

```
Networks1> show crypto dynamic ike policy all
```

Step 6: Display dynamic IKE policies in detail

```
Networks1> show crypto dynamic ike policy all detail
```

Step 7: Configure dynamic IPSec policy for a group of mobile users

```
Networks1/configure/crypto>
Networks1/configure/crypto> dynamic
Networks1/configure/crypto/dynamic> ipsec policy sales modecfg-group
Networks1/configure/crypto/dynamic/ipsec/policy sales> match address
10.0.1.0 24
Networks1/configure/crypto/dynamic/ipsec/policy sales> proposal 1
Networks1/configure/crypto/dynamic/ipsec/policy sales/proposal 1>
encryption-algorithm aes256-cbc
Networks1/configure/crypto/dynamic/ipsec/policy sales/proposal 1> exit
Networks1/configure/crypto/dynamic/ipsec/policy sales> exit
Networks1/configure/crypto/dynamic> exit
```

Step 8: Display dynamic IPSec policies

```
Networks1> show crypto dynamic ipsec policy all
```

Step 9: Display dynamic IPSec policies in detail

```
Networks1> show crypto dynamic ipsec policy all detail
```

Step 10: Configure firewall policies to allow IKE negotiation through untrusted interface

```
Networks1/configure> firewall internet
Networks1/configure/firewall internet> policy 1000 in service ike self
Networks1/configure/firewall internet/policy 1000 in> exit
```

Step 11: Display firewall policies in the Internet map

```
Networks1> show firewall policy internet
```

Step 12: Display firewall policies in the Internet map in detail

```
Networks1> show firewall policy internet detail
```

Step 13: Configure firewall policies for a group of mobile users to allow access to the local LAN

```
Networks1/configure> firewall corp
Networks1/configure/firewall corp> policy 1000 in address 20.1.1.100
20.1.1.150 10.0.1.0 24
Networks1/configure/firewall corp/policy 1000 in> exit
```

Note:

The address range in this command typically matches the address range configured in the dynamic IKE policy (see Step 4).

Step 14: Display firewall policies in the corp map

```
Networks1> show firewall policy corp
```

Step 15: Display firewall policies in the corp map in detail

```
Networks1> show firewall policy corp detail
```

Step 16: Test the IPSec tunnel between the VPN client and the server by passing traffic from the client to the 10.0.1.0 network

Step 17: After passing traffic through the tunnel, display the list of clients logged onto the VPN server and the IKE and IPSec SA tables

```
Networks1> show crypto dynamic clients
```

```
Networks1> show crypto ike sa all
```

```
Networks1> show crypto ike sa all detail
```

```
Networks1> show crypto ipsec sa all
```

```
Networks1> show crypto ipsec sa all detail
```

IKE Dead Peer Detection

IKE Dead Peer Detection (DPD) is a traffic-based method of detecting dead IKE peers. It utilizes on-demand IPSec traffic patterns to minimize the number of IKE messages sent to confirm activity. The purpose of DPD is to detect whether a peer is still alive. In the event a peer has died, the system can regain valuable resources, thereby improving network performance and availability. Secure Router supports On-demand DPD only.

For the best DPD performance, configure the crypto keepalive according to the sense of urgency. DPD occurs when a peer has not acknowledged its presence within (transmit-interval +3 * retry-interval) seconds. For faster DPD, set the transmit-interval and retry-interval to smaller values.

Example

Configuring IKE Dead Peer Detection

```
SR/config/crypto> keepalive enable
SR/config/crypto> keepalive transmit-interval 30
SR/config/crypto> keepalive retry-interval 10
```

Example

Displaying Dead Peer Detection status

```
SR/show/crypto> keepalive
```

PMTU Support for IPSec tunnels

PMTU is a configurable option. If enabled and fragmentation is required and DF bit s set, it sends an ICMP error to the packet originator. The DF bit from the inner IP header is copied to the outer IP header; this allows intermediate routers to fragment or not depending on the value of the DF bit. IP fragmentation is supported for IP packets that exceed the MTU after insertion of GRE/IPIP header. IP fragmentation if applicable is based on the MTU of the outbound physical interface.

Disabling the IPSec Anti-replay service

The ability to disable the anti-replay service is useful when using Diff-serv marking on a IPSec tunnel where you want to support voice traffic at a higher priority then data traffic. As the voice

call level (high priority) increases then the data traffic is delayed sufficiently where the anti-replay service starts affecting the amount of (lower priority) data traffic that is delivered properly. By disabling the anti-replay service more data traffic can get through.

There is a new configuration command under crypto command tree which enables or disables the anti-replay service. By default the anti-replay service is enabled. Also a new show crypto configuration command was added. Below is an example of toggling the service on and off.

```
R1/configure > crypto lay-service

R1/configure/crypto > antireplay-service

R1/configure/crypto > show crypto configuration

Crypto Configuration
------------------------
Anti-Replay Service: ON

R1/configure/crypto > no antireplay-service

R1/configure/crypto > show crypto configuration

Crypto Configuration
------------------------
Anti-Replay Service: OFF
```

VPN-only mode

Earlier releases of the Secure Router required that the firewall be configured to use the IPSec VPN features. To overcome this limitation, the Secure Router supports a VPN-only mode. Enabling VPN-only mode allows the traffic to skip firewall-related checks. IPSec services are provided based on the policies configured.

To switch to VPN-Only mode, you must reboot the router for the change to take effect. After the router is rebooted in VPN-only mode, all the commands under the firewall section do not exist.

The following sections show example configurations for converting the router to and from VPN-only mode.

Conversion procedure to VPN-only mode

```
Host> file
Host/file >
Host/file > copy system.cfg firewall.cfg
exit Host> conf t
Host/configure > system security firewall-disable
Host/configure> write mem
Host/configure> exit
Host> reboot
```

Converting back to firewall mode

```
Host> file
Host/file > copy system.cfg vpnonly.cfg
Host/file > copy firewall.cfg system.cfg
Host/file >
exit Host> conf t
Host/configure > no system security firewall-disable
Host/configure> exit
Host> reboot
```

Displaying the configuration

```
Host > show system security
```

Chapter 25: IPSec APPENDIX

This appendix provides information about IPSec supported protocols and modes, encryption algorithms and block sizes, and Avaya IPSec and IKE default values.

IPSec Supported Protocols and Algorithms

The following tables provide supported protocol and algorithm information.

Table 29: IPSec Protocols Support

Supported Security Protocols	Mode
ESP	Tunnel Transport
AH	Tunnel Transport

Table 30: Encryption Algorithms

Encryption Algorithms for ESP	Block Size
Data Encryption Standard (DES)	56-bits
Triple Data Encryption Standard (3DES)	168-bits
Advanced Encryption Standard (AES-128)	128-bits
Advanced Encryption Standard (AES-192)	192-bits
Advanced Encryption Standard (AES-256)	256-bits
Null Encryption	

Table 31: Authentication Algorithms

Authentication Algorithms for AH/ESP	Hash Size
HMAC-MD5-96	96-bits
HMAC-HSHA1-96	96-bits
Null Authentication	

Table 32: Diffie-Hellman Group

Diffie-Hellman Groups for Authentication	Key Size
Group 1	768-bits
Group 2	1024-bits
Group 5	1536-bits

Avaya IKE and IPSec Defaults

To minimize configuration required by the user, default IKE and IPSec values are implemented in the Avaya encryption scheme.

IKE Defaults

The following table lists IKE defaults. When the user creates an IKE policy specifying an IKE peer, an IKE proposal with priority 1 is automatically created. However, to make the IKE policy fully functional, the user must enter a pre-shared key.

Table 33: IKE Default Values

Parameter Name	Avaya Default Value
Mode	Main mode
Perfect forward secrecy	Disabled
Hash algorithm	SHA1
Encryption algorithm	DES
Authentication method	PreShared
DH Group	Group 1
Lifetime	86400 seconds
Response type	Initiator and responder

IPSec Defaults

The following table lists IPSec defaults. When the user creates an IPSec policy and provides the match address, an IPSec proposal with priority 1 is automatically created. When an outbound policy is specified, an inbound policy is automatically created.

Table 34: IPSec Default Values

Parameter Name	Avaya Default Value
Key management type	Automatic
Hash algorithm	SHA1
Encryption algorithm	3DES
Protocol	ESP
Mode	Tunnel
Lifetime	3600 seconds
Direction	Out
Position in SPD where policy added	End
Perfect forward secrecy	Disabled

Chapter 26: PKI Certificate Support

Manual Certificate Enrollment:

1. Create a trustpoint.

   ```
   R1/configure/crypto> ca trustpoint ms2003
   ```

2. Configure the enrollment mode to terminal.

   ```
   R1/configure/crypto/ca/trustpoint ms2003> enrollment
   terminal
   ```

3. Configure the subject name, ip address, fqdn, email address and key pair details.

   - `R1/configure/crypto/ca/trustpoint ms2003> subject-name "cn=orion,ou=security,o=avaya,c=us"`
   - `R1/configure/crypto/ca/trustpoint ms2003> ip-address 10.1.1.1`
   - `R1/configure/crypto/ca/trustpoint ms2003> fqdn avaya.com`
   - `R1/configure/crypto/ca/trustpoint ms2003> email test@test.com`
   - `R1/configure/crypto/ca/trustpoint ms2003> keypair key1 rsa 1024`

4. Import the CA certificate.

   ```
   R1/configure/crypto> ca authenticate ms2003
   ```

   Paste the Certificate in PEM format. Finger print is computed on the CA certificate, and displayed to the user

5. Enroll the certificate request.

   ```
   R1/configure/crypto> ca enroll ms2003
   ```

   This command generates the certificate request in PEM format.

6. Import the router certificate.

   ```
   R1/configure/crypto/ca/import ms2003> router-certificate
   ```

   This command generates the certificate request in PEM format.

Certificate enrollment using SCEP

1. Create a trustpoint.

   ```
   R1/configure> ca trustpoint ms2003
   ```

2. Configure the enrollment URL.

   ```
   R1/configure/crypto/ca/trustpoint ms2003> enrollment url
   http://192.168.114.2/certsrv/mscep/mscep.dll/
   ```

3. Configure the subject name, ip address, fqdn, email address and key pair details.

   - `R1/configure/crypto/ca/trustpoint ms2003> subject-name cn=orion,ou=security,o=tasmannetworks,c=us`
   - `R1/configure/crypto/ca/trustpoint ms2003> ip-address 10.1.1.1`
   - `R1/configure/crypto/ca/trustpoint ms2003> fqdn tasmannetworks.com`
   - `R1/configure/crypto/ca/trustpoint ms2003> email test@test.com`
   - `R1/configure/crypto/ca/trustpoint ms2003> keypair key1 rsa 1024`

4. Fetch the Certificate Authority (CA) Certificate.

   ```
   R1/configure/crypto> ca authenticate ms2003
   ```

   Finger print is computed on the CA certificate, and displayed to the user.

5. Generate the Certificate request, send it to CA and import the certificate. Since here the enrollment method is SCEP, everything is done in a single command.

   ```
   R1/configure/crypto> ca enroll ms2003
   ```

   Receive the router certificate from the CA server

IKE negotiation with DSS

1. Configure the authentication method to Digital Signature Standard (DSS)

   ```
   R1/configure/crypto/ike/policy test1 11.1.1.1/proposal 1>
   authentication-method dss-signature
   ```

2. Certificate validation can be done using CRLs or OCSP. OCSP supports real time certificate validation.

IKE negotiation with RSA

1. Configure the authentication method to RSA Signature

   ```
   R1/configure/crypto/ike/policy test1 11.1.1.1/proposal 1>
   authentication-method dss-signature
   ```

2. Certificate validation can be done using CRLs or OCSP. OCSP supports real time certificate validation.

OCSP Configuration

1. Configure OCSP Responder URL.

   ```
   R1/configure/crypto/ca/trustpoint ms2003> ocsp url http://
   192.168.114.3:2560/
   ```

2. Enable OCSP

   ```
   R1/configure/crypto/ike/policy test1 11.1.1.1> ocsp
   ```

   OCSP enabled for this policy

CRL Configuration

1. CRL can be retrieved through LDAP client and SCEP client. We can also import the CRL manually (cut and paste method). LDAP client supports the periodic download of CRLs.
2. Ldap Client configuration.

   ```
   R1/configure/crypto/ca/trustpoint ms2003> crl query ldap://
   192.168.114.3/ou=security,o=tasman,c=us
   ```

3. SCEP Client configuration.

   ```
   R1/configure/crypto/ca/trustpoint ms2003> enrollment url
   http://192.168.114.2/certsrv/mscep/mscep.dll/
   ```

4. Manual CRL download

   ```
   R1/configure/crypto/ca/trustpoint ms2003> enrollment
   terminal
   ```

5. Fetch the CRL. If configured, Ldap client is used to fetch the CRL. If SCEP client is configured, CRL will be downloaded using SCEP. If the enrollment mode is manual, user will be prompted to paste the CRL in PEM format.

   ```
   R1/configure/crypto/ca/crl> request ms2003
   ```

Note:
All the certificates are saved in Certificates.dat file and private keys are stored in Keys.dat file. Since private keys need to be securely stored, private keys are stored in an encrypted format.

Chapter 27: Configuring GRE

Generic Routing Encapsulation (GRE) is a standards-based (RFC1701, RFC2784) tunneling protocol that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link between routers at remote points over an IP network. A tunnel is a logical interface that provides a way to encapsulate passenger packets inside a transport protocol. By connecting multiprotocol subnetworks in a single-protocol backbone environment, IP tunneling using GRE allows network expansion across a single-protocol backbone environment.

IPSec and GRE complement each other well, while IPSec provides a secure method of transporting data across the Internet GRE provides the capability to transport routing protocols (for example: OSPF) that use broadcast and multicast.

GRE tunnel can now be configured over the Ethernet that supports 1500 bytes of user data without having to fragment the packet over the tunnel.

Installing Licenses

Licenses control access to:

Example

- Basic VPN Management (vpn_mgmt) — allows users to manage a remote Secure Router.
- Advanced VPN (advance_vpn) — allows users to manage remote LANs.

To see the licenses available in this release, enter:

```
/configure> system licenses ?
NAME
licenses - Configure feature upgrade licenses
SYNTAX
licenses license_type <cr>
DESCRIPTION
license_type        -- Specifies the type of feature upgrade license
The parameter may have any of the following values:
vpn_mgmt      -- Enable VPN Mgmt License
advance_vpn   -- Enable Advance VPN
```

To install the advanced VPN license and use all the security features available in this release, enter:

```
/configure> system licenses advance_vpn
Enter Security Upgrade License key: 024f3bc296b4ea7265
```

GRE Configuration Examples

This example explains how to configure a basic GRE tunnel as shown in Figure 27: Fig 2 Simple GRE configuration on page 178.

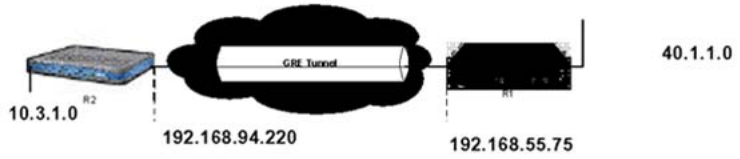


Figure 27: Fig 2 Simple GRE configuration

Configuring Site to Site Tunnel

To configure GRE in a site to site tunnel configuration:

1. Configure the interface.

```
SR> configure terminal
Router/configure> interface bundle wan1
Router/configure/interface/bundle wan1> link t1 1
Router/configure/interface/bundle wan1> encapsulation ppp
Router/configure/interface/bundle wan1> ip address
192.168.94.220 255.255.255.0
Router/configure/interface/bundle wan1> exit
```

2. Configure the tunnel.

```
Router/configure> interface tunnel t0
Router/configure/interface/tunnel t0> ip 103.1.1.2 24
Router/configure/interface/tunnel t0> tunnel source
192.168.94.220
Router/configure/interface/tunnel t0> tunnel destination
192.168.55.75
Router/configure/interface/tunnel t0> exit
```

3. Configure the IP routes.

```
Router/configure> ip route 0.0.0.0 0.0.0.0 192.168.94.254
Router/configure> ip route 40.1.1.0 24 t0
```

Note:
The peer of a local WAN interface cannot be used as a tunnel destination.

4. Verify that the tunnel is up and running. If it is not, check the Gateway and Source Address fields.

```
SR> show ip interface t0
t0 (unit number 5)
Type: TUNNEL
Flags: (0x74243) UP, RUNNING, MULTICAST-ROUTE
Internet Address: 103.1.1.2
Internet Netmask: 255.255.255.0
Internet Broadcast: 103.1.1.255
Maximum Transfer Unit: 1476 bytes
Source Address: 192.168.94.220
Destination Address: 192.168.55.75
Gateway: wan1
Protocol: GRE
Mac Address 00:50:52:60:00:00
```

For more information enter:

```
SR> show interface tunnel t0
Tunnel: t0 Status: up
Internet Address: 103.1.1.2 Internet Netmask: 255.255.255.0
Source Address: 192.168.94.220 Destination Address: 192.168.55.75
MTU: 1476 bytes                       Protocol: GRE
ICMP unreachable: will be sent        ICMP redirect: will be sent
Crypto Snet: not set                  Protection: policy grecisco
key ****
TTL: 30                               Keepalive: disabled
TOS: not set                          Path MTU discovery: disabled
Key Value: not set                    Checksum: disabled
Sequence Datagrams: disabled
Tunnel Statistics:
Bytes Rx                  95112    Bytes Tx                  60016
Packets Rx                  860    Packets Tx                  499
```

```
Err Packets Rx                  0    Output Errs
0
```

5. Configure the Cisco-compatible router side:

```
cisco > config t
cisco(config)>interface Ethernet2/0
cisco(config-if)>ip address 192.168.55.75255.255.255.0
cisco(config-if)>exit

cisco(config)>interface Tunnel 0
cisco(config-if)>ip address 103.1.1.1 255.255.255.0
cisco(config-if)>tunnel source 192.168.55.75
cisco(config-if)>tunnel destination 192.168.94.220
cisco(config-if)>exit
cisco(config)>ip route 0.0.0.0 0.0.0.0 192.168.55.254
cisco(config)>ip route 10.3.1.0 255.255.255.0 Tunnel0
```

Bridging across GRE

This configuration is useful for transport of non-IP protocols such as VLANs across MPLS, or VPN. In both configurations, untagged packets on Ethernet1 are bridged across the Ethernet0 "WAN" through a GRE tunnel terminated on the Ethernet0 endpoints.

Router 1

```
interface ethernet 0
ip address 10.1.1.2 255.255.255.0
exit ethernet
interface ethernet 1
ip address 192.168.4.1 255.255.255.0
vlan
vlanid 10
exit vlan
exit ethernet
interface tunnel gre1
ip address 192.168.1.2 255.255.255.0
tunnel source 10.1.1.2
tunnel destination 10.1.1.1
```

Router 2

```
interface ethernet 0
ip address 10.1.1.1 255.255.255.0
exit ethernet
interface ethernet 1
ip address 192.168.3.1 255.255.255.0
vlan
vlanid 10
exit vlan
exit ethernet
interface tunnel gre1
ip address 192.168.1.1 255.255.255.0
tunnel source 10.1.1.1
tunnel destination 10.1.1.2
```

Configuring GRE Site to Site with IPSec

This example extends the first example by adding encryption to the tunnel.

1. Prepare the WAN link:

```
SR> configure terminal
Router/ configure> interface bundle wan1
Router/ configure/interface/bundle wan1> link t1 1
Router/ configure/interface/bundle wan1> encapsulation ppp
Router/ configure/interface/bundle wan1> ip address
192.168.94.220 255.255.255.0
Router/ configure/interface/bundle wan1> crypto untrusted
Router/ configure/interface/bundle wan1> exit
```

2. Configure the tunnel:

```
Router/ configure> interface tunnel t0
Router/ configure/interface/tunnel t0> ip address 103.1.1.2 24
Router/ configure/interface/tunnel t0> tunnel source
192.168.94.220
Router/ configure/interface/tunnel t0> tunnel destination
192.168.55.75
Router/ configure/interface/tunnel t0> tunnel protection
grecisco secretkeyfortest
Router/ configure/interface/tunnel t0> crypto untrusted
Router/ configure/interface/tunnel t0> exit
```

3. Configure the routes:

```
Router/ configure> ip route 0.0.0.0 0.0.0.0 192.168.94.254
Router/ configure> ip route 40.1.1.0 24 t0
```

4. Define the policy:

```
SR/ configure > firewall internet
SR/configure/firewall internet> policy 100 in proto gre self
SR/configure/firewall internet/policy 100 in> exit
SR/configure/firewall internet> policy 101 in service ike self
SR/configure/firewall internet/policy 101 in> exit 2
SR/configure> firewall corp
SR/configure/firewall corp> policy 100 in self
```

5. Check the status of the tunnel by entering:

```
SR> show ip interface tunnel t0
```

6. Validate the tunnel configuration by entering:

```
SR> show crypto ipsec policy all
```

Or enter:

```
SR> show crypto ike policy all
```

Configuring GRE Site to Site with IPSec and OSPF

This example extends the previous IPSec configuration example by enabling Open Shortest Path First (OSPF) protocol which provides redundant paths for the tunnel.

1. To enable OSPF, add to the Secure Router configuration above:

```
SR> configure terminal
Router/configure> router routerid 2.2.2.2
Router/configure> router ospf
Router/configure/router/ospf> interface t0 area 0
Router/configure/router/ospf> exit
```

2. Add to the Cisco-compatible configuration above:

```
cisco > config t
cisco(config)>router ospf 1
cisco(config-router)> network 103.1.1.0 0.0.0.255 area 0
```

3. To verify the OSPF configuration, enter:

```
SR> show ip ospf interface all
```

Note:
Using the redistribute connected command adds a recursive route to the tunnel destination. This will cause the tunnel to shut down. To prevent this, add a 32-bit static route for the tunnel destination.

Multicast over GRE

The Secure Routers now support Multicast Routing (PIM-SM) over GRE tunnels. Typically, multicast routing protocol control traffic and multicast data traffic are sent over GRE tunnels when there has to be a data exchange between two multicast-enabled routers that are separated by an IP cloud that does not have multicast capabilities. In such scenarios, the ability to configure PIM over GRE tunnels helps in transporting multicast packets (both control and data) across a non-Multicast aware IP cloud.

Configuration

When configuring PIM over GRE tunnels, you must adhere to the following points:

1. When PIM is to be run over GRE tunnels, configure the GRE tunnel IP addresses at both the tunnel end-points in the same subnet. Also, it is mandatory to ensure the reachability to the tunnel destination either using IGP or a static route.
2. When configuring PIM over GRE tunnels (with the tunnel source configured the same as the IP address of the tunnel) configuring the tunnel interface as the CBSR or CRP is not allowed, as the RPF check will fail at the tunnel end point routers (as result of having the 32-bit static route to the tunnel destination).
3. If the BSR/ RP/ Multicast Source networks reside on the other side of the tunnel, the reachability towards the BSR/ RP/ Multicast Source **should be** ensured with the next-hop as tunnel's other end, either by IGP or static routes.

Note:

There are no new CLI commands added for the purpose of supporting multicast over GRE tunnels. The PIM related configurations have the same syntax as they do for Ethernet or WAN interfaces.

The following figure shows an example configuration:

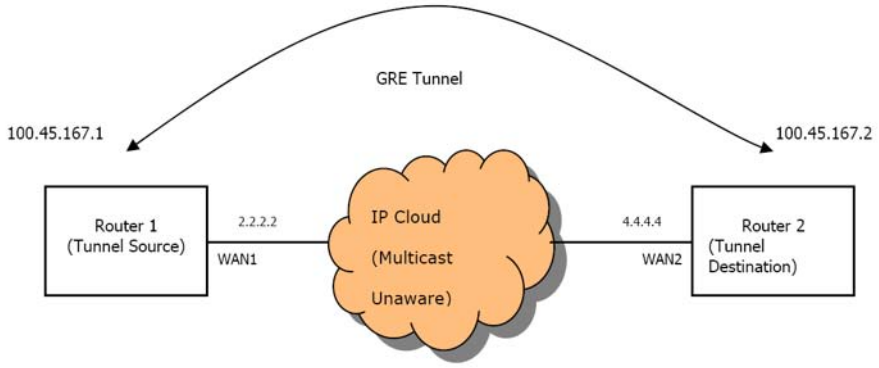


Figure 28: Multicast over GRE

Below is the sample configuration.

Router 1 configuration

```
Router 1/configure> interface tunnel t1
Router 1/configure/interface t1> ip address 100.45.167.1 255.255.255.0
Router 1/configure/interface t1> tunnel source 2.2.2.2
Router 1/configure/interface t1> tunnel destination 4.4.4.4
Router 1/configure/interface t1> exit tunnel
```

Router 2 configuration

```
Router 2/configure> interface tunnel t2
Router 2/configure/interface t2> ip address 100.45.167.2 255.255.255.0
Router 2/configure/interface t2> tunnel source 4.4.4.4
Router 2/configure/interface t2> tunnel destination 2.2.2.2
Router 2/configure/interface t2> exit tunnel
```

Chapter 28: Multipath Multicast

Configuration Guide

The multicast multipath feature allows load balancing on multicast traffic across equal cost paths. Equal cost multipath routing is useful when multiple equal cost routes to the same destination exist. These routes can be discovered and be used to provide load balancing among redundant paths. Commonly used methods for multipath forwarding are Round-Robin and Random. While these methods do provide a form of load balancing, but variable path MTUs, variable latencies, and debugging can limit the effectiveness of these methods.

The following methods are developed to deal with the load balancing limitations of the Round-Robin and Random methods:

- Modulo-N Hash — To select a next-hop from the list of N next-hops, the router performs a modulo-N hash over the packet header fields that identify a flow."
- Hash-Threshold— The router first selects a key by performing a hash over the packet header fields that identify the flow. The N next-hops are assigned unique regions in the hash functions output space. By comparing the hash value against region boundaries the router can determine which region the hash value belongs to and thus which next-hop to use.
- Highest Random Weight (HRW)— The router computes a key for each next-hop by performing a hash over the packet header fields that identify the flow, as well as over the address of the next-hop. The router then chooses the next-hop with the highest resulting key value.

The Round-Robin and Random methods are disruptive by design (that is, if there is no change to the set of next-hops, the path a flow takes changes every time). Modulo-N, Hash Threshold, and HRW are not disruptive.

RFC 2991 recommends to use HRW method to select the next-hop for multicast packet forwarding. or this reason, Avaya-only scenarios apply the HRW method as the default. This is similar to the Cisco Systems IPv6 multicast multipath implementation.

Multipath Commands

The following table lists the multipath commands:

Task	Command
Enabling HRW method	**Router/configure/ip/multicast> multipath**
Enabling Cisco-compatible method	**Router/configure/ip/multicast> multipath cisco**
Disabling Multipath	**Router/configure/ip/multicast> no multipath** **Router/configure/ip/multicast> no multipath cisco**
Display RPF selection	**SR>show ip rpf <addr> <addr> - source or RP address**

When multipath is disabled, Avaya selects the nexthop address with lowest ip address. For equal cost routes the nexthops are stored in the increasing (ascending) order of IP address. **show ip rpf** command displays the selected path, based on the configured multipath method and the nexthops of the best route to the IP address passed.

Multipath Examples

The following examples illustrate how the multicast commands are used:

The following command enables compatibility between the Secure Router and equipment running Cisco IOS.

Router/configure/ip/multicast> multipath mode cisco

The following command enables HRW compatibility.

Router/configure/ip/multicast> multipath

The following example shows how to see the reverse path forwarding information for the RP at 201.1.1.99:

```
SR> show ip rpf 201.1.1.99
```

Chapter 29: Multilink Frame Relay

Frame Relay service providers can use Avaya products to offer multimegabit service of 3 to 24 Mb/s using standard T1 local loops. These services can be used for both Intranet and Internet applications. Depending on the needs of the customer, Avaya products can perform router functionality or be installed simply as a Multilink Frame Relay multiplexer in front of an existing router. This application describes both configurations.

Features:

- Low cost and scalable multimegabit access using standard T1 access
- Flexible feature support (NAT, packet filtering, CBQ, routing) depending on customer needs
- Internet and Intranet access supported in one box
- Command Line Interface (CLI) and SNMP support for easy management
- Aggregation of FT1, T1, and NxT1 customers using PPP/MLPPP, HDLC, and Frame Relay/MFR in one product
- Low cost aggregation for remote POPs in emerging markets

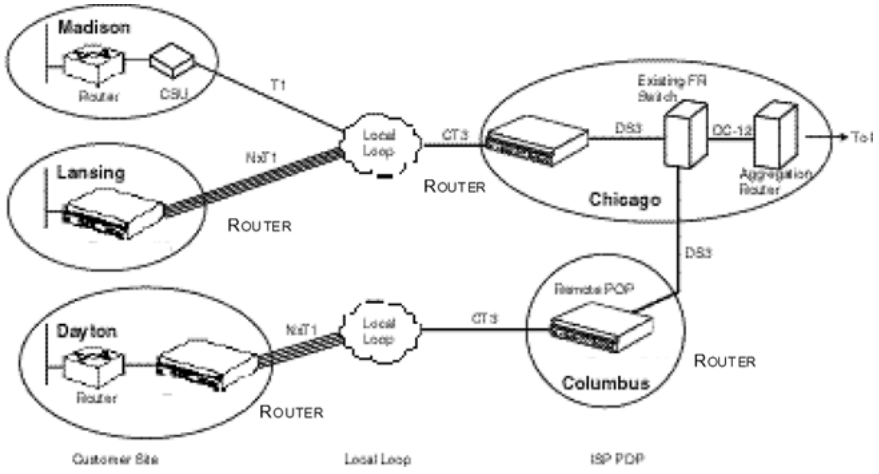


Figure 29: Multilink Frame Relay Switching

The illustration above shows a customer with three sites being served by a frame relay provider. The site in Madison requires only one T1, and it relies on Lansing for Internet access. At Lansing, the Secure Router provides MFR access to the Internet as well as a PVC to the other sites. Dayton has an existing router and needs the Secure Router to provide NxT1 access. By using IP multiplexing, the Secure Router can transparently provide access to the router while also performing CPU intensive tasks such as packet filtering, NAT, or CBQ.

In this example, the Madison, Lansing, and Dayton sites could be serviced with the Secure Router 1000 series and the Columbus and Chicago sites with the Secure Router 3120.

The service provider has a primary POP in Chicago and a smaller, remote POP in Columbus. The Routers at both locations provide frame relay switching from the MFR bundle to the existing frame switch through clear channel DS3. Switching between the PVCs is done in the existing frame switch to keep management and monitoring consistent with previous practices. In addition to enabling a new service (MFR), the Secure Routers lower the channelized T3 port cost. The Secure Router in Columbus is deployed as a means for reducing the up-front cost of entering a new market. After installing a few Routers there, the provider may decide to install a backbone switch at that location and upgrade the link between Columbus and Chicago.

A service provider who does not have an existing frame backbone might simply choose to uplink the Secure Router 3120s directly into the aggregation router. In that case, the Secure Router 3120 may be preferable when uplinking to an Ethernet-only router.

Note:

Reducing the CIR on frame relay interfaces too low can cause insufficient buffers for classes when configuring class-based queueing.

Chicago - Secure Router Configuration

```
SR/configure> hostname Chicago-Router
Chicago-SR/configure> interface bundle mad1
Chicago-SR/configure/interface/bundle mad1> link ct3 1/1/1
Chicago-SR/configure/interface/bundle mad1> encapsulation fr
Chicago-SR/configure/interface/bundle mad1> fr
Chicago-SR/configure/interface/bundle mad1/fr> intf_type dce
Chicago-SR/configure/interface/bundle mad1/fr> pvc 100
Chicago-SR/configure/interface/bundle mad1/fr/pvc 100> desc "to lansing"
Chicago-SR/configure/interface/bundle mad1/fr/pvc 100> switch 100 uplink
Chicago-SR/configure/interface/bundle mad1/fr/pvc 100> policing cir
1536000 bc 1536000 be 1536000
Chicago-SR/configure/interface/bundle mad1/fr/pvc 100> shaping cir
1536000 bcmax 1536000 bcmin 1536000 be 1536000
Chicago-SR/configure/interface/bundle mad1/fr/pvc 100> exit 3
```

Configuring bundle lans1

```
Chicago-SR/configure> interface bundle lans1
Chicago-SR/configure/interface/bundle lans1> link ct3 1/1/2-6
Chicago-SR/configure/interface/bundle lans1> encapsulation fr
Chicago-SR/configure/interface/bundle/lans1> fr
Chicago-SR/configure/interface/bundle lans1/fr> intf_type dce
```

Configuring pvc 101

```
Chicago-SR/configure/interface/bundle lans1/fr> pvc 101
Chicago-SR/configure/interface/bundle lans1/fr/pvc 101> desc "to internet
" Chicago-SR/configure/interface/bundle lans1/fr/pvc 101> switch 101
uplink
Chicago-SR/configure/interface/bundle lans1/fr/pvc 101> policing cir
1536000 bc 1536000 be 6144000
Chicago-SR/configure/interface/bundle lans1/fr/pvc 101> shaping cir
1536000 bcmax 1536000 bcmin 1536000 be 6144000
Chicago-SR/configure/interface/bundle lans1/fr/pvc 101> exit
```

Configuring pvc 102

```
Chicago-SR/configure/interface/bundle lans1/fr> pvc 102
Chicago-SR/configure/interface/bundle lans1/fr/pvc 102> desc "to madison
" Chicago-SR/configure/interface/bundle lans1/fr/pvc 102> switch 102
uplink
Chicago-SR/configure/interface/bundle lans1/fr/p[vc 102> policing cir
1536000 bc 1536000 be 6144000
Chicago-SR/configure/interface/bundle lans1/fr/pvc 102> shaping cir
1536000 bcmax 1536000 bcmin 1536000 be 6144000
Chicago-SR/configure/interface/bundle lans1/fr/pvc 102>exit
```

Configuring pvc 103

```
Chicago-SR/configure/interface/bundle lans1/fr> pvc 103
Chicago-SR/configure/interface/bundle lans1/fr/pvc 103> desc "to dayton"
Chicago-SR/configure/interface/bundle lans1/fr/pvc 103> switch 103 uplink
Chicago-SR/configure/interface/bundle lans1/fr/pvc 103> policing cir
3072000 bc 3072000 be 6144000
Chicago-SR/configure/interface/bundle lans1/fr/pvc 103> shaping cir
3072000 bcmax 3072000 bcmin 3072000 be 6144000
Chicago-SR/configure/interface/bundle lans1/fr/pvc 103> exit 3
```

Configuring bundle uplink

```
Chicago-SR/configure> interface bundle uplink
Chicago-SR/configure/interface/bundle uplink> link t3 1/2
Chicago-SR/configure/interface/bundle uplink> encapsulation fr
Chicago-SR/configure/interface/bundle uplink> fr
Chicago-SR/configure/interface/bundle uplink/fr> intf_type nni
```

Configuring bundle uplink pvc 100

```
Chicago-SR/configure/interface/bundle uplink/fr> pvc 100
Chicago-SR/configure/interface/bundle uplink/fr> desc "madison-lansing"
```

```
Chicago-SR/configure/interface/bundle uplink/fr> switch 100 mad1
Chicago-SR/configure/interface/bundle uplink/fr> policing cir 1536000 bc
1536000 be 1536000
Chicago-SR/configure/interface/bundle uplink/fr> shaping cir 1536000
bcmax 1536000 bcmin 1536000 be 1536000
Chicago-SR/configure/interface/bundle uplink/fr> exit
```

Configuring bundle uplink pvc 101

```
Chicago-SR/configure/interface/bundle uplink/fr> pvc 101
Chicago-SR/configure/interface/bundle uplink/fr> desc "lansing-internet"
Chicago-SR/configure/interface/bundle uplink/fr> switch 101 lans1
Chicago-SR/configure/interface/bundle uplink/fr> policing cir 1536000 bc
1536000 be 6144000
Chicago-SR/configure/interface/bundle uplink/fr> shaping cir 1536000
bcmax 1536000 bcmin 1536000 be 6144000
Chicago-SR/configure/interface/bundle uplink/fr> exit
```

Configuring bundle uplink pvc 102

```
Chicago-SR/configure/interface/bundle uplink/fr> pvc 102
Chicago-SR/configure/interface/bundle uplink/fr> desc "lansing-madison"
Chicago-SR/configure/interface/bundle uplink/fr> switch 102 uplink
Chicago-SR/configure/interface/bundle uplink/fr> policing cir 1536000 bc
1536000 be 6144000
Chicago-SR/configure/interface/bundle uplink/fr> shaping cir 1536000
bcmax 1536000 bcmin 1536000 be 6144000
```

Configuring bundle uplink pvc 103

```
Chicago-SR/configure/interface/bundle uplink/fr> pvc 103
Chicago-SR/configure/interface/bundle uplink/fr> desc "lansing-dayton"
Chicago-SR/configure/interface/bundle uplink/fr> switch 103 uplink
Chicago-SR/configure/interface/bundle uplink/fr> policing cir 3072000 bc
3072000 be 6144000
Chicago-SR/configure/interface/bundle uplink/fr> shaping cir 3072000
bcmax 3072000 bcmin 3072000 be 6144000
Chicago-SR/configure/interface/bundle uplink/fr> exit 3
```

Configuring interface ethernet 0/1

```
Chicago-SR/configure> interface ethernet 0/1
Chicago-SR/configure/interface ethernet 0/1> speed 100 full_duplex
Chicago-SR/configure/interface ethernet 0/1> ip address 10.1.1.2 255.255.255.0
Chicago-SR/configure/interface ethernet 0/1> exit
```

Configuring snmp

```
Chicago-SR/configure> snmp
Chicago-SR/configure/snmp> community public ro
Chicago-SR/configure/snmp> system_id chi-Router
Chicago-SR/configure/snmp> trap_host 10.2.1.1 public
Chicago-SR/configure/snmp> exit
```

Configuring IP routes

```
Chicago-SR/configure> ip
Chicago-SR/configure/ip> route 0.0.0.0 0.0.0.0 10.1.1.1 1
Chicago-SR/configure/ip> exit
Chicago-SR/configure>
```

Lansing - Secure Router Configuration

```
SR> configure term
SR/configure> hostname lans1-Router
```

Configuring interface bundle wan1

```
lans1-SR/configure> interface bundle wan1
lans1-SR/configure/interface/bundle wan1> link t1 1/1-4
lans1-SR/configure/interface/bundle wan1> encapsulation fr
lans1-SR/configure/interface/bundle wan1> fr
```

Configuring interface bundle wan1 pvc 101

```
lans1-SR/configure/interface/bundle wan1/fr> pvc 101
lans1-SR/configure/interface/bundle wan1/fr/pvc 101> desc "to internet"
lans1-SR/configure/interface/bundle wan1/fr/pvc 101> ip addr 205.100.1.2
255.255.255.252
lans1-SR/configure/interface/bundle wan1/fr/pvc 101> policing cir 1536000
bc 1536000 be 6144000
lans1-SR/configure/interface/bundle wan1/fr/pvc 101> shaping cir 1536000
bcmax 1536000 bcmin 1536000 be 6144000
lans1-SR/configure/interface/bundle wan1/fr/pvc 101> exit
```

Configuring interface bundle wan1 pvc 102

```
lans1-SR/configure/interface/bundle wan1/fr> pvc 102
lans1-SR/configure/interface/bundle wan1/fr/pvc 102> desc "to madison"
lans1-SR/configure/interface/bundle wan1/fr/pvc 102> ip addr 205.1.1.129
255.255.255.252
lans1-SR/configure/interface/bundle wan1/fr/pvc 102> policing cir 1536000
bc 1536000 be 6144000
lans1-SR/configure/interface/bundle wan1/fr/pvc 102> shaping cir 1536000
vcmax 1536000 bcmin 1536000 be 6144000
lans1-SR/configure/interface/bundle wan1/fr/pvc 102> exit
```

Configuring interface bundle wan1 pvc 103

```
lans1-SR/configure/interface/bundle wan1/fr> pvc 103
lans1-SR/configure/interface/bundle wan1/fr/pvc 103> desc "to dayton"
lans1-SR/configure/interface/bundle wan1/fr/pvc 103> ip addr 205.1.1.133
255.255.255.252
lans1-SR/configure/interface/bundle wan1/fr/pvc 103> policing cir 3072000
bc 3072000 be 6144000
lans1-SR/configure/interface/bundle wan1/fr/pvc 103> shaping cir 3072000
bcmax 3072000 bcmin 3072000 be 6144000
lans1-SR/configure/interface/bundle wan1/fr/pvc 103> exit 3
```

Configuring ethernet 0/1

```
lans1-SR/configure> interface ethernet 0/1
lans1-SR/configure/interface/ethernet 0/1> ip addr 205.1.2.1 255.255.255.0
lans1-SR/configure/interface/ethernet 0/1> exit
```

Configuring IP routing

```
lans1-SR/configure> ip
lans1-SR/configure/ip> routing
lans1-SR/configure/ip> route 205.1.1.0 255.255.255.128 205.1.1.130 1
lans1-SR/configure/ip> route 205.1.3.0 255.255.255.0 205.1.1.134 1
lans1-SR/configure/ip> route 0.0.0.0 0.0.0.0 205.100.1.1 1
lans1-SR/configure/ip> exit
```

Columbus - Secure Router Configuration

```
SR> configure term
SR/configure> hostname Columbus-Router
Columbus-SR/configure> interface bundle dayt1
Columbus-SR/configure/interface/bundle dayt1> link ct3 1/1/1-3
```

```
Columbus-SR/configure/interface/bundle dayt1> encapsulation fr
Columbus-SR/configure/interface/bundle dayt1> fr
Columbus-SR/configure/interface/bundle dayt1/fr> intf_type dce
```

Configuring interface bundle dayt1 pvc 104

```
Columbus-SR/configure/interface/bundle dayt1/fr> pvc 104
Columbus-SR/configure/interface/bundle dayt1/fr/pvc 104> desc "to
internet
" Columbus-SR/configure/interface/bundle dayt1/fr/pvc 104> switch 104
uplink
Columbus-SR/configure/interface/bundle dayt1/fr/pvc 104> policing cir
1536000 bc 1536000 be 4608000
Columbus-SR/configure/interface/bundle dayt1/fr/pvc 104> shaping
cir
1536000 bcmax 1536000 bcmin 1536000 be 4608000
Columbus-SR/configure/interface/bundle dayt1/fr/pvc 104> exit
```

Configuring interface bundle dayt1 pvc 105

```
Columbus-SR/configure/interface/bundle dayt1/fr> pvc 105
Columbus-SR/configure/interface/bundle dayt1/fr/pvc 105> desc "to
lansing
" Columbus-SR/configure/interface/bundle dayt1/fr/pvc 105> switch 105
uplink
Columbus-SR/configure/interface/bundle dayt1/fr/pvc 105> policing cir
3072000 bc 3072000 be 4608000
Columbus-SR/configure/interface/bundle dayt1/fr/pvc 105> shaping cir
3072000 bcmax 3072000 bcmin 3072000 be 4608000
Columbus-SR/configure/interface/bundle dayt1/fr/pvc 105> exit 3
```

Configuring interface bundle uplink

```
Columbus-SR/configure> interface bundle uplink
Columbus-SR/configure/interface bundle uplink> link t3 1/1
Columbus-SR/configure/interface bundle uplink> encapsulation fr
Columbus-SR/configure/interface bundle uplink> fr
Columbus-SR/configure/interface bundle uplink/fr> intf_type nni
```

Configuring interface bundle uplink pvc 104

```
Columbus-SR/configure/interface bundle uplink/fr> pvc 104
Columbus-SR/configure/interface bundle uplink/fr/pvc 104> desc "to
internet
" Columbus-SR/configure/interface bundle uplink/fr/pvc 104> switch 104
dayt1
Columbus-SR/configure/interface bundle uplink/fr/pvc 104> policing cir
1536000 bc 1536000 be 4608000
Columbus-SR/configure/interface bundle uplink/fr/pvc 104> shaping cir
```

```
1536000 bcmax 1536000 bcmin 1536000 be 4608000
Columbus-SR/configure/interface bundle uplink/fr/pvc 104> exit
```

Configuring interface bundle uplink pvc 105

```
Columbus-SR/configure/interface bundle uplink/fr> pvc 105
Columbus-SR/configure/interface bundle uplink/fr/pvc 105> desc "dayton-lansing"
Columbus-SR/configure/interface bundle uplink/fr/pvc 105> switch 105
dayt1
Columbus-SR/configure/interface bundle uplink/fr/pvc 105> policing cir
3072000 bc 3072000 be 4608000
Columbus-SR/configure/interface bundle uplink/fr/pvc 105> shaping cir 3072000 bcmax
3072000 bcmin 3072000 be 4608000
Columbus-SR/configure/interface bundle uplink/fr/pvc 105> exit 3
```

Configuring interface ethernet 0/2

```
Columbus-SR/configure> interface ethernet 0/2
Columbus-SR/configure/interface/bundle ethernet 0/2> speed 100 full_duplex
Columbus-SR/configure/interface/bundle ethernet 0/2> ip address 10.1.2.2
255.255.255.0
Columbus-SR/configure/interface/bundle ethernet 0/2 exit
```

Configuring snmp

```
Columbus-SR/configure> snmp
Columbus-SR/configure/snmp> community public ro
Columbus-SR/configure/snmp> system_id col-Router
Columbus-SR/configure/snmp> trap_host 10.2.1.1 public
Columbus-SR/configure/snmp> exit
```

Configuring IP routes

```
Columbus-SR/configure> ip
Columbus-SR/configure/ip> route 0.0.0.0 0.0.0.0 10.1.2.1 1
Columbus-SR/configure/ip> exit
Columbus-SR/configure>
```

Dayton- Secure Router Configuration

```
SR> configure term
SR/configure> hostname lans1-Router
```

Configuring interface bundle wan1

```
lans1-SR/configure> interface bundle wan1
lans1-SR/configure/interface/bundle wan1> link t1 1-3
lans1-SR/configure/interface/bundle wan1> encapsulation fr
lans1-SR/configure/interface/bundle wan1> fr
```

Configuring IP routing

```
lans1-SR/configure> ip
lans1-SR/configure/ip> routing
lans1-SR/configure/ip> route 205.1.2.0 255.255.255.0 205.1.1.133 1
lans1-SR/configure/ip> route 0.0.0.0 0.0.0.0 205.100.1.5 1
lans1-SR/configure/ip> exit
lans1-SR/configure>
```

Configuring bundle wan1 pvc 104

```
lans1-SR/configure/interface/bundle wan1/fr> pvc 104
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> desc "to internet"
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> ip addr 205.100.1.6
255.255.255.252
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> policing cir 153600
bc 153600 be 4608000
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> shaping cir 1536000
bcmax 1536000 bcmin 1536000 be 4608000
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> exit
```

Configuring bundle wan1 pvc 105

```
lans1-SR/configure/interface/bundle wan1/fr> pvc 105
lans1-SR/configure/interface/bundle wan1/fr/pvc 105> desc "to lansing"
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> ip addr 205.1.1.134
255.255.255.252
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> policing cir 3072000
bc 3072000 be 4608000
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> shaping cir 3072000
bcmax 3072000 bcmin 3072000 be 4608000
lans1-SR/configure/interface/bundle wan1/fr/pvc 104> exit 3
```

Configuring interface ethernet 0/1

```
lans1-SR/configure> interface ethernet 0/1
lans1-SR/configure/interface ethernet 0/1> ip addr 205.1.3.1 255.255.255.0
lans1-SR/configure/interface ethernet 0/1> exit
```

Configuring FRF.12

FRF.12

FRF.12 is a backward compatible, lightweight fragmentation protocol and is configurable on one or more PVCs of a Frame Relay bundle. With FRF.12 Frame Relay DTEs and DCEs fragment long frames into sequences of shorter frames. Upon arrival, the fragments are reassembled into the original frame by the receiving peer DTE or DCE.

When used in combination with interleaving, serialization delay of voice packets over low speed frame links can be reduced resulting in performance enhancements. Interleaving is configurable on a Frame Relay bundle basis.

FRF.12 can be helpful when a single VC in a branch office location is needed to transport both voice and data over a FR WAN or for separate VCs in a FR interface used in carrying voice or data.

FRF.12 design considerations:

- On AVCs (FRF.15 End-to-end MFR), the FRF.12 feature is not supported.
- Oversubscription of bandwidth on the Frame Relay bundle is not allowed when interleaving is turned on.
- Class Based Queuing (CBQ) over Frame Relay feature is not recommended on Frame Relay bundles with interleaving enabled.
- 0 –CIR PVCs can not be supported on the FR bundles with interleaving enabled.
- FRF.12 fragmentation is not currently supported on the bridged PVCs.

Note:
It is not recommended to configure classed-based queueing (CBQ) and Frame relay interleaving (FRF.12) simultaneously on the same interface.

DTE-DCE FRF.12 where DCE terminates the traffic

Configure DTE-1:

Example

```
Configure a bundle:
DTE-1/configure> interface bundle todce1
DTE-1/configure/interface todce1> link t1 1:1-2
DTE-1/configure/interface todce1> encap fr
DTE-1/configure/interface todce1> fr
```

```
Configure PVC 100 and provision FRF.12 fragmentation on it:
DTE-1/configure/interface todce1/fr> pvc 100
DTE-1/configure/interface todce1/fr/pvc 100> ip addr 2.2.2.2 24
DTE-1/configure/interface todce1/fr/pvc 100> frf12 framesize 128
DTE-1/configure/interface todce1/fr/pvc 100> exit
Configure interleaving on the bundle:
DTE-1/configure/interface todce1/fr> interleave
DTE-1/configure/interface todce1/fr/interleave> enable
DTE-1/configure/interface todce1/fr/interleave> hiprio crpcnt 50 brpcnt
100
DTE-1/configure/interface todce1/fr/interleave> exit 3
```

Configure DCE-1 to provision PVC 100 to terminate FR traffic:

Example

```
Configure a bundle:
DCE-1/configure> interface bundle todte1
DCE-1/configure/interface todte1> link t1 1:1-2
DCE-1/configure/interface todte1> encap fr
DCE-1/configure/interface todte1> fr
DCE-1/configure/interface todte1/fr> intf_type dce
Configure the PVC 100:
DCE-1/configure/interface todte1/fr> pvc 100
DTE-1/configure/interface todte1/fr/pvc 100> ip addr 2.2.2.3 24
DTE-1/configure/interface todte1/fr/pvc 100> frf12
DCE-1/configure/interface todte1/fr/pvc 100> exit 3
```

DTE-DTE FRF.12 with an FR cloud in the middle

Configure DTE-1:

Example

```
Configure a bundle:
DTE-1/configure> interface bundle todce1
DTE-1/configure/interface todce1> link t1 1:1-2
DTE-1/configure/interface todce1> encap fr
DTE-1/configure/interface todce1> fr
Configure PVC 100 and provision FRF.12 fragmentation on it:
DTE-1/configure/interface todce1/fr> pvc 100
DTE-1/configure/interface todce1/fr/pvc 100> ip addr 2.2.2.2 24
DTE-1/configure/interface todce1/fr/pvc 100> frf12 framesize 128
DTE-1/configure/interface todce1/fr/pvc 100> exit
Configure interleaving on the bundle:
DTE-1/configure/interface todce1/fr> interleave
DTE-1/configure/interface todce1/fr/interleave> enable
DTE-1/configure/interface todce1/fr/interleave> hiprio crpcnt 50 brpcnt
100
DTE-1/configure/interface todce1/fr/interleave> exit 3
```

Configure DTE-2:

Example

```
Configure a bundle:
DTE-1/configure> interface bundle todce2
DTE-1/configure/interface todce2> link t1 1:1-2
DTE-1/configure/interface todce2> encap fr
```

```
DTE-1/configure/interface todce2> fr
Configure PVC 100 and provision FRF.12 fragmentation on it:
DTE-1/configure/interface todce2/fr> pvc 200
DTE-1/configure/interface todce2/fr/pvc 200> ip addr 2.2.2.2 24
DTE-1/configure/interface todce2/fr/pvc 200> frf12 framesize 128
DTE-1/configure/interface todce2/fr/pvc 200> exit 4
```

Configure DCE-1 to provision PVC 100 for switching:

Example

```
Configure a bundle:
DCE-1/configure> interface bundle todte1
DCE-1/configure/interface todte1> link t1 1:1-2
DCE-1/configure/interface todte1> encap fr
DCE-1/configure/interface todte1> fr
DCE-1/configure/interface todte1/fr> intf_type dce
Configure the switched PVC 100:
DCE-1/configure/interface todte1/fr> pvc 100
DCE-1/configure/interface todte1/fr/pvc 100> switch nnibundle:200
DCE-1/configure/interface todte1/fr/pvc 100> exit 3
```

Configure PVC 200 on an NNI interface used for switching traffic between PVC 100 on todte1 interface:

Example

```
Configure a bundle: D
CE-1/configure> interface bundle nnibundl
DCE-1/configure/interface nnibundle> link t1 1:1-2
DCE-1/configure/interface nnibundle> encap fr
DCE-1/configure/interface nnibundle> fr
DCE-1/configure/interface nnibundle/fr> intf_type nni
Configure the switched PVC 200: D
CE-1/configure/interface nnibundle/fr> pvc 200
DCE-1/configure/interface nnibundle/fr/pvc 200>switch todte1:100
DCE-1/configure/interface nnibundle/fr/pvc 100>exit 3
```

Chapter 30: OSPF Routing Protocol - Frame Relay

The following example shows OSPF running between a Secure Router 1000 Series and a router over a serial T1 link with back-to-back Frame Relay.

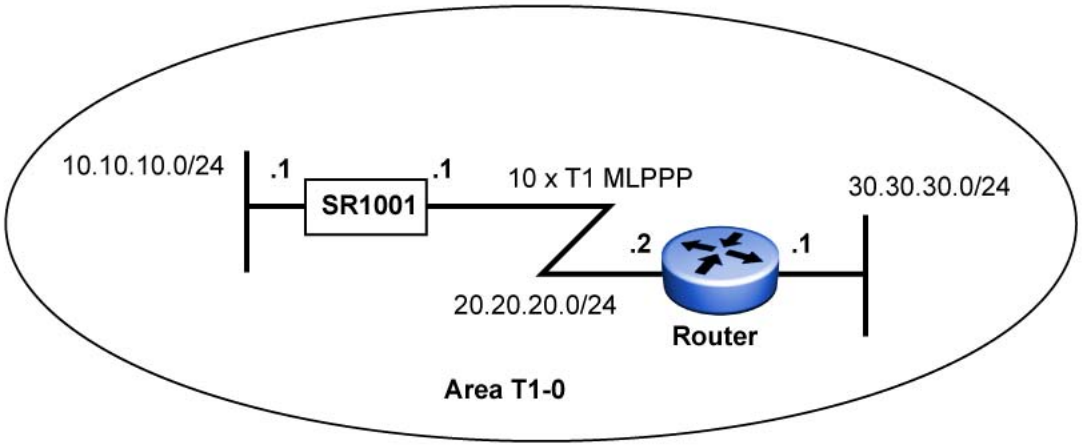


Figure 30: OSPF Over a Single T1 with Frame Relay

Configuring the host name

```
SR> configure terminal

SR/configure> hostname SR
```

Configuring interface ethernet 0

```
SR/configure> interface ethernet 0
SR/configure/interface/ethernet0> ip address 10.10.10.1
24
SR/configure/interface/ethernet0> exit
```

Configuring interface bundle Dallas

```
SR/configure> interface bundle Dallas
SR/configure/interface/bundle Dallas> link t1 1
SR/configure/interface/bundle Dallas> encapsulation
frelay
SR/configure/interface/bundle Dallas> fr
SR/configure/interface/bundle Dallas/fr> intf_type dce
SR/configure/interface/bundle Dallas/fr> pvc 16
SR/configure/interface/bundle Dallas/fr/pvc 16> ip
address 20.20.20.1 255.255.255.0
SR/configure/interface/bundle Dallas/fr/pvc 16> exit 3
```

Configuring ospf

```
SR/configure> router routerid 10.10.10.1
SR/configure> router ospf
SR/configure/router/ospf> area 760
SR/configure/router/ospf/area 760> exit
```

Configuring interface Dallas parameters

```
SR/configure/router/ospf> interface Dallas dlci 16
area_id 760
SR/configure/router/ospf/interface Dallas> cost 10
SR/configure/router/ospf/interface Dallas> exit
```

Configuring interface ethernet 0 parameters

```
SR/configure/router/ospf> interface ethernet0 area_id
760
SR/configure/router/ospf/interface ethernet0> cost 10
SR/configure/router/ospf/interface ethernet0> exit 3
```

Displaying ospf parameters

Execute **`show ip ospf int bundle`** to display interface specific OPSPF parameters.

Chapter 31: PIM Quick Configuration

Protocol Independent Multicast (PIM)

Protocol Independent Multicast (PIM) protocols route multicast packets to multicast groups. PIM is protocol independent because it can leverage whichever unicast routing protocol is used to populate unicast routing table. There are two modes of PIM protocol – Dense mode (DM) and Sparse mode (SM). Avaya supports SM only.

PIM-DM floods multicast traffic throughout the network initially and then generates prune messages as required. PIM-SM attempts to send multicast data only to networks which have active receivers. This is achieved by having a common Rendezvous Point (RP) known to the senders and receivers and by forming shared trees from the RP to the receivers.

PIM-SM is described in RFC 2362.

PIM Commands

The general PIM commands supported in this release are:

Global parameters	
Enable PIM	**Router/configure/ip> pim**
Configure PIM mode	**Router/configure/ip/pim> mode [sparse | dense]**
Configure Assert Holdtime	**Router/configure/ip/pim>assert-holdtime <time>**
Configure Hello Interval	**Router/configure/ip/pim>hello-interval <time**
Configure Hello Holdtime	**Router/configure/ip/pim>hello-holdtime <time>**
Configure Hello priority	**Router/configure/ip/pim>hello-priority <value>**
Configure Join/Prune Holdtime	**Router/configure/ip/pim>join-prune-holdtime <time>**

Configure Join /Prune Interval	**Router/configure/ip/pim>join-prune-interval <time>**
Configure MRT Period	**Router/configure/ip/pim>mrt-period <time>**
Configure MRT Stale Multiplier	**Router/configure/ip/pim>mrt-stale-mult <number>**
Configure MRT SPT Multiplier	**Router/configure/ip/pim>mrt-spt-multiplier <number>**
Configure Probe Period	**Router/configure/ip/pim>probe-period <time>**
Configure Registration suppression timeout	**Router/configure/ip/pim>register-suppress-timeout <time>**
Configure DR to switch immediate	**Router/configure/ip/pim>dr-switch-immediate**
Configure RP to switch immediate	**Router/configure/ip/pim>rp-switch-immediate**
Configure Threshold for DR	**Router/configure/ip/pim>threshold-dr <bps>**
Configure Threshold for RP	**Router/configure/ip/pim>threshold-rp <bps>**
Configure to calculate whole packet checksum (for Cisco Systems interoperability)	**Router/configure/ip/pim>whole-packet-checksum**
Bootstrap Router related Commands	
Configure as candidate BSR	**Router/configure/ip/pim/cbsr> address <A.B.C.D>**
Configure CBSR period	**Router/configure/ip/pim/cbsr> period <time>**
Configure CBSR holdtime	**Router/configure/ip/pim/ cbsr>holdtime <time>**
Configure CBSR priority	**Router/configure/ip/pim/ cbsr>priority <value>**
RP commands	
Configure as candidate RP	**Router/configure/ip/pim>crp**
Configure as candidate RP address	**Router/configure/ip/pim/crp> address <A.B.C.D>**

Configure candidate RP group for advertisement	**Router/configure/ip/pim/crp> group-add <A.B.C.D> [mask] [priority]**
Configure as candidate RP holdtime	**Router/configure/ip/pim/ crp>holdtime <time>**
Configure as candidate RP period	**Router/configure/ip/pim/crp>period <time>**
Configure as candidate RP priority	**Router/configure/ip/pim/ crp>priority <value>**
Configure a static RP address	**Router/configure/ip/pim/> rp <A.B.C.D> <gaddress> [mask]**
Inteface based parameters	
Configure PIM for an interface	**Router/configure/ip/pim>interface <interface_name>[:dlci_no]**
Configure PIM mode for an interface	**Router/configure/ip/pim/interface wan1> mode [sparse | dense | ssm | sparse-ssm]**
Configure PIM interface assert holdtime	**Router/configure/ip/pim/interface wan1>assert-holdtime <time>**
Configure PIM interface hello holdtime	**Router/configure/ip/pim/interface wan1>hello-holdtime <time>**
Configure PIM interface hello interval	**Router/configure/ip/pim/interface wan1>hello-interval <time>**
Configure PIM interface Join/ Prune Delay Timeout	**Router/configure/ip/pim/interface wan1>join-prune-timeout <time>**
Configure PIM interface Join/ Prune Interval	**Router/configure/ip/pim/interface wan1>join-prune-interval <time>**
Configure PIM interface Join/ Prune holdtime	**Router/configure/ip/pim/interface wan1>join-prune-holdtime <time>**
Configure PIM interface as border of PIM domain	**Router/configure/ip/pim/interface wan1>boundary**
PIM SSM	
Configure the SSM range	**Router/configure/ip/pim> ssm-range <group-address> <group-mask>**

The show and debug PIM commands are:

Display PIM global configuration	**SR>show ip pim global**

Display PIMC timers	**SR>show ip pim timers**
Display PIM interfaces	**SR>show ip pim interfaces**
Display PIM neighbors	**SR>show ip pim neighbors**
Display PIM Bootstrap info	**SR>show ip pim bsr-info**
Display PIM Candidate RP info	**SR>show ip pim crp-info**
Display PIM statistics	**SR>show ip pim statistics**
Display PIM RP set	**SR>show ip pim rp-set**
Display PIM Static RP	**SR>show ip pim rp**
Trace PIM packets	**SR> debug ip pim packet <pkt_type> <direction> [interface_name] [dlci]**
Trace PIM state changes	**SR> debug ip pim state**
Trace PIM routes	**SR> debug ip pim route**
Trace PIM detail	**SR> debug ip pim detail**
Trace PIM debug	**SR> debug ip pim debug**
All Traces	**SR>debug ip pim all**

PIM Configuration Examples

This section shows examples of how the PIM commands are used.

To access PIM mode, enter:

```
Router/configure/ip> pim

Router/configure/ip/pim>
```

The following example enters the BSR mode.

```
Router/configure/ip/pim> cbsr

Router/configure/ip/pim/cbsr>
```

The following command sets Ethernet1 as the BSR interface.

```
Router/configure/ip/pim/cbsr> interface ethernet1
```

The following example sets the holdtime to 33 seconds.

```
Router/configure/ip/pim/cbsr> holdtime 33
```

```
Router/configure/ip/pim/cbsr>
```

To configure the DLCI for Ethernet0 to 100, enter:

```
Router/configure/ip/pim/cbsr> interface ethernet0 dlci 100
```

To set the CBSR priority to 45, enter:

```
Router/configure/ip/pim/cbsr> priority 45
```

To enter the candidate Rendezvous Point mode, enter:

```
Router/configure/ip/pim> crp

Router/configure/ip/pim/crp>
```

To set the group IP address for CRP advertisements to 224.1.1.0, enter:

```
Router/configure/ip/pim/crp> group-add 224.1.1.0
```

To set the flag at the DR to switch to the SPT on receiving the first packet (default on), enter:

```
Router/configure/ip/pim> dr-switch-immediate
```

The following example configures the MRT SPT Mult value to be 25.

```
Router/configure/ip/pim> mrt-spt-mult 25
```

The following example configures the probe period to 30 seconds.

```
Router/configure/ip/pim> probe-period 30

Router/configure/ip/pim>
```

The following example configures the Register Suppression Timeout to be 70 seconds.

```
Router/configure/ip/pim> register-suppress-timeout 70
```

To set the RP static IP address to 10.10.1.1, enter:

```
Router/configure/ip/pim> rp 10.10.1.1
```

To set the flag for the RP to switch to the SPT for (S,G) upon receipt of the first Register message (default: on). To turn on this feature, enter:

```
Router/configure/ip/pim> rp-switch-immediate
```

The following example configures this feature.

```
Router/configure/ip/pim> rp-switch-immediate
```

To configure the router such that the data from S addressed to G must exceed an average of 1024 KBytes per second before an SPT switch is initiated, enter:

```
Router/configure/ip/pim> threshold-dr 1024
```

To configure the threshold-dr option such that the data from S addressed to G must exceed an average of 1500 KBytes per second before an SPT switch is initiated. If this router is a DR for the pair (S,G), then the same data must exceed an average of 1500 KBytes per second before an SPT switch is initiated. The period over which the average will be calculated will be the **mrt-period** times the **mrt-spt-mult**, or 60 seconds.

```
Router/configure/ip/pim> threshold-rp 1500
```

To specify that the message checksum will be calculated over the entire encapsulated packet, rather than just over the Register message header, enter:

```
Router/configure/ip/pim> whole-packet-checksum
```

The following example configures a global assert-holdtime value of 600.

```
Router/configure/ip/pim> assert-holdtime 600
```

To set the holdtime to 60 seconds, enter:

```
Router/configure/ip/pim> hello-holdtime 60
```

To set the hello interval time to 145 seconds, enter:

```
Router/configure/ip/pim> hello-interval 145
```

To set the priority to 15, enter:

```
Router/configure/ip/pim> hello-priority 15
```

To set the holdtime to 30 seconds, enter:

```
Router/configure/ip/pim> join-prune-holdtime 30
```

To send messages every five minutes, enter:

```
Router/configure/ip/pim> join-prune-interval 300
```

To check the router table every 15 seconds, enter:

```
Router/configure/ip/pim> mrt-period 15
```

To set the mrt-spt-mult value to be ten times that of the mrt-period value, enter:

```
Router/configure/ip/pim> mrt-spt-mult 10
```

To set the time out (S, G) entries at 5 times the mrt-period value, enter:

```
Router/configure/ip/pim> mrt-spt-mult 5
```

To display PIM global configuration settings, enter:

```
Router/configure> show ip pim global
PIM: Enabled
Mode: Sparse
Timers:
Hello Interval: 145
Hello Hold Time: 60
Hello Priority: 15
```

```
Join/Prune Interval: 300
Join/Prune Hold Time: 30
Assert Hold Time: 200
Probe Period: 15
Register Suppress Timeout: 90
MRT Interval: 15
MRT SPT Multiplier : 10
MRT Stale Multiplier: 5
Thresholds:
Threshold DR: 2400
Threshold RP: 1500
RP Switch Immediate: enabled
DR Switch Immediate: enabled
Whole packet checksum: enabled
SSM Range: 224.20.12.1 24
Router/configure>
```

To display information for all interfaces, enter:

```
Router/configure> show ip pim interface all
```

To see all IP PIM interface information for Ethernet1, enter:

```
 Router/configure/ip/pim/interface ethernet1> show ip pim interface
ethernet1
```

To display IP PIM statistics for ethernet1, enter:

```
Router/configure/ip/pim/interface ethernet1> show ip pim statistics

PIM Statistics:

Total PIM msgs recvd 0 (0 bytes)
Recvd msgs too short 0
Recvd msgs bad checksum 0
Recvd msgsg bad version 0
Recvd register msgs 0 (0 bytes)
Recvd registers wrong iif 0
Recvd bad registers 0
Sent register msgs 0 (0 bytes)

 Router/configure/ip/pim/interface ethernet1>
```

To display information on PIM neighbors, enter:

```
Router/configure> show ip pim neighbors
Neighbor Interface Uptime Expires Hello Priority
--------------- ------------ -------- ----------------
--------------
Router/configure>
```

To display RP information, enter:

```
Router/configure> show ip pim rp
Group/Mask RP
------------------ ---------------
224.0.0.0/4 10.10.1.1
Router/configure>
```

To view RP-set information, enter:

```
Router/configure> show ip pim rp-set
Group/mask Src/RP Pri Uptime Expires
---------- ------ --- ------ -------
224/4 10.10.1.1 1 Static RP
Dependencies: None
Router/configure>
```

To view PIM counters, enter:

```
Router/configure> show ip pim statistics
PIM Statistics:
Total PIM msgs recvd 0 (0 bytes)
Recvd msgs too short 0
Recvd msgs bad checksum 0
Recvd msgsg bad version 0
Recvd register msgs 0 (0 bytes)
Recvd registers wrong iif 0
Recvd bad registers 0
Sent register msgs 0 (0 bytes)
Router/configure>
```

To display PIM timer information, enter:

```
Router/configure> show ip pim timers
PIM Timers:
Hello Interval: 145
Hello Hold Time: 60
Hello Priority: 15
```

```
Join/Prune Interval: 300
Join/Prune Hold Time: 30
Assert Hold Time: 200
Probe Period: 15
Register Suppress Timeout: 90
MRT Interval: 15
MRT SPT Multiplier : 10
MRT Stale Multiplier: 5
Router/configure>
```

To examine PIM BSR statistics, enter:

```
Router/configure/ip/pim> show ip pim bsr-info
Candidate BSR Information
-------------------------
Candidate BSR Status: Disabled
Candidate BSR Interface: NOT CONFIGURED
Candidate BSR Priority: 45
Candidate BSR Period: 30
Candidate BSR Hold Time: 2048
Candidate BSR Admin Scope: Disabled
No BSR’s
Router/configure/ip/pim>
```

To reset PIM counters, enter:

```
SR> clear ip pim statistics
```

Chapter 32: OSPF Routing Protocol

The following example shows a SR1001 connected to a router over a single T1 link. IP addresses 10.10.10.0, 20.20.20.0, and 30.30.30.0 are assigned to area 760.

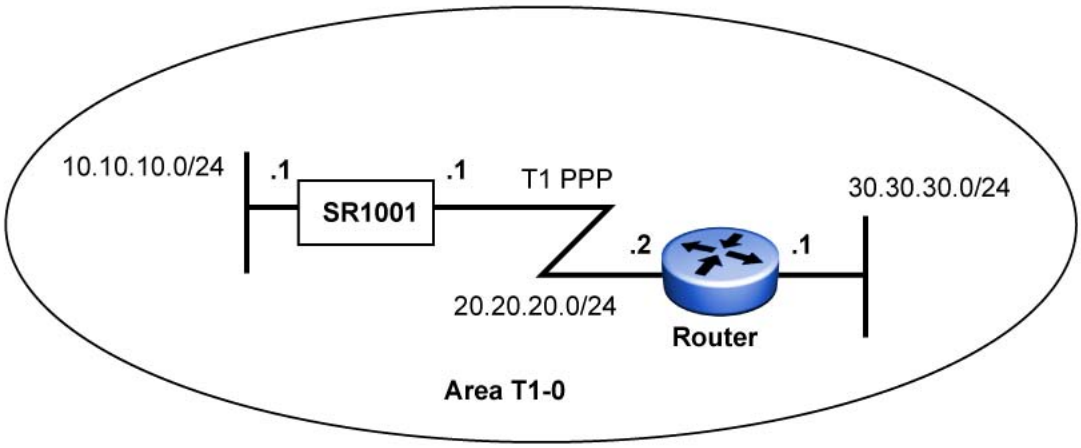


Figure 31: Configuring OSPF Between a SR1001 and a Router

Configuring the host name

```
SR>configure terminal

SR/configure/hostname SR
```

Configuring interface ethernet 0

```
SR/configure> interface ethernet 0
SR/configure/interface/ethernet 0> ip address 10.10.10.1 24
SR/configure/interface/ethernet 0> exit
```

Configuring interface bundle Dallas

```
SR/configure> interface bundle Dallas
SR/configure/interface/bundle Dallas> link t1 1
SR/configure/interface/bundle Dallas> encapsulation ppp
SR/configure/interface/bundle Dallas> ip address
20.20.20.1 24
SR/configure/interface/bundle Dallas> exit
```

Configuring ospf

```
SR/configure> router routerid 10.10.10.1
SR/configure> router ospf
SR/configure/router/ospf> area 760
SR/configure/router/ospf/area 760> exit
```

Configuring ospf interface parameters

```
SR/configure/router/ospf> interface Dallas area_id 760
SR/configure/router/ospf/interface Dallas> exit
SR/configure/router/ospf> interface ethernet0 area_id
760
SR/configure/router/ospf/interface ethernet0> exit 3
```

Displaying neighbors

Note that "display" and "show" can be used interchangeably in the CLI tree hierarchy.

Execute **`show ip ospf neighbor list`** on the SR1001 to display the neighbor information. In this example, the state is in FULL adjacency with the router.

```
SR> show ip ospf neighbor list

Neighbor ID   PRI  State     Dead Time  Address       Interface
-----------   ---  -----     ---------  -------       ---------
30.30.30.1     1    FULL/ -   00:00:30  20.20.20.2    TMan1
SR>
```

Figure 32: show ip ospf neighbor list Command

Displaying ospf routes

Execute **show ip routes ospf** on the SR1001 to display the OSPF routes learned from neighbors. The following display shows the route 30.30.30.0/24, which was learned through OSPF from the router advertisements.

```
SR> show ip ospf routes

OSFP ROUTE TABLE

Codes: A - OSPF intra area   IA - OSPF inter area,
   E1 - OSPF external type 1, E2 - OSPF external type 2

Destination       Gateway     Interface Protocol  Type  Metric  Preference
-----------       -------     --------- --------  ----  ------  ----------
10.10.30.0/24     20.20.20.2  TMan1     OSPF        A      2    10

SR>
```

Figure 33: show ip routes ospf Command

The metric shows a value of 2. By default, Avaya assigns a cost value of 1 to all interfaces. The cost can be changed by entering it under the appropriate interface in the OSPF command tree structure. For example:

```
SR/configure> router ospf
SR/configure/router/ospf> interface Dallas area_id 760
SR/configure/router/ospf/interface/Dallas> cost 10
SR/configure/router/ospf/interface/Dallas> exit 3
```

This would change the cost of bundle link Dallas from default (1) to 10. If the interface is already configured, then entering **area_id 760** is optional.

Displaying IP routes

Execute **show ip routes** to display all the active routes in the routing table.

OSPF NBMA over Ethernet

The Secure Router 1000 Series and 3120 will provide support for OSPF non-broadcast multi-access (NBMA) over Ethernet. While it's well known that OSPF operates in peer-to-peer and broadcast networks, its role in another kind of network can be just as important. A non-broadcast network operates between point-to-point and broadcast networks, and doesn't include broadcast or multicast functionality. Its purpose is to connect more than two devices

to the same physical media device and, by nature, it is multi-access. Some examples of this are Frame Relay networks, ATM networks and x.25 networks.

To achieve this functionality, some components of OSPF have been modified in an attempt to mirror functionality found in OSPF broadcast networks. Two modes of operation on these types of OSPF networks are NBMA and P2MP. When using NBMA, operation over a broadcast network is emulated by OSPF. The NBMA network has a router designated to originate a network LSA. NBMA mode is the most efficient way to run OSPF over non-broadcast networks, both in terms of link-state database size and in terms of the amount of routing protocol traffic.

When deploying OSPF on a network, neighbor discovery is achieved using multicast hello packets. Designated Routers (DR) and Backup Designated Routers (BDR) are elected for each multicast network in order to optimize adjacency building. All routers in a segment should communicate directly with a DR or BDR for proper adjacency. For a neighbor to be successfully discovered on a segment, broadcast and multicast packet sending must be allowed on the network.

When using NBMA technology, neighbors are not discovered automatically due to the non-broadcast nature of the feature. Instead, OSPF attempts to designate a DR and a BDR, but the election fails since no neighbors are discovered. In order to overcome this issue, neighbors must be manually configured.

Broadcast vs non-broadcast networks

One difference between broadcast and non-broadcast networks is in the functionality of the hello protocol. On a broadcast network, a router advertises itself using hello packets allowing itself to be discovered dynamically. These packets include the router's DR identity and a list of routers who have recently send Hello packets. On NBMA networks, some configuration must take place before successful operation of the hello protocol. Routers that are potential DRs have a list of all other routers currently attached. If a DR candidate, a router sends Hello packets to other candidates in an attempt to find a DR. If elected DR, a router sends hello packets to all other routers on the network. To minimize the number of hello packets sent, the number of eligible routers on a NBMA network should be kept to a minimum.

The behavior of any router's hello packet sending depends on its status as potential DR. If eligible, it must send hello packets to eligible neighbors periodically. If the router becomes the DR or BDR, it expands distribution of hello packets to include all neighbors, regardless of eligibility. If a router is not eligible, it must send hello packets to the DR and BDR periodically, along with sending a reply hello packet to any hello packet received from an eligible neighbor. Frequency of hello packets depends on a neighbor's state. When down, hello packets are sent at Poll Interval, otherwise they are sent at Hello Interval.

Another difference comes when identifying a neighbor address. In a point-to-point network or virtual link, the neighbor is identified by router ID. However, in a broadcast, point-to-multipoint or NBMA network, the neighbor is identified by IP source address.

Finally, in an OSPF operation specific to NBMA, OSPF generates a start event to a neighbor when the neighbor command is issued. When this occurs, hello packets begin to be sent to a neighbor using the Hello Interval as a frequency. This causes the neighbor to receive an ATTEMPT message that indicates no recent information has been received from the neighbor and that a greater effort is to be to contact that neighbor. To achieve this, up to four hello packets

are sent to the neighbor. If no response is received, a DOWN state is entered, where packet frequency is reduced to that of the Poll Interval.

Configuring OSPF NBMA over Ethernet

Use the following procedure to configure OSPF NBMA over Ethernet. There are 3 main components to configuring OSPF NBMA. First, you specify the interface network type. This is followed by specifying neighbors and a poll interval.

Procedure steps

1. To configure OSPF NBMA, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Specify a router ID for OSPF.

   ```
   router router-id <X.X.X.X>
   ```

3. Enable OSPF.

   ```
   router ospf
   ```

4. Configure the OSPF area.

   ```
   interface <interface> area <areaid>
   ```

5. Specify the network type.

   ```
   network <type>
   ```

6. Configure neighbors, repeating this step for each neighbor you want to add.

   ```
   neighbor <A.B.C.D>
   ```

7. Configure the poll interval.

   ```
   poll_interval <interval>
   ```

Table 35: Variable definitions

Variable	Value
<A.B.C.D>	The IP address.
<areaid>	The OSPF area ID.
<interface>	The interface to work with.
<interval>	The poll interval.
<type>	The network type.
<X.X.X.X>	The router ID IP address.

Chapter 33: QOS Configuration

Overview

Avaya QoS ensures bandwidth guarantees throughout the Secure Router by implementing Random Early Detection (RED) to address congestion and Class Based Queuing (CBQ) to address traffic policing. This document discusses the CBQ features.

Avaya's bandwidth management capability allows multiple agencies or customers to share access bandwidth on a WAN link in a controlled fashion to effectively and efficiently utilize available bandwidth. Even during times of congestion, each customer is guaranteed a share of the access bandwidth and is allowed to borrow unused bandwidth from other customers. This bandwidth management capability allows service providers to offer their customers Internet access based on the amount of guaranteed bandwidth-committed rate (CR) and the amount of bandwidth-borrowed burst rate (BR). Similarly, an organization can share its access bandwidth among its different departments.

Features

The network administrator manages bundle bandwidth across various customers by defining traffic classes. Each traffic class is assigned the desired committed bandwidth as well as the burst bandwidth. The sum of the CRs of all classes must be less than or equal to the total bundle bandwidth. CBQ can be deployed only in the WAN outbound direction.

A traffic class is characterized by the following parameters:

- Class name
- Parent class
- Committed rate (CR)
- Burst rate (BR)
- Classification type based on:
    - Application level

        Application ports (TCP or UDP)

- Network level

  Source or destination IP addresses, address ranges, or subnets

- Ethernet MAC level

  • VLAN identifiers

  • dot1q

Traffic classes are arranged in a hierarchical manner. A class has a parent class and can have one or more child classes. The root class has no parent and is identified as root-out or root-in. There is no theoretical limit to the number of classes that can be created. The only limitation that can arise is due to available memory in the Secure Router.

Definitions

• Committed Rate

Each traffic class can be assigned a CR parameter in Kbps or as a percent of link bandwidth. This is the amount of bandwidth that the class or flow is guaranteed at all times, even during congestion. The sum of the CRs for all classes in a given direction cannot exceed the access bandwidth of their parent class. By maintaining a moving average of the bandwidth for each class, the class is not "strictly" policed at CR Kbps, and momentary bursts in the flow are permitted. The goal is that each class with sufficient demand will be able to receive roughly its allocated bandwidth over some interval of time.

• Burst Rate

Every traffic class can be configured with a burst parameter, which is the bandwidth that can be offered to that class if unused bandwidth is available from other classes. This provides for very efficient bandwidth utilization. A class can also be configured to borrow whatever bandwidth is available from its parent class up to the BR limit set for that class. To prevent a class from borrowing, set the CR equal to the BR. Also, note that a class cannot borrow more than the bundle bandwidth.

Classification Types

The example in Figure 1 reserves the largest CR (1536 Kbps) for two servers, 10.1.1.1 and 10.1.1.2, which are members of the SrcOne class. The remainder of the 10.1.1.0/24 subnet is assigned to the SrcTwo class and is configured with a CR of 1024 Kbps. Additionally, the SrcTwo class is further divided into application port classes. All other hosts in Figure 1, the default class, are configured for a CR of 512 Kbps.

The classification type must be the same across a given level of traffic class. Note in Figure 1, that the classification type at the first level traffic class is the source IP address; for the second

level, the classification type is the application port. Because bandwidth limitations are evaluated from most specific to least specific, 10.1.1.1/32 falls within the SrcOne class.

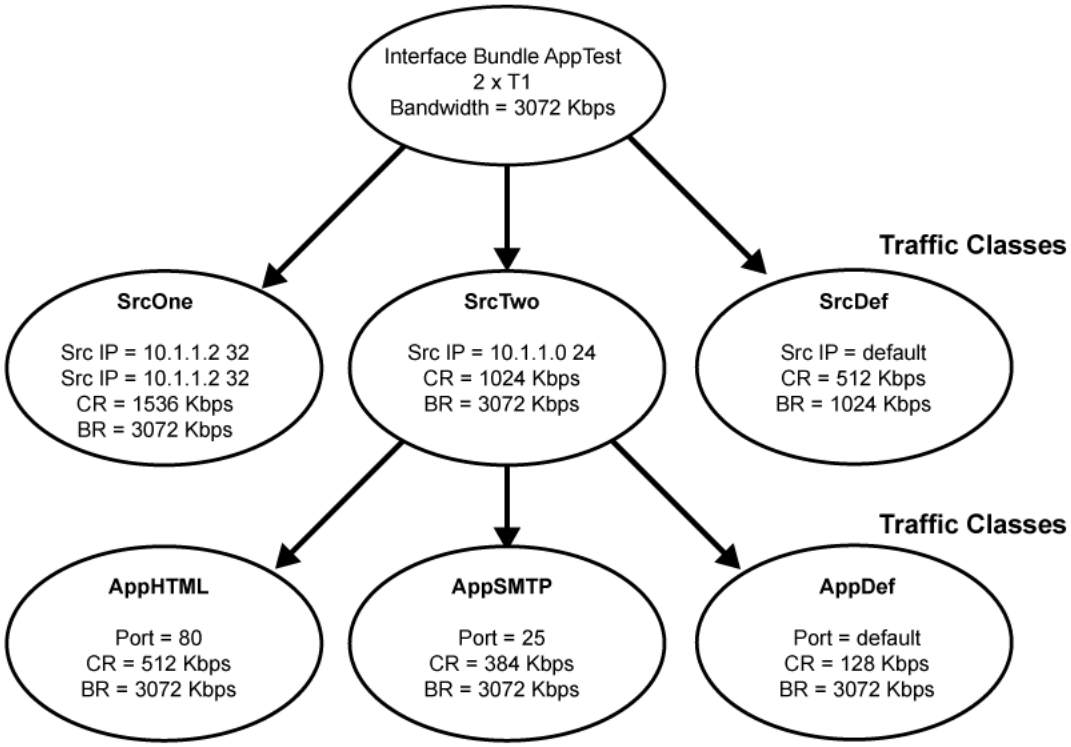


Figure 34: Assigning Classification Types

Configuration for the example in Figure 1

Create bundle AppTest

```
SR/configure> interface bundle AppTest
SR/configure/interface/bundle AppTest> link t1 1
SR/configure/interface/bundle AppTest> encap ppp
SR/configure/interface/bundle AppTest> ip addr
199.1.1.1 255.255.255.252
```

Create traffic classes

```
SR/configure/interface/bundle AppTest> qos
SR/configure/interface/bundle AppTest/qos> add_class
SrcOne root-out cr 1536 br 3072
SR/configure/interface/bundle AppTest/qos> add_class
SrcTwo root-out cr 1024 br 3072
SR/configure/interface/bundle AppTest/qos>add_class
SrcDef root-out cr 512 br 1024
SR/configure/interface/bundle AppTest/qos> add_class
AppHTML SrcTwo cr 512 br 3072
SR/configure/interface/bundle AppTest/qos> add_class
AppSMTP SrcTwo cr 384 br 3072
SR/configure/interface/bundle AppTest/qos> add_class
AppDef SrcTwo cr 128 br 3072
```

Assign classification types

```
SR/configure/interface/bundle AppTest/qos> class SrcOne
SR/configure/interface/bundle AppTest/qos/class SrcOne> add_src_ip 10.1.1.1
255.255.255.255
SR/configure/interface/bundle AppTest/qos/class SrcOne> add_src_ip 10.1.1.2
255.255.255.255
SR/configure/interface/bundle AppTest/qos/class SrcOne> exit
SR/configure/interface/bundle AppTest/qos> class SrcTwo
SR/configure/interface/bundle AppTest/qos/class SrcTwo> add_src_ip 10.1.1.0
255.255.255.0
SR/configure/interface/bundle AppTest/qos/class SrcTwo> exit
SR/configure/interface/bundle AppTest/qos> class SrcDef
SR/configure/interface/bundle AppTest/qos/class SrcDef> add_src_ip default
SR/configure/interface/bundle AppTest/qos/class SrcDef> exit
SR/configure/interface/bundle AppTest/qos> class AppHTML
SR/configure/interface/bundle AppTest/qos/class AppHTML> add_port 80
SR/configure/interface/bundle AppTest/qos/class AppHTML> exit
SR/configure/interface/bundle AppTest/qos> class AppSMTP
SR/configure/interface/bundle AppTest/qos/class AppSMTP> add_port 25
SR/configure/interface/bundle AppTest/qos/class AppSMTP> exit
SR/configure/interface/bundle AppTest/qos> class AppDef
SR/configure/interface/bundle AppTest/qos/class AppDef> add_port default
SR/configure/interface/bundle AppTest/qos/class AppDef> exit
SR/configure/interface/bundle AppTest/qos> enable
SR/configure/interface/bundle AppTest/qos> exit 3
```

VLAN Identifiers

Figure 2 illustrates the classification based on VLAN identifiers. Note that these classes are leaf classes and do not have child classes.

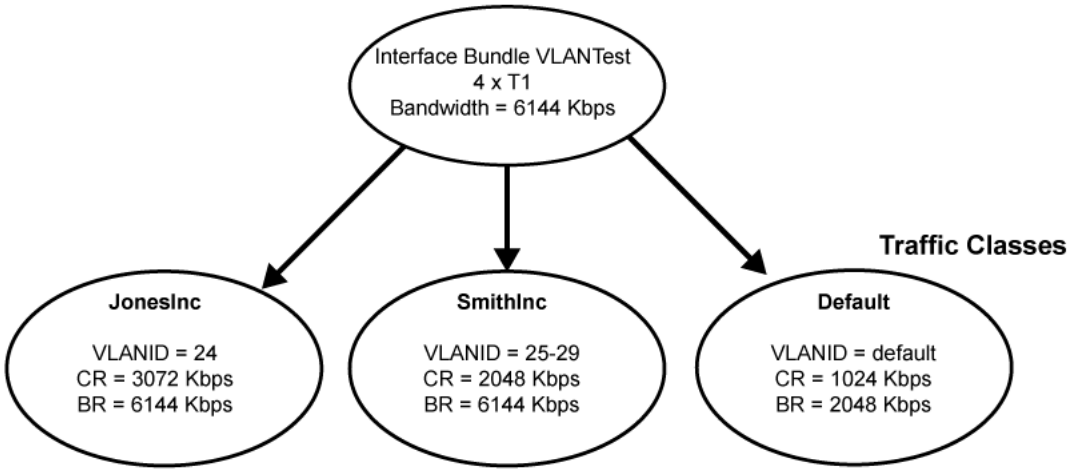


Figure 35: Assigning VLAN Identifiers

Configuration for Figure 2

Create bundle VLANtest

```
SR> conf t
SR/configure> interface bundle VLANtest
SR/configure/interface/bundle VLANtest> link t1 1
SR/configure/interface/bundle VLANtest> encap ppp
SR/configure/interface/bundle VLANtest> ip addr
200.1.1.1 255.255.255.252
```

Create traffic classes and assign classifications

```
SR/configure/interface/bundle VLANtest> qos
SR/configure/interface/bundle VLANtest/qos> add_class JonesInc root-out cr 3072 br
6144
SR/configure/interface/bundle VLANtest/qos> add_class SmithInc root-out cr 2048 br
6144
SR/configure/interface/bundle VLANtest/qos> add_class
Default root-out cr 1024 br 2048
SR/configure/interface/bundle VLANtest/qos> class JonesInc
SR/configure/interface/bundle VLANtest/qos/class JonesInc> add_vlan_id 24
SR/configure/interface/bundle VLANtest/qos/class JonesInc> exit
SR/configure/interface/bundle VLANtest/qos> class SmithInc
SR/configure/interface/bundle VLANtest/qos/class SmithInc> add_vlan_id 25-29
SR/configure/interface/bundle VLANtest/qos/class SmithInc> exit
```

```
SR/configure/interface/bundle VLANtest/qos> class Default
SR/configure/interface/bundle VLANtest/qos/class Default> add_vlan_id default
SR/configure/interface/bundle VLANtest/qos/class Default> exit
SR/configure/interface/bundle VLANtest/qos> enable
SR/configure/interface/bundle VLANtest/qos> exit 4
```

Historical Statistics

The **historical_stats** command enables users to collect statistics for every N hours (N =< 4 hours) and upload the statistics for every class to an FTP server. The data is sent in an ASCII format file with three sections. The first section includes the upload time, Secure Router IP address, sample interval in minutes, and the upload interval in minutes. The second section includes class-based statistics for all bundles configured and enabled with QoS. The third section includes class-based statistics. Empty lines and header lines in the file start with the ">" character. The bundle statistics start with the character "B," and class statistics start with the character "C." These designations allow easier parsing of the file.

Configuring bulk statistics

```
SR/configure/qos/historical_stats> ftp_parameters
Primary FTP server: 10.1.3.1
Secondary FTP server: 10.1.18.1
FTP user name: bjones
FTP password: xxxxxxxx
SR/configure/qos/historical_stats > sample_interval
5
SR/configure/qos/historical_stats > upload interval 2
stats
SR> show qos historical_stats configuration
```

```
Historical Statistics Configuration
---------------------------

status               : ENABLED
Primary FTP server   : 10.10.1.1
Secondary FTP server : 10.2.2.1
FTP user name        : joeuser
FTP password         : ******
Upload interval      : 1 hrs
Sample interval      : 5 mins
```

Figure 36: Screen Display for show qos historical_stats configuration Command

Traffic Policing versus Traffic Shaping

Policing controls the traffic by dropping packets or marking down their priority when the configured rate is exceeded. Shaping controls the traffic by delaying packets using a queuing

mechanism when they arrive faster than the configured rate. Therefore shaping can smooth out the burstiness in a flow, helping to minimize buffer overruns in intermediate routers. Policing does not smooth out bursts, but it can control bursts by dropping packets or marking them down.

Since shaping delays packets, thereby throttling the packet rate instead of dropping them, adaptive applications (those using TCP) perform better. Dropping packets can cause exponential back-off of TCP which can affect the throughput. This problem with policing can be minimized by configuring the burst parameter of the policer to be a sufficiently large value.

Need for Traffic Policing

Policing has the advantage of providing low latency since it does not queue packets. This makes policing a good choice for interactive and streaming voice and video applications. Policing also uses much less resources in the router than shaping. It is a better and more practical method to provide QoS for incoming traffic on an interface.

On Secure Routers, policing can be used to limit the maximum rate of a traffic flow received on a WAN interface. This is especially useful when the POP/CO router does not do any shaping or policing of traffic before transmitting it onto the WAN or it is under a different administrative domain which prevents access to the QoS configuration.

Traffic Policing Functionality on Secure Routers

Customers will now be able to rate limit inbound traffic on the WAN links using policing while doing CBQ for outbound traffic. This means the Secure Router can now provide QoS for traffic in both directions, eliminating dependency on the upstream router. Policing of outbound WAN traffic is also possible, but using CBQ to shape traffic is recommended because of reasons mentioned in section " Traffic Policing versus Traffic Shaping on page 220 ". Additionally, CBQ also provides bandwidth guarantee, bandwidth borrowing and prioritization. It provides "Total Link Access Control" instead of just rate limiting. Traffic policing is also supported on ethernet interfaces. Please check section " Verifying Policing Status and Configuration on page 223 " for limitations.

Traffic policing is implemented using a token bucket algorithm. Users will be able specify two parameters when configuring traffic policing, Rate (token fill rate) and Burst (number of tokens). Rate is specified in Kbps or percent of link bandwidth. Burst can be specified in kilobits or as a duration (based on the configured rate) in milliseconds. "Rate" determines the average bandwidth for the policed flow and "Burst" determines the maximum burst (in bits or bytes) permitted for the flow. Packets conforming to these limits will be forwarded and those violating these limits will be dropped. Other conform-actions and violate-actions, like marking down the TOS or DSCP value, are not be supported for now.

We will not support additional parameters that Cisco systems supports, like "extended burst" for CAR (committed access rate) and "excess burst" for policing. We feel, they introduce more

complexity (in configuration and implementation as well) without adding much usefulness. Specifying just the "Rate" and "Burst" makes the policing feature simple and effective. The "extended burst" parameter is needed to permit a large packet, by loaning tokens, when there are not enough tokens available at a given time for the entire packet. The Avaya policing algorithm allows for such "loaning of tokens" by default.

Configuring Traffic Policing

As with CBQ, the first step is traffic classification. Flows should be defined by creating traffic classes. To classify based on multiple fields, for e.g. source IP address and port, a hierarchy of classes should be created. Traffic policing for a class can be configured using the "police" command at the class level as shown below. Policing for non-leaf classes (which define more aggregate flows) is currently not be supported. Multi-level policing will be supported in a later release.

The policing rate and burst limit can be configured using the **police** command. Only the "rate" parameter is mandatory. Burst limit can be specified in Kbits or milliseconds using the "burst" parameter or the "burst-time" parameter respectively. If not specified, a default value of 1000 ms is used for the burst limit. "no" form of the command can be used to unconfigure policing for a class.

Syntax

```
police rate <rate in kbps or percent> [burst <burst kbits> |
burst-time <burst msec>]
no police
```

The example below shows the different variations of the **police** command. Class c1 is configured for policing with a rate of 512 Kbps and a burst of 768 Kbits. This translates to a permissible burst of 1.5 sec at the rate of 512 Kbps.

```
R1/configure/interface/bundle wan1/qos/class c1>
police rate 512 burst 768
R1/configure/interface/bundle wan1/qos/class c1>
exit
R1/configure/interface/bundle wan1/qos> class c2
R1/configure/interface/bundle wan1/qos/class c2>
police rate 1024 burst-time 800
R1/configure/interface/bundle wan1/qos/class c2>
exit
R1/configure/interface/bundle wan1/qos> class c3
R1/configure/interface/bundle wan1/qos/class c3>
police rate 256
```

After configuring policing parameters for the individual classes, policing needs to be enabled at the "/interface/qos" level on that interface for a specific traffic direction. It is not mandatory for policing to be configured for all classes on the interface for a given traffic direction. If policing

is enabled on the interface, but is not configured for a given class, no policing will be performed on packets matching that class, for example, all packets matching the class will be allowed.

```
R1/configure/interface/bundle wan1/qos> enable
policing in
R1/configure/interface/bundle wan1/qos> enable
policing out
```

In the absence of a default class, with policing enabled, packets not matching any configured class will be allowed. This is also true for Monitoring but not for CBQ. The reason is that, unlike CBQ, policing does not manage the entire link bandwidth but only imposes bandwidth limits on certain traffic classes.

Policing and CBQ cannot be simultaneously enabled at the same time for a given traffic direction on an interface. One has to be disabled before the other is enabled. (Note: CBQ can only be enabled for outbound traffic).

Verifying Policing Status and Configuration

The two CLI display commands below can be used to check the policing configuration and status for a class and for an interface.

show qos bundle bundle_name shows the policing status on the interface and the bandwidth each class is getting along with its configured policing rate.

show qos bundle bundle_name class class_name shows detailed information for the class including configured policing parameters and packet drops due to policing.

```
R87> show qos bundle wan1
```

```
Interface: Bundle wan1 (Bandwidth = 3072Kbps)
Interface Outbound Configuration & Statistics
---------------------------------------------
CBQ: on  Policing: off  MON: off
+------------------+------+------+------+-------+-------+----------+----------
Traffic Class       CBQ-CR CBQ-BR Police Avg Out  Avg In    Packets    Packets
(kbps) (kbps) (kbps)  (kbps)  (kbps)      Fwded    Dropped
+------------------+------+------+------+-------+-------+----------+----------
s1                    500   3072      -  1781.2  2666.6       4451       2220
s1-def               100   3072    800   891.6  1333.1       2229       1107
s1-web               100   3072    900   889.5  1333.4       2222       1113
def-o                 500   3072   1333  1290.7  1333.4       3230        109
Interface Inbound Configuration & Statistics
--------------------------------------------
Policing: on  MON: off
+------------------+------+------+------+-------+-------+----------+----------
Traffic Class       CBQ-CR CBQ-BR Police Avg Out  Avg In    Packets    Packets
(kbps) (kbps) (kbps)  (kbps)  (kbps)      Fwded    Dropped
+------------------+------+------+------+-------+-------+----------+----------
def-in                   -      -   1100   999.7   999.7        481          0
d1                       -      -      -  1901.2  1999.6       1096         57
d1-def                 -      -      -   998.9   998.9        576          0
d1-web                 -      -    900   902.2  1000.6        520         57
R87>
```

```
R87> show qos bundle wan1 d1-web

Class: d1-web (Inbound )
Parent Class: d1 (CR = 0 kbps ; BR = 0 kbps)
Interface: Bundle wan1 (bandwidth = 3072 kbps; MON IN: off)
Configuration
--------------
Policing - Rate: 900 Kbps  Burst: 1000 msec
Application Ports assigned to this Class:
80 8080
Traffic Statistics
------------------
Avg Rate Out: 901.8 Kbps
Avg Rate In: 1000 Kbps
Counters since last boot/clear:
Packets Forwarded: 26222            -- Packet drop details
Bytes Forwarded: 34088600           Queue overflow: 0
Packets Dropped: 2913               No buffers: 0
Bytes Dropped: 3786900              Policing: 2913
RED: 0
R87>
```

Limitations

The following limitations apply for this release:

- Policing is not supported for outbound traffic on ethernet interfaces.
- Multi-level policing is not supported. Policing is done only for leaf classes. Any policing configuration on non-leaf classes is ignored.
- When adding a new outbound class, it is mandatory to specify CR and BR, which are CBQ parameters, even though the class is intended to be used only for policing.

QoS Monitor Mode QoS Configuration

When the traffic policies are defined on an interface, there are three options which can be applied to the policies:

- Mon (for traffic monitoring and statistics)
- CBQ (for Class-Based Queuing)
- Policing

This section provides a typical example of using the monitor option to collect policy statistics; including step-by-step CLI configuration details.

Configuration Steps:

The following are the step-by-step CLI configuration details to achieve this configuration.

```
Config term
interface bundle wan
```

```
link t1 1
link t1 2
encapsulation ppp
ip address 10.19.5.2 255.255.255.0
ip multicast ospfrip2
red
exit red
qos
add_class VoIP root-out cr_percent 30 br_percent 50
priority 3
add_class Video root-out cr_percent 50 br_percent 100
priority 5
add_class OtherSrcIP root-out cr_percent 11 br_percent
100
add_class FTP OtherSrcIP cr_percent 10 br_percent 100
priority 7
add_class other OtherSrcIP cr_percent 1 br_percent 100
class VoIP
add_src_ip 10.10.10.0 255.255.255.0
mark_dscp ef
exit class
class Video
add_src_ip 20.20.20.0 255.255.255.0
mark_dscp af41
exit class
class OtherSrcIP
add_src_ip default
exit class
class FTP
add_port 21
mark_dscp af11
exit class
class other
add_port default
mark_dscp cs0
exit class
enable mon outbound
exit qos
exit bundle
```

Trusted Core Configuration

Configuration Steps

The following are the step-by-step CLI configuration details to achieve a trusted core configuration:

```
Config term
interface bundle wan
qos
add_class critical root-out cr_percent 6 br_percent 20
priority 1
add_class network root-out cr_percent 10 br_percent 30
priority 2
add_class premium root-out cr_percent 30 br_percent 50
priority 3
add_class platinum root-out cr_percent 10 br_percent 50
priority 4
add_class gold root-out cr_percent 30 br_percent 50
```

```
priority 5
add_class silver root-out cr_percent 10 br_percent 100
priority 6
add_class bronze root-out cr_percent 2 br_percent 100
priority 7
add_class default root-out cr_percent 2 br_percent 100
priority 8
class critical
add_dscp cs7
exit class
class network
add_dscp cs6
exit class
class premium
add_dscp cs5
add_dscp ef
no enable red
exit class
class platinum
add_dscp cs4
add_dscp af41
add_dscp af42
add_dscp af43
exit class
class gold
add_dscp cs3
add_dscp af31
add_dscp af32
add_dscp af33
exit class
class silver
add_dscp cs2
add_dscp af21
add_dscp af22
add_dscp af23
exit class
class bronze
add_dscp cs1
add_dscp af11
add_dscp af12
add_dscp af13
exit class
class default
add_dscp default
mark_dscp 0
exit class
enable cbq outbound
exit
```

Un-trusted Access Configuration

This section provides a typical example of untrusted access configuration, including step-by-step CLI configuration details.

Configuration Steps:

The following are the step-by-step CLI configuration details to achieve the configuration.

```
Config term
interface bundle wan
link t1 1
link t1 2
encapsulation ppp
ip address 10.19.5.2 255.255.255.0
ip multicast ospfrip2
red
exit red
qos
add_class VoIP root-out cr_percent 30 br_percent 50
priority 3
add_class Video root-out cr_percent 50 br_percent 100
priority 5
add_class OtherSrcIP root-out cr_percent 11 br_percent
100
add_class FTP OtherSrcIP cr_percent 10 br_percent 100
priority 7
add_class other OtherSrcIP cr_percent 1 br_percent 100
class VoIP
add_src_ip 10.10.10.0 255.255.255.0
mark_dscp ef
exit class
class Video
add_src_ip 20.20.20.0 255.255.255.0
mark_dscp af41
exit class
class OtherSrcIP
add_src_ip default
exit class
class FTP
add_port 21
mark_dscp af11
exit class
class other
add_port default
mark_dscp cs0
exit class
enable cbq outbound
exit qos
exit bundle
```

Traffic Policing Configuration

The following are the step-by-step CLI configuration details to achieve a traffic policing configuration:

```
Config term
interface ethernet 0
qos
add_class Video root-in
add_class DataVoice root-in
add_class UDP1500 Video
add_class FTP DataVoice
add_class Allother DataVoice
class Video
exit class
```

```
class DataVoice
exit class
class UDP1500
add_port 1500
exit class
class FTP
police rate 500
add_port 21
exit class
class Allother
add_port default
exit class
enable policing inbound
exit qos
exit ethernet
```

Burst Tolerance for FR and PPP

You can configure the burst tolerance capacity on PPP and Frame Relay (FR) interfaces in milliseconds ranging from 15 to 200 ms. The burst tolerance configuration tunes the maximum and minimum thresholds for interface RED and class RED accordingly. Burst tolerance can be configured regardless of the CBQ status on the bundle.

For PPP interface burst tolerance is always configured at the bundle command level.

For FR interfaces, when CBQ is enabled on the bundle, burst tolerance is configured at the bundle command level . When CBQ is not enabled on the FR bundle, then the burst tolerance must be configured at the PVC command level.

This feature is recommended for low speed links (T1/E1) on Secure Router 1004 and Secure Router 3120 series only.

CLI Command Syntax

Burst tolerance updates the maximum and minimum threshold values for interface RED (when CBQ is not enabled) and class RED (when CBQ is enabled on the interface) accordingly.

Burst tolerance can be configured at bundle level or, with an FR bundle, at PVC level . In either case, the command syntax remains same.

The default value of burst tolerance is 15 ms. The maximum value can be configured is 200 ms. The values can be configured in multiples of 5 ms.

1. Enter configuration mode:

   ```
   configuration terminal
   ```

2. Specify the PPP or FR WAN bundle to configure:

   ```
   interface bundle <wan>
   ```

   OR

   Specify the FR PVC to configure:

```
interface bundle <wan>
fr
pvc <pvc>
```

3. Specify the burst-tolerance:

```
burst-tolerance <15-200>
```

QOS Strict Priority Queuing (SPQ)

Secure Router supports Strict Priority Queuing (SPQ) to minimize latency and jitter for traffic on Main Ethernet interfaces and MLPPP ,PPP, and HDLC bundle interfaces. SPQ uses the shaping and scheduling infrastructure currently used with Class Based Queuing (CBQ), so there is minimal change to QoS configuration. Traffic is classified and marked as before. When SPQ is enabled, instead of queuing the classified traffic into class queues, the traffic flows through one of the interface queues based on the configuration. SPQ supports up to eight different queues per physical interface in which each queue has a separate priority. This means that multiple flows can be queued into a single queue if their priority values are the same. SPQ can be enabled or disabled at the interface level for outbound flows, just like CBQ. Only CBQ or SPQ can be active on any interface, yet both can be active at the same time on different interfaces. The SPQ queues are named 1 through 8 where queue 1 is the highest priority and the lowest priority is 8. All unclassified flows are dropped unless a default class is configured and mapped to the lowest priority queue (queue 8).

Unlike CBQ where the committed rate percentage and peak rate percentage are specified globally in the class map, with SPQ the committed rate percentage is specified for each queue at the interface level with the shape command. The peak rate percentage is 100% for all SPQ interface queues. After the committed rate percentage for all the queues has been observed the remaining bandwidth is serviced by priority.

By default, the committed rate for any SPQ interface queue is zero. The latency for traffic for a SPQ queue can increase once it exceeds the committed rate percentage for that queue.

CLI QOS Commands

To configure SPQ, you must configure the traffic classes for each traffic flow under the interface tree and assign an SPQ queue number for each class with the assign_queue command. When SPQ is enabled on an interface, the committed rate (cr_percent) and peak rate (br_percent) values defined for the class are ignored. For SPQ, the committed rate is specified using the queue command under the QoS section of the interface tree.

All the clear and show commands are equivalent for SPQ as for CBQ.

Mapping a traffic class to an SPQ queue on Ethernet interface

1. To enter configuration mode, enter:

```
configure terminal
```

2. To select the interface, enter:

```
interface ethernet <port number, 0 or 1>
```

3. To enter the QoS tree, enter:

```
qos
```

4. To create a traffic class, enter:

```
add_class <class-name> <parent-name> <cr_percent>
<br_percent>
```

5. To configure the traffic class, enter:

```
class <class-name>
```

6. To classify traffic, use the "add" commands. For example, to classify traffic according to diffServ code point, use:

```
add_dscp <value>
```

7. To assign an SPQ queue, enter:

```
assign-queue <queue number, range 1-8>
```

8. To exit class configuration mode, enter:

```
exit
```

9. To configure the committed rate for the queue:

```
queue <queue number, range 1-8> <cr_percent>
```

10. To exit QoS configuration mode, enter:

```
exit
```

Enable SPQ for Ethernet Interface

1. To enter the configuration mode, enter:

```
configure terminal
```

2. To select the ethernet interface, enter:

```
interface ethernet <port number, 0 or 1>
```

3. To specify qos chassis configuration, enter:

```
qos
```

4. To enable SPQ, enter:

```
enable spq output
```

Capacity of QoS over Ethernet

The SR 3120, 1004 and 1002 support QOS Buffering up to 50000 Kbs.

SR 1001and SR1001S support QOS Buffering up to 20000 Kbs

Chapter 34: Remote Access VPN

Secure Remote Access Using IPSec VPN

The corporate network no longer has a clearly defined perimeter inside secure building and locked equipment closets. Increasingly, companies have a need to provide remote access to their corporate resources for the employees on the move.

Traditionally, remote users could access the corporate LAN through dial-up and ISDN lines which were terminated in the corporate remote access servers. However, these point-to-point connection technologies do not scale well to the growing number of remote users and the corresponding increase in the infrastructure investments and maintenance costs.

A solution to meeting the needs of increasing numbers of remote users and for controlling access costs is to provide remote access through the Internet using firewalls and a Virtual Private Network (VPN). Internet Protocol Security (IPSec) keeps the connection safe from unauthorized users.

In a typical IPSec remote access scenario, the mobile user has connectivity to Internet and an IPSec VPN client loaded on their PC. The remote user connects to the Internet through their Internet service provider and then initiates a VPN connection to the IPSec security gateway (the VPN server) of the corporate office, which is typically an always-on Internet connection.

One of the main limitations in providing remote access is the typical remote user connects with a dynamically assigned IP address provided by the ISP. IPSec uses the IP address of users as an index to apply the Internet Key Exchange (IKE) and IPSec policies to be used for negotiation with each peer. When the VPN client has a dynamic IP address, the VPN server cannot access the policies based on the IP address of the client. Instead, the VPN server uses the identity of the VPN client to access the policies.

Access Methods

Avaya supports two types of IPSec remote access using VPNs.

Remote Access: User Group

One of the methods to achieve IPSec remote access in Avaya is the user group method. In this method, the administrator creates an IKE policy for a logical group of users such as a department in an organization. Each user in the group is identified with unique information that is uniquely configured in the IKE policy. Also, an IPSec template is attached to the user group.

Once the VPN user is authenticated using IKE, the users dynamically-assigned IP address is added to the destination address field in the IPSec template attached to the user group. The VPN user now has the required IPSec policy that allows access through the gateway to the corporate LAN.

Remote Access: Mode Configuration

The other method to achieve IPSec remote access in Avaya is the mode configuration method.

This method makes the VPN client an extension of the LAN being accessed by the VPN client. The remote client appears as a network accessing some resource behind the VPN server.

The VPN client is allocated a private IP address by the VPN server and the client uses this as the source IP address in the inner IP header in tunnel mode.

In tunnel mode, at each IKE end point, the IP traffic to be protected is completely encapsulated with another IP packet. In this, the inner IP header remains the same as seen in the original traffic to be protected. In the outer IP header, the source and destination addresses are the addresses of the tunnel end points.

Typically, for a remote user, the source address of the outer IP header is the dynamic public IP address provided by the ISP. When mode configuration is enabled, the source address of the inner IP header is the private address allocated by the VPN server to the VPN client.

As in the case of user group method, the administrator creates an IKE policy for a logical group of users such as a department in an organization. The identity information used to identify each user uniquely is configured in the IKE policy. The IKE policy is attached to a mode configuration record. The mode configuration record contains an IPSec policy template to be used for creating dynamic IPSec policy. Also, the record contains one or more pools of private IP addresses to be used for allocating the addresses to the VPN clients. Besides the private IP address, the VPN server can also provide WINS and DNS server addresses.

Upon successful IKE authentication of a VPN client, the server checks whether the IKE policy used to authenticate the VPN client is enabled for mode configuration. If so, the server allocates a private IP address from one of the IP pools in the mode configuration record to the VPN client. The destination address field in the IPSec template attached to the user group is filled in with the private IP address allocated to the VPN client and this is installed as an IPSec policy.

Configuration Examples

The following examples illustrate configurations for creating secure remote VPN access to:

- An individual SNMP user managing the gateway (user group method)
- The corporate LAN for multiple users (mode configuration method)

IPSec Remote Access User Group Method: Single Proposal, Pre-shared Key Authentication

The following example demonstrates how to manage the Avaya gateway from a secure VPN management host. An application would look like a host in a remote site is interested in managing the Secure Router using SNMP. But the remote host is interested in doing securely. The SNMP response that is generated in Secure Router for a request from the management host is called self-generated traffic.

The Avaya gateway provides a map called Self for self-generated traffic. This map is created automatically when the gateway comes up.

The security requirements for the management tunnel are:

- 3DES with SHA1,Pre-shared key authentication, XAuth
- IPSec ESP with AES128 and HMAC-SHA1

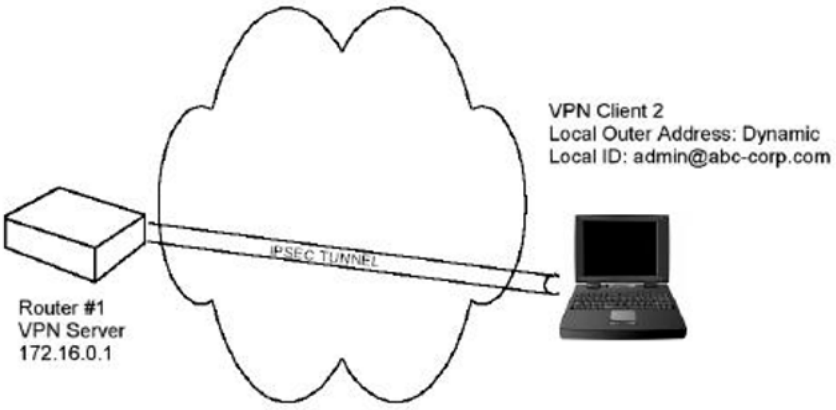


Figure 37: User Group Remote Access Configuration

To create the user group configuration enter:

```
Router>1> configure term
Router>1/configure> interface bundle wan
Router>1/configure/interface/bundle wan> link t1 1-2
Router>1/configure/interface/bundle wan> ip address
172.16.0.1 32
Router>1/configure/interface/bundle wan> crypto internet
```

Note:

1.error message saying Bundle is not yet encapped.

To configure the IKE policy for negotiating with the remote VPN client needing access (note that the IKE and IPSec policies for management (self) tunnel need to be defined in the "Self" map):

```
Router>1/configure> crypto Self
Router>1/configure/crypto> dynamic
Router>1/configure/crypto/dynamic> ike policy admin
user-group
Router>1/configure/crypto/dynamic/ike/policy admin>
local-address 172.16.0.1
Router>1/configure/crypto/dynamic/ike/policy admin>
remote-id email-id sampledata admin@abc-corp.com
Router>1/configure/crypto/dynamic/ike/policy admin> key
pskforadminuser
Router>1/configure/crypto/dynamic/ike/policy admin>
proposal 1
Router>1/configure/crypto/dynamic/ike/policy
admin/proposal 1> encryption-algorithm 3des-cbc
Router>1/configure/crypto/dynamic/ike/policy
admin/proposal 1> client authentication radius pap
```

To configure the IPSec policy for negotiating with VPN client needing access to the security gateway.

```
Router/configure/crypto/dynamic> ipsec policy admin
user-group
Router/configure/crypto/dynamic/ipsec/policy admin>
```

```
match address 172.16.0.1 32
Router>1/configure/crypto/dynamic/ipsec/policy admin>
proposal 1
Router>1/configure/crypto/dynamic/ipsec/policy
admin/proposal 1> encryption-algorithm aes128-cbc
```

IPSec Remote Access Mode Configuration Group Method

The following example demonstrates how to configure a Secure Router to be an IPSec VPN server using mode-configuration method. The client could be any standard mode config enabled IPSec VPN client.

In this example, the client needs to access the corporate private network 10.0.1.0/24 through the VPN tunnel. The server has a pool of IP addresses from 20.1.1.100 through 20.1.1.150 to be allocated for mode config enabled VPN clients. The assigned IP address is used by the VPN client as the source address in the inner IP header. The outer IP header will carry the dynamic IP address assigned by the Internet Service Provider as the source address. The security requirements are as follows:

3DES with SHA1, Mode Config

IPSec ESP tunnel with AES256 and HMAC-SHA1

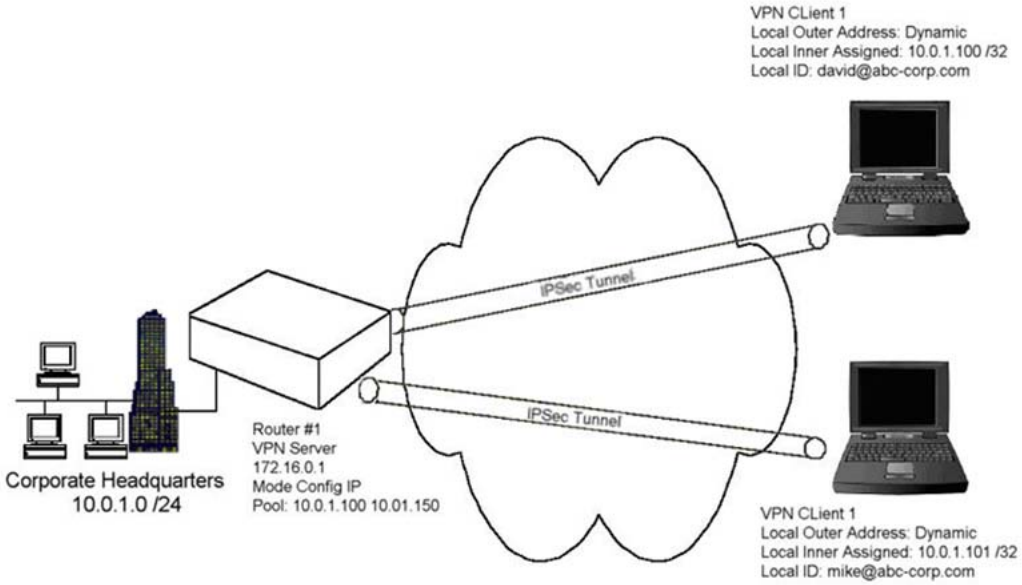


Figure 38: Configuration Mode Remote Access Configuration

To configure the VPN gateway:

```
Router>1> configure term
Router>1/configure> interface ethernet 1
Router>1/configure/interface/ethernet 1> ip address
10.0.1.1 24
Router>1/configure/interface/ethernet 1> crypto corp
Router>1/configure> interface bundle wan
Router>1/configure/interface/bundle wan> link t1 1-2
Router>1/configure/interface/bundle wan> ip address
172.16.0.1 32
Router>1/configure/interface/bundle wan> crypto internet
```

To configure the IKE policy for negotiating with VPN clients needing access to the corporate private network 10.0.1.0.

Note:

Bundle must be encapsulated first.

```
Router>1/configure> crypto corp
Router>1/configure/crypto> dynamic
Router>1/configure/crypto/dynamic> ike policy IDCsales
modecfg-group
Router>1/configure/crypto/dynamic/ike/policy
IDCsales> modeconfig-group
Router>1/configure/crypto/dynamic/ike/policy IDCsales>
local-address 172.16.0.1
```

To configure the user name (optional) for remote-id:

```
Router>1/configure/crypto/dynamic/ike/policy IDCsales>
remote-id email-id sampledata david@abc-corp.com
```

```
Router>1/configure/crypto/dynamic/ike/policy IDCsales>
remote-id email-id sampledata mike@abc-corp.com
Router>1/configure/crypto/dynamic/ike/policy IDCsales>
key pskforsalesusers
Router>1/configure/crypto/dynamic/ike/policy IDCsales>
proposal 1
Router>1/configure/crypto/dynamic/ike/policy
IDCsales/proposal 1> encryption-algorithm 3des-cbc
Router>1/configure/crypto/dynamic/ike/policy
IDCsales/proposal 1> exit
Router>1/configure/crypto/dynamic/ike/policy IDCsales/propo
sal 1>client authentication radius pap
Router>1/configure/crypto/dynamic> client configuration
> configure address pool for modecfg client
address-pool 1 20.1.1.100 20.1.1.150
```

To configure the IPSec policy for negotiating with VPN clients needing access to the corporate private network 10.0.1.0.

```
Router>1/configure/crypto/dynamic> ipsec policy IDCsales
Router>1/configure/crypto/dynamic/ipsec/policy
IDCSales> match address 10.0.1.0 24
Router>1/configure/crypto/dynamic/ipsec/policy
IDCSales> proposal 1
Router>1/configure/crypto/dynamic/ipsec/policy
IDCSales/proposal 1> encryption-algorithm aes256-cbc
```

Chapter 35: Routing Information Protocol

Configuring Routing Information Protocol for Ethernet 0 and WAN 1 Interfaces

```
SR> configure terminal
SR/configure> router rip
SR/configure/router rip> interface ethernet0
SR/configure/router rip/interface ethernet0> exit
SR/configure/router rip> interface wan1
SR/configure/router rip> exit
```

Displaying RIP Configuration

Execute **`show ip rip global`** to display RIP configuration information

```
> show ip rip global
Router RIP is enabled
        Mode: RIP 2
        Distance: 100
        Default Metric: 1
        Timers:
                Update: 30 seconds
                Holddown: 120 seconds
                Flush: 180 seconds
>
```

Figure 39: show ip rip global Command

Displaying All Configured RIP Interfaces

Execute **`show ip rip interface all`** to display information about all configured RIP interfaces.

```
> show ip rip interface all
RIP is configured for interface <ethernet0>
        Mode: RIP 2
        Metric: 5
        Authentication: None
        Split Horizon: Poison
        Routers : None
        Interface state:  Up Broadcast Multicast Active

>
```

Figure 40: show ip rip interface all Command

Chapter 36: Static Routing

All Secure Routers support IP routing utilizing static routes. The following diagram shows a remote Secure Router "A" connected over an MLPPP bundle to the main Secure Router "B". Secure Router B in turn routes to the customer router.

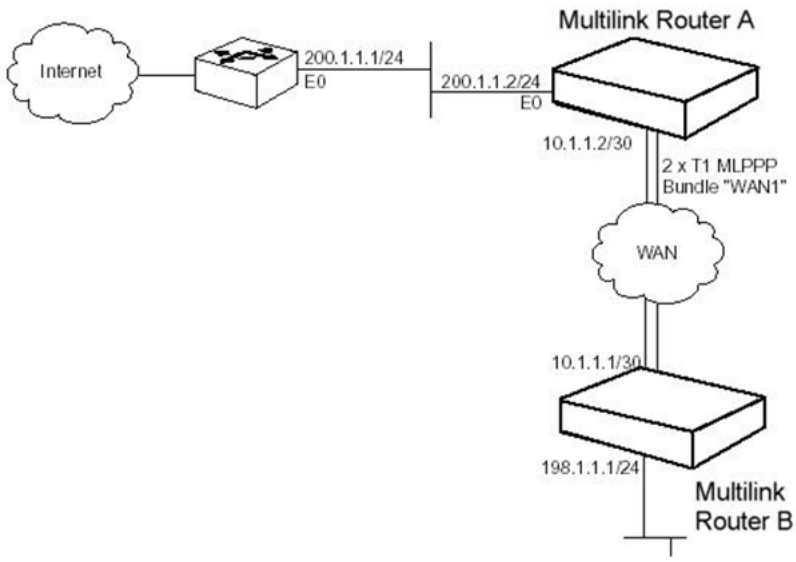


Figure 41: IP Routing

The customer router Ethernet 0 IP address is 200.1.1.1 255.255.255.0, and the IP route is 198.1.1.0 255.255.255.0 200.1.1.2 2.

Configure the Multilink Router A at Site A

```
Multilink_Router_A> configure term
Multilink_Router_A/configure> interface ethernet 0
Multilink_Router_A/configure/interface/ethernet> ip addr 200.1.1.2 24
Multilink_Router_A/configure/interface/ethernet> exit
Multilink_Router_A/configure> interface bundle wan1
Multilink_Router_A/configure/interface/bundle> link t1 1
Multilink_Router_A/configure/interface/bundle> encap ppp
Multilink_Router_A/configure/interface/bundle> ip addr 10.1.1.2 255.255.255.252
Multilink_Router_A/configure/interface/bundle> exit
Multilink_Router_A/configure> ip route 0.0.0.0 0.0.0.0 10.1.1.1 1
```

Configure the Multilink Router B at site B

```
Multilink_Router_B> configure term
Multilink_Router_B/configure> interface ethernet 0
Multilink_Router_B/configure/interface/ethernet> ip addr 198.1.1.1 255.255.255.0
Multilink_Router_B/configure/interface/ethernet> exit

Multilink_Router_B/configure> interface bundle wan 1
Multilink_Router_B/configure/interface/bundle> link t1
1
Multilink_Router_B/configure/interface/bundle> encapp ppp
Multilink_Router_B/configure/interface/bundle> ip addr 10.1.1.1 255.255.255.252
Multilink_Router_B/configure/interface/bundle> exit

Multilink_Router_B/configure> ip route 200.1.1.0 255.255.255.0 10.1.1.2 1
```

Chapter 37: VRRP enhancements

The Secure Router 1000 Series and 3120 provide support for multiple VRRP enhancements. By design, VRRP eliminates a common point of failure present in static routing environments by specifying an election protocol to dynamically assign routing responsibility to a VRRP router on a LAN. VRRP is used to maintain availability at the IP address level. In a VRRP setup, one router is elected the master. When the master goes down, backup routers hold an election for a replacement. VRRP is applicable only to primary ethernet interfaces and VLAN enabled subinterfaces, with a maximum of 10 VRRP groups per router.

The nature of VRRP has several routers performing as one virtual router that has a Virtual Router ID and virtual IP addresses. Any of these routers could act as master at any given time, provided it wins the election. The master sends advertisements to backup routers informing them of its state. If advertisements fail to be received, an election is called. The backup with the highest priority value wins and assumes position as master. As of this release, the interval at which these advertisements are sent is configurable via CLI.

The Secure Router supports VRRP authentication types "no authentication" and "clear text password authentication" for VLAN enabled subinterfaces. Using the "no authentication" type, VRRP exchanges are not authenticated, while with "clear text password authentication", the receiver checks to make sure VRRP authentication packet data matches the configured authentication string. If there is no match, the exchange is discarded.

In addition to this, VRRP interface monitoring on VLAN enabled subinterfaces functionality has been included. VRRP groups can be configured to monitor external interfaces in case they go down. The reason for this is to calculate VRRP priority based on a router’s tracking priority. When a router’s external interface goes down, the number value given to tracking priority is subtracted from the VRRP priority value, giving it as new priority and ultimately affecting its chances in an election.

Finally, several VRRP load-balancing mode types are present so users can choose which mode best suits their needs. Full load-balancing is supported, and users can further choose from one of the following options:

- Mode 0: Gratuitous ARP Mode

  Relies on gratuitous ARP to redirect traffic in the event of a failover. This mode uses the MAC address of a physical ethernet port as a virtual MAC.

- Mode 1: Active/Standby Mode

  Applicable to primary ethernet interfaces only, this mode allows a VRRP interface to participate in a single VRRP group at any given time using a virtual MAC address. If Active/Standby Mode is used, only 1 VRRP group can be configured.

- Mode 2: Promiscuous Mode

All packets that reach the interface are accepted, including packets not intended for it. Potentially generates performance overhead as packets are processed.

Configuring VRRP over VLAN

Use the following procedure to configure VRRP over VLAN.

Procedure steps

1. To configure VRRP over VLAN, enter Configuration Mode.

   ```
   configure terminal
   ```

2. Enter Interface Mode.

   ```
   interface <interface>
   ```

3. In the case of a 802.1q (VLAN) interface/subinterface, apply encapsulation.

   ```
   encapsulation <type>
   ```

4. In the case of a subinterface, specify an IP address.

   ```
   ip address <A.B.C.D>
   ```

5. Specify VRRP mode.

   ```
   vrrp_mode <mode>
   ```

6. Specify a VRRP group.

   ```
   vrrp <group>
   ```

7. Specify a virtual IP address.

   ```
   ipaddr <virtual IP>
   ```

8. Configure tracking.

   ```
   track <interface> <priority>
   ```

9. Configure a priority level.

   ```
   priority <level>
   ```

10. Enable VRRP.

    ```
    enable
    ```

Table 36: Variable definitions

Variable	Value
<A.B.C.D>	The IP address of the subinterface.
<group>	The VRRP group number, in the range 1 to 255.
<interface>	The interface to work with.
<level>	The priority level, in the range 1 to 254.

Variable	Value
<mode>	The VRRP mode. Possible choices are: • 0 - Gratuitous ARP • 1 - Active/Standby Mode • 2 - Promiscuous Mode
<priority>	The track priority.
<type>	The type of encapsulation to apply.
<virtual IP>	The virtual IP address to be used.

Chapter 38: Trunk Group/Failover

Redundant connections are often required between Secure Routers and the switches to which they connect. The following diagram illustrates Ethernet redundancy between a Secure Router 1000 Series and a Layer 3 switch using failover on the Secure Router and a trunk group configuration on the switch.

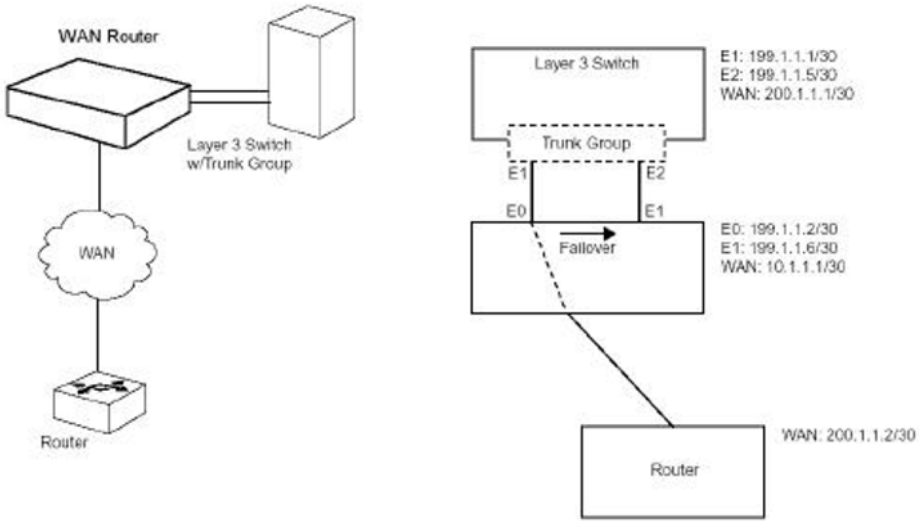


Figure 42: Trunk Group/Failover Configuration

Configuration Details

- Secure Router Ethernet 0 and 1 are connected to ports 1 and 2 of a trunk group configured switch.
- The trunk group is configured with three IP addresses and a single MAC address. One IP address is utilized for WAN connectivity; the second address provides for communication between the switch and Secure Router Ethernet 0. For this configuration, a third IP address is utilized for the failover path.
- The Secure Router 1000 Series is configured for failover on E0. When E0 loses link conectivity, it will failover to E1 and continue to pass traffic. When E0 recovers, traffic will be switched back.
- To manage the Secure Router 1000 Series from the switch during normal mode, ping, telnet, or SNMP to the Ethernet 0 IP address; during failover mode, ping, telnet, or SNMP to the Ethernet 1 IP address.

Configure the WAN Router for Failover Operation

```
WAN Router> configure term
WAN Router/configure> interface ethernet 0
WAN Router/configure/interface/ethernet> ip address
199.1.1.2 255.255.255.252
WAN Router/configure/interface/ethernet> failover
WAN Router/configure/interface/ethernet> exit

WAN Router/configure> interface ethernet 1
WAN Router/configure/interface/ethernet> ip address
199.1.1.1.6 255.255.255.252
WAN Router/configure/interface/ethernet> exit

WAN Router/configure> interface bundle wan
WAN Router/configure/interface/bundle> link t1 1
WAN Router/configure/interface/bundle> enc ppp
WAN Router/configure/interface/bundle> ip address
10.1.1.1 255.255.255.252
WAN Router/configure/interface/bundle> exit
```

Chapter 39: VLAN Tagging

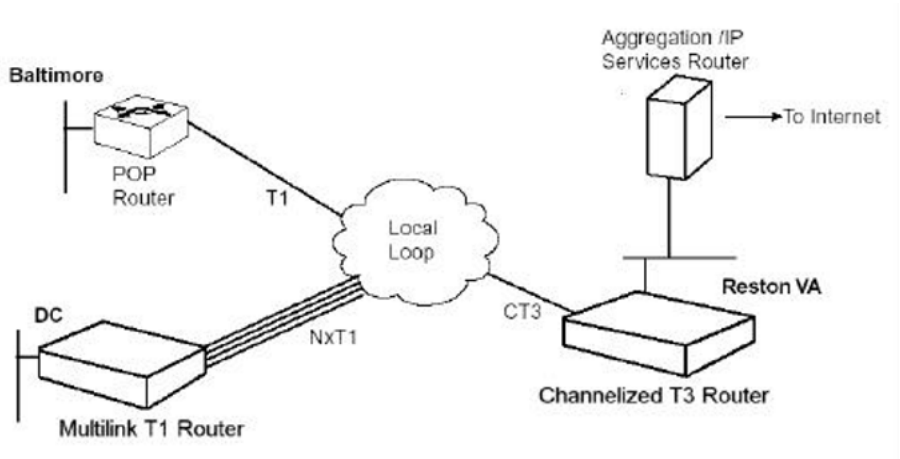


Figure 43: Aggregation Using VLAN Tagging

The illustration above shows two customers connected to an aggregation/IP services router using a SR3120. All packets coming into the SR3120 on the single T1 bundle are tagged with VLAN ID 5. All packets coming across the 4 T1 bundle from DC are tagged with a VLAN tag of 10.

In this example, the VLAN tags are only relevant from the SR3120 to the VLAN-enabled POP router. The tags are removed in the reverse direction. Avaya's IP multiplexing technology enables both remote customers to operate as if they are directly connected to the POP router residing on a tagged VLAN. In this scenario, the provider can offer HDLC, PPP, MLPPP, frame relay, and MFR connections. (The sample configurations in this document assume Baltimore uses frame relay and DC uses MLPPP.) Upgrading customer service by adding T1s to an Avaya product can be accomplished remotely (for example, at DC) after the T1 cable is connected. Thus, deploying a technician to reconfigure the unit is not necessary.

By connecting the SR3120 using a VLAN switch, additional SR3120s and POP routers can be easily added. If additional SR3120s are desired, the appropriate uplink from the VLAN switch is Gigabit Ethernet. Redundancy for the POP routers can be provided using either the second fast Ethernet port on the SR3120, in conjunction with Avaya's failover feature, or using HSRP/VRRP between the two routers. In the latter case, a VLAN switch is required.

Special configuration is not required at the CPE for this application. At the POP, traffic from each bundle or frame relay PVC is tagged and forwarded to a VLAN trunk port on the Ethernet interface. In the other direction, the SR3120 proxies for the CPE routers after learning their IP addresses through link control protocol or inverse ARP. Routing between customer VLANs, firewall functions, and traffic management can be provided by the POP router subinterfaces so there is only one location to monitor customer traffic.

Network administrators have the option of not adding tagged VLAN interfaces to the VLAN management group. This option allows network administrators to control access to the network by end users, who have access through the tagged interface by default.

See the *Command Reference* **configure interface vlan management** commands for more information. See the **show vlanfwd management** command to view all the management disabled interfaces of the router.

In this example application, the POP router is configured with the following three subinterfaces:

- 205.1.1.1
- 205.1.1.5
- 10.1.1.5

Reston configuration: Channelized T3 Router

```
Channelized_T3/configure> hostname reston
reston/configure> no ftp_server
reston/configure> no autoconf
```

Configure interface bundle balt1

```
reston/configure> interface bundle balt1
reston/configure/interface/balt1> link ct3 1 1
reston/configure/interface/balt1> encapsulation fr
reston/configure/interface/balt1> fr
reston/configure/interface/balt1/fr> intf_type dce
```

Configure interface balt1 pvc 100

```
Configure interface balt1 pvc 100
reston/configure/interface/balt1/fr> pvc 100
reston/configure/interface/balt1/fr/pvc 100> policing
cir 1536000 bc 1536000 be 1536000
reston/configure/interface/balt1/fr/pvc 100> shaping cir
1536000 bcmax 1536000 bcmin 1536000 be 1536000
```

>The Baltimore router is 205.1.1.2/30.

> The PVC uses a private address on the Reston end.

```
reston/configure/interface/balt1/fr/pvc 100> ip addr
10.1.1.1 255.255.255.252
```

> The POP router is 205.1.1.1/30

```
reston/configure/interface/balt1/fr/pvc 100> ip
source_forwarding 205.1.1.1
reston/configure/interface/balt1/fr/pvc 100> vlan
reston/configure/interface/balt1/fr/pvc 100/vlan> vlanid
```

```
5
reston/configure/interface/balt1/fr/pvc 100/vlan> exit 4
```

Configure interface bundle dc1

```
reston/configure> interface bundle dc1
reston/configure/interface/bundle dc1> link ct3 1 2-5
reston/configure/interface/bundle dc1> encapsulation ppp
reston/configure/interface/bundle dc1> ip unnumbered
ethernet0
```

> DC is 205.1.1.6/30.

```
reston/configure/interface/bundle dc1> ip source_forwarding
205.1.1.5
reston/configure/interface/bundle dc1> vlan
reston/configure/interface/bundle dc1/vlan> vlanid 10
reston/configure/interface/bundle dc1>/vlan> exit 2
```

Configure interface ethernet 0

```
reston/configure> interface ethernet 0
reston/configure/interface/ethernet0> speed 100
full_duplex
reston/configure/interface/ethernet0> ip address
10.1.1.6 255.255.255.252
reston/configure/interface/ethernet0> exit
```

Configure ip routing

```
reston/configure> ip
reston/configure/ip> route 205.1.1.0 255.255.255.0
ethernet0 1
reston/configure/ip> route 0.0.0.0 0.0.0.0 10.1.1.5 1
reston/configure/ip> exit
```

> The above route summarizes the customer access subnets.

DC configuration: Multilink T1 Router

```
Multilink_T1> configure terminal
Multilink_T1/configure> hostname dc1
dc1/configure>
```

Configure interface ethernet 0

```
dc1/configure> interface ethernet 0
dc1/configure/interface/ethernet0> ip addr 205.100.1.1
255.255.255.0
dc1/configure/interface/ethernet0> exit
```

Configure interface bundle mip

```
dc1/configure> interface bundle mlp
dc1/configure/interface/bundle mlp> link t1 1-4
dc1/configure/interface/bundle mlp> encapsulation ppp
dc1/configure/interface/bundle mlp> ip addr 205.1.1.6
255.255.255.252
dc1/configure/interface/bundle mlp> exit
```

Configure ip routing

```
dc1/configure> ip
dc1/configure/ip> routing
dc1/configure/ip> route 0.0.0.0 0.0.0.0 205.1.1.5 1
dc1/configure/ip> exit
dc1/configure>
```

VLAN Tagging and Forwarding over Ethernet

The Ethernet interface on the Secure Router 3120 can be configured with several different forwarding modes or options. While the Secure Routers each support two Ethernet ports, Ethernet0 is utilized for Local Area Network (LAN) connectivity in the following examples.

- Packets are either tagged or un-tagged on the Ethernet interface
- Un-tagged packets can be tagged to a default VLAN
- Packets can be either routed or forwarded (bridged) by VLAN
- MAC Learning is optional for bridged packets

Note:

The Ethernet interface must always have an IP address configured, or the interface will not function properly. If IP routing is not required, a dummy IP address must still be configured.

VLAN Forwarding - Packets are already tagged at the Ethernet interface

Packets are received and transmitted in tagged format. Forwarding decision is made by the a vlanfwd mapping table of VLAN ID to Interface. Optional: The VLAN forwarding decision can also be made by L2 MAC Address.

```
interface ethernet 0
ip address 192.168.0.1 30
exit
vlanfwd
add vlanid 75 ethernet0
add vlanid 75 wan
add vlanid 130 ethernet0
add vlanid 130 wan
macbridge
exit
exit
```

Note:

In this example, the statement ip address 192.168.0.1 30 is a dummy address and will not be used for forwarding or IP access. It is required for the interface to come active.

Note:

Packets are received on VLANs 75 and 130 from a switch upstream from ethernet 0. They are forwarded based upon the vlanfwd rules.

Note:

The macbridge statement cause the forwarding decision to be made based upon the VLAN Tag and L2 MAC Address. Otherwise the VLAN will be forwarded based upon the tag.

VLAN Tagging - Interface will add and remove tags

The interface will transmit and receive untagged packets. On receipt, the packets will be tagged, and tags will be stripped on transmission. This example would be identical to the VLAN Forwarding example, except that only a single VLAN is supported on the Ethernet.

```
interface ethernet 0
ip address 192.168.0.1 30
vlan
vlanid 10
exit
exit
vlanfwd
add vlanid 10 wan
```

```
macbridge
exit
exit
```

Note:

The statement ip address 192.168.0.1 30 is a dummy address and will not be used for forwarding or IP access. It is required for the interface to come active.

Note:

Packets are received untagged, placed into VLAN 10, and then forwarded per the rules in the vlanfwd table.

802.1Q VLAN Routing - Packets are tagged and IP routed per VLAN

Packets are received and transmitted in tagged format. Individual subinterfaces are individually routed. The below configuration is for 3 Tagged interfaces. Note that although tagged in this example, the primary interface does not have to be tagged.

```
interface ethernet 0
encapsulation dot1q 10
ip address 216.138.115.193 29
exit ethernet
interface ethernet 0.1
encapsulation dot1q 20
ip address 216.138.115.201 29
exit ethernet
interface ethernet 0.2
encapsulation dot1q 30
ip address 216.138.115.209 29
exit ethernet
```

Note:

Each VLAN is fully routed as a subinterface.

Note:

All interfaces use the same Source MAC. The Ethernet switch used in the example must support Independent VLAN Learning of MAC addresses.

Multinetting (IP Subinterfaces) Configuration

Another variation for multiple subnet support that does not involve VLAN Tagging is IP Multi-Net.

```
interface ethernet 0
ip address 47.17.187.15 255.255.255.0
exit ethernet
interface ethernet 0.1
ip address 192.168.44.44 255.255.255.0
exit ethernet
```

Note:
The Ethernet subinterface shares the same broadcast domain as the primary interface

Note:
Each subinterface is assigned a unique Source MAC address. The final octet of the MAC address encodes the internal unit number.

Note:
The Ethernet interface is placed into Promiscuous mode. It will receive all packets on the interface including non-broadcast packets that are not addressed to the router. This can result in higher workload on the router.

VLAN Tagging and Forwarding over Ethernet Summary

- To support a single untagged bridged VLAN on the Ethernet
  - do not specify any encapsulation on the Ethernet
  - add a dummy IP address on the interface
  - configure the vlan and vlanid on the Ethernet
  - do not add the ethernet interface to the vlanfwd list
- To support multiple tagged bridged VLANs on the Ethernet
  - do not specify any encapsulation on the Ethernet
  - add a dummy IP address on the interface
  - do not configure the vlan and vlanid on the Ethernet
  - add each ethernet interface / tag to the vlanfwd list

- To support multiple tagged routed VLANs on the Ethernet
  - set the encapsulation type to dot1q for the interface
  - set the encapsulation type to dot1q for each subinterface
- To support multinetting on the Ethernet

  do not set dot1q encapsulation on the interface or any subinterface

Independent VLAN Learning (IVL) Support

Independent Vlan Learning allows the router to split the ARP table according to VLAN so that the ARP table lookup is based on both MAC and VLAN Id. The default mode for ARP for backward functionality is Shared VLAN learning mode.

When vlan forwarding and macbridging are enabled, to show whether IVL is set, issue the **`show vlanfwd macbridge config`** command .

The following figure shows the output when IVL is enabled:

```
host/configure/vlanfwd > show vlanfwd macbridge config
Macbridge: Enabled
MacAge(in minutes): 5
MAC Learning Type: Independent VLAN learning (ivl)
```

The following figure shows the output when macbridge is disabled:

```
host/configure/vlanfwd > show vlanfwd macbridge config
Macbridge: Disabled
MacAge(in minutes): 5
```

Queue-in-Queue VLAN support

The Queue-in-Queue VLAN feature has backwards compatibility with previous commands and there are no new CLI commands to support it.

In previous releases, Ethernet interfaces could be configured as Vlan Tagged or Vld Tagged interfaces. When untagged packets were to be switched using a VLAN, the interface had to be configured as Vlan Tagged. When a VLAN tagged packet was to be switched with another level of VLAN, the interface had to be configured as Vld Tagged. The Vlan and Vld interfaces had their own configuration and forwarding tables. Additionally, there was a limit of 2 levels of Vlan tagging (Vlan + Vld) allowed.

Using the Queue-in-Queue VLAN feature, there is a single type of tagged interface which allows packets to be switched with any number of VLAN tags. Packets are then switched on the outermost level of VLAN tags. However, the VLAN for management can only accept single tagged packets.

Chapter 40: Serial Interface

High-Speed Serial Interface

This chapter outlines the configuration of module parameters (Layer 1) and, to a lesser degree, the configuration of bundle parameters (Layer 2). The bundle configuration examples demonstrate linking of physical interfaces (modules) to logical interfaces (bundles). Module configuration occurs within the configure module tree of the Avaya CLI, and bundle configuration occurs within the configure interface bundle tree.

Bundle Configuration

Configuration of an interface bundle is required for use of any of the Secure Router 3120 serial interfaces. The interface bundle specifies the physical connection to be linked, an encapsulation protocol (Layer 2) and, optionally, Layer 3 parameters.

Serial Configuration

DCE

Figure 44: V.35 DCE on page 260 shows a connection between a router with V.35 interface and a Secure Router. In the figure, the router is configured for DTE operation and the Secure Router V.35 is configured for DCE operation. Configuring the V.35 interface for DTE operation is similar to the DCE configuration, though no clock speed is set in DTE mode.

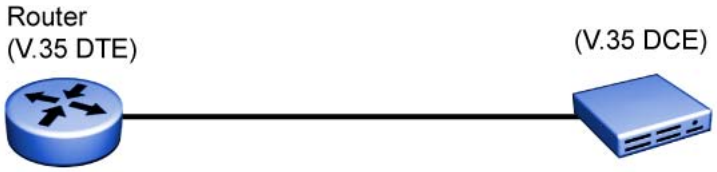


Figure 44: V.35 DCE

Configure the V.35 Module as a DCE

Example

```
SR/configure> module
SR/configure/module/uw/v35> mode dce
SR/configure/module/uw/v35> clock_rate 7000000
SR/configure/module/uw/v35> clock_source internal
SR/configure/module/uw/v35> data_mode normal
SR/configure/module/uw/v35> exit
```

HDLC

The following example creates a simple interface bundle utilizing the V.35 interface with HDLC (layer two) parameters.

Configure a V.35 HDLC bundle

Example

```
SR/configure> interface bundle toRouter
SR/configure/interface/bundle> link serial 1
SR/configure/interface/bundle> encap hdlc
SR/configure/interface/bundle> exit
```

x.21 Serial Configuration

The following examples illustrates x.21 bundle configuration:

Note:

Cables types RS-232, RS-449, EIA-530, EIA-530A and v35 are also supported. If cable_type is configured for RS-232 the clock rate will be limited to 115200 Hz.

Note:

Saving the configuration and rebooting is recommended after any serial port changes.

Example

Configuring the Secure Router at Site A

```
Select a cable type for each port.
SR/configure> module serial 1
SR/configure/module/serial 1> cable_type x21
SR/configure/module/serial 1 > exit
SR> save local
Create the bundle.
SR/configure > interface bundle MLPPP-A
Configuring a new WAN bundle interface MLPPP-A.
Begin by adding WAN link(s) to this bundle and selecting encapsulation.
SR/configure/interface/bundle MLPPP-A > link serial 1
SR/configure/interface/bundle MLPPP-A > link serial 2
SR/configure/interface/bundle MLPPP-A > encapsulation ppp
SR/configure/interface/bundle MLPPP-A > ip address 192.168.100.100
24
SR/configure/interface/bundle MLPPP-A > exit
SR> save local
SR > show interface bundles
```

Note:

Bundle link provisioning is dependant on "clock_rate", "crc", and "data_inversion" settings. Bundles must be provisioned again if any of these settings changes.

Example

Configuring the Secure Router at Site B

```
Select the cable type and mode for each port.
SR/configure > module serial 1
SR/configure/module/serial 1 > cable_type x21
SR/configure/module/serial 1 > mode dce
SR/configure/module/serial 1 > exit
SR/configure > module serial 2
SR/configure/module/serial 2 > cable_type x21
SR/configure/module/serial 2 > mode dce
SR/configure/module/serial 2 > exit
SR> save local
Create the bundle.
SR/configure > interface bundle MLPPP-B
Configure a new WAN bundle interface MLPPP-B.
Begin by adding WAN link(s) to this bundle and selecting encapsulation.
SR/configure/interface/bundle MLPPP-B > link serial 1
SR/configure/interface/bundle MLPPP-B > link serial 2
SR/configure/interface/bundle MLPPP-B > encapsulation ppp
SR/configure/interface/bundle MLPPP-B > ip address 192.168.100.101
24
SR> save local
SR/configure/interface/bundle MLPPP-B > exit
```

```
SR/configure > exit
SR > show interface bundles
```

Note:
Serial port hardware is enabled only when the physical cable matches the cable_type setting.

Display details of each bundle:

```
SR > show interface bundle MLPPP-A
SR > show interface bundle MLPPP-B
```

Troubleshooting the Serial link

If the Serial Port link is unusable, perform the following troubleshooting:

- Verify the correct cable_type and mode configuration settings.

  Use the command "**show module config serial <slot>/<port>**"

- Verify Wan Status LED state with cable connected:

  - Green - Cable type and mode are OK.
  - Red - Incorrect cable type or mode
  - Off - Cable type is undetectable or not compatible

- Verify remote equipment connected and properly configured.

  The remote equipment should mirror the Secure Router's configuration.

- Verify port alarm states

  Use the command "**show module alarm serial <slot>/<port>**"

  DTR/DSR alarms cause link outages.

- Verify bundle configuration and statistics

  Use the command "**show interface bundle <bundlename>**"

Chapter 41: VLAN Forwarding with QOS

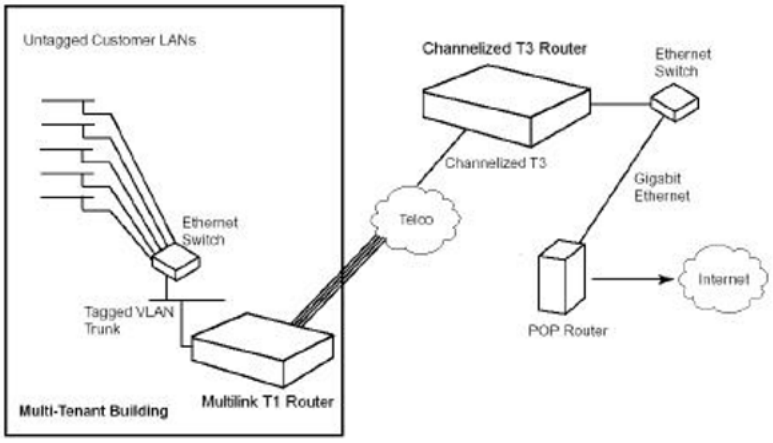


Figure 45: VLAN Forwarding: Multi-Tenant Internet Access

The example above shows each multi-tenant customer represented as a separate VLAN on the Ethernet switch. The connection in the customer office can be routed or bridged, depending on whether the provider will be hosting customer applications at the POP. The Ethernet switch passes a VLAN trunk to the Secure Router 1000 Series that forwards traffic, based on the VLAN tags, from this interface to the multilink bundle.

At the POP, tagged traffic is forwarded to a VLAN trunk port on the Ethernet switch. Routing between customer VLANs is provided by the POP router using subinterfaces on the Gigabit Ethernet VLAN trunk. The customer LAN subnet is extended all the way to the POP router making remote management of LAN services (for example, DHCP, file servers. SMTP) possible.

The VLAN forwarding feature has the added benefit of being able to support non-IP traffic since all traffic is forwarded based only on the Layer 2 VLAN tag. Although Avaya products do not communicate using non-IP Layer 3 protocols, Secure Routers can forward these protocols.

The management VLAN feature provides in-band communication with the Secure Routers as well as the Ethernet switches while remaining separate from customer traffic. The Secure Routers will examine the destination IP address of any packets received on the management VLAN. If the destination is the Secure Router, the address of the packet will be forwarded to the IP layer for local processing. If the address does not match the address of the Secure Router, the packet will be forwarded to all interfaces configured for the management VLAN with the exception of the interface where it was received. This allows all transmission equipment to be managed in a single, flat VLAN.

When the Secure Router generates traffic on to the management VLAN, an ARP request is generated in the direction of the VLAN's default route. If no default is configured, the ARP request will be generated in all possible directions, and the interface receiving the response will be cached with the reply. The source

MAC address used by the Secure Router will be associated with the Ethernet port associated with the management VLAN.

In a multi-tenant unit (MTU) where customer Internet access is through the Ethernet interface, some form of bandwidth control is necessary to prevent a high bandwidth customer from blocking others since the uplink out of the building will typically be less than 10 Mb/s. Avaya provides QoS support to limit customer bandwidth using a committed rate and burst rate, ensuring that customers get consistent bandwidth performance as other customers are activated. Avaya's QoS can be configured based on VLAN IDs, in increments of 64 kbps providing greater control than what is normally available in Ethernet switches.

Virtual LAN Domain

This version supports VLAN-based data forwarding. Essentially, this feature forwards the packets from one network to another based on the VLAN identifier (rather than routing) contained in the VLAN header, as defined by IEEE 802.1q. The VLAN packets are generally termed as tagged packets referring to the VLAN encapsulation of the Ethernet packets.

This version also supports VLD (Virtual LAN Domain). VLD allows a VLAN packet to be tagged with another level of VLAN header. This is used by service providers to carry the subscriber's VLAN packets transparently through the provider's VLAN network. The VLD packets are generally termed as double tagged packets referring to the two levels of VLAN encapsulation of the Ethernet packets.

Ethernet packets arriving on an interface (say an Ethernet) configured for VLD tagging, are also tagged (only one level) with the tag ID configured for VLD tagging on that interface. These single level tagged VLAN packets can now be forwarded on the trunk port (say a WAN interface) using the VLD forwarding table itself. In the return path, packets arriving on the trunk port as VLAN packets can be forwarded to the Ethernet interface (based on VLD forwarding table) and when they exit through the Ethernet, they are untagged one level (since this VLD tagged interface would have performed one level of untagging anyway).

This enhancement allows users to handle two different types of subscriber traffic on the same interface. One traffic type is the VLAN traffic generated by the subscriber and typically meant for office-to-office communication through a service provider network. The other traffic type is the Ethernet traffic generated by the subscriber for Internet access through the same service provider.

The management of Secure Routers through a VLAN remains unaffected due to this enhancement.

POP Configuration: Channelized T3 Router

```
Channelized_T3/configure> hostname POP-SR3120
POP-SR3120/configure> no ftp_server
POP-SR3120/configure> no autoconf
```

Configure mlppp bundle interface

```
POP-SR3120/configure> interface bundle bldg1
POP-SR3120/configure/interface/bundle bldg1> link ct3 1
1-4
POP-SR3120/configure/interface/bundle bldg1>
encapsulation ppp
POP-SR3120/configure/interface/bundle bldg1> ip
unnumbered ethernet0
POP-SR3120/configure/interface/bundle bldg1> exit
```

Configure interface ethernet 0

```
POP-SR3120/configure> interface ethernet 0
POP-SR3120/configure/interface/ethernet0> speed 100
full_duplex
POP-SR3120/configure/interface/ethernet0> ip address
10.1.1.2 255.255.255.0
POP-SR3120/configure/interface/ethernet0> exit
```

Configure in-band vlan forwarding table

```
POP-SR3120/configure> vlanfwd
POP-SR3120/configure/vlanfwd > add vlanid 4092 ethernet0
POP-SR3120/configure/vlanfwd > add vlanid 4092 bldg1
POP-SR3120/configure/vlanfwd > add vlanid 11-18
ethernet0
POP-SR3120/configure/vlanfwd > add vlanid 11-18 bldg1
POP-SR3120/configure/vlanfwd > management
POP-SR3120/configure/vlanfwd/management> vlanid 4092
POP-SR3120/configure/vlanfwd/management> disable_ipfwd
POP-SR3120/configure/vlanfwd/management> default_route
10.1.1.1 ethernet0
POP-SR3120/configure/vlanfwd/management> exit 2
```

Configure rate limiting for vlans

```
POP-SR3120/configure> interface bundle bldg1
POP-SR3120/configure/interface bundle bldg1> no
```

```
enable_cbq out
POP-SR3120/configure/interface bundle bldg1> add_class
mgmt-vlan root-out vlan_id 4092 cr 10 be 3072
POP-SR3120/configure/interface bundle bldg1> add_class
custA root-out vlan_id 11 cr 128 br 1024
POP-SR3120/configure/interface bundle bldg1> add_class
custB root-out vlan_id 12 cr 128 br 1024
POP-SR3120/configure/interface bundle bldg1> add_class
custC root-out vlan_id 13 cr 128 br 1024
POP-SR3120/configure/interface bundle bldg1> add_class
custD root-out vlan_id 14 cr 128 br 1024
POP-SR3120/configure/interface bundle bldg1> add_class
custE root-out vlan_id 15 cr 128 br 1024
POP-SR3120/configure/interface bundle bldg1> add_class
custF root-out vlan_id 16 cr 128 br 1024
POP-SR3120/configure/interface bundle bldg1> add_class
custG root-out vlan_id 17 cr 128 br 1024
POP-SR3120/configure/interface bundle bldg1> add_class
custH root-out vlan_id 18 cr 128 br 1024
POP-SR3120/configure/interface bundle bldg1> enable_cbq
out
POP-SR3120/configure/interface bundle bldg1> exit
```

Bldg1 configuration: Multilink T1 Router

```
Multilink_T1/configure> hostname bldg1-SR
bldg1-SR/configure> interface ethernet 0
bldg1-SR/configure/interface/ethernet0 > ip addr
10.1.1.3 255.255.255.0
bldg1-SR/configure> interface ethernet0> exit
```

Configure interface bundle uplink

```
bldg1-SR/configure> interface bundle uplink
bldg1-SR/configure/interface/bundle uplink> link t1 1-4
bldg1-SR/configure/interface/bundle uplink>
encapsulation ppp
bldg1-SR/configure/interface/bundle uplink> ip
unnumbered ethernet0
bldg1-SR/configure/interface/bundle uplink> exit
```

Configure inband VLAN forwarding table

```
bldg1-SR/configure/interface> vlanfwd
bldg1-SR/configure/interface/vlanfwd> add vlanid 4092
ethernet0
bldg1-SR/configure/interface/vlanfwd> add vlanid 4092
uplink
bldg1-SR/configure/interface/vlanfwd> add vlanid 11-18
ethernet0
bldg1-SR/configure/interface/vlanfwd> add vlanid 11-18
uplink
```

```
bldg1-SR/configure/interface/vlanfwd> management
bldg1-SR/configure/interface/vlanfwd/management> vlanid
4092
bldg1-SR/configure/interface/vlanfwd> disable_ipfwd
bldg1-SR/configure/interface/vlanfwd> default_route
10.1.1.1 uplink
bldg1-SR/configure/interface/vlanfwd> exit 2
```

Configure rate limiting for vlans

```
bldg1-SR/configure> interface bundle uplink
bldg1-SR/configure/interface/bundle uplink> qos
bldg1-SR/configure/interface/bundle uplink> no
enable_cbq
bldg1-SR/configure/interface/bundle uplink> add_class
mgmt-vlan root-out vlan_id 4092 cr 10 br 3072
bldg1-SR/configure/interface/bundle uplink> add_class
custA root-out vlan_id 11 cr 128 br 1024
bldg1-SR/configure/interface/bundle uplink> add_class
custB root-out vlan_id 12 cr 128 br 1024
bldg1-SR/configure/interface/bundle uplink> add_class
custC root-out vlan_id 13 cr 128 br 1024
bldg1-SR/configure/interface/bundle uplink> add_class
custD root-out vlan_id 14 cr 128 br 1024
bldg1-SR/configure/interface/bundle uplink> add_class
custE root-out vlan_id 15 cr 128 br 1024
bldg1-SR/configure/interface/bundle uplink> add_class
custF root-out vlan_id 16 cr 128 br 1024
bldg1-SR/configure/interface/bundle uplink> add_class
custG root-out vlan_id 17 cr 128 br 1024
bldg1-SR/configure/interface/bundle uplink> add_class
custH root-out vlan_id 18 cr 128 br 1024
bldg1-SR/configure/interface/bundle uplink> enable_cbq
bldg1-SR/configure/interface/bundle uplink> exit
```

Configure SNMP

```
bldg1-SR/configure> snmp
bldg1-SR/configure/snmp> community public ro
bldg1-SR/configure/snmp> system_id bldg1-SR
bldg1-SR/configure/snmp> trap_host 10.2.1.1 public
bldg1-SR/configure/snmp> exit
```

Chapter 42: WAN Interfaces

T1/E1

Secure Routers are available with T1/E1 WAN interfaces. Secure Router 1000 does not support Channelized T3 or DS3. Consult the SR1000 Installation Guide for details on WAN interface types, cabling, and pinouts.

This document outlines the configuration of module parameters (Layer 1) and, to a lesser degree, the configuration of bundle parameters (Layer 2). The bundle configuration examples demonstrate linking of physical interfaces (modules) to logical interfaces (bundles). Module configuration occurs within the configure module tree of the Secure Router CLI, and bundle configuration occurs within the configure interface bundle tree.

Secure Router T1 interfaces support logical interfaces made up of fractional T1, single T1, and multi-link T1 connections.

Module Configuration

T1

The following example configures the operational and descriptive parameters for T1 number 6.

Configure T1 Parameters

```
SR/configure> module t1 6
SR/configure/module/t1> circuitId X1234567890
SR/configure/module/t1> contactInfo George_Anderson
SR/configure/module/t1> description T1_to_Troy
SR/configure/module/t1> framing esf
SR/configure/module/t1> linecode b8zs
SR/configure/module/t1> clock_source line
SR/configure/module/t1> exit
```

Bundle Configuration

Configuration of an interface bundle is required for use of any of the Secure Router WAN interfaces. Multiple physical interfaces may be linked to a single interface bundle; multi-link protocols, including MLPPP and Multilink Frame Relay, make use of NxT1 interfaces to create single logical interfaces.

The interface bundle specifies the physical connection(s) to be linked, an encapsulation protocol (Layer 2) and, optionally, Layer 3 parameters.

Fractional T1

The following example creates a 384 Kbps fractional T1 bundle utilizing DS0s 1-3 and 8-10 of T1 number 3.

Configure a Fractional T1 HDLC Bundle

```
SR/configure> interface bundle demo 1
SR/configure/interface/bundle> link t1 3:1-3,8-10
SR/configure/interface/bundle> encap hdlc
SR/configure/interface/bundle> ip addr 10.1.1.1 255.255.255.252
SR/configure/interface/bundle> exit
```

T1

The following example creates a 1536 Kbps T1 bundle utilizing T1 number 4. This bundle uses IP unnumbered.

Configure a T1 PPP Bundle

```
SR/configure> interface bundle demo2
SR/configure/interface/bundle> link t1 4
SR/configure/interface/bundle> encap ppp
SR/configure/interface/bundle> ip unnumbered ethernet0
SR/configure/interface/bundle> exit
```

NxT1

The following example creates a 4.5 Mb/s N x T1 bundle utilizing T1s 6-8. MLPPP is not explicitly specified, a PPP bundle with two or more linked T1s uses the multi-link protocol by definition.

Configure an N x T1 MLPPP Bundle

```
SR/configure> interface bundle demo3
SR/configure/interface/bundle> link t1 6-8
SR/configure/interface/bundle> encap ppp
SR/configure/interface/bundle> ip addr
10.1.1.5 255.255.255.252
SR/configure/interface/bundle> exit
```

Chapter 43: Backup Interface-ISDN

1000-series routers support an ISDN (Integrated Services Digital Network) interface that functions as a backup interface. The advantage of this design is to provide an alternate WAN interface in the event that the T1/E1 leased line WAN interfaces are down. Dial-up ISDN is an established broadband technology that can be configured as a secondary connection (while the primary connections are down) to support small to medium sized remote site connections to enterprise locations as well as general Internet connectivity for groups of from 25 to 250 users.

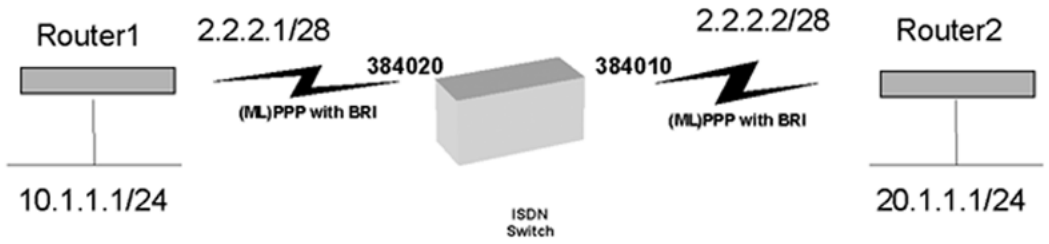
ISDN configured with a BRI interface supports two 64 Kbps channels or one 128Kbps connection. ISDN supports standard networking protocols such as Network Address Translation (NAT), PAP/CHAP, and PPP/MLPPP (for 128 Kbps channel configurations). ISDN supports dial on demand and bandwidth on demand capability.

ISDN as Primary Interface

ISDN interface can be used as Primary interface, similar to T1/E1 links with encapsulation of PPP or MLPPP. The bundle can consist of single BRI channel or two channels. Once the call is established & (ML) PPP negotiations are successful, the data can be transmitted similar to any other interface.

Configuring ISDN as a 128Kbps Primary Interface

The network topology map show below is used for this configuration.



To configure the ISDN interface as the primary 128Kbps interface:

1. Configure the ISDN switch type.

```
Router1/configure> isdn switch-type basic-ni
ISDN Switch type changed. Please reboot the box
for this change to take effect.
```

2. Save the configuration and reboot the router.

```
Router1/configure> exit
Router1> save local
```

3. Verify that the switch type is correct:

```
Router1> show isdn global ISDN Global Configuration
-------------------------
Switch Type: basic-ni Interface Type: U Interface
```

4. Configure a WAN bundle (refer to the illustration above).

```
Router1/configure> interface bundle wan1
Configuring new bundle
Router1/configure/interface/bundle wan1> link bri
128
Router1/configure/interface/bundle wan1> encapsulation
ppp Router1/configure/interface/bundle
wan1> ip address 2.2.2.1 24
```

5. Configure ISDN.

```
Router1/configure/interface/bundle wan1> isdn
Router1/configure/interface/bundle wan1/isdn>
callednum 384010
Router1/configure/interface/bundle wan1/isdn>
idle-timeout 5
Router1/configure/interface/bundle wan1/isdn>
connect-delay 10
Router1/configure/interface/bundle wan1/isdn>
spid1 3840200001
Router1/configure/interface/bundle wan1/isdn> tei
p
Please save and reboot the box for this change to
take effect
Router1/configure/interface/bundle wan1/isdn> exit
4
Router1> reboot
```

Note:
Spid configuration is optional for most switch types, but not for basic ni.

6. Verify the ISDN interfaces.

```
Router1> show isdn interfaces ISDN Information:
wan1
-------------------------
caller -
answer1 -
answer2 -
called-number 384010
spid1 3840200001
spid2 -
idle-timeout 5 minutes
connect-delay 10 seconds
keep-alive 10000 ms
disconnect-cause 17
```

```
tei point-to-point
tei-value 0
```

```
caller              -
answer1             -
answer2             -
called-number       384010
spid1               3840200001
spid2               -
idle-timeout        5 minutes
connect-delay       10 seconds
keep-alive          10000 ms
disconnect-cause    17
tei                 point-to-point
tei-value           0
```

7. Configure a route over the ISDN interface.

```
Router1/configure> ip route 20.1.1.0 24 wan1
```

Note:

You cannot use the peer IP address for this route. You must use the interface name for routes dedicated to ISDN.

8. Verify the IP interfaces.

```
Router1> show ip interfaces wan1 (unit number 4)
Type: S/W LOOPBACK
Flags: (0x20074243) UP, UNNUMBERED_S, RUNNING,
MULTICAST-ROUTE
Internet Address: 2.2.2.1
Internet Netmask: 255.255.255.0
Internet Broadcast: 2.2.2.255
Maximum Transfer Unit: 1500 bytes
Mac Address: 00:50:52:b2:c8:04
_wan1 (unit number 5)
Type: PT2PT
Flags: (0x1a07c203) UP, UNNUMBERED, MULTICAST-ROUTE
RED is enabled
Maximum Transfer Unit: 1500 bytes
Mac Address: 00:50:52:b2:c8:05
```

9. Display the ISDN statistics.

```
Router1> show isdn statistics BRI channel 0
statistics :
--------------------------
call not yet established
BRI channel 1 statistics :
--------------------------
call not yet established
D channel statistics--------------------
Bytes Rx: 3062
Frames Rx: 509
Error Frames Rx: 0
Bytes Tx: 2050
Frames Tx: 510
Fail Tx: 0
```

Note:
The ISDN call will get disconnected if there is no traffic passing through the ISDN interface for idle timeout period.

ISDN as backup Interface

ISDN interface can be used as Backup interface, for another primary interface. This ISDN backup interface will be tracking the primary interface. When the primary interface goes down, then this backup interface will dial and make a call establishment. Data, which is supposed to go through the primary interface, will be diverted and starts to flow through the ISDN backup interface. When the primary interface is back up and running, then the ISDN call gets disconnected and data starts to flow through the primary interface.

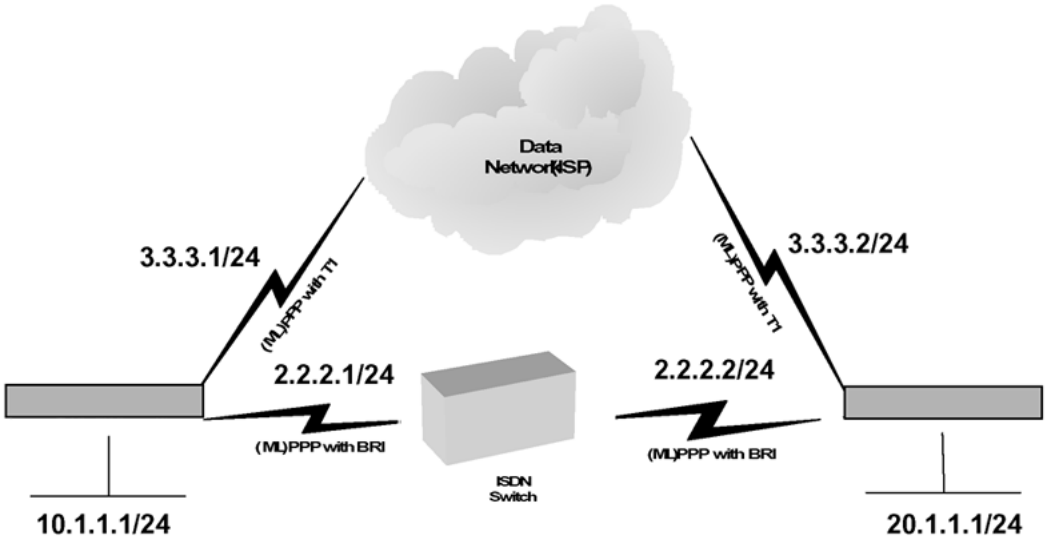
Primary interface might go down due to any of the following reasons:

- Physical links go down
- Negotiation failure occurs
- Link-drop feature (e.g., Link errors might exceed the configured thresholds)

Configuring ISDN as a 64Kbps Backup Interface

The network topology map show below is used for this configuration.

I



To configure the ISDN interface as the primary 128Kbps interface:

1. Configure the ISDN switch type).

```
Router1/configure> isdn switch-type basic-ni ISDN
Switch type changed. Please reboot the box for
this change to take effect.
```

2. Reboot the router.
3. Verify that the switch type is correct:

```
Router1> show isdn global
ISDN Global Configuration
-------------------------
Switch Type: basic-ni
Interface Type: U Interface
```

4. Configure a primary WAN bundle (refer to the illustration above).

```
Router1/configure> interface bundle wan1
Configuring new bundle
Router1/configure/interface/bundle wan1> link t1 1
Router1/configure/interface/bundle wan1>
encapsulation ppp
Router1/configure/interface/bundle wan1> ip
address 3.3.3.1 24
Router1/configure/interface/bundle wan1> exit
```

5. Configure a backup WAN bundle (refer to the illustration above).

```
Router1/configure> interface bundle wan2
Configuring new bundle
Router1/configure/interface/bundle wan2> link bri
64
Router1/configure/interface/bundle wan2> encapsulation
ppp
Router1/configure/interface/bundle
wan2> ip address 2.2.2.1 24
Router1/configure/interface/
bundle wan2> exit
```

6. Configure ISDN.

```
Router1/configure/interface/bundle wan2> isdn
Router1/configure/interface/bundle wan2/isdn>
callednum 384010
Router1/configure/interface/bundle
wan2/isdn> idle-timeout 5
Router1/configure/interface/
bundle wan2/isdn> connect-delay 10
Router1/configure/interface/bundle wan2/isdn>
spid1 3840200001
Router1/configure/interface/bundle
wan2/isdn> exit
```

Note:
Spid configuration is optional for most switch types, but not for basic ni.

7. Verify the ISDN interfaces.

```
Router1> show isdn interfaces
ISDN Information: wan1
--------------------------
caller -
answer1 -
```

```
answer2 -
called-number 384010
spid1 3840200001
spid2 -
idle-timeout 5 minutes
connect-delay 10 seconds
keep-alive 10000 ms
disconnect-cause 17
tei multipoint
tei-value 0
```

8. Configure a route over the ISDN interface.

```
Router1/configure> ip route 20.1.1.0 24 wan1
Router1/configure> ip route 20.1.1.0 24 wan2
metric 10
```

Note:

By assigning a lower priority to the backup interface, packets will always take the primary interface if it is available. Only when the primary interface fails will the ISDN interface become active.

Note:

You cannot use the peer IP address for this route. You must use the interface name for routes dedicated to ISDN.

9. Verify the IP interfaces.

```
Router1> show ip interfaces
Router1> show ip interfaces
wan1 (unit number 3)
Type: PT2PT
Flags: (0xa07c203) UP, MULTICAST-ROUTE
Internet Address: 3.3.3.1
Internet Netmask: 255.255.255.0
Internet Broadcast: 3.3.3.255
RED is enabled
Maximum Transfer Unit: 1500 bytes
Mac Address: 00:50:52:b2:c8:03
wan2 (unit number 4)
Type: S/W LOOPBACK
Flags: (0x20074243) UP, UNNUMBERED_S, RUNNING,
MULTICAST-ROUTE
Internet Address: 2.2.2.1
Internet Netmask: 255.255.255.0
Internet Broadcast: 2.2.2.255
Maximum Transfer Unit: 1500 bytes
Mac Address: 00:50:52:b2:c8:04
_wan2 (unit number 5)
Type: PT2PT
Flags: (0x1a07c203) UP, UNNUMBERED, MULTICASTROUTE
RED is enabled
Maximum Transfer Unit: 1500 bytes
Mac Address: 00:50:52:b2:c8:05
```

10. Display the ISDN statistics.

```
Router1> show isdn statistics BRI channel 0
statistics :
-------------------------
call not yet established BRI channel 1 statistics
:
```

```
-------------------------
call not yet established D channel statistics
--------------------
Bytes Rx: 3062
Frames Rx: 509
Error Frames Rx: 0 Bytes Tx: 2050
Frames Tx: 510
Fail Tx: 0
```

Note:

The ISDN call will get disconnected if there is no traffic passing through the ISDN interface for idle timeout period.

ISDN enhancements

The Secure Router 1001 provides support for multiple ISDN enhancements.

Unnumbered IP over ISDN interfaces will now be supported, as well as the ability to modify the ISDN parameters SWITCH TYPE, TEI TYPE, and TEI without having to reboot the router. The purpose of unnumbered IP over ISDN is to conserve IP addresses by borrowing an IP address from another configured interface. The unnumbered interface is the interface that borrows, and it should do so from an interface that is physically up and running. In the event the unnumbered interface attempts to borrow from an non-functioning interface, the unnumbered interface will not function.

The **`activate`** command is used to activate and change ISDN configurations. ISDN configurations not affected by this command are callingnum, callednum,, idle-timeout, and connect-delay.

Finally, the Secure Router 1001 provides support for configuration of the calling party number and various ISDN timers. In prior releases, the calling party number was provided automatically. This is now user configurable. Similarly, ISDN layer 2 and 3 timers were not configurable. This has changed, allowing the user to maximize performance of the ISDN network.

Three items users should note prior to implementation are:

- When configuring unnumbered IP over ISDN, the loopback interface cannot be used as the source.
- When users upgrade from pre-Release 9.3 they will need to reconfigure any existing ISDN parameters. This is a result of changes in the way ISDN is configured.
- Only static routing is supported on ISDN interface.

Configuring Unnumbered IP over ISDN BRI

Use the following procedure to configure unnumbered IP over ISDN BRI.

Procedure steps

1. Enter Configuration Mode.

   ```
   configure terminal
   ```

2. Enter Interface Mode.

   ```
   interface <interface>
   ```

3. Configure an IP address for the Ethernet interface.

   ```
   ip address <A.B.C.D>
   ```

4. Give a name to the ISDN interface.

   ```
   interface bundle <bundle_name>
   ```

5. Provide BRI link bandwidth.

   ```
   link bri <link-spec>
   ```

6. Add encapsulation.

   ```
   encapsulation <encap-type>
   ```

7. Borrow an IP address from the Ethernet interface.

   ```
   ip unnumbered <interface>
   ```

8. Enter the isdn sub-tree.

   ```
   isdn
   ```

9. Configure sub-tree commands.

   ```
   switch-type <type>
   idle-timeout <timeout>
   connect-delay <delay>
   callednum <number>
   tei <tei-type>
   ```

10. Activate the ISDN interface.

    ```
    activate
    ```

Table 37: Variable definitions

Variable	Value
<A.B.C.D>	The IP address.
<bundle_name>	The bundle name, maximum 8 characters.
<delay>	The connect delay in seconds, in the range 1 to 60.
<encap_type>	The encapsulation protocol. Only PPP is supported.
<interface>	The interface name.

Variable	Value
<link-spec>	BRI bandwidth, 64 or 128.
<number>	The called number, maximum 20 digits.
<tei-type>	The ISDN TEI type. Options are: • multipoint - Automatic TEI • point-to-point - Static TEI
<timeout>	The idle timeout in minutes, in the range 0 to 60. 0 disables the feature.
<type>	The ISDN switch type. Available options are: • basic-ni - National ISDN Switch Type • • basic-dms - NT DMS-100 switch type • basic-5ess - AT & T basic rate switch type (default) • basic-1tr6 - German 1tr6 switch type • basic-ntt - ntt switch type • basic-vn3 - French vn3 switch type • basic-etsi - ETSI [EURO] basic switch type • basic-ccitt - CCITT basic switch type

Multiple BRI bundles

You can now configure two 64 kb/s BRI bundles, which was not possible in previous releases. Both bundles should be configured identically.

Interface-based backup using ISDN

Interface-based backup is a feature which enables you to configure an ISDN interface as a backup for the primary WAN link. When the primary link goes down, the backup interface comes up. An ISDN call is triggered as soon as the primary WAN link goes down. Once the ISDN call is established, all traffic begins to flow through the ISDN interface with the static routes configured. When the primary link is restored, the ISDN call is dropped and the traffic passes through the primary link as it did before.

CLI

Use the following procedure to configure ISDN interface-based backup

1. Enter configuration mode:

   ```
   configure terminal
   ```

2. Specify the ISDN bundle to configure as the backup:

   ```
   interface bundle <name>
   ```

3. Specify ISDN configuration:

   ```
   isdn
   ```

4. Specify the WAN interface to back up:

   ```
   backup <bundle-name>
   ```

Example

The following shows an example configuration

```
Host/configure > interface bundle bri
configuring existing WAN bundle interface bri
Host/configure/interface/bundle bri > isdn
Host/configure/interface/bundle bri/isdn > backup ?
NAME
backup - Configure interface to backup (bundle name)
SYNTAX
backup bundle_name <cr>
DESCRIPTION
bundle_name -- bundle name to backup
(enter a word )
Host/configure/interface/bundle bri/isdn > backup wan
Warning: Idle timer will be disabled..
```

The preceding configuration configures the bundle **bri** as a backup for the bundle **wan** which is the primary link.

Time of day scheduling for ISDN

With time of day scheduling, you can configure the date and time for triggering any ISDN call. This feature allows you to configure the time schedule in two different ways: periodic and absolute. With periodic scheduling, you can configure a time range that reoccurs every week. With absolute scheduling, you can configure a specific time range on the calendar.

This feature can work with the backup feature. In this case, when the Serial interface is down, an ISDN call is triggered based on the configured schedule. Only if the current time is within the time range schedule that is configured on the bundle is the ISDN call triggered; otherwise, the ISDN call is not initiated.

CLI Display

The threshold for triggering the 2nd bundle can be configured using the following CLI.

```
Host/configure >time-range <time-range name>
NAME
time-range - configure time-range
SYNTAX
time-range timeRangeName <cr>
```

```
DESCRIPTION
timeRangeName -- Time-Range name, max 8 characters ( enter a word )
Host/configure/time-range test> ?
COMMANDS -- Any of the following commands can be used
absolute -- Configure specific scheduling for isdn
periodic -- Configure periodic scheduling for isdnHost/configure/time-range test >
absolute ?
NAME
absolute - Configure specific scheduling
SYNTAX
absolute start startdate starttime end enddate endtime
DESCRIPTION
start -- start
The parameter may have any of the following values:
start -- start
startdate -- start date in the format of dd/mm/yyyy
( enter a word )
starttime -- start time in the format of hh:mm (24 hours time format)
( enter a word )
end -- end
The parameter may have any of the following values:
end -- end
enddate -- end date in the format of dd/mm/yyyy
( enter a word )
endtime -- end time in the format of hh:mm (24 hours time format)
( enter a word )

Host/configure/time-range test > periodic ?
NAME
periodic - Configure periodic scheduling
SYNTAX
periodic days starttime to endtime <cr>
DESCRIPTION
days -- list of days : weekdays weekends, monday, tuesday,
wednessday, thursday, friday, saturday, sunday
The parameter may have any of the following values:
daily -- daily
weekdays -- weekdays
weekends -- weekends
monday -- monday
tuesday -- tuesday
wednessday -- wednessday
thursday -- thursday
friday -- friday
saturday -- saturday
sunday -- sunday
starttime -- start time in the format of hh:mm (24 hours time format)
( enter a word )
to -- time range
The parameter may have any of the following values:
to -- specify the end time
endtime -- end time in the format of hh:mm (24 hours time format)
( enter a word )
Host/configure/interface/bundle bri/isdn > trigger-schedule ?
NAME
trigger-schedule - Configure time schedule for ISDN
SYNTAX
trigger-schedule timeRangeName <cr>
DESCRIPTION
timeRangeName -- Time Range name for ISDN time scheduling
( enter a word )
```

Examples

The following example shows a sample periodic configuration.

```
configure# time-range periodic
configure/time-range periodic# periodic weekdays 9:00 to 20:30
configure/time-range periodic# exit
```

The following example shows a sample absolute configuration:

```
configure# time-range absolute
configure/time-range absolute# absolute start 17/07/2008 12:00 end 18/07/2008 12:45
configure/time-range absolute# exit
```

Filtering idle timeout with ISDN

The Secure Router can filter routing updates and keepalive packets so that these packets do no impact the idle timer for ISDN connections. A **`filter`** option in the ISDN command tree allows filters to be configured for incoming and outgoing packets. By default all the filtering is in the disabled state. You can enable filtering for any specific multicast protocol. On enabling the filtering for a particular protocol, keepalive and control packets specific to that protocol no longer impact the idle timer.

CLI

The following example shows the command options available for idle timeout filtering for ISDN.

```
Host/configure/interface/bundle bri/isdn > filter
Host/configure/interface/bundle bri/isdn/filter > ?
NAME
filter -- Configures the ISDN command
SYNTAX
COMMANDS <cr>
DESCRIPTION
COMMANDS -- Any of the following commands can be used
incoming -- Configure incoming filter
outgoing -- Configure outgoing filter
Host/configure/interface/bundle bri/isdn/filter > incoming ?
NAME
incoming - Configure incoming filter
SYNTAX
incoming enable <cr>
DESCRIPTION
enable -- enable or disable the filter for IGRP, OSPF, VRRP,
ICMP, IGMP, PIM, RIP, BGP
The parameter may have any of the following values:
enable -- enable
IGRP -- IGRP
OSPF -- OSPF
VRRP -- VRRP
ICMP -- ICMP
IGMP -- IGMP
PIM -- PIM
RIP -- RIP
BGP -- BGP
Host/configure/interface/bundle bri/isdn/filter > outgoing ?
NAME
outgoing - Configure outgoing filter
```

```
SYNTAX
outgoing enable <cr>
DESCRIPTION
enable -- enable or disable the filter for IGRP, OSPF, VRRP,
ICMP, IGMP, PIM, RIP, BGP
The parameter may have any of the following values:
enable -- enable
IGRP -- IGRP
OSPF -- OSPF
VRRP -- VRRP
ICMP -- ICMP
IGMP -- IGMP
PIM -- PIM
RIP -- RIP
BGP -- BGP
```

Numbering Plan And Type Of Number for ISDN

The Secure Router CLI provides options to configure the Numbering Plan and Type of Number. This allows you to select the Numbering Plan and Type of Number for the Called Party Number.

Use the following procedure to configure these parameters.

1. Enter configuration mode:

   ```
   configure terminal
   ```

2. Specify an existing ISDN bundle:

   ```
   interface bundle <name>
   ```

3. Specify ISDN configuration:

   ```
   isdn
   ```

4. To specify the numbering plan, enter:

   ```
   numplan <numplan>
   ```

5. To specify the type of number, enter:

   ```
   typeofnum <typeofnum>
   ```

Table 38: Variable definitions

Variable	Value
numplan <numplan>	Specifies the numbering plan. The <numplan> parameter can have any of the following values: • unknown: Unknown plan • isdn: ISDN/Telephony Numbering plan (default) • reserved: Telephony Numbering plan

Variable	Value
	• data: Data Numbering plan • telex: Telex Numbering plan • national: National Standard Numbering plan • privacy: Private Numbering plan
typeofnum <typeofnum>	Specifies the type of number. The parameter may have any of the following values: • unknown: Unknown type • international: International type (default) • national: National type • network: Network Specific type • subscriber: Subscriber type • abbreviated: Abbreviated type • reserved: Reserved value 5

Chapter 44: PPP Over Ethernet Client

PPPoE is a commonly used application in the deployment of DSL. One of the main advantages of PPPoE is that it offers authentication based on Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP).

The main purpose of PPPoE is to provide a back-up/fail-over solution. When primary connectivity goes down, traffic will switch-over to the back-up interface, in this case a virtual PPPoE interface. A PPPoE client session is established with a PPPoE server and traffic will be routed through this path until the primary connectivity is restored.

The following considerations are important in configuring PPPoE:

- This implementation will only support client PPPoE.
- There will be one client PPPoE interface for each physical Ethernet interface.
- PPPoE client will not support subinterfaces.
- PPPoE will drop any data packets it receives if it is not the primary interface.
- SNMP is not supported.
- This feature will only be available in routing mode.
- IPSec over PPPoE is supported when the address is assigned dynamically to the PPPoE interface.
- There is no NAT, ACL, VLAN supported on PPPoE interface.
- PPPoE will not run on parent Ethernet interface configured for VLAN encapsulation.
- PPPoE will not run on parent Ethernet interface configured as subinterface.
- QoS will not be supported on PPPoE virtual access interface in this release.
- The PPPoE Client implementation cannot be used to connect to two DSL connections for load sharing.
- Unicast and Multicast routing protocols are not supported on virtual access interfaces.
- PPPoE assumes that the netmask for IP address to be 30 bits wide when the address is negotiated, for example, supplied by the server.
- The PPPoE virtual access interface will be automatically configured in the Internet security zone for transit traffic to flow from the trusted/corp side the inbound interface needs to be configured as trusted.

Sample PPPoE Configuration

The following is a sample minimum configuration required:

```
1001> config terminal
1001/configure> interface ethernet 1
1001/configure/interface ethernet 1> ip address
192.168.20.101 24

1001/configure> interface virtual-access towan2
1001/configure/interface/virtual-access towan2> ip
negotiated
1001/configure/interface/virtual-access towan2>
protocol pppoe client
1001/configure/interface/virtual-access towan2>
pppoe ethernet 1
1001/configure/interface/virtual-access towan2>
ppp authentication pap sent-username test
password mypass

1001/configure> ip route 10.1.1.0 24 towan2 10
```

Note:
For security purposes, there must be a trusted interface for transit traffic.

Sample Configuration for Transit Traffic

```
1001> config terminal
1001/configure> interface ethernet 0
1001/configure/interface ethernet 0> ip address
10.10.1.1 24
1001/configure/interface ethernet 0> crypto
trusted

1001/configure> interface ethernet 1
1001/configure/interface ethernet 1> ip address
192.168.20.101 24
1001/configure> interface virtual-access towan2
1001/configure/interface/virtual-access towan2> ip
negotiated
1001/configure/interface/virtual-access towan2>
protocol pppoe client
1001/configure/interface/virtual-access towan2>
pppoe ethernet 1
1001/configure/interface/virtual-access towan2>
ppp authentication pap sent-username test
password mypass

1001/configure> ip route 10.1.1.0 24 towan2 10
```

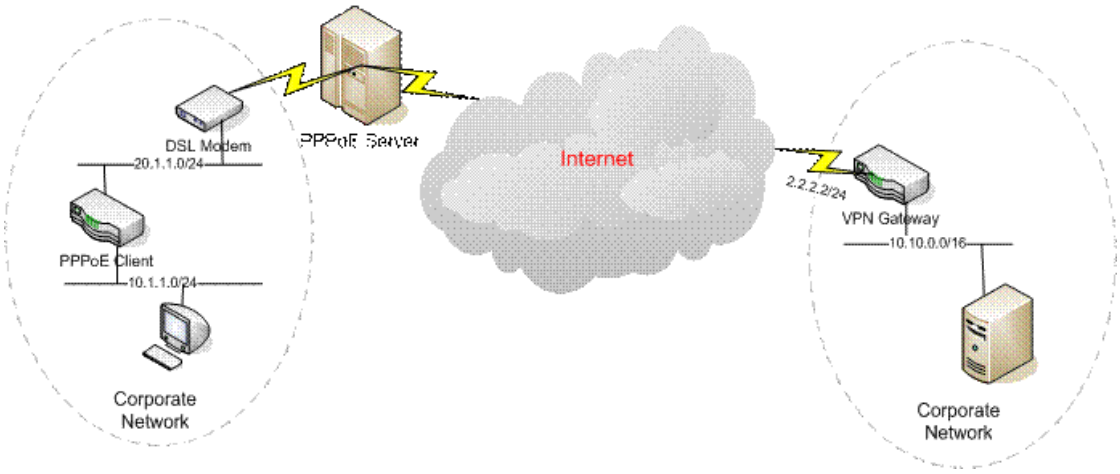
IPSec over PPPoE between two Secure Routers

The remote office uses PPPoE client for connecting Internet and by acquiring dynamic ip address from ISP. The IP address is assigned dynamically to the PPPoE interface. Remote Office to access corporate network of the headquarters uses site to site IPSec using following security suites.

Note:

This configuration is not currently supported with the ABOT feature in the VPN Router.

- Local address 0.0.0.0 (as PPPoE Clients obtains dynamic IP address)
- DES, SHA1, DH group 1 and Preshared key authentication method as Phase 1 security suite
- Aggressive mode
- 3DES, SHA1, tunnel mode and ESP as phase 2 security suite.



PPPoE Client Configuration

```
config term
# internet interface configuration
interface ethernet 0
ip address 20.1.1.2 255.255.255.0
crypto untrusted
exit
# trusted/corporate n/w interface configuration
interface ethernet 1
ip address 10.1.1.1 255.255.255.0
crypto trusted
exit
# PPPoE Client configuration
interface virtual-access test
```

```
ip negotiated
protocol pppoe
pppoe ethernet 0
ppp authentication pap sent-username test password test
exit
ip route 0.0.0.0 0.0.0.0 test 10
# IKE and IPSec policy configuration
crypto
ike policy test 2.2.2.2
local-id domain-name client.tasman.com
local-address 0.0.0.0
key testing12345
exit policy
ipsec policy test 2.2.2.2
match address 10.10.0.0 255.255.0.0 10.1.1.0
255.255.255.0
exit policy
exit crypto
```

Peer VPN Gateway configuration

```
#Corporate n/w interface configuration
interface ethernet 1
ip address 10.10.44.91 255.255.255.0
crypto trusted
exit ethernet
#internet interface configuration
interface bundle wan1
link t1 1
encapsulation ppp
ip address 2.2.2.2 24
crypto untrusted
exit
ip route 0.0.0.0 0.0.0.0 2.2.2.1 1
crypto
ike policy test 0.0.0.0
remote-id domain-name client.tasman.com
local-address 2.2.2.2
key testing12345
exit
ipsec policy test 0.0.0.0
match address 10.10.0.0 255.255.0.0 10.1.1.0
255.255.255.0
exit policy
exit crypto
```

IPSec over PPPoE between Secure router and Cisco

Note:
In this example, the Secure Router is configured as a PPPoE client.

Configure the Secure Router:

```
interface ethernet 0/1
ip address 50.1.1.1 255.255.255.0
qos
exit qos
crypto trusted
exit ethernet
interface ethernet 0/2
speed 100 full_duplex
ip address 192.168.29.101 255.255.254.0
qos
exit qos
crypto untrusted
exit ethernet
interface virtual-access pppoe1
ip negotiated
protocol pppoe
pppoe ethernet 0/2
ppp authentication pap sent-username opal password opla
ppp keepalive 0
exit virtual-access
hostname Hurricane
log utc
system display-boot-config no
system alarm_relay closed
system hdlc_error 1000 hold 20 min 900
system logging
console debugging
exit logging
ip
load_balance per_flow
route 30.1.1.0 255.255.255.0 pppoe1 1
route 2.2.2.0 255.255.255.0 pppoe1 1
route 0.0.0.0 0.0.0.0 pppoe1 1
exit ip
crypto
ike policy test 2.2.2.2
local-id domain-name client.tasman.com
local-address 0.0.0.0
key test12345
mode aggressive
exchange-type initiator-only
proposal 1
exit proposal
exit policy
ipsec policy test 2.2.2.2
match address 50.1.1.0 255.255.255.0 30.1.1.0
255.255.255.0
proposal 1 esp
exit proposal
anti-replay 32
exit policy
exit crypto
firewall global
algs
exit algs
max-connection-limit self 2048
exit firewall
firewall internet
nterface ethernet0/2 pppoe1
policy 101 in permit self
exit policy
exit firewall
firewall corp
```

```
interface ethernet0/1
policy 100 in permit
exit policy
policy 1024 out permit
exit policy
exit firewall
snmp-server
chassis-id Hurricane
trap-version 1
exit snmp-server
```

Configure IPSec:

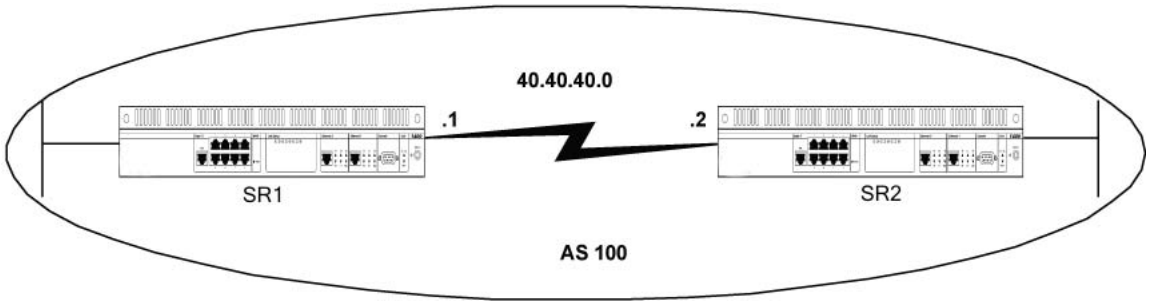
```
crypto isakmp policy 11
authentication pre-share
crypto isakmp key test12345 address 0.0.0.0 0.0.0.0
crypto isakmp key test12345 hostname client.tasman.com
crypto ipsec transform-set pppoe esp-3des esp-sha-hmac
crypto dynamic-map dynmap 5
set transform-set pppoe
crypto map dynmap2 20 ipsec-isakmp dynamic dynmap
interface
GigabitEthernet0/1
mtu 1500
ip address 30.1.1.1 255.255.255.0
duplex full
speed 100
interface Serial0/0/0:0
ip address 2.2.2.2 255.255.255.0
encapsulation ppp
crypto map dynmap2
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0/0:0
ip route 50.1.1.0 255.255.255.0 Serial0/0/0:0
ip http server
no ip http secure-server
access-list 108 permit ip 30.1.1.0 0.0.0.255 50.1.1.0
0.0.0.255
```

Chapter 45: Configuring BGP Features

Configuring IBGP Sessions

An IBGP Session is established between 2 BGP peers if they both belong to the same autonomous system number. They need not be directly connected to make any peer relationship. IBGP Sessions need to be fully meshed to get EBGP routes advertised to all peers in the autonomous system.

Configuring an IBGP Session between 2 Avaya Secure Routers.



CONFIGURATION OF Avaya1:

```
interface bundle ToAvaya2
link t1 4
encapsulation ppp
ip address 40.40.40.1 255.255.255.0
exit
router bgp 100
neighbor 40.40.40.2 100
exit
```

CONFIGURATION OF Avaya2:

```
interface bundle ToAvaya1
link t1 4
encapsulation ppp
ip address 40.40.40.2 255.255.255.0
exit
router bgp 100
neighbor 40.40.40.1 100
exit
```

The above configuration should bring up an IBGP Session between Avaya1 and Avaya2.

To verify the session status, use the command:

```
show ip bgp neighbors
```

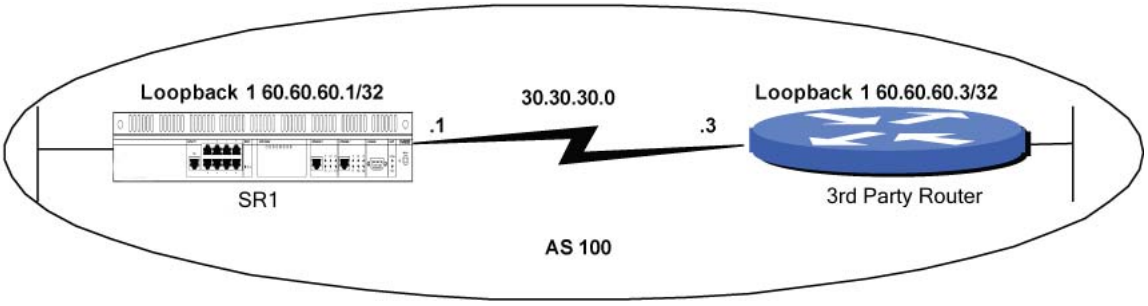
```
Avaya1> show ip bgp neighbors address 40.40.40.2
BGP neighbor is 40.40.40.2, peer group is BgpInternal
Remote AS 100, local AS 100, Group Type is Internal
BGP version 4, local router ID 60.60.60.5, remote
router ID 40.40.40.2
Current state = Established, up for 00:03:11
Last state = OpenConfirm, last event = RecvKeepAlive
Last error = None (subcode : None)
Configured Hold time is 180 seconds, keepalive interval
is 60 seconds
Active Holdtime is 180 seconds, keepalive interval is
60 seconds
Maximum prefixes is set to 50000 (warning : 41500)
Current number of prefixes from this neighbor is 0
Received 4 messages (57 bytes), 0 notifications, 0
updates
Sent 5 messages (105 bytes), 0 notifications, 0 updates
Connections established 1
Connection state is ESTABLISHED
Local host: 40.40.40.1, Local port: 179
Foreign host: 40.40.40.2, Foreign port: 1780
```

Group Type is Internal means that the session is an IBGP Session and the peer group is BgpInternal which is the default group the neighbor is assigned to by default in an IBGP session. Connection state in the above output shows it as ESTABLISHED, which means the two Avaya's have successfully formed a IBGP Connection.

```
Avaya1> show ip bgp summary

Avaya2> show ip bgp summary
```

```
Avaya1> show ip bgp summary

Avaya2> show ip bgp neighbors address 40.40.40.1
BGP neighbor is 40.40.40.1, peer group is BgpInternal
Remote AS 100, local AS 100, Group Type is Internal
BGP version 4, local router ID 40.40.40.2, remote
router ID 60.60.60.5
Current state = Established, up for 00:02:40
Last state = OpenConfirm, last event = RecvKeepAlive
Last error = None (subcode : None)
Configured Hold time is 180 seconds, keepalive interval
is 60 seconds
Active Holdtime is 180 seconds, keepalive interval is
60 seconds
Maximum prefixes is set to 50000 (warning : 41500)
Current number of prefixes from this neighbor is 0
Received 2 messages (38 bytes), 0 notifications, 0
updates
Sent 4 messages (86 bytes), 0 notifications, 0 updates
Connections established 1
Connection state is ESTABLISHED
Local host: 40.40.40.2, Local port: 1780
Foreign host: 40.40.40.1, Foreign port: 179

Avaya2> show ip bgp summary
```

Configuring an IBGP Session between an Avaya Router and a 3rd Party Router.



CONFIGURATION OF Avaya1:

```
interface bundle To3rdPartyRouter
link t1 4
encapsulation ppp
ip address 30.30.30.1 255.255.255.0
exit
router bgp 100
neighbor 30.30.30.3 100
exit
```

CONFIGURATION OF 3RD PARTY ROUTER:

```
interface Serial3/0
ip address 30.30.30.3 255.255.255.0
encapsulation ppp
exit
router bgp 100
neighbor 30.30.30.1 remote-as 100
exit
```

The above configuration should bring up an IBGP Session between Avaya1 and the 3rd party router.

To verify the session status, use the command:

```
Avaya1> show ip bgp neighbors
```

```
BGP neighbor is 30.30.30.3, peer group is BgpInternal
Remote AS 100, local AS 100, Group Type is Internal
BGP version 4, local router ID 60.60.60.5, remote
router ID 30.30.30.3
Current state = Established, up for 00:16:26
Last state = OpenConfirm, last event = RecvKeepAlive
Last error = Open Message Error (subcode : unsupported
optional parameter)
Configured Hold time is 180 seconds, keepalive interval
is 60 seconds
Active Holdtime is 180 seconds, keepalive interval is
60 seconds
Maximum prefixes is set to 50000 (warning : 41500)
Current number of prefixes from this neighbor is 0
Received 18 messages (323 bytes), 0 notifications, 0
updates
```

```
Sent 20 messages (390 bytes), 1 notifications, 0
updates
Connections established 1
Connection state is ESTABLISHED
Local host: 30.30.30.1, Local port: 179
Foreign host: 30.30.30.3, Foreign port: 11000
```

Group Type is Internal means that the session is an IBGP Session and the peer group is BgpInternal which is the default group the neighbor is assigned to by default in an IBGP session. Connection state in the above output shows it as ESTABLISHED, which indicates the two Avaya's have successfully formed a IBGP Connection.

```
Avaya1> show ip bgp summary
```

The following 3rd party router outputs shows internal link , which means neighbor 30.30.30.1 is an IBGP neighbor and the BGP State shows it as ESTABLISHED.

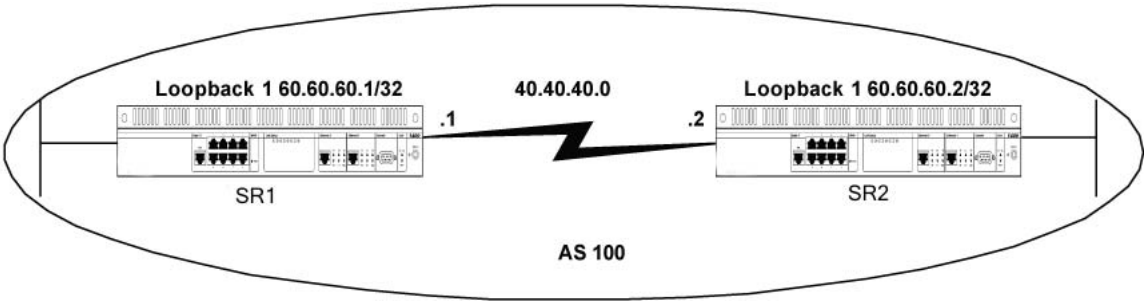
```
3rdPartyRouter>show ip bgp neighbors
BGP neighbor is 30.30.30.1, remote AS 100, internal
link
BGP version 4, remote router ID 60.60.60.5
BGP state = Established, up for 00:17:07
Last read 00:00:36, hold time is 180, keepalive
interval is 60 seconds
Received 23 messages, 1 notifications, 0 in queue
Sent 22 messages, 0 notifications, 0 in queue
Route refresh request: received 0, sent 0
Minimum time between advertisement runs is 5 seconds
For address family: IPv4 Unicast
BGP table version 1, neighbor version 1
Index 1, Offset 0, Mask 0x2
0 accepted prefixes consume 0 bytes
Prefix advertised 0, suppressed 0, withdrawn 0
Connections established 1; dropped 0
Last reset 00:17:13, due to BGP Notification received,
unsupported optional parameter
Connection state is ESTAB, I/O status: 1, unread input
bytes: 0
Local host: 30.30.30.3, Local port: 11000
Foreign host: 30.30.30.1, Foreign port: 179
```

```
3rdPartyRouter>show ip bgp summary
```

Note:

If the State is not in "Established" condition, make sure the peer ip address is reachable by doing a PING to the neighbor address.

Configuring an IBGP Multi-Hop Session between 2 Avaya Secure Routers



CONFIGURATION OF Avaya1:

```
interface bundle ToAvaya2
link t1 4
encapsulation ppp
ip address 40.40.40.1 255.255.255.0
exit
interface loopback 1
ip address 60.60.60.1 255.255.255.255
exit
ip route 60.60.60.2 255.255.255.255 40.40.40.2 1
router bgp 100
neighbor 60.60.60.2 100
exit
exit
```

CONFIGURATION OF Avaya2:

```
interface bundle ToAvaya1
link t1 4
encapsulation ppp
ip address 40.40.40.2 255.255.255.0
exit
interface loopback 1
p address 60.60.60.2 255.255.255.255
exit
ip route 60.60.60.1 255.255.255.255 40.40.40.2 1
router bgp 100
neighbor 60.60.60.1 100
exit
```

Note in the above configuration we have added an ip route command to reach the other side loopback interface. We need to have a route to reach the bgp peer address, either through a static route or through any other protocol like rip / ospf.

Now we can ping the peer address 60.60.60.1 from Avaya2, but the "show ip bgp summary" still shows the connection to be in Active state.

Reachability to the peer address is achieved, but the session is still in an Active state. The BGP Session is not established because there is one thing that is missing still. When BGP Initiates a connection with another peer, it would always use its outgoing interface as its source address. In this case Avaya2 would use 40.40.40.2 and Avaya1 would use 40.40.40.1. But

BGP is configured with neighbor address as 60.60.60.1 in Avaya2 and 60.60.60.2 in Avaya1 instead of 40.40.40.1 and .2. So we need to instruct BGP to use 60.60.60.1 and .2 as source address instead of 40.40.40.x

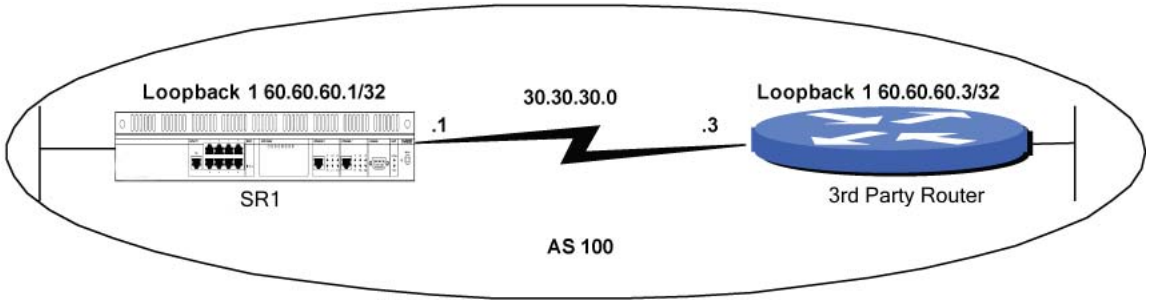
By putting an update_source command under the neighbor, BGP would start using the 60.60.60.x address.

```
Avaya1> conf term
Avaya1/configure> router bgp 100
Avaya1/configure/router/bgp
100> neighbor 60.60.60.2 100
Avaya1/configure/router/bgp 100/neighbor
60.60.60.2 100> update_source 60.60.60.1

Avaya2> conf term
Avaya2/configure> router bgp 100
Avaya2/configure/router/bgp
100> neighbor 60.60.60.1 100
Avaya2/configure/router/bgp 100/neighbor
60.60.60.1 100> update_source 60.60.60.2
```

Examine the session status now:

```
Avaya1> show ip bgp summary

Avaya2> show ip bgp summary
```

Configuring an IBGP Multi-Hop Session between an Avaya Router and a 3rd Party Router



CONFIGURATION OF Avaya1

```
interface  bundle To3rdPartyRouter
link  t1 4
encapsulation ppp
ip address 30.30.30.1 24
exit
interface  loopback 1
ip  address 60.60.60.2 32
exit
ip route 60.60.60.3 32 30.30.30.3 1
router  bgp 100
neighbor60.60.60.3 100
neighbor 60.60.60.3 100> update_source 60.60.60.1
exit
```

CONFIGURATION OF 3RD PARTY ROUTER

```
interface Loopback1
ip address 60.60.60.3 255.255.255.255
interface Serial3/0
ip address 30.30.30.3 255.255.255.0
encapsulation ppp
exit
ip route 60.60.60.1 255.255.255.255 30.30.30.1 1
router bgp 100
neighbor 60.60.60.1 remote-as 100
neighbor 60.60.60.1 update-source loopback 1
exit
```

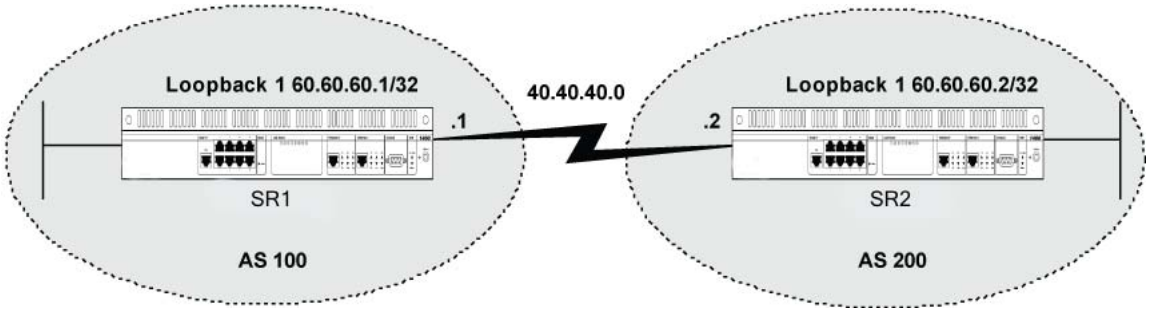
By adding update-source on Avaya and the 3rdPartyRouter we could establish an IBGP session between Avaya and 3rdPartyRouter. Show ip bgp summary on 3rdPartyRouter shows the State/PrefixRcd as 0 which could be Idle/Active otherwise.

```
3rdPartyRouter>show ip bgp summary
```

Configuring EBGP Sessions

An EBGP Session is established between 2 BGP peers if they belong to two different autonomous system numbers. They need to be directly connected to make a peer relationship. If an EBGP Peer is not directly connected and it is of Multi-hops away, it has to be specially configured under that neighbor to take care of peer relationship.

Configuring an EBGP Session between 2 Avaya Secure Routers



CONFIGURATION OF Avaya1:

```
interface  bundle ToAvaya2
link  t1 4
encapsulation ppp
ip  address 40.40.40.1 255.255.255.0
exit
router bgp 100
neighbor 40.40.40.2 200
```

CONFIGURATION OF Avaya2:

```
interface  bundle ToAvaya1
link  t1 4
encapsulation ppp
ip  address 40.40.40.2 255.255.255.0
exit
router bgp 200
neighbor 40.40.40.1 100
```

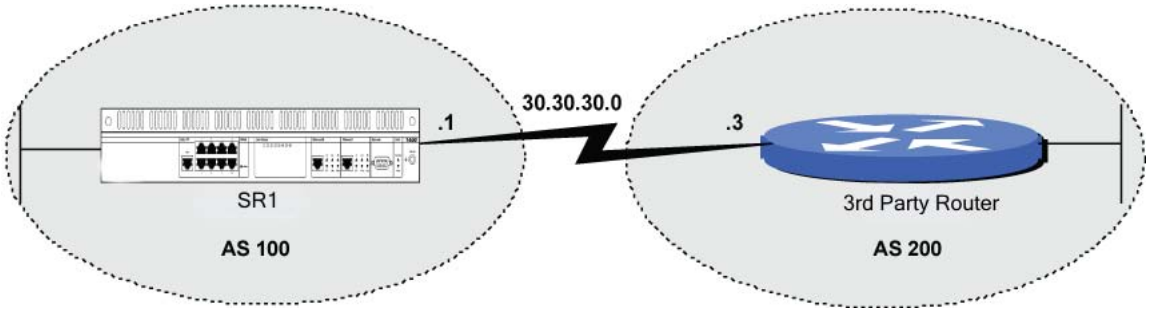
The above configuration should bring up an EBGP Session between Avaya1 and Avaya2.

```
Avaya1> show ip bgp summary
```

```
BGP neighbor is 40.40.40.2, peer group is BgpExternal
Remote AS 200, local AS 100, Group Type is External
BGP version 4, local router ID 60.60.60.5, remote
router ID 60.60.60.2
Current state = Established, up for 00:01:36
Last state = OpenConfirm, last event = RecvKeepAlive
Last error = None (subcode : None)
Configured Hold time is 180 seconds, keepalive interval
is 60 seconds
Active Holdtime is 180 seconds, keepalive interval is
60 seconds
Maximum prefixes is set to 50000 (warning : 41500)
Current number of prefixes from this neighbor is 0
Received 2 messages (19 bytes), 0 notifications, 0
updates
Sent 3 messages (67 bytes), 0 notifications, 0 updates
Connections established 1
Connection state is ESTABLISHED
Local host: 40.40.40.1, Local port: 179
Foreign host: 40.40.40.2, Foreign port: 1801
```

```
Avaya2> show ip bgp summary
```

Configuring an EBGP Session between an Avaya Router and a 3rd Party Router



CONFIGURATION OF Avaya1:

```
interface  bundle To3rdPartyRouter
link  t1 4
encapsulation ppp
ip address 30.30.30.1 255.255.255.0
exit
```

```
router bgp 100
neighbor 30.30.30.3 200
```

CONFIGURATION OF 3RDPARTYROUTER:

```
interface Serial3/0
ip address 30.30.30.3 255.255.255.0
encapsulation ppp
exit
router bgp 200
neighbor 30.30.30.1 remote-as 100
```

The above configuration should bring up an EBGP Session between Avaya1 and 3rdPartyRouter.

View the neighbor status:

```
Avaya1> show ip bgp neighbors
BGP neighbor is 30.30.30.3, peer group is BgpExternal
Remote AS 200, local AS 100, Group Type is External
BGP version 4, local router ID 60.60.60.5, remote
router ID 60.60.60.3
Current state = Established, up for 00:01:20
Last state = OpenConfirm, last event = RecvKeepAlive
Last error = Open Message Error (subcode : unsupported
optional parameter)
Configured Hold time is 180 seconds, keepalive interval
is 60 seconds
Active Holdtime is 180 seconds, keepalive interval is
60 seconds
Maximum prefixes is set to 50000 (warning : 41500)
Current number of prefixes from this neighbor is 0
Received 3 messages (38 bytes), 0 notifications, 0
updates
Sent 3 messages (67 bytes), 1 notifications, 0 updates
Connections established 1
Connection state is ESTABLISHED
Local host: 30.30.30.1, Local port: 179
Foreign host: 30.30.30.3, Foreign port: 11048
```

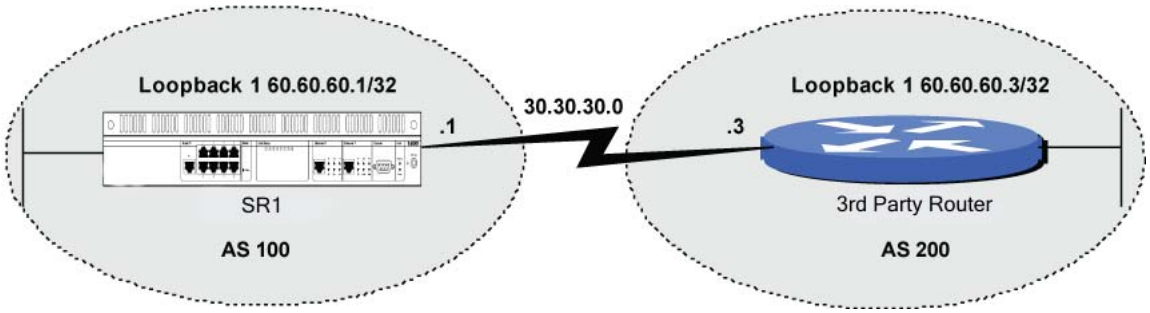
Group Type is External means that the session is an EBGP Session and the peer group is BgpExternal which is the default group the neighbor is assigned to by default in an EBGP session. Connection state in the above output shows it as ESTABLISHED, which means the Avaya and 3rdPartyRouter have successfully formed a EBGP Connection.

The following 3rdPartyRouter outputs shows external link, which means neighbor 30.30.30.1, is an EBGP neighbor and the BGP State shows it as ESTABLISHED.

```
3rdPartyRouter>show ip bgp neighbors
BGP neighbor is 30.30.30.1, remote AS 100, external link
BGP version 4, remote router ID 60.60.60.5
BGP state = Established, up for 00:01:33
Last read 00:00:33, hold time is 180, keepalive interval is 60 seconds
Received 5 messages, 1 notifications, 0 in queue
Sent 6 messages, 0 notifications, 0 in queue
Route refresh request: received 0, sent 0
Minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
BGP table version 1, neighbor version 1
Index 1, Offset 0, Mask 0x2
0 accepted prefixes consume 0 bytes
Prefix advertised 0, suppressed 0, withdrawn 0
```

```
Connections established 1; dropped 0
Last reset 00:02:02, due to BGP Notification received, unsupported optional
parameter
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 30.30.30.3, Local port: 11048
Foreign host: 30.30.30.1, Foreign port: 179
```

Configuring an EBGP Multi-Hop Session between an Avaya Router and a 3rd Party Router



CONFIGURATION OF Avaya1:

```
interface  bundle To3rdPartyRouter
link  t1 4
encapsulation ppp
ip address 30.30.30.1 255.255.255.0
exit
interface  loopback 1
ip  address 60.60.60.2 32
exit
ip route 60.60.60.3 32 30.30.30.3 1
conf term
router bgp 100
neighbor 60.60.60.3 200
ebgp_multihop
update_source 60.60.60.1
```

On an IBGP Multihop session we need to take care of only the update_source for getting the BGP to an Established state, but in case of EBGP neighbors we have to specify the session to be EBGP_MULTIHOP in their neighbor configuration itself.

CONFIGURATION OF 3RDPARTYROUTER:

```
interface Loopback1
ip address 60.60.60.3 255.255.255.255
interface Serial3/0
ip address 30.30.30.3 255.255.255.0
encapsulation ppp
exit
ip route 60.60.60.1 255.255.255.255 30.30.30.1 1
conf term
router bgp 200
neighbor 60.60.60.1 remote-as 100
neighbor 60.60.60.1 ebgp-multihop
neighbor 60.60.60.1 update-source loopback 1
```

The above configuration should bring up an EBGP Session over multi-hop between Avaya1 and 3rdPartyRouter.

View the neighbor status:

```
Avaya1> show ip bgp neighbors
```

```
BGP neighbor is 60.60.60.3, peer group is BgpExternalRt
Remote AS 200, local AS 100, Group Type is External
Routing
BGP version 4, local router ID 60.60.60.5, remote router ID 60.60.60.3
Current state = Established, up for 00:01:23
Last state = OpenConfirm, last event = RecvKeepAlive
Last error = Open Message Error (subcode : unsupported
optional parameter)
Configured Hold time is 180 seconds, keepalive interval
is 60 seconds
Active Holdtime is 180 seconds, keepalive interval is
60 seconds
Maximum prefixes is set to 128000 (warning : 106240)
Current number of prefixes from this neighbor is 0
External BGP neighbor may be multi hops away
Received 2 messages (38 bytes), 0 notifications, 0
updates
Sent 5 messages (117 bytes), 1 notifications, 0 updates
Connections established 1
Connection state is ESTABLISHED
Local host: 60.60.60.1, Local port: 1049
Foreign host: 60.60.60.3, Foreign port: 179
```

Group Type is External means that the session is an EBGP Session and the peer group is BgpExternalRt which is the default group the neighbor is assigned to by default in an EBGP session if the remote peers is not directly connected and it is of multi-hops away. External BGP neighbor may be multi hops away in the above output means that the remote peer session is an EBGP Multi-Hop session.

Connection state in the above output shows it as ESABLISHED, which means the Avaya and 3rdPartyRouter have successfully formed an EBGP Connection.

By default Avaya puts its neighbor in to one of the three groups:

- IBGP Neighbor > BgpInternal
- EBGP Neighbor > BgpExternal
- EBGP Multi-Hop Neighbor > BgpExternal_Rt

The following 3rdPartyRouter output shows external link, in this case neighbor 30.30.30.1, is an EBGP neighbor and the BGP State shows it as ESTABLISHED.

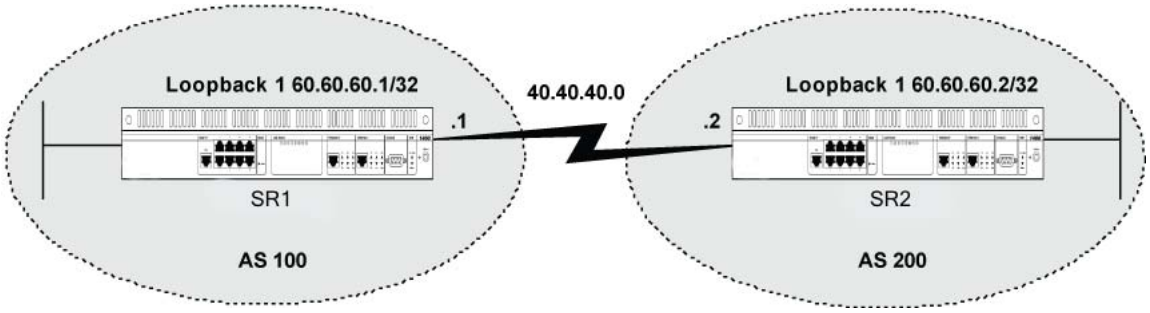
```
3rdPartyRouter>show ip bgp neighbors
```

```
BGP neighbor is 60.60.60.1, remote AS 100, external
link
BGP version 4, remote router ID 60.60.60.5
BGP state = Established, up for 00:01:41
Last read 00:00:40, hold time is 180, keepalive
interval is 60 seconds
Received 5 messages, 1 notifications, 0 in queue
Sent 9 messages, 0 notifications, 0 in queue
Route refresh request: received 0, sent 0
```

```
Minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
BGP table version 1, neighbor version 1
Index 2, Offset 0, Mask 0x4
0 accepted prefixes consume 0 bytes
Prefix advertised 0, suppressed 0, withdrawn 0
Connections established 1; dropped 0
Last reset 00:01:51, due to Peer closed the session
External BGP neighbor may be up to 255 hops away
Connection state is ESTAB, I/O status: 1, unread input
bytes: 0
Local host: 60.60.60.3, Local port: 179
Foreign host: 60.60.60.1, Foreign port: 1049
```

External BGP neighbor may be up to 255 hops away in the above output means that the remote neighbor is an EBGP-Multi-hop neighbor. EBGP Peers over an unnumbered WAN interface needs to be configured as EBGP MULTI-HOP only.

Configuring an EBGP Multi-Hop Session between 2 Avaya Secure Routers



CONFIGURATION OF Avaya1:

```
interface  bundle ToAvaya2
link  t1 4
encapsulation ppp
ip  address 40.40.40.1 255.255.255.0
exit
interface  loopback 1
ip  address 60.60.60.1 255.255.255.255
exit
ip route 60.60.60.2 255.255.255.255 40.40.40.2 1
conf term
router bgp 100
neighbor 60.60.60.2 200
update_source 60.60.60.1
ebgp_multihop
```

On an IBGP Multihop session we need to take care of only the update_source for getting the BGP to an Established state, but in case of EBGP neighbors we have to specify the session to be EBGP_MULTIHOP in their neighbor configuration itself.

CONFIGURATION OF Avaya2:

```
interface bundle ToAvaya1
link t1 4
```

```
encapsulation ppp
ip address 40.40.40.2 255.255.255.0
exit
interface loopback 1
ip address 60.60.60.2 255.255.255.255
exit
ip route 60.60.60.1 255.255.255.255 40.40.40.2 1
conf term
router bgp 200
neighbor 60.60.60.1 100
update_source 60.60.60.2
ebgp_multihop
```

The above configuration should bring up an EBGP Session over multi-hop between Avaya1 and Avaya2.

View the neighbor status:

```
Avaya1> show ip bgp neighbors
```

```
Avaya1> show ip bgp neighbors
BGP neighbor is 60.60.60.2, peer group is BgpExternalRt
Remote AS 200, local AS 100, Group Type is External
Routing
BGP version 4, local router ID 60.60.60.5, remote
router ID 60.60.60.2
Current state = Established, up for 00:00:41
Last state = OpenConfirm, last event = RecvKeepAlive
Last error = None (subcode : None)
Configured Hold time is 180 seconds, keepalive interval
is 60 seconds
Active Holdtime is 180 seconds, keepalive interval is
60 seconds
Maximum prefixes is set to 128000 (warning : 106240)
Current number of prefixes from this neighbor is 0
External BGP neighbor may be multi hops away
Received 0 messages (0 bytes), 0 notifications, 0
updates
Sent 2 messages (48 bytes), 0 notifications, 0 updates
Connections established 1
Connection state is ESTABLISHED
Local host: 60.60.60.1, Local port: 1053
Foreign host: 60.60.60.2, Foreign port: 179
```

```
Avaya2> show ip bgp summary
```

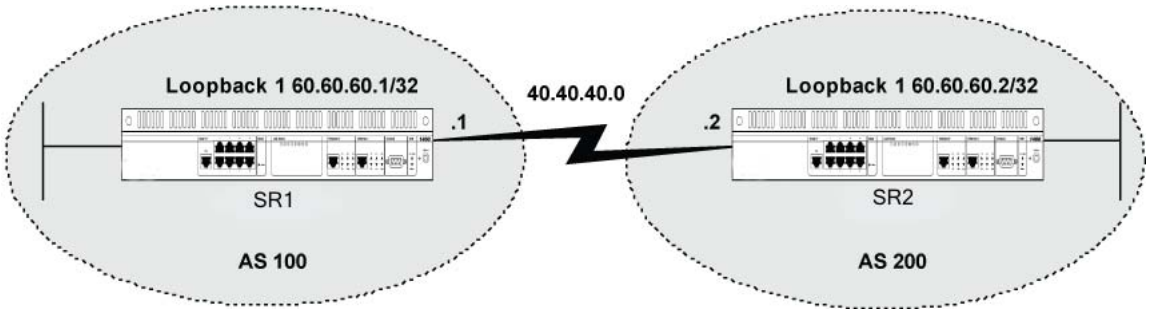
```
Avaya2> show ip bgp neighbors
```

```
BGP neighbor is 60.60.60.1, peer group is BgpExternalRt
Remote AS 100, local AS 200, Group Type is External
Routing
BGP version 4, local router ID 60.60.60.2, remote
router ID 60.60.60.5
Current state = Established, up for 00:00:59
Last state = OpenConfirm, last event = RecvKeepAlive
Last error = None (subcode : None)
Configured Hold time is 180 seconds, keepalive interval
is 60 seconds
Active Holdtime is 180 seconds, keepalive interval is
60 seconds
Maximum prefixes is set to 128000 (warning : 106240)
Current number of prefixes from this neighbor is 0
External BGP neighbor may be multi hops away
Received 1 messages (0 bytes), 0 notifications, 0
```

```
updates
Sent 2 messages (48 bytes), 0 notifications, 0 updates
Connections established 1
Connection state is ESTABLISHED
Local host: 60.60.60.2, Local port: 179
Foreign host: 60.60.60.1, Foreign port: 1053
```

Group Type is External means that the session is an EBGP Session and the peer group is BgpExternalRt which is the default group the neighbor is assigned to by default in an EBGP session if the remote peers is not directly connected and it is of multi-hops away. Connection state in the above output shows it as ESABLISHED, which means the Avaya and 3rdPartyRouter have successfully formed an EBGP Connection. External BGP neighbor may be multi hops away in the above output means that the remote peer session is an EBGP Multi-Hop session.

Clearing BGP Sessions



A BGP Session needs to be cleared if there is any policy change or to bring up a peer which is already in an IDLE state. In Avaya BGP implementation a BGP Session can be cleared with respect to a peer, or multiple peers in a group together or all the peers in that particular unit.

Avaya1 and Avaya2 are in ESTABLISHED state. Clear that session by specifying the peer ip address of Avaya2, which is 60.60.60.2.

```
Avaya1> clear ip bgp neighbor 60.60.60.2 200
```

This would clear the session between Avaya1 and Avaya2. In the above command 60.60.60.2 is the remote peer address and 200 is the AS number. By clearing a bgp peer group, all the neighbor sessions belonging to that group will get cleared.

Show ip bgp groups command displays all the groups and the peers assigned to each group.

```
Avaya1> show ip bgp groups
```

```
BGP group is EBGP-1
group type is External Routing, total peers 2,
established peers 2, members: 60.60.60.2 100.1.1.3
options set :
None
BGP group is BgpExternal
group type is External, total peers 1, established
peers 1, members: 30.30.30.3
options set :
```

```
None
BGP group is BgpExternalRt
group type is External Routing, total peers 0,
established peers 0
options set :
None
BGP group is BgpInternal
group type is Internal, total peers 0, established
peers 0
options set :
None
```

In the above show command output, 60.60.60.2 and 100.1.1.3 belong to EBGP-1 group and 30.30.30.3 belong to BgpExternal group, which is a default group for external peers. Now to clear peer sessions 60.60.60.2 and 100.1.1.3, we can issue a clear command on that group instead of giving it individually.

```
Avaya1> clear ip bgp group EBGP-1
```

The above command should clear bgp sessions with 60.60.60.2 and 100.1.1.3

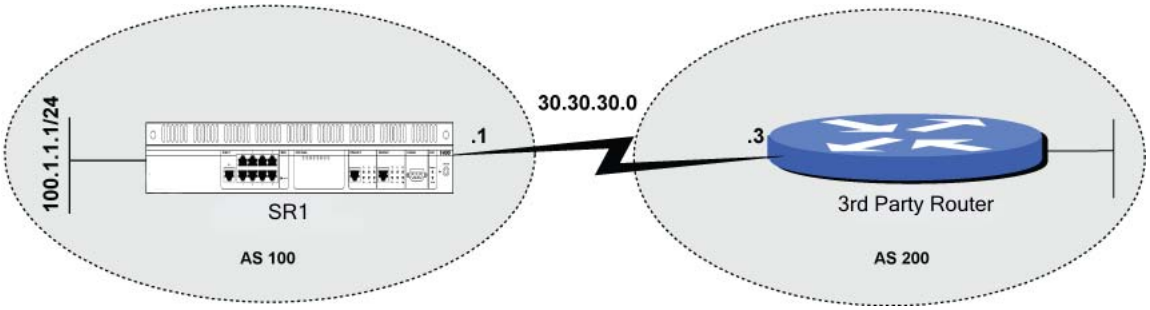
All bgp sessions can be cleared by giving the command:

```
Avaya1> clear ip bgp all
```

Configuring Advertising Routes to BGP

BGP announces routes to its IBGP and EBGP Peers connected to it. Even though it receives routes from multiple peers, it pickups the best route and announces to its other peers. It runs a PATH calculation algorithm in selecting the best routes. In Avaya implementation we have different ways of advertising routes to other peers, redistribution of static , connected, OSPF and RIP routes.

Announcing Static routes to BGP




```
Avaya1> show ip bgp summary
```

Avaya1 has an EBGP session with 3rdPartyRouter in ESTABLISHED state.

Avaya1's Ethernet interface is configured with an ip address of 100.1.1.1/24. Configure a static route of 200.1.1.0/24 pointing to 100.1.1.254 on Avaya1.

```
Avaya1> conf term Avaya1/configure> ip route 200.1.1.0 24 100.1.1.254
1
```

View the bgp table of 3rdPartyRouter to check the routes announced by Avaya1.

```
3rdPartyRouter>show ip bgp
```

Announce the static route 200.1.1.0 on to BGP

```
Avaya1/configure> router bgp 100
```

```
Avaya1/configure/router/bgp 100> redistribute static
```

Examine the bgp table of 3rdPartyRouter to see whether Avaya1 has announced any routes.

```
3rdPartyRouter>show ip bgp
```

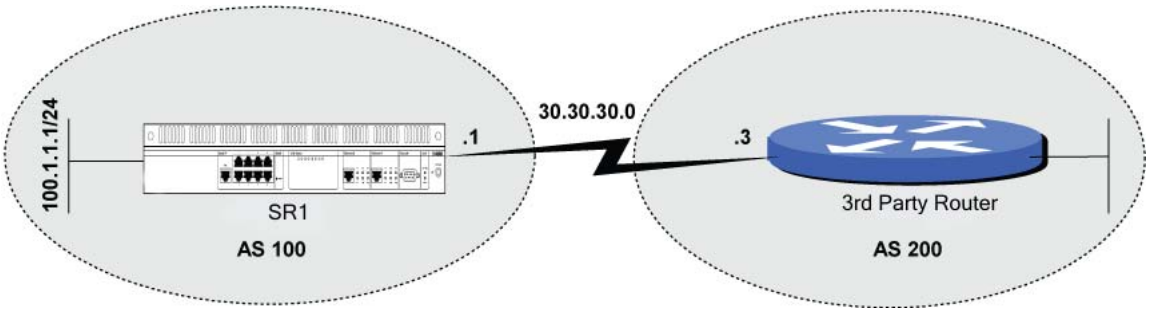
Pull back the route 200.1.1.0 announced to 3rdPartyRouter by doing a no redistribute static on Avaya1.

```
Avaya1/configure/router/bgp 100> no redistribute static
```

Examine the bgp table of 3rdPartyRouter to see whether we have any routes.

```
3rdPartyRouter> show ip bgp
```

Announcing Connected routes to BGP




```
Avaya1> show ip bgp summary
```

Avaya1 has an EBGP session with 3rdPartyRouter in ESTABLISHED state. Avaya1's Ethernet interface is configured with an ip address of 100.1.1.1/24. Lets announce the connected interface route of 100.1.1.0/24 to 3rdPartyRouter through redistribution.

View the bgp table of 3rdPartyRouter to check the routes announced by Avaya1.

```
3rdPartyRouter>show ip bgp
```

Announce the connected interface route 100.1.1.0 on to BGP

```
Avaya1/configure> router bgp 100

Avaya1/configure/router/bgp 100> redistribute connected
```

Examine the bgp table of 3rdPartyRouter to see Avaya1 has announced any routes.

```
3rdPartyRouter>show ip bgp
```

3rdPartyRouter has two routes which are announced by 30.30.30.1 (Avaya1). 100.1.1.0/24 is the Ethernet interface connected route and 30.30.30.0/24 is the wan bundle interface route.

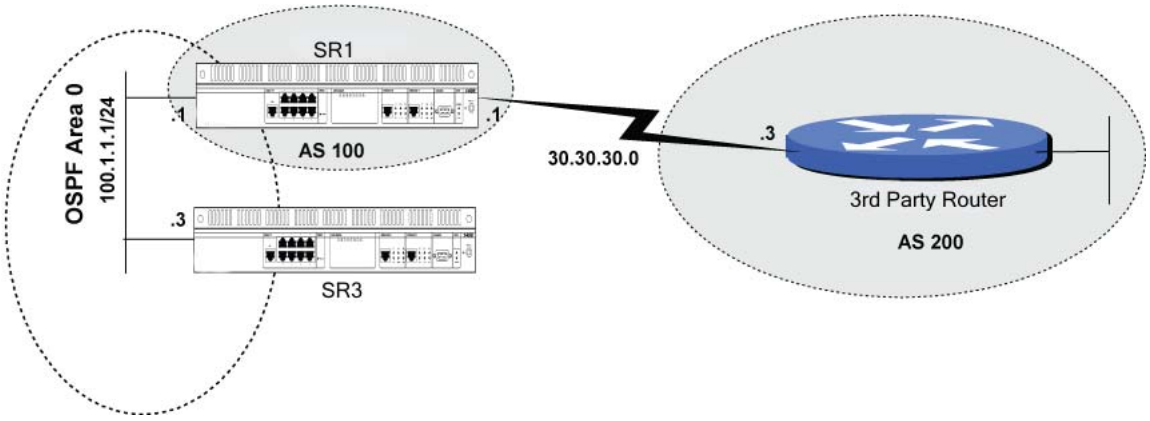
Pull back the routes 30.30.30.0 and 100.1.1.0 announced to 3rdPartyRouter by doing a no redistribute connected on Avaya1.

```
Avaya1/config term>router bgp 100

Avaya1/config term/router bgp 100>no redistribute connected
```

Examine the bgp table of Avaya2 to see whether we have any routes.

```
3rdPartyRouter>show ip bgp
```

Announcing OSPF routes to BGP




```
Avaya1> show ip bgp summary
```

Avaya1 has an EBGP session with 3rdPartyRouter in ESTABLISHED state.

Configure OSPF between Avaya1 and Avaya3 in OSPF area 0.

```
Avaya1/configure> router routerid 100.1.1.3

Avaya1/configure> router ospf

Avaya1/configure/router/ospf> int ethernet1 area 0
```

Redistribute static route 150.1.1.0/24 in Avaya3 on to OSPF using redistribute static.

```
Avaya3/configure> router routerid 100.1.1.3
Avaya3/configure> router ospf
Avaya3/configure/router/ospf> int ethernet1 area 0
Avaya3/configure/router/ospf/interface ethernet1> exit
Avaya3/configure/router/ospf> redistribute static
Avaya3/configure/router/ospf> exit
Avaya3/configure> ip route 150.1.1.0 24 10.1.5.254 1
Avaya3/configure>
```

Avaya1 and Avaya2 are adjacent to each other in OSPF Area 0.

View the route table of Avaya1 to check the OSPF routes received from Avaya3.

```
Avaya1> show ip route
```

View the bgp table of 3rdPartyRouter.

```
3rdPartyRouter>show ip bgp
```

Announce the OSPF learned routes from Avaya3 to BGP

```
Avaya1/config term>router bgp 100 A
```

```
Avaya1/config term/router bgp 100>redistribute ospf
```

Examine the bgp table of 3rdPartyRouter now.

```
3rdPartyRouter>show ip bgp
```

Pull back the announced route 150.1.1.0/24

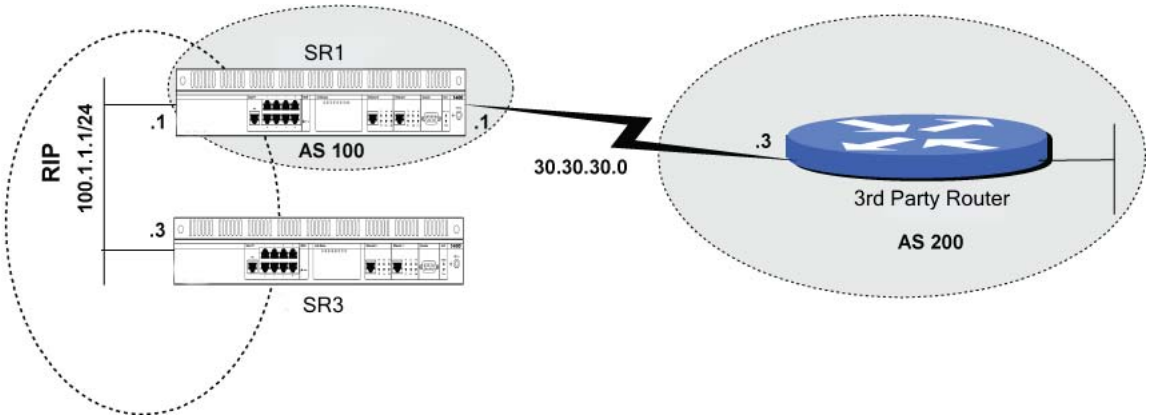
```
Avaya1/config term>router bgp 100
```

```
Avaya1/config term/router bgp 100>no redistribute ospf
```

Examine the bgp table of 3rdPartyRouter to see the routes.

```
3rdPartyRouter>show ip bgp
```

Announcing RIP routes to BGP




```
Avaya1> show ip bgp summary
```

Avaya1 has an EBGP session with 3rdPartyRouter in ESTABLISHED state.

Configure RIP between Avaya1 and Avaya3.

```
Avaya1/configure> router rip
Avaya1/configure/router/rip> int ethernet1
```

Redistribute static route 150.1.1.0/24 in Avaya3 on to RIP using redistribute static.

```
Avaya3/configure> router rip
Avaya3/configure/router/rip> int ethernet1
Avaya3/configure/router/rip/interface ethernet1> exit
Avaya3/configure/router/rip> redistribute static
Avaya3/configure> ip route 150.1.1.0 24 10.1.5.254 1
```

View the route table of Avaya1 to check the RIP routes received from Avaya3.

```
Avaya1> show ip route
```

View the bgp table of 3rdPartyRouter.

```
3rdPartyRouter>show ip bgp
```

Announce the RIP learned routes from Avaya3 on to BGP.

```
Avaya1/config term>router bgp 100
Avaya1/config term/router bgp 100>redistribute rip
```

View the bgp table of 3rdPartyRouter now.

```
3rdPartyRouter>show ip bgp
```

Pull back the announced route 150.1.1.0/24

```
Avaya1/config term>router bgp 100

Avaya1/config term/router bgp 100>no redistribute rip
```

View the bgp table of 3rdPartyRouter to see the routes.

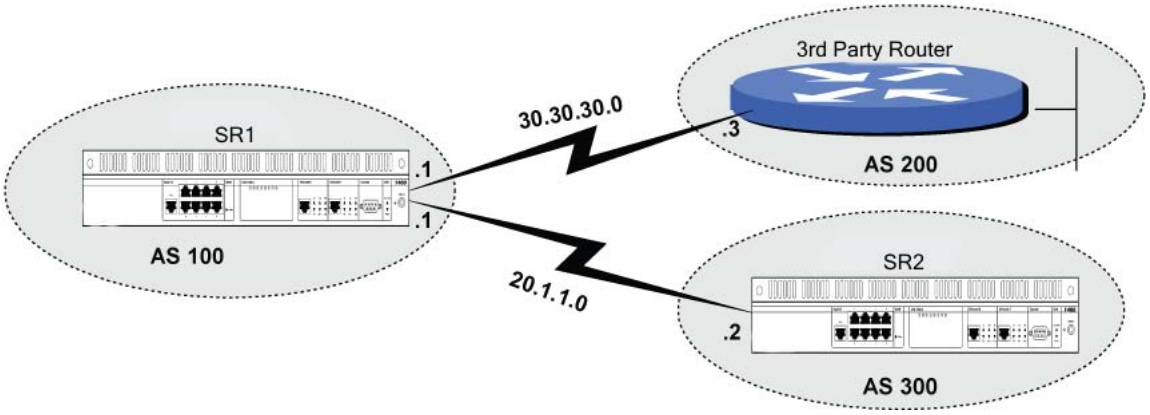
```
3rdPartyRouter>show ip bgp
```

Configuring BGP Policies

BGP policies can be applied to a neighbor or a group of neighbors. Policies are used to control the inflow/outflow mechanism of routes. The policy can be configured to filter in/out routes, change PATH information, route attributes, communities, metric values etc.

In Avaya's BGP Implementation the policy can be applied in different ways:

- OutBound ROUTE_MAP Policy can only be applied to a Neighbor group. InBound Policy can only be applied to a neighbor directly.
- Filter list and Distribute list also can be applied only to a neighbor group for OUTBOUND and per neighbor for INBOUND.



Avaya1 has established EBGP Session with neighbors, 3rdPartyRouter and Avaya2.

```
Avaya1> show ip bgp summary

BGP router identifier 30.30.30.1, local AS number 100
```

Neighbor	V	AS	MsgRcvd	MsgSent	State	Up/Down
20.1.1.2	4	300	5	8	Established	00:04:11
30.30.30.3	4	200	27	25	Established	00:20:43

3rdPartyRouter is announcing some BGP routes in the range 15.1.1.0 to 15.1.2.0/24, 18.1.1.0 to 18.1.7.0/24 to Avaya1.

```
Avaya1> show ip bgp table
```

```
BGP route table, local router ID is 30.30.30.1 Status codes: * valid, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
15.1.2.0/24	30.30.30.3	100	170	200 ?
18.1.1.0/24	30.30.30.3	100	170	200 ?
18.1.2.0/24	30.30.30.3	100	170	200 ?
18.1.3.0/24	30.30.30.3	100	170	200 ?
18.1.4.0/24	30.30.30.3	100	170	200 ?
18.1.5.0/24	30.30.30.3	100	170	200 ?
18.1.4.0/24	30.30.30.3	100	170	200 ?

The above show command output shows all the routes received from AS 200 which is 3rdPartyRouter.

Avaya2 has received all the above routes from Avaya1.

```
Avaya2> show ip bgp summary
```

```
BGP router identifier 100.1.1.3, local AS number 300
```

Neighbor	V	AS	MsgRcvd	MsgSent	State	Up/Down
20.1.1.1	4	100	5	8	Established	00:04:11

The above output shows that Avaya2 is in ESTABLISHED state with Avaya1.

```
Avaya2> show ip bgp table
```

```
Avaya2> show ip bgp table Status codes: * valid, > best, i - internal Origin codes:
i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
15.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.4.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?

18.1.6.0/24	20.1.1.1	20.1.1.1	100 200 ?

All the routes above show the PATH information as 100 200 which means the routes were originally originated from AS 200 and it is learned through AS 100.

Route Aggregation

Aggregation in BGP is used to summarize more specific routes into a single less specific route so that the route table size is reduced or optimized.

Aggregate all the 18.1.1.0 to 18.1.7.0 routes in to one range 18.1.0.0/21 on Avaya1.

```
Avaya1/configure/router/bgp 100> aggregate_address 18.1.0.0
255.255.248.0
```

The above command aggregates all the routes 18.1.1.0 to 18.1.1.7.0 as 18.1.0.0/21.

View the bgp table of Avaya2 to see whether the aggregated route is announced by Avaya1 or not.

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid,> best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.4.0/24	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

Entry 18.1.0.0/21 in Avaya2's bgp table shows that the aggregated route is advertised by Avaya1. Note the PATH information for that route, it says 100. Even though the routes 18.1.1.0 to 18.1.7.0 were advertised originally by 3rdPartyRouter, since Avaya1 is aggregating it, the source AS is 100 now.

Do a summary only on Avaya1.

```
Avaya1/configure/router/bgp 100>aggregate_address 18.1.0.0
255.255.248.0 summary_only
```

The above command would send only an aggregated summary route and suppress all other more specific routes under that.

View the Avaya2 bgp table now.

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.4.0/24	20.1.1.1	20.1.1.1			100 i

The above output shows only 18.1.0.0/21 instead of all the 18.1.1.0 to 18.1.7.0 routes. The originating AS is still 100 in this case.

In order to know which AS routes were used to do this aggregation 18.1.0.0/21 specify the keyword as_set, telling Avaya to send the AS_SET information to other peers.

```
Avaya1/configure/router/bgp 100> aggregate_address 18.1.0.0
255.255.248.0 summary_only as_set
```

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.0.0/24	20.1.1.1	20.1.1.1			100 200 ?

In the above output the PATH info for 18.1.0.0/21 is shown as "100 200", which means that routes from AS200 is used for announcing the aggregate 18.1.0.0/21.

Suppress Map

Suppress maps in aggregate address command is used to specify what are the more specific routes that needs to be suppress from sending along with the aggregated route.

select a few routes to be advertised along with aggregate address by using the suppress map. Remove the aggregate address command on Avaya1.

```
Avaya1/configure/router/bgp 100> no aggregate_address 18.1.0.0
255.255.248.0 summary_only as_set
```

This would have remove the aggregate announcement and send all the more specific routes. Look at Avaya2's bgp table, it has all the routes except the 18.1.0.0/21 route.

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.4.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

To apply a suppress map, select a block of routes that needs to be suppressed. This can be done using IP_ACCESS_LISTS. Create an ip access list that blocks a set of routes in the 18.1.1.0 to 18.1.7.0 range.

```
Avaya1/configure> policy ip_access_list 1 1 permit network 18.1.0.0
netmask 0.0.3.0 mask 255.255.255.0 maskmask 0.0.0.0
```

The above access list would permit routes 18.1.1.0, 18.1.2.0, and 18.1.3.0 and deny everything else.

Configure a route_map with the name ToAvaya2 to match the ip access list in it.

```
Avaya1/configure> policy route_map ToAvaya2 1 permit
```

```
Avaya1/configure/policy/route_map ToAvaya2 1> match ip ip_address 1
```

```
Avaya1/configure/policy/route_map ToAvaya2 1> exit
```

The above route map would allow routes by matching them with the ip access list 1. All other routes would get denied. Suppress map applies a route map to to get the routes that needs to be suppressed.

Apply the route map ToAvaya2 in the suppress map command to our previous aggregate address command.

```
Avaya1/configure/router/bgp 100> aggregate_address 18.1.0.0
255.255.248.0 suppress_map ToAvaya2
```

The above command would apply a route-map on the more specific routes and permit what matches the ip access list 1. Then the routes that were permitted by the route map are

suppressed by suppress_map and the routes denied by route map are advertised to other peers.

View the bgp table of Avaya2. Since the ip access list permits 18.1.1.0 , 18.1.2.0 and 18.1.3.0 , these 3 routes will get suppressed and all the other 18.1.4.0 to 18.1.7.0 range will get advertised with the aggregated route 18.1.0.0/21.

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.4.0/24	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

Attribute Map

Attribute map allows the user to set attributes for the aggregated address.

```
Avaya1/configure/router/bgp 100> aggregate_address 18.1.0.0
255.255.248.0
```

The above command configures an aggregate address of 18.1.0.0/21 for all the more specific routes like 18.1.1.0 to 18.1.7.0. Avaya2 has received those routes with aggregate address 18.1.0.0/21.

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.4.0/24	20.1.1.1	20.1.1.1			100 i

18.1.5.0/24	20.1.1.1	20.1.1.1	100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1	100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1	100 200 ?

Check whether Avaya2 has any routes with community value of 100 in it.

```
Avaya2> show ip bgp community number 100
```

No routes have got a community set with value 100.

In order to set a community value of 100 to the aggregated address 18.1.0.0/21 in Avaya1, use an attribute_map. Attribute_map also uses a route map to select a set of routes and then set some attribute values to it.

In this case, set community value 100 to aggregated address 18.1.0.0/21. Use a route map called ToAvaya2 which matches all routes and sets community 100.

```
Avaya1/configure> policy route_map ToAvaya2 1 permit
```

```
Avaya1/configure/policy/route_map ToAvaya2 1> set community aa_nn
0:10
```

Apply that route map to our aggregate address using attribute_map:

```
Avaya1/configure> router bgp 100
```

```
Avaya1/configure/router/bgp 100> aggregate_address 18.1.0.0
255.255.248.0 attribute_map ToAvaya2
```
Avaya1 should have advertised 18.1.0.0/21 with a community value of 0:100 or 100. View the bgp table of Avaya2:

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

Avaya2 has received the aggregate address 18.1.0.0/21.

View the routes that have the community value of 10. In this case it should be just the route 18.1.0.0/21:

```
Avaya2> show ip bgp community aa_nn 0:100

Avaya2> show ip bgp community number 100
```

The above output shows that 18.1.0.0/21 has a community value of 100 set to it. Avaya has two ways of displaying the communities. Either by just giving a community number value or in a x:x notation like 0:100. Both give the same results.

Route Map

Route maps are used to match and modify certain routing information. A Match can be done on one of the following:

- as_path > match BGP AS path list
- community > match BGP community list
- ip > ip specific information

One a Match is done, the following attributes can be set on those matched information:

- as_path > prepend string for a BGP AS-path attribute
- commit > commit configuration
- community > bgp community attribute
- distance > BGP preference path attribute
- local_preference > BGP local preference path attribute
- metric > Metric value for destination routing protocol
- metric_type > metric type for destination routing protocol
- origin > bgp origin code

Create a route_map by name AS-200Filter and apply it to neighbor Avaya2. The route_map AS-200Filter has a sequence number of 100 which has a condition that denies everything that matches the ip access list 10.

```
Avaya1/configure> policy route_map AS-200Filter 100
deny
Avaya1/configure/policy/route_map AS-200Filter 100>
match ip ip_address 10
```

Issue a show command to verify the routemap and ip_access list:

```
Avaya1/configure> show policy route_map
route-map AS-200Filter, deny, sequence 100
Match clauses:
ip-address (ip-access-list filter): 10
Set clauses:
```

Create an ip_access_list 10 so that the route map can use it. By default if a ip_access_list is specified under the route_map and no acess list is created with that number, everything is denied implicitly. Create an ip_access_list with the number 10:

```
Avaya1/configure> policy ip_access_list 10 1 permit
network 18.1.0.0 netmask 0.0.3.0 mask 255.255.255.0
maskmask 0.0.0.0
Avaya1> show policy ip_access_list
IP access list 10
permit 18.1.0.0 0.0.3.0 255.255.255.0 0.0.0.0
```

View the Avaya2 bgp table:

```
Avaya2> show ip bgp table

BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

Avaya2 has all the 18.1.x.x routes in its table. Apply the route map ToAvaya2 on Avaya1 to filter routes 18.1.1.0, 2.0 and 3.0 announced to Avaya2. To do this, apply this route_map to a group since it is OUTBOUND. Create a external group called ToAvaya2 and apply that route_map to ToAvaya2 group outbound:

```
Avaya1/configure/router/bgp 100/group ToAvaya2
external> route_map AS-200Filter out
Avaya1/configure/router/bgp 100/group ToAvaya2
external> exit
```

After applying the route_map to that group, assign Avaya2 neighbor ip address to that group. In this case 20.1.1.2 is Avaya2's ip address.

```
Avaya1/configure/router/bgp 100> neighbor 20.1.1.2 300
Avaya1/configure/router/bgp 100/neighbor 20.1.1.2 300>
neighbor_group ToAvaya2
```

View the route table of Avaya2 now.

```
Avaya2> show ip bgp table
Avaya2>
```

Avaya2 has not received any routes from Avaya1. The route map that we applied on Avaya1 should have filtered 18.1.1.0, 18.1.2.0 and 18.1.3.0. By default an implicit deny all gets added to the route_map.

Add another statement that says permit others:

```
Avaya1/configure> policy route_map AS-200Filter 101 permit
```

The above statement states that after going through the sequence 100 do this sequence 101 that says permit any. The permit any route_map statement should have an higher sequence number than the deny, or else everything will get permitted.

```
Avaya1> show policy route_map
```

This command will bring back all the routes except 18.1.1.0/2.0 and 3.0 to Avaya2:

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

Announce a community of 100 to all routes that get advertised to Avaya2 on Avaya1. Add a set statement under the route_map sequence 101. This would permit any other routes that were not denied by route_map sequence 100 and set a community of 0:100.

```
Avaya1/configure> policy route_map AS-200Filter 101
permit
Avaya1/configure/policy/route_map AS-200Filter 101>
set community aa_nn 0:100
Avaya2> show ip bgp community aa_nn 0:100
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

```
Avaya2> show ip bgp community number 10
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i

18.1.5.0/24	20.1.1.1	20.1.1.1	100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1	100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1	100 200 ?

The above output shows that all routes that came from Avaya1 have the community value of 100 or 0:100 set. Apply a route-map INBOUND on Avaya2 to filter any routes that comes from AS 200. Currently Avaya2 has got routes except 18.1.0.0/21 coming from AS 200.

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

By putting a filter to block routes from AS 200 we should see only 18.1.0.0/21 which comes from 100. To do this, use as_path filter to deny routes that come from AS 200. Again INBOUND route-maps are applied to neighbors directly and outbounds are applied to GROUPs.

```
Avaya2/configure> policy route_map From100 1 permit
Avaya2/configure/policy/route_map From100 1> match
as_path 1
Avaya2/configure/policy/route_map From100 1> exit
Avaya2/configure> policy as_path 1 1 deny ".* 200"
```

The above commands create a route_map named From100 with sequence number 1 to permit anything that matches as_path list 1. We have two sequence numbers in as_path, 1 1 is to deny anything from originates from AS200 and 1 2 is to permit others.

The route map needs to be applied INBOUND under neighbor 20.1.1.1 which is Avaya1.

```
Avaya2/configure/router/bgp 300> neighbor 20.1.1.1 100
Avaya2/configure/router/bgp 300/neighbor 20.1.1.1 100>
route_map From100 in
Avaya2/configure> show policy route_map
Avaya2/configure> show policy as_path
AS path access list 1
permit .* 200
permit .*
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i

18.1.5.0/24	20.1.1.1	20.1.1.1	100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1	100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1	100 200 ?

The above routes appear in the bgp table but not in the route table. Only active routes are pushed on to the FIB. In the above *> 18.1.0.0/21 is the only one active.

```
Avaya2> show ip route
```

18.1.0.0/21 is the only route that got installed in the routing table.

Community List Filters

One other way filtering routes is to use community filters and decide based on the community whether to permit/deny routes. Add a community of 100:200 to all routes announced by 3rdPartyRouter.

```
3rdPartyRouterA(config)>route-map AS-100Community permit
3rdPartyRouterA(config-route-map)>set community 200:100
3rdPartyRouterA(config)>router bgp 200
3rdPartyRouterA(config-router)>neighbor 30.30.30.1
route-map AS-100Community out
3rdPartyRouterA(config-router)>neighbor 30.30.30.1
send-community
3rdPartyRouterA>clear ip bgp *
```

View Avaya1's bgp table to check whether it has received all the routes with community 200:100 from 3rdPartyRouter. Perform a show using x:x format.

```
Avaya1> show ip bgp community aa_nn 200:100
```

```
BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	30.30.30.3	100	170	100 200 ?
18.1.5.0/24	30.30.30.3	100	170	100 200 ?
18.1.6.0/24	30.30.30.3	100	170	100 200 ?
18.1.7.0/24	30.30.30.3	100	170	100 200 ?

The same can be obtained by giving an absolute number 13107300 in the show ip bgp community , which is equal to 200:100

```
Avaya1> show ip bgp community number 13107300
```

```
BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	30.30.30.3	100	170	100 200 ?
18.1.5.0/24	30.30.30.3	100	170	100 200 ?
18.1.6.0/24	30.30.30.3	100	170	100 200 ?
18.1.7.0/24	30.30.30.3	100	170	100 200 ?

Filter all routes with a community of 200:100 in INBOUND using community list filter on Avaya11.

```
Avaya1> conf term
Avaya1/configure> policy community_list standard_community
1 1 permit aa_nn 200:100
```

The above command creates a community_list filter 1 with sequence number 1 which permits any community with the value 200:100.

The community_list filter can be displayed by issuing the command below:

```
Avaya1/configure> show policy community_list
```

Community standard list 1 permit 200:100 Apply that community_list filter to a route_map From200 and put it under the neighbor 30.30.30.3, which is 3rdPartyRouter INBOUND.

```
Avaya1/configure> policy route_map From200 1 deny
Avaya1/configure/policy/route_map From200 1> match
community standard_community 1
Avaya1/configure/policy/route_map From200 1> exit
Avaya1/configure> policy route_map From200 2 permit
Avaya1/configure/policy/route_map From200 2>
```

Route map From200 sequence 1 denies any routes that has the community 200:100 set. Route map From200 sequence 2 permits all other routes.

```
Avaya1> conf term Avaya1/configure> router bgp 100
Avaya1/configure/router/bgp 100> neighbor 30.30.30.3 200
Avaya1/configure/router/bgp 100/neighbor 30.30.30.3 200> route_map From200 in
```

The above command applies the route map to neighbor 30.30.30.3 INBOUND. This will filter all routes from 3rdPartyRouter, since every routes from 3rdPartyRouter has community 200:100 set.

```
Avaya1> show ip bgp table

BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	30.30.30.3	100	0	200 ?
18.1.5.0/24	30.30.30.3	100	0	200 ?
18.1.6.0/24	30.30.30.3	100	0	200 ?
18.1.7.0/24	30.30.30.3	100	0	200 ?

All routes received from 3rdPartyRouter having a community of 200:100 are filtered. The bgp table shows all of those routes as INACTIVE.

```
Avaya1> show ip route
```

Show ip route does not show any BGP routes in its table. Change the routemap in 3rdPartyRouter to set community value of 200:100 , only for routes in 18.1.x.x range.

```
3rdPartyRouterA>conf term
3rdPartyRouterA(config)>ip access-list standard 1
3rdPartyRouterA(config-std-nacl)>permit 18.1.0.0
0.0.255.255
3rdPartyRouterA(config-std-nacl)>exit
3rdPartyRouterA(config)>route-map AS-100Community
permit 10
3rdPartyRouterA(config-route-map)>match ip address 1
3rdPartyRouterA(config-route-map)>set community 200:100
3rdPartyRouterA(config)>route-map AS-100Community
permit 20
3rdPartyRouterA>clear ip bgp *
3rdPartyRouterA>
```

Route map From200 on Avaya1 will filter everything in 18.1.x.x range and allow 15.1.x.x range to be installed in to the routing table.

```
Avaya1> show ip bgp community aa_nn 200:100

BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	30.30.30.3	100	0	200 ?
18.1.5.0/24	30.30.30.3	100	0	200 ?
18.1.6.0/24	30.30.30.3	100	0	200 ?
18.1.7.0/24	30.30.30.3	100	0	200 ?

The above shows that all 18.1.x.x range has the community of 200:100 set, but all of them are marked INACTIVE which means got filtered properly.

```
Avaya1> show ip bgp table

BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
15.1.2.0/24	30.30.30.3	100	170	200 ?
15.1.3.0/24	30.30.30.3	100	170	200 ?
18.1.1.0/24	30.30.30.3	100	0	200 ?
18.1.5.0/24	30.30.30.3	100	0	200 ?
18.1.6.0/24	30.30.30.3	100	0	200 ?

18.1.7.0/24	30.30.30.3	100	0	200 ?

Show ip bgp table above shows that 15.1.1.0 and 15.1.2.0 which do not have the community value of 200:100 are being permitted and made ACTIVE.

```
Avaya1> show ip route
```

Remove the routemap configured INBOUND to 3rdPartyRouter.

```
Avaya1/configure> router bgp 100
Avaya1/configure/router/bgp 100> neighbor 30.30.30.3
200
Avaya1/configure/router/bgp 100/neighbor 30.30.30.3
200> no route_map From200 in
Avaya1> show ip bgp table
```

```
BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
15.1.2.0/24	30.30.30.3	100	170	200 ?
15.1.3.0/24	30.30.30.3	100	170	200 ?
18.1.1.0/24	30.30.30.3	100	0	200 ?
18.1.5.0/24	30.30.30.3	100	0	200 ?
18.1.6.0/24	30.30.30.3	100	0	200 ?
18.1.7.0/24	30.30.30.3	100	0	200 ?

All routes are back ACTIVE.

Distribute Lists

Distribute Lists can be applied to neighbors/neighbor groups to permit/deny routes based on ip route information. By using distribute-list you do not have to apply it to a route map and then apply to a neighbor/group. A distribute-list can be directly applied either to a neighbor INBOUND or a group OUTBOUND. Avaya1 has received the following routes from 3rdPartyRouter.

```
Avaya1> show ip bgp table
```

```
BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
15.1.2.0/24	30.30.30.3	100	170	200 ?
15.1.3.0/24	30.30.30.3	100	170	200 ?
18.1.1.0/24	30.30.30.3	100	170	200 ?

18.1.5.0/24	30.30.30.3	100	170	200 ?
18.1.6.0/24	30.30.30.3	100	170	200 ?
18.1.7.0/24	30.30.30.3	100	170	200 ?

Distribute_list uses ip_access_list filter rules to define the filter set. Configure an IP_ACCESS_LIST for distribute_list to use.

```
Avaya1/configure> policy ip_access_list 10 1 deny
network 15.1.0.0 netmask 0.0.255.255 mask 255.255.255.0
maskmask 0.0.0.0
Avaya1/configure> policy ip_access_list 10 2 permit
```

The above commands configure the filter_list which denies routes in the range of 15.1.0.0 to 15.1.255.255 having netmask 255.255.255.0. IP_ACCESS_LIST 10 with sequence number 2 is configured to allow routes other than 15.1.x.x range or else every route will get denied because of the IMPLICIT deny all. Put a distribute_list INBOUND to block routes in the range of 15.1.x.x coming from 3rdPartyRouter.

```
Avaya1/configure/router/bgp 100>
Avaya1/configure/router/bgp 100> neighbor 30.30.30.3 200
Avaya1/configure/router/bgp 100/neighbor 30.30.30.3
200> distribute_list 10 in
```

The above command applies the distribute_list 10 INBOUND to neighbor 30.30.30.3 which is 3rdPartyRouter.

```
Avaya1/configure> show ip bgp table
```

```
BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
15.1.2.0/24	30.30.30.3	100	0	200 ?
15.1.3.0/24	30.30.30.3	100	0	200 ?
18.1.1.0/24	30.30.30.3	100	170	200 ?
18.1.5.0/24	30.30.30.3	100	170	200 ?
18.1.6.0/24	30.30.30.3	100	170	200 ?
18.1.7.0/24	30.30.30.3	100	170	200 ?

Avaya1 has filtered routes in the range of 15.1.x.x learnt from 3rdPartyRouter. All other routes are made ACTIVE in the bgp table. Remove the distribute list from neighbor 30.30.30.3.

```
Avaya1/configure/router/bgp 100/neighbor 30.30.30.3
200> no distribute_list 10 in
Avaya1/configure/router/bgp 100/neighbor 30.30.30.3.
200> Avaya1> show ip bgp table
```

```
BGP route table, local router ID is 30.30.30.1 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Path
15.1.2.0/24	30.30.30.3	100	170	200 ?
15.1.3.0/24	30.30.30.3	100	170	200 ?
18.1.1.0/24	30.30.30.3	100	170	200 ?
18.1.5.0/24	30.30.30.3	100	170	200 ?
18.1.6.0/24	30.30.30.3	100	170	200 ?
18.1.7.0/24	30.30.30.3	100	170	200 ?

POLICIES applied to Avaya are dynamic and the user does not have to clear the bgp session to take effect. All the policies from Avaya1 and Avaya2 are removed. Avaya2 gets all the routes received from 3rdPartyRouter through Avaya1.

```
Avaya2>show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

Apply a distribute list OUTBOUND going to Avaya2 on Avaya1 blocking all the routes in the range 18.1.x.0/24.

```
Avaya1/configure> policy ip_access_list 20 1 deny network 18.1.0.0
netmask 0.0.255.255 mask 255.255.255.0 maskmask 0.0.0.0
Avaya1/configure> policy ip_access_list 20 2 permit
```

Avaya2 belongs to a group ToAvaya2, so the distribute list OUTBOUND needs to be applied to the group name ToAvaya2.

```
Avaya1> show ip bgp groups
```

```
Avaya1/configure/router/bgp 100> group ToAvaya2 external
Avaya1/configure/router/bgp 100/group ToAvaya2
external> distribute_list 20 out
Avaya1/configure/router/bgp 100/group ToAvaya2
external> exit
```

Check whether the distribute list is applied to the group ToAvaya2 or not.

```
Avaya1> show ip bgp groups
```

The above output shows that Distribute_list 20 is applied to group ToAvaya2.

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
15.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
15.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i

Avaya2 bgp table output shows that all 18.1.0.0/24 range is filtered by Avaya1 while sending the bgp updates. Since 18.1.0.0 has a mask of /21 it was not filtered by Avaya1. Remove the distribute list from Avaya1.

```
Avaya1/configure> router bgp 100
Avaya1/configure/router/bgp 100> group ToAvaya2
external
Avaya1/configure/router/bgp 100/group ToAvaya2
external> no distribute_list 20 out
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
18.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

Filter Lists

Filter-Lists can be applied to neighbors/neighbor groups to permit/deny routes based on AS_PATH information. By using filter-list you do not have to apply it to a route map and then apply to a neighbor/group. A filter-list can be directly applied either to a neighbor INBOUND or a group OUTBOUND. Avaya2 has the following routes received from Avaya1.

```
Avaya2> show ip bgp table
```

```
BGP route table, local router ID is 100.1.1.3 Status codes: * valid, > best, i -
internal Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Orig Next Hop	Next Hop	Metric	LocPrf	Path
15.1.1.0/24	20.1.1.1	20.1.1.1			100 200 ?
15.1.2.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.3.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.0.0/21	20.1.1.1	20.1.1.1			100 i
18.1.5.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.6.0/24	20.1.1.1	20.1.1.1			100 200 ?
18.1.7.0/24	20.1.1.1	20.1.1.1			100 200 ?

Apply a filter_list out to Avaya2 using group ToAvaya2 in Avaya1.

```
Avaya1/configure> policy as_path 1 1 deny ".* 200"
Avaya1/configure> policy as_path 1 2 permit ".*"
```

The above commands configures a AS_PATH filter which denies all routes originating from AS200 and permitting other routes. Apply that AS_PATH 1 filter OUTBOUND to group ToAvaya2.

```
Avaya1/configure> router bgp 100
Avaya1/configure/router/bgp 100> group ToAvaya2
external
Avaya1/configure/router/bgp 100/group ToAvaya2
external> filter_list 1 out
```

Avaya1 should have filtered all the routes that originated from AS200 while sending bgp updates to Avaya2.

```
Avaya2> show ip bgp table
```

Configuring Peer Groups

Peer Groups are used in Avaya to apply common policies to a set of neighbors. OUTBOUND policies can only be applied to peer groups. INBOUND policies are neighbor based.In Avaya BGP Implementation the policy can be applied in different ways. Avaya supports 3 types of Peer Groups.

By default Avaya has 3 peer groups configured on the system once BGP is configured:

- BgpInternal
- BgpExternal
- BgpExternal_rt

By default Avaya assigns a particular neighbor in to one of the following 3 neighbor groups depending on their type:

- IBGP Neighbor > BgpInternal
- EBGP Neighbor > BgpExternal
- EBGP Multi-Hop Neighbor > BgpExternal_Rt

Other than the above 3 default groups, a user can also create user-defined groups and assign neighbors to it.

A user-defined group can be created by using the following command:

```
router  bgp 100
group ToAvaya2 external
group ToAvaya2 external
```

A neighbor can be assigned to a group by using the following command:

```
router  bgp 100
neighbor 20.1.1.2 300
neighbor_group ToAvaya2
```

All the groups defined in the system can be displayed using the following show command:

```
show ip bgp groups

BGP group is BgpExternal
group type is  External, total peers 1, established
peers 1, members:
30.30.30.3
options set : None

BGP group is ToAvaya2
group type is  External, total peers 1, established
peers 1, members:
20.1.1.2
options set :
None
```

In the above show command output, ToAvaya2 is a user-defined group which is external and BgpExternal is the default group. Since neighbor 30.30.30.3 was not assigned to any user specific peer group, Avaya assigns it to the default group which is BgpExternal.

Any EBGP Multi-Hop neighbor needs to be applied to a group, which is of type External_Rt only. Directly connected EBGP neighbors can be assigned to group type External.

Under a neighbor group we have the following commands that can be set:

```
Avaya1/configure/router/bgp 100/group To3rdPartyRouter external_rt>
?

commit > commit configuration
distribute_list > filter updates to this group
filter_list > establish BGP filters
next_hop_self > Disable nexthop calculation for this
group
```

```
password > TCP MD5 password for the group
remove_private_AS > remove private AS number from
updates
route_map > apply route map to group
```

If any of the above parameters needs to be set for all or a set of neighbors then it makes sense to put those policies under a group and apply those neighbor to that group. OUTBOUND Policies applied to Group improves the performance rather than having it on Peer based.

Chapter 46: Route tags for route redistribution

The Secure Router allows you to match and set route tags in routing protocols. This allows you to, for example, tag routes redistributed from BGP and RIP into OSPF so that you can later match on those tags to exclude the routes from being redistributed back into BGP.

To configure the route tagging, two options, namely **match tag** and **set tag**, are provided in the **policy route_map** CLI command . These commands allow you to create a route map with a match condition for particular route tags. The route map can be used with the **redistribution** command to permit or deny the redistribution of routes based on the tag value of the routes. You can also configure the route map with a set tag option, to apply a new tag or modify an existing tag on the redistributed routes.

Some usage scenarios are:

1. If a route map is created with **match tag 500** and **set tag 1000** and is used for redistributing routes, then all the routes with route tag 500 are redistributed with route tag 1000. Other routes do not get redistributed
2. If a route map is created with **match tag 500** and no **set tag** and is used for redistributing routes, then all the routes with tag matching 500 are redistributed with the same tag 500. Other routes do not get redistributed.
3. If a route map is created with no **match tag** and **set tag 1000** and is used for redistributing routes, then all the routes are redistributed with tag 1000.

The **match tag** and **set tag** commands can also be used along with other match conditions and set options in the route-maps. The CLI command **show policy route-map** displays the match and set configuration for route tags.

The following output shows the **route_map** command tree from which you can configure the **match tag** and **set tag** commands.

```
configure/policy/route_map rmap1 10 > match ?
as_path
community
ip
source-protocol
tag
```

```
configure/policy/route_map rmap1 10 > set ?
as_path
community
distance
local_preference
metric
metric_type
origin
tag
```

Chapter 47: Configuring Packet Capture

The packet capture feature is used to capture the packets sent and received on any supported interface in the router. This feature aids in debugging and troubleshooting network traffic issues. Capture format is libpcap format (version 2.4) supported by the most commonly used third party packet analyzers like ethereal and tcpdump.

The captured packets will contain all the protocol headers including the L2 header. This feature can be enabled on a network layer interface while traffic is flowing through the system. Packets can be captured in inbound and outbound directions, while the default is 'both' when direction is not specified. Packets can be captured simultaneously on one more L3 interfaces. The packet feature can use filtering rules and enable a user to view the captured packets. It is possible to specify whether non-ip packets should be captured on a given device. Packet capture can be disabled on a network layer interface while packet capture session is ongoing.

The number of packets to be captured and the maximum capture length can be specified. It is possible to configure the total amount of memory to be reserved for all packet capture sessions configured in the system, and the max size of the packet capture session on an interface.

It is possible to save the captured packets to a local or remote file and specify whether the packet capture buffers should be overwritten when the buffer is full.

The following packet capture feature link layers are supported:

- Ethernet physical interfaces
- Ethernet subinterfaces
- Single and multilink PPP bundle interfaces
- Cisco HDLC interfaces
- Frame Relay PVCs
- PCs over multilink FR (FRF.16) bundle interfaces

Important:
PCAP is primarily a debug tool and should be used only in this fashion as it can generate a substantial amount of output and can use a substantial amount of system resources.

Example

Configuring Packet Capture

```
SR/debug/pcap> capture cap1

SR/debug/pcap/capture cap1> attach ethernet 0

SR/debug/pcap/capture cap1> count 1000
```

```
SR/debug/pcap/capture cap1> size 400
SR/debug/pcap/capture cap1> drop-nonip
SR/debug/pcap/capture cap1> direction in
SR/debug/pcap/capture cap1> filter fil1 in
SR/debug/pcap/capture cap1> snaplen 256
SR/debug/pcap/capture cap1> commit
SR/debug/pcap/capture cap1> show-config

Packet Capture : cap1
=====================================
Interface attached : ethernet0
State : Disabled.
Configurations committed
Duration of session : 0 secs
Direction : IN
Buffer Wrap OFF
Number of packets to capture :1000
Capture length is 256 bytes in each packet
Capture non-IP packets OFF
0 packets captured in this session
Buffer size for this session: 400KB
Inbound Filter : fil1
Filter Rule List : fil1
1. permit ip any any
2. permit icmp any any
Outbound Filter : Not configured
SR/debug/pcap/capture cap1> exit
SR/debug/pcap> enable

Enabled session cap1
SR/debug/pcap> stats cap1

Packet Capture : cap1
=====================================
Interface attached : ethernet0
State : Enabled
Session active since : 44 secs
Direction : IN
Number of inbound packets captured : 135
Number of outbound packets captured : 0
Actual capture size : 14896 bytes

IP Packet Filter Statistics
===========================
Inbound Packets :
Matched 135
Unmatched 0
Outbound Packets :
Matched 0
Unmatched 0
```

Packet Capture Considerations:

- The number of packets to be captured in a packet capture session can be specified. The maximum number of packets allowed can be up to 10000. The default value is 0 which means all packets are to be captured.
- The maximum capture length for a packet can be specified. A value 0 implies entire contents of a packet has to be captured. The entire contents of packets will be captured by default.
- Ability to configure the total amount of memory to be reserved for all the packet capture sessions configured in the system. Default and maximum limit is 5 MB. This can be reconfigured to a lower limit.
- Ability to configure the max size of the packet capture session on an interface. Default is 1MB. This can be reconfigured, subject to availability of memory in the pool allocated for all packet capture sessions
- Packet capture filtering rules cannot be applied on an ethernet subinterface.
- In Frame Relay interfaces, LMI packets can not be captured because there is no network layer interface associated with DLCI 0 or 1023 over which LMI packets are sent/received.
- MFR Link Integrity Protocol packets and packets over the switched PVCs cannot be captured (frame relay traffic) because there is no associated network layer interface.
- Packet capture is not supported for the tunnel and loopback interfaces.
- Packet capture is not supported for ISDN D channel as it is not associated with a network layer interface.
- The **upload** , **saveto**, and **dump** commands are not available during an active packet capture session.

Statistics for dropped packets support

Data Packets can be dropped by an interface while data traffic is passing for several reasons. The dropped packet statistics will be collected for the following set of packet drop scenarios:

- Tunnel encapsulation errors
- VLAN packets dropped due to QoS(classification errors,queuing errors)
- VLAN packets dropped due to RED
- VLAN Input/Output interface
- VLAN errors such as unrecognized VLAN id, multiple dot1q encapsulations in the packet, etc
- Internal (system) errors

Note:

The following MIBs apply:

- Counter for LAN Drop packet is defined in private enterprise MIB ntEnterpriseDataTasmanMgmtethernet.mib.
- Counter for WAN Drop packet is defined in private enterprise MIB ntEnterpriseDataTasmanMgmtbundle.mib.

Packet Capture of VLAN Packet with Filter Rules

The capturing of VLAN traffic for a specific VLAN ID is done using packet capture access-list rules and applying them to a capture buffer specifying the direction. The access-list rule now supports fields to filter on the MAC portion of the packet header. The MAC accesss-list can filter on the source and destination MAC address, Ether type, CoS user defined field, VLAN, and second VLAN. Only the MAC source address and MAC destination address require the proto field to be of MAC type. All other MAC fields can be applied to any proto type of the access-list rule.

The syntax of the mac access-list rule is:

```
add permit mac <src-mac> <dest-mac> [<ethertype>] [<cos>] [<vlan>]
[<vlan2>]
```

The following table describes the variables in this command.

Table 39: Variable definitions

Variable	Value
<src-mac>	Specifies the source MAC address.
<dest-mac>	Specifies the destination MAC address
[<ethertype>]	Specifies the ethertype as a four digit hexadecimal number.
[<cos>]	Specifies Class of Service (CoS). Values range from 0 to 7. This parameter applies to the outer VLAN tag.
[<vlan>]	Specifies the VLAN ID. This parameter applies to the outer VLAN tag.
[<vlan2>]	Specifies the inner VLAN ID for a double tagged frame.

The following example show the configuration required to capture all the TCP traffic over VLAN ID 10 with an ethertype of 0x8100 on Ethernet 0/1.

```
Host/debug/pcap > show-config
Packet capture global configurations:
=====================================
Maximum size reserved for packet capture : 5120KB
Alloted for packet capture sessions : 0KB
Available for packet capture sessions : 5120KB
Maximum number of sessions allowed : 5
capture configuration session interface: buffer size total pkts
name : committed : active : (Kb) : captured :
================================================================================
================================================================================
Host/debug/pcap > access-list invlan10
Host/debug/pcap/access-list invlan10 > add permit tcp any any ethertype 0x8100 vlan
10
Host/debug/pcap/access-list invlan10 > exit
Host/debug/pcap > access-list outvlan10
Host/debug/pcap/access-list outvlan10 > add permit tcp any any ethertype 0x8100 vlan
10
Host/debug/pcap/access-list outvlan10 > exit
Host/debug/pcap > capture vlan10
Host/debug/pcap/capture vlan10 > attach ethernet 0/1
Host/debug/pcap/capture vlan10 > filter invlan10 in
Host/debug/pcap/capture vlan10 > filter outvlan10 out
Host/debug/pcap/capture vlan10 > wrap
Host/debug/pcap/capture vlan10 > direction both
Host/debug/pcap/capture vlan10 > commit

Host/debug/pcap/capture vlan10 > show-config
Packet Capture : vlan10
=====================================
Interface attached : ethernet0/1
State : Disabled.
Configurations committed
Duration of session : 0 secs
Direction : IN, OUT
Buffer Wrap : ON
Capture all packets
Capture entire contents of each packet
Capture non-IP packets : ON
packets captured in this session : 0
Buffer size for this session : 1024KB
Inbound Filter : invlan10
Pcap Filter Rule List : invlan10
1. permit tcp any any
Outbound Filter : outvlan10
Pcap Filter Rule List : outvlan10
1. permit tcp any any

Host/debug/pcap/capture vlan10 > exit

Host/debug/pcap > show-config
Packet capture global configurations :
=====================================
Maximum size reserved for packet capture : 5120KB
Alloted for packet capture sessions : 1024KB
Available for packet capture sessions : 4096KB
Maximum number of sessions allowed : 5
capture configuration session interface: buffer size total pkts
name : committed : active : (Kb) : captured :
================================================================================
vlan10 yes no ethernet0/1 1024 0
```

```
================================================================================

Host/debug/pcap > enable
Enabled session vlan10

Host/debug/pcap > show-config
Packet capture global configurations :
=====================================
Maximum size reserved for packet capture : 5120KB
Alloted for packet capture sessions : 1024KB
Available for packet capture sessions : 4096KB
Maximum number of sessions allowed : 5
capture configuration session interface: buffer size total pkts
name : committed : active : (Kb) : captured :
================================================================================
vlan10 yes yes ethernet0/1 1024 128
================================================================================

Host/debug/pcap > no enable
Disabled session vlan10

Host/debug/pcap > show int ethernet 0/1
ethernet 0/1
ipaddr 60.1.1.1
netmask 255.255.255.0
description -
status up
configured auto
speed -
mode -
actual
speed 100
mode full_duplex
mss 600
mtu 1500
ethernet0/1 (vlan:10) (unit number 0)
Type: ETHERNET (802.1q)
Flags: (0x880fc343) Up, RUNNING, MULTICAST-ROUTE
Internet Address: 60.1.1.1
Internet Netmask: 255.255.255.0
Internet Broadcast: 60.1.1.255
Maximum Transfer Unit: 1500 bytes
Mac Address: 00:50:52:02:02:01
port counters since last boot/clear
Bytes Rx 71244 Bytes Tx 54498
Packets Rx 842 Packets Tx 625
Runts Rx 40 Collisions 0
Babbels Rx 0 Late Collisions 0
Err Packets Rx 0 Up/Down States (Phys) 13
Up/Down States (Admin) 5
port counters for the last five minutes
Bytes Rx 5463 Bytes Tx 4521
Packets Rx 64 Packets Tx 64
Runts Rx 0 Collisions 0
Babbels Rx 0 Late Collisions 0
Err Packets Rx 0 Up/Down States (Phys) 0
Up/Down States (Admin) 0
```

Chapter 48: Secure Router Configuration for Dynamic Route Exchange over IPSec Tunnel interoperability with VPN Router

Both Secure Router and VPN router currently support dynamic routing over IPSec. Secure router configuration for dynamic route exchange over IPSec Tunnel allows interoperability by using IP-on-IP over a transport mode IPSec connection.

Capabilities

Secure router configuration for dynamic route exchange over IPSec Tunnel, has the following capabilities:

- IPSec transport mode is used, not tunnel mode
- The Secure Router default IPIP tunnel MTU needs to be set to 1500 for OSPF, to match the VPN Router tunnel MTU..
- If both “ip mtu” and “tunnel path-mtu-discovery” are configured/ enabled on Secure Router the mtu value set by “ip mtu” configuration will be in effect.

Secure router configuration for BGP

Configure secure routing for BGP as follows:

```
interface ethernet 0
ip address 10.10.10.1 24
crypto trusted
exit

interface ethernet 1
ip address 192.168.26.100 24
crypto untrusted
exit

interface tunnel toCes
ip address 100.1.1.1 24
tunnel source 192.168.26.100
tunnel destination 192.168.27.100
```

```
tunnel mode ipip
tunnel protection toCe Avaya
crypto untrusted
interface loopback ll1
ip address 50.1.1.1 24
exit

router ospf
redistribute connected
exit

router bgp 100
neighbor 50.1.1.10 100
update_source 50.1.1.1
exit
exit
interface loopback ll1
ip address 50.1.1.1 24
exit
ip route 50.1.1.10 32 toCes
```

Secure router configuration for OSPF

Configure secure routing for OSPF as follows:

```
interface ethernet 0
ip address 10.10.10.1 24
crypto trusted
exit

interface ethernet 1
ip address 192.168.26.100 24
crypto untrusted
exit

interface tunnel toCes
Ip unnumbered ethernet1
ip mtu 1500
tunnel source 192.168.26.100
tunnel destination 192.168.27.100
tunnel mode ipip
tunnel protection toCes Avaya
crypto untrusted
exit

router ospf
interface toCes area 0
exit
```

Secure router configuration for RIPv2

Configure secure routing for RIPv2 as follows:

```
interface ethernet 0
ip address 10.10.10.1 24
crypto trusted
exit

interface ethernet 1
ip address 192.168.26.100 24
crypto untrusted
exit

interface tunnel toCes
ip address 100.1.1.1 24
tunnel source 192.168.26.100
tunnel destination 192.168.27.100
tunnel mode ipip
tunnel protection toCes Avaya
crypto untrusted
exit

ip route 192.168.27.0 24 192.168.26.101
router routerid 192.168.26.100
router rip
interface toCes
mode 2
exit

firewall corp
policy 101 in
exit
exit

firewall internet
policy 100 in self
exit
exit
```

Chapter 49: Management Configuration Guide

Simple Network Management Protocol

Secure Router provides two classes of SNMP Management Information Bases (MIBs) to provide you with the ability to manage Secure Routers. The two MIBs are the Enterprise MIB and the Standard MIB.

Enterprise MIBs

The Enterprise MIB folder includes the following individual MIB..

ntEnterpriseDataTasmanMgmtbundle.mib

Use bundle.mib to manage bundles and links within bundles. Compile bundle.mib before you compile ppp.mib, fr.mib, ghdlc.mib, or qos.mib.

Use bundle.mib to:

- Be notified when a bundle goes down (ntSRbundleDownTrap)
- Be notified when a bundle comes up (ntSRbundleUpTrap)
- Be notified when a link goes down (ntSRlinkDownTrap)
- Be notified when a link comes up (ntSRlinkUpTrap)

ntEnterpriseDataTasmanMgmtchassis.mib

Use chassis.mib to manage the platform.

ntEnterpriseDataTasmanMgmtconfig.mib

Use config.mib to manage configuration data on the router, in memory, or on the network.

Use this MIB to:

- Be notified when the configuration changes (ntSRcfgEventChangeNotification)
- Be notified when the configuration is saved (ntSRcfgEventSaveNotification)

ntEnterpriseDataTasmanMgmtdos.mib

Use dos.mib to manage basic Denial of Service variables in the router.

ntEnterpriseDataTasmanMgmtdsx-tc.mib

Compile this MIB before any other DSX MIBs. It does not contain any traps.

ntEnterpriseDataTasmanMgmtdsx-te1.mib

This MIB manages T1/E1 interfaces. Use this MIB to

- Be notified when an alarm is generated (ntSRdsxT1E1AlarmOnTrap)
- Be notified when an alarm is turned off (ntSRdsxT1E1AlarmOffTrap)

ntEnterpriseDataTasmanMgmtenvironment.mib

This MIB manages the chassis environment such as temperature, fan function. Use this MIB to:

- Be notified if the trap is set to true and if temperature reaches a critical level (ntSRenvTemperatureNotification).
- Be notified if the trap is set to true and the fan state changes (ntSRenvFanNotification).
- Be notified if the trap is set to true and the specified power supply shuts off (ntSRenvPowerSupply1DownNotification, ntSRenvPowerSupply2DownNotification).
- Be notified if the trap is set to true and the specified power supply turns on (ntSRenvPowerSupply1UpNotification, ntSRenvPowerSupply2UpNotification).

ntEnterpriseDataTasmanMgmtdsx-te3.mib

This MIB manages T3/E3 interfaces. Use this MIB to

- Be notified when an alarm is generated (ntSRdsxT3E3AlarmOnTrap)
- Be notified when an alarm is turned off (ntSRdsxT3E3AlarmOffTrap)

ntEnterpriseDataTasmanMgmtethernet.mib

This MIB manages Ethernet parameters and does not contain any traps.

ntEnterpriseDataTasmanMgmtfr.mib

This MIB manages Frame Relay and Multilink Frame Relay bundles and does not contain any traps.

ntEnterpriseDataTasmanMgmtghdlc.mib

This MIB manages generic HDLC encapsulated bundles and does not contain any traps.

ntEnterpriseDataTasmanMgmtip.mip

This MIB manages IP addressable interfaces and static routes and does not contain any traps.

ntEnterpriseDataTasmanMgmtppp.mib

This MIB manages PPP and MLPPP bundles and does not contain any traps.

ntEnterpriseData.mib

This MIB manages registration objects and does not contain any traps.

ntEnterpriseDataTasmanMgmtqos.mib

This MIB defines objects to access QOS parameters. These include CBQ (Class based queuing) status and statistics variables. Read parameters further include historical statistics gathering. This MIB does not contain any traps.

nortel.mib

This MIB manages internal MIB processes. It must be compiled before any other MIBs are compiled. This does not contain any traps.

ntEnterpriseDataTasmanMgmtsnAg.mib

This MIB allows file uploads/downloads, reload of router, configuration of telnet/welcome banners, TACACS/Radius servers, Syslog server, trap receive managers, DNS configurations etc.

ntEnterpriseDataTasmanMgmtsnmp.mib

This MIB defines objects related to SNMP configuration mainly community and trap_host configurations. This MIB does not contain any traps.

ntEnterpriseDataTasmanMgmtsystem.mib

This MIB defines system objects such as IP Address, hostName and DNS server. Use this MIB to:

- Be notified when an SNTP client is enabled (ntSRsntpEnableNotification).
- Be notified when an SNTP client is disabled (ntSRsntpDisableNotification).
- Be notified when an SNTP client is connected to a network time server (ntSRsntpSuccessNotification)
- Be notified when an SNTP client is having problems connecting to a network time server (ntSRsntpErrorNotification)
- Be notified when an there is a system shutdown (ntSRshutDownNotification)
- Be notified when a user logs in successfully (ntSRuserAccessNotification)
- Be notified when a user logs off(ntSRuserLogOffNotification)
- Be notified when a user is having trouble logging in (ntSRuserLoginFailNotification)

ntEnterpriseDataTasmanMgmtMgmtserial.mib

This MIB manages serial (v.35 type) interfaces.

Use this MIB to:

- Be notified when a serial interface alarm is on (ntSRserialIfAlarmOnTrap).
- Be notified when a serial interface alarm is turned off (ntSRserialIfAlarmOffTrap).

Standard MIBs

Refer to the README file for details. Be sure to compile rfc1214.mib before you compile any standard MIBs. The Standard MIB folder contains the following MIBs:

iana-iftype.mib

This contains the ifType enumerated values needed for rfc1213.mib and rfc2233.mib. Compile iana-iftype.mib before you compile rfc2233.mib. This MIB does not contain any traps.

rfc1213.mib

Standard MIB-II objects for TCP/IP networks. This MIB does not contain any traps.

rfc1315.mib

This MIB manages specified Frame Relay DLCI parameters. Use this MIB to:

Be notified when a Virtual Circuit changes state (frDLCIStatusChange)

rfc1406.mib

MIB objects for DS1 interface. This MIB does not contain any traps.

rfc1407.mib

MIB objects for DS3 interface. This MIB does not contain any traps.

rfc1643.mib

MIB objects for Ethernet-like interface. This MIB does not contain any traps.

rfc1657.mib

This MIB manages specified BGP parameters. This MIB does not contain any traps.

rfc1724.mib

The objects in this MIB manage the RIP2 V2 Protocol in the router.

rfc1850.mib

This MIB manages specified OSPF parameters. This MIB does not contain any traps.

rfc2233.mib

MIB objects for Interface Table extensions including StackTable and ifXTable. The IfStackTable shows the sub-layer relationships of interfaces. This MIB does not contain any traps.

rfc2787.mib

This MIB describes objects used for managing Virtual Router Redundancy Protocol (VRRP) routers.

SNMP Applications Supported

These SNMP v1 and v2 MIBs can be compiled and used with many popular SNMP managers including, but not limited to:

- HP Openview
- MRTG
- SNMPvC
- NetID
- NetCool