

Advanced Juniper Networks Routing in the Enterprise

8.a

Detailed Lab Guide



1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Course Number: EDU-JUN-AJRE

Juniper Networks, the Juniper Networks logo, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOS and JUNOSe are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Advanced Juniper Networks Routing in the Enterprise Detailed Lab Guide, Revision 8.a

Copyright © 2006, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Revision History:

Revision 8.a—December 2006

The information in this document is current as of the date listed above.

The information in this document has been carefully verified and is believed to be accurate for software Release 8.0R2. Juniper Networks assumes no responsibilities for any inaccuracies that may appear in this document. In no event will Juniper Networks be liable for direct, indirect, special, exemplary, incidental or consequential damages resulting from any defect or omission in this document, even if advised of the possibility of such damages.

Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

YEAR 2000 NOTICE

Juniper Networks hardware and software products do not suffer from Year 2000 problems and hence are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

Contents

Lab 1:	JUNOS Policy (Detailed)	1-1
	Part 1: Load Router Configuration	1-2
	Part 2: Establish ISP Connectivity	1-4
	Part 3: Configure Static Routes to ISPs	1-5
	Part 4: Configure Load Balancing	1-7
	Part 5: Configure BGP Policy	1-11
	Part 6: Establish Connectivity to a Partner	1-17
	Part 7: Protect the Router	1-21
Lab 2:	BGP Configuration (Detailed)	2-1
	Part 1: Prepare the Routers	2-2
	Part 2: Establish Internal Connectivity	2-4
	Part 3: Configure IBGP Sessions	2-7
	Part 4: Establish ISP Connectivity	2-9
	Part 5: Configure Intra-AS Routing for BGP Speakers	2-13
	Part 6: Configure Intra-AS Routing for Non-BGP Speakers	2-17
	Part 7: Establish Primary/Secondary Inbound Routing Policy	2-22
	Part 8: Establish Primary/Secondary Outbound Routing Policy	2-30
Lab 3:	IGP Conversion (Detailed)	3-1
	Part 1: Prepare the Routers	3-2
	Part 2: Configure RIP	3-4
	Part 3: Prepare for the Overlay Transition	3-8
	Part 4: Deactivate RIP	3-14
	Part 5: Perform Additional OSPF Configuration	3-17
Lab 4:	Layer 2 Services (Detailed)	4-1
	Part 1: Configure CRTP	4-2
	Part 2: Verify the Operation of the Link Services Interface	4-3
	Part 3: Configure MLPPP	4-8
	Part 4: Configure the MLPPP Interface to Optimize Voice Traffic	4-14
Lab 5:	Stateful Firewall and NAT (Detailed)	5-1
	Part 1: Prepare the Routers	5-2
	Part 2: Configure a Virtual Router	5-8
	Part 3: Configure a Next-Hop-Style Service Set	5-8
	Part 4: Configure NAT Rules	5-10
	Part 5: Verify NAT Operation	5-13
	Part 6: Configure Stateful Firewall Rules	5-15
	Part 7: Verify Stateful Firewall Operation	5-16
	Part 8: Configure an ALG	5-17
	Part 9: Configure an Interface-Style Service Set	5-20
Lab 6:	IPSec VPN (Detailed)	6-1
	Part 1: Prepare the Router	6-2
	Part 2: Configure an IPSec-over-GRE Tunnel	6-4
	Part 3: Configure a Service Filter	6-8
	Part 4: Configure a Stateful Firewall Filter	6-13
	Part 5: Configure a Next-Hop-Style VPN	6-19

Lab 7:	Class of Service (Detailed)	7-1
	Part 1: Prepare the Router.....	7-2
	Part 2: Configure Queues and Scheduler Maps.....	7-3
	Part 3: Configure Multifield Classification.....	7-7
	Part 4: Verify the Operation of the Multifield Classifier.....	7-8
	Part 5: Configure BA Rewrite Rules.....	7-14
	Part 6: Configure BA Classifiers.....	7-16
	Part 7: Configure Virtual Channels.....	7-23
Lab 8:	Branch Office (Optional) (Detailed)	8-1
Appendix A:	Lab Diagrams	A-1

Course Overview

Advanced Juniper Networks Routing in the Enterprise (AJRE) is an instructor-led course designed to provide enterprise network engineers with the knowledge and skills necessary to use Juniper Networks routers to meet their networks' requirements. It covers advanced routing and services configurations of Juniper Networks J-series and M-series platforms, focussing specifically on advanced configurations commonly used in the enterprise environment.

Objectives

After successfully completing this course, you should be able to:

- Explain how Juniper Networks routers evaluate policies.
- Implement a stateless firewall filter to protect the Routing Engine (RE).
- Implement both an inbound and outbound routing policy for traffic over multiple ISP connections using BGP.
- Monitor and troubleshoot BGP sessions and policy application.
- Successfully transition a network running a distance-vector IGP to use a link-state protocol.
- Configure Layer 2 services, including the Multilink Point-to-Point (MLPPP) protocol, Multilink Frame Relay (MLFR), and the Compressed Real-Time Transport Protocol (CRTP).
- Implement a stateful firewall to enforce a firewall policy.
- Implement Network Address Translation (NAT).
- With a virtual private network (VPN) design, create the VPNs between some combination of J-series and M-series routers.
- Given a class-of-service (CoS) design, configure appropriate CoS policies.
- List two branch-office connectivity solutions and the traffic considerations that apply to each.
- Given a list of technical constraints, implement certain branch-office connectivity solutions.
- Explain the interaction of various configuration elements with CoS configuration.

Intended Audience

The primary audience for this course is network engineers who work in an enterprise environment.

Course Level

This is an advanced-level course.

Prerequisites

The following are the prerequisites for this course:

- The *Operating Juniper Networks Routers in the Enterprise* (OJRE) course or equivalent experience;
- Knowledge, familiarity, and comfort with the JUNOS CLI;
- Experience managing routers (not necessarily Juniper Networks) in an enterprise environment;
- Understanding of destination-based, hop-by-hop IP routing in a Classless Inter-Domain Routing (CIDR) environment; and
- Experience with IGPs in an enterprise environment.

Not For Reproduction

Course Agenda

Day 1

Lab 1: JUNOS Policy (Detailed)

Day 2

Lab 2: BGP Configuration (Detailed)

Lab 3: IGP Conversion (Detailed)

Lab 4: Layer 2 Services (Detailed)

Day 3

Lab 5: Stateful Firewall and NAT (Detailed)

Lab 6: IPSec VPN (Detailed)

Day 4

Lab 7: Class of Service (Detailed)

Lab 8: Branch Office (Optional) (Detailed)

Document Conventions

CLI and GUI Text

Frequently throughout this course, we refer to text that appears in a command-line interface (CLI) or a graphical user interface (GUI). To make the language of these documents easier to read, we distinguish GUI and CLI text from chapter text according to the following table.

Style	Description	Usage Example
Franklin Gothic	Normal text.	Most of what you read in the Lab Guide and Student Guide.
Courier New	Console text: <ul style="list-style-type: none">• Screen captures• Noncommand-related syntax	<code>commit complete</code> Exiting configuration mode
Century Gothic	GUI text elements: <ul style="list-style-type: none">• Menu names• Text field entry	Select File > Open, and then click Configuration.conf in the Filename text box.

Input Text Versus Output Text

You will also frequently see cases where you must enter input text yourself. Often this will be shown in the context of where you must enter it. We use bold style to distinguish text that is input versus text that is simply displayed.

Style	Description	Usage Example
Normal CLI	No distinguishing variant.	Physical interface:fxp0, Enabled
Normal GUI		View configuration history by clicking Configuration > History.
CLI Input GUI Input	Text that you must enter.	lab@San_Jose> show route Select File > Save, and enter config.ini in the Filename field.

Defined and Undefined Syntax Variables

Finally, this course distinguishes between regular text and syntax variables, and it also distinguishes between syntax variables where the value is already assigned (defined variables) and syntax variables where you must assign the value (undefined variables). Note that these styles can be combined with the input style as well.

Style	Description	Usage Example
<i>CLI Variable</i>	Text where variable value is already assigned.	<code>policy my-peers</code>
<i>GUI Variable</i>		Click on <i>my-peers</i> in the dialog.
<u><i>CLI Undefined</i></u>	Text where the variable's value is the user's discretion and text where the variable's value as shown in the lab guide might differ from the value the use must input.	Type set policy <u>policy-name</u> .
<u><i>GUI Undefined</i></u>		ping 10.0.1.1 Select File > Save, and enter <u>filename</u> in the Filename field.

Additional Information

Education Services Offerings

You can obtain information on the latest Education Services offerings, course dates, and class locations from the World Wide Web by pointing your Web browser to:
<http://www.juniper.net/training/education/>.

About This Publication

The *Advanced Juniper Networks Routing in the Enterprise Detailed Lab Guide* was developed and tested using software version 8.0R2. Previous and later versions of software may behave differently so you should always consult the documentation and release notes for the version of code you are running before reporting errors.

This document is written and maintained by the Juniper Networks Education Services development team. Please send questions and suggestions for improvement to training@juniper.net.

Technical Publications

You can print technical manuals and release notes directly from the Internet in a variety of formats:

- Go to <http://www.juniper.net/techpubs/>.
- Locate the specific software or hardware release and title you need, and choose the format in which you want to view or print the document.

Documentation sets and CDs are available through your local Juniper Networks sales office or account representative.

Juniper Networks Support

For technical support, contact Juniper Networks at <http://www.juniper.net/customers/support/>, or at 1-888-314-JTAC (within the United States) or 408-745-2121 (from outside the United States).

JUNOS Policy (Detailed)

Overview

This lab explores policy configuration. First, you will establish connectivity to an ISP with static routing. You will configure the router to load-balance between multiple next hops to a destination. You will then activate two preconfigured BGP sessions and will use policy to control the way routes are advertised to the ISP. You will establish connectivity to a partner router with a static route and will configure unicast reverse-path forwarding (RPF) checks. Finally, you will configure and apply a firewall filter to control access to the router.

The lab is available in two formats: a high-level format that is designed to make you think through each step and a detailed format that offers step-by-step instructions complete with sample output from most commands.

By completing this lab, you will perform the following tasks:

- Load a router configuration.
- Establish ISP connectivity.
- Configure static routes to ISPs.
- Configure load balancing.
- Configure BGP policy.
- Establish connectivity to a partner.
- Protect the router.

This lab requires you to use the Sydney router as a route server. You use it to view the routing table of the ISPs, each of which is available as a virtual router on Sydney.

Key Commands

Key operational-mode commands used in this lab include the following:

```
ping
show bgp summary
show configuration
show route
show route advertising-protocol
show route forwarding-table
show route table
telnet
traceroute
```

Part 1: Load Router Configuration

In this part, you will load a base configuration file that includes some basic system configuration and two preconfigured inactive BGP sessions.

Step 1.1

Log in to the router with the username *lab* and the password supplied by your instructor. Note that both the name and the password are case sensitive.

```
HongKong (ttyd0)
```

```
login: lab
Password:
```

```
--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC
lab@HongKong>
```

Step 1.2

Perform a **load override** of the configuration file located on the router at */var/home/lab/ajre/lab1-reset.conf*.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# load override ajre/lab1-reset.conf
load complete
```

Step 1.3

View the configuration file. You should see a deactivated BGP configuration.

```
[edit]
lab@HongKong# show
version 8.0R2.8;
system {
    host-name HongKong;
    root-authentication {
```

```

        encrypted-password "$1$KI99zGk6$MbYFuBbpLffu9tn2.sI7l1"; ## SECRET-DATA
    }
    login {
        user lab {
            uid 2000;
            class super-user;
            authentication {
                encrypted-password "$1$84J5Maes$cni5Hrazbd/IEHr/50oY30"; ##
SECRET-DATA
            }
        }
    }
    services {
        ftp;
        ssh;
        telnet;
        web-management {
            http;
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
interfaces {
    fe-0/0/0 {
        description "MGMT INTERFACE - DO NOT DELETE";
        unit 0 {
            family inet {
                address 10.210.9.177/28;
            }
        }
    }
}
routing-options {
    autonomous-system 65108;
}
protocols {
    inactive: bgp {
        group isp {
            neighbor 172.17.39.17 {
                description isp-a;
                peer-as 65010;
            }
            neighbor 172.17.55.17 {

```

```

        description isp-c;
        peer-as 65030;
    }
}
}

```

Step 1.4

Configure the lo0 interface using the IP addresses found on the “Lab 1a: Policy” diagram. Make the first loopback address listed the primary and preferred loopback address.

```

[edit]
lab@HongKong# edit interfaces lo0 unit 0

[edit interfaces lo0 unit 0]
lab@HongKong# set family inet address 10.14.8.1/32 primary preferred

[edit interfaces lo0 unit 0]
lab@HongKong# set family inet address 10.14.8.254

```

Part 2: Establish ISP Connectivity

In this part, you configure connectivity to two ISPs.

Step 2.1

Configure Frame Relay connectivity to ISP A according to the “Lab 1a: Policy” diagram. Use the IP address found in the following table:

IP Addresses for ISP A Connectivity

Router	IP Address
Hong Kong	172.17.39.18/30
Tokyo	172.17.39.22/30
London	172.17.39.26/30
Amsterdam	172.17.39.30/30
Montreal	172.17.39.34/30
San Jose	172.17.39.38/30
Denver	172.17.39.42/30
Sao Paulo	172.17.39.46/30

```

[edit]
lab@HongKong# edit interfaces se-1/0/0

[edit interfaces se-1/0/0]
lab@HongKong# set encapsulation frame-relay

```

```
[edit interfaces se-1/0/0]
lab@HongKong# edit unit 101

[edit interfaces se-1/0/0 unit 101]
lab@HongKong# set dlci 101

[edit interfaces se-1/0/0 unit 101]
lab@HongKong# set family inet address 172.17.39.18/30
```

Step 2.2

Configure Frame Relay connectivity to ISP C according to the “Lab 1a: Policy” diagram. Use the IP address found in the following table:

IP Addresses for ISP B Connectivity

Router	IP Address
Hong Kong	172.17.55.18/30
Tokyo	172.17.55.22/30
London	172.17.55.26/30
Amsterdam	172.17.55.30/30
Montreal	172.17.55.34/30
San Jose	172.17.55.38/30
Denver	172.17.55.42/30
Sao Paulo	172.17.55.46/30

```
edit interfaces se-1/0/0 unit 101]
lab@HongKong# up

[edit interfaces se-1/0/0]
lab@HongKong# edit unit 201

[edit interfaces se-1/0/0 unit 201]
lab@HongKong# set dlci 201

[edit interfaces se-1/0/0 unit 201]
lab@HongKong# set family inet address 172.17.55.18/30
```

Part 3: Configure Static Routes to ISPs

In this part, you configure a static default route with multiple next hops.

Step 3.1

Add a static default route with next hops to both ISP A and ISP C.

```
[edit routing-options]
lab@HongKong# set static route 0.0.0.0/0 next-hop se-1/0/0.101

[edit routing-options]
lab@HongKong# set static route 0.0.0.0/0 next-hop se-1/0/0.201
```

Step 3.2

Commit the changes and determine which next hop the router chose for the default route.

```
[edit routing-options]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode

lab@HongKong> show route 0/0 exact extensive
```

```
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
0.0.0.0/0 (1 entry, 1 announced)
```

TSI:

```
KRT in-kernel 0.0.0.0/0 -> {se-1/0/0.101}
  *Static Preference: 5
    Next-hop reference count: 2
    Next hop: via se-1/0/0.101, selected
    Next hop: via se-1/0/0.201
    State: <Active Int Ext>
    Local AS: 65108
    Age: 1:03:00
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I
```

```
lab@HongKong> show route forwarding-table destination 0/0
```

Routing table: inet

Internet:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	user	0		ucst	323	2	se-1/0/0.101
default	perm	0		rjct	14	1	

Question: Which next hop did the router choose for the default route?

Answer: One of the two available next hops is selected. The next hop selected will vary. In this case, HongKong selected the `se-1/0/0.101` next hop to ISP A.

Step 3.3

Verify the route(s) the router chose by tracing the route to 172.17.24.1 (an IP address in ISP B's network).


```
lab@HongKong> traceroute 172.17.24.1
traceroute to 172.17.24.1 (172.17.24.1), 30 hops max, 40 byte packets
 1  172.17.39.17 (172.17.39.17)  15.213 ms  15.221 ms  10.002 ms
 2  172.17.24.1 (172.17.24.1)   9.887 ms  29.774 ms  9.525 ms
```

Part 4: Configure Load Balancing

In this part, you will configure the router to load-balance traffic across all available next hops for the default route.

Step 4.1

Create a policy called *load-balance-default* that marks all default routes to be load-balanced.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit policy-options

[edit policy-options]
lab@HongKong# edit policy-statement load-balance-default

[edit policy-options policy-statement load-balance-default]
lab@HongKong# set term def-route from route-filter 0.0.0.0/0 exact

[edit policy-options policy-statement load-balance-default]
lab@HongKong# set term def-route then load-balance per-packet

[edit policy-options policy-statement load-balance-default]
lab@HongKong# show
term def-route {
  from {
    route-filter 0.0.0.0/0 exact;
  }
  then {
    load-balance per-packet;
  }
}
```

Step 4.2

Apply the policy to the routes as they are exported to the forwarding table. Commit the configuration.

```
[edit policy-options policy-statement load-balance-default]
lab@HongKong# top

[edit]
lab@HongKong# edit routing-options forwarding-table

[edit routing-options forwarding-table]
lab@HongKong# set export load-balance-default
```

```
[edit routing-options forwarding-table]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 4.3

Determine which next hops the router chose for the default route.

```
lab@HongKong> show route 0/0 exact extensive
```

```
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
0.0.0.0/0 (1 entry, 1 announced)
```

TSI:

```
KRT in-kernel 0.0.0.0/0 -> {se-1/0/0.101, se-1/0/0.201}
```

```
*Static Preference: 5
```

```
Next-hop reference count: 2
```

```
Next hop: via se-1/0/0.101, selected
```

```
Next hop: via se-1/0/0.201
```

```
State: <Active Int Ext>
```

```
Local AS: 65108
```

```
Age: 58:50
```

```
Task: RT
```

```
Announcement bits (1): 0-KRT
```

```
AS path: I
```

```
lab@HongKong> show route forwarding-table destination 0/0
```

```
Routing table: inet
```

Internet:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	user	0		ulst	262142	1	
				ucst	323	2	se-1/0/0.101
				ucst	336	2	se-1/0/0.201
default	perm	0		rjct	14	1	

Question: Which next hop(s) did the router choose for the default route?

Answer: The router now installed both next hops in the forwarding table.

Step 4.4

Verify connectivity by tracing the route to 172.17.24.1 (an IP address in ISP B's network).

```
lab@HongKong> traceroute 172.17.24.1
```

```
traceroute to 172.17.24.1 (172.17.24.1), 30 hops max, 40 byte packets
```

```
1 172.17.55.17 (172.17.55.17) 9.108 ms 8.790 ms 172.17.39.17 (172.17.39.17)
10.137 ms
```

```
2 172.17.24.1 (172.17.24.1) 9.309 ms 9.252 ms 10.200 ms
```

Question: How do you explain this output?

Answer: Packets originated by the Routing Engine are actually load-balanced per packet rather than per flow. You can see this load balancing in the first hop of the **traceroute** output in which one packet was sent to 172.17.55.17 and one was sent to 172.17.39.17.

Step 4.5

Verify that the router has other routes in the forwarding table.

```
lab@HongKong> show route forwarding-table
```

Routing table: inet

Internet:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	user	0		ulst	262142	1	
				ucst	323	2	se-1/0/0.101
				ucst	336	2	se-1/0/0.201
default	perm	0		rjct	14	1	
10.14.8.1/32	intf	0	10.14.8.1	locl	320	1	
10.210.9.176/28	intf	0		rslv	333	1	fe-0/0/0.0
10.210.9.176/32	dest	0	10.210.9.176	recv	331	1	fe-0/0/0.0
10.210.9.177/32	intf	0	10.210.9.177	locl	332	2	
10.210.9.177/32	dest	0	10.210.9.177	locl	332	2	
10.210.9.178/32	dest	0	0:5:85:ca:2a:d0	ucst	342	1	fe-0/0/0.0
10.210.9.190/32	dest	1	0:10:db:ff:20:50	ucst	356	2	fe-0/0/0.0
10.210.9.191/32	dest	0	10.210.9.191	bcst	330	1	fe-0/0/0.0
172.17.39.16/30	intf	0	dlci: 101	ucst	325	1	se-1/0/0.101
172.17.39.16/32	dest	0	172.17.39.16	recv	327	1	se-1/0/0.101
172.17.39.18/32	intf	0	172.17.39.18	locl	324	1	
172.17.39.19/32	dest	0	172.17.39.19	bcst	326	1	se-1/0/0.101
172.17.55.16/30	intf	0	dlci: 201	ucst	329	1	se-1/0/0.201
172.17.55.16/32	dest	0	172.17.55.16	recv	335	1	se-1/0/0.201
172.17.55.18/32	intf	0	172.17.55.18	locl	328	1	
172.17.55.19/32	dest	0	172.17.55.19	bcst	334	1	se-1/0/0.201
224.0.0.0/4	perm	0		mdsc	13	1	
224.0.0.1/32	perm	0	224.0.0.1	mcst	9	1	
255.255.255.255/32	perm	0		bcst	10	1	

Routing table: __juniper_private1__.inet

Internet:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	62	1	
10.0.0.1/32	intf	1	10.0.0.1	locl	321	2	
10.0.0.16/32	intf	0	10.0.0.16	locl	322	1	
224.0.0.0/4	perm	0		mdsc	61	1	
224.0.0.1/32	perm	0	224.0.0.1	mcst	57	1	
255.255.255.255/32	perm	0		bcst	58	1	

Routing table: iso

ISO:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	50	1	

Routing table: inet6

Internet6:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	26	1	
ff00::/8	perm	0		mdsc	25	1	
ff02::1/128	perm	0	ff02::1	mcst	21	1	

Routing table: __juniper_private1__.inet6

Internet6:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	74	1	
ff00::/8	perm	0		mdsc	73	1	
ff02::1/128	perm	0	ff02::1	mcst	69	1	

Routing table: mpls

MPLS:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		dscd	36	1	

Question: If your policy only has one term that matches the default route exactly, why are there other routes in the forwarding table? (If your policy has additional terms that accepted additional routes, remove those additional terms, and verify that the router still has other routes in the forwarding table. Then return to this question.)

Answer: The default export policy to the forwarding table accepts all routes.

Question: If multiple next hops were available for these other routes, would the router install multiple next hops in the forwarding table for these additional routes?

Answer: No, it would not install multiple next hops for these additional routes. It chooses a single next hop except for those specific routes marked for load-balancing through an export policy.

Part 5: Configure BGP Policy

In this part, you will activate preconfigured BGP sessions and control advertisements to your providers through the use of policy.

Step 5.1

Create an aggregate route to announce via BGP. You should create an aggregate route for the /24 that includes your loopback addresses.

```
[edit]
lab@HongKong# edit routing-options

[edit routing-options]
lab@HongKong# set aggregate route 10.14.8.0/24
```

Step 5.2

Next, create a prefix list called *announce-to-ISP* that includes the aggregate you created in the last step.

```
[edit policy-options]
lab@HongKong# edit prefix-list announce-to-ISP

[edit policy-options prefix-list announce-to-ISP]
lab@HongKong# set 10.14.8.0/24
```

Step 5.3

Create a policy called *to-ISP* that accepts the routes in the *announce-to-ISP* prefix list.

```
[edit policy-options prefix-list announce-to-ISP]
lab@HongKong# up

[edit policy-options]
lab@HongKong# edit policy-statement to-ISP term announce-to-ISP

[edit policy-options policy-statement to-ISP term announce-to-ISP]
lab@HongKong# set from prefix-list announce-to-ISP

[edit policy-options policy-statement to-ISP term announce-to-ISP]
lab@HongKong# set then accept
```

Step 5.4

Create a policy called *reject-all* that rejects all routes.

```
[edit policy-options policy-statement to-ISP term announce-to-ISP]
lab@HongKong# up 2 edit policy-statement reject-all

[edit policy-options policy-statement reject-all]
lab@HongKong# set then reject
```

Step 5.5

Apply a policy chain to control the routes exported to the *isp* group. The routes should be processed by the *to-ISP* and *reject-all* policies.

```
[edit policy-options policy-statement reject-all]
lab@HongKong# top edit protocols bgp group isp
```

```
[edit protocols bgp group isp]
lab@HongKong# set export [ to-ISP reject-all ]
```

Step 5.6

Activate the BGP sessions and commit the configuration.

```
[edit protocols bgp group isp]
lab@HongKong# top
```

```
[edit]
lab@HongKong# activate protocols bgp
```

Step 5.7

Commit your changes and verify that the BGP sessions are established.

```
[edit]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

```
lab@HongKong> show bgp summary
```

```
Groups: 1 Peers: 2 Down peers: 0
```

Table	Tot Paths	Act Paths	Suppressed	History	Damp	State	Pending
inet.0	20	10	0	0	0	0	0
Peer	AS	InPkt	OutPkt	OutQ	Flaps	Last Up/Dwn	
State #Active/Received/Damped...							
172.17.39.17	65010	13	4	0	0	15 9/10/0	
0/0/0							
172.17.55.17	65030	13	3	0	0	11 1/10/0	
0/0/0							

Step 5.8

Ensure that the router is receiving routes from the ISPs.

```
lab@HongKong> show route receive-protocol bgp 172.17.39.17
```

```
inet.0: 20 destinations, 30 routes (20 active, 0 holddown, 0 hidden)
```

Prefix	Nexthop	MED	Lclpref	AS path
* 10.14.9.0/24	172.17.39.17			65010 I
* 10.14.10.0/24	172.17.39.17			65010 I
* 10.14.11.0/24	172.17.39.17			65010 I
* 10.14.12.0/24	172.17.39.17			65010 I
* 10.14.13.0/24	172.17.39.17			65010 I
* 10.14.14.0/24	172.17.39.17			65010 I
* 10.14.15.0/24	172.17.39.17			65010 I
* 172.17.24.0/21	172.17.39.17			65010 65020 I
* 172.17.32.0/20	172.17.39.17			65010 I
172.17.48.0/20	172.17.39.17			65010 65020
65030 I				

```
__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

```
lab@HongKong> show route receive-protocol bgp 172.17.55.17
```

```
inet.0: 20 destinations, 30 routes (20 active, 0 holddown, 0 hidden)
```

Prefix	Nexthop	MED	Lclpref	AS path
10.14.9.0/24	172.17.55.17			65030 I
10.14.10.0/24	172.17.55.17			65030 I
10.14.11.0/24	172.17.55.17			65030 I
10.14.12.0/24	172.17.55.17			65030 I
10.14.13.0/24	172.17.55.17			65030 I
10.14.14.0/24	172.17.55.17			65030 I
10.14.15.0/24	172.17.55.17			65030 I
172.17.24.0/21	172.17.55.17			65030 65020 I
172.17.32.0/20	172.17.55.17			65030 65020
65010 I				
* 172.17.48.0/20	172.17.55.17			65030 I

```
__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Step 5.9

Remove the static default route and commit your changes.

```
lab@HongKong> configure
```

```
Entering configuration mode
```

```
[edit]
```

```
lab@HongKong# delete routing-options static route 0/0
```

```
[edit]
```

```
lab@HongKong# commit and-quit
```

```
commit complete
```

```
Exiting configuration mode
```

Step 5.10

Check that the aggregate route is being created.

```
lab@HongKong> show route protocol aggregate
```

```
inet.0: 19 destinations, 29 routes (19 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
10.14.8.0/24      *[Aggregate/130] 00:10:02
                  Reject
```

```
__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Step 5.11

Determine which routes the router is sending to the ISP peers.

```
lab@HongKong> show route advertising-protocol bgp 172.17.39.17
```

```
inet.0: 19 destinations, 29 routes (19 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref    AS path
* 10.14.8.0/24          Self              0

```

```
lab@HongKong> show route advertising-protocol bgp 172.17.55.17
```

```
inet.0: 19 destinations, 29 routes (19 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref    AS path
* 10.14.8.0/24          Self              0

```

Step 5.12

Reference the management network diagram provided by your instructor. Use Telnet to access the Sydney router's fe-0/0/0 interface IP address. Log in using the username *routeserver* and the password provided by your instructor. This account gives you read-only access, which you can use to view routing tables and to ping and traceroute from several different networks. The *isp-a*, *isp-b*, and *isp-c* virtual routers are on those three networks. View the routing table for the *isp-a* and *isp-c* virtual routers and ensure that they are receiving these routes.

```
lab@HongKong> telnet 10.210.9.185
```

```
Trying 10.210.9.185...
```

```
Connected to 10.210.9.185.
```

```
Escape character is '^]'.
```

```
Sydney (ttyp2)
```

```
login: routeserver
```

```
Password:
```

```
--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC
```

NOTE: This router is divided into many virtual routers used by different teams. Please only configure your own virtual router.

You must use 'configure private' to configure this router.

```
routeserver@Sydney> show route table isp-a protocol bgp terse
```

```
isp-a.inet.0: 30 destinations, 46 routes (30 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
* 10.14.8.0/24	B 160	100		>172.17.39.18	65108 I
	B 170	80		>172.17.25.1	65020 65030
65108 I					
* 10.14.9.0/24	B 160	100		>172.17.39.22	65109 I
	B 170	80		>172.17.25.1	65020 65030
65109 I					
* 10.14.10.0/24	B 160	100		>172.17.39.26	65110 I
	B 170	80		>172.17.25.1	65020 65030
65110 I					
* 10.14.11.0/24	B 160	100		>172.17.39.30	65111 I


```

65111 I
* 10.14.12.0/24 B 170 80 >172.17.25.1 65020 65030
B 170 80 >172.17.39.34 65112 I
>172.17.25.1 65020 65030
65112 I
* 10.14.13.0/24 B 160 100 >172.17.39.38 65113 I
B 170 80 >172.17.25.1 65020 65030
65113 I
* 10.14.14.0/24 B 160 100 >172.17.39.42 65114 I
B 170 80 >172.17.25.1 65020 65030
65114 I
* 10.14.15.0/24 B 160 100 >172.17.39.46 65115 I
B 170 80 >172.17.25.1 65020 65030
65115 I
* 172.17.24.0/21 B 170 80 >172.17.25.1 65020 I
* 172.17.48.0/20 B 170 80 >172.17.25.1 65020 65030 I

```

```
routeserver@Sydney> show route table isp-c protocol bgp terse
```

```
isp-c.inet.0: 30 destinations, 38 routes (30 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
* 10.14.8.0/24	B 160	100		>172.17.55.18	65108 I
* 10.14.9.0/24	B 160	100		>172.17.55.22	65109 I
* 10.14.10.0/24	B 160	100		>172.17.55.26	65110 I
* 10.14.11.0/24	B 160	100		>172.17.55.30	65111 I
* 10.14.12.0/24	B 160	100		>172.17.55.34	65112 I
* 10.14.13.0/24	B 160	100		>172.17.55.38	65113 I
* 10.14.14.0/24	B 160	100		>172.17.55.42	65114 I
* 10.14.15.0/24	B 160	100		>172.17.55.46	65115 I
* 172.17.24.0/21	B 170	80		>172.17.25.5	65020 I
* 172.17.32.0/20	B 170	80		>172.17.25.5	65020 65010 I

Note

Depending on how many teams have completed the previous step, your output might be different than the capture shown.

Step 5.13

On your router, create a policy called *add-community-65000* that sets community AS:65000. Apply this to the beginning of the export policy chain for the *isp* group and commit the configuration.

```
lab@HongKong> configure
Entering configuration mode
```

```
[edit]
lab@HongKong# edit policy-options
```

```
[edit policy-options]
lab@HongKong# set community community-65000 members 65108:65000
```

```

[edit policy-options]
lab@HongKong# edit policy-statement add-community-65000

[edit policy-options policy-statement add-community-65000]
lab@HongKong# set then community set community-65000

[edit policy-options policy-statement add-community-65000]
lab@HongKong# top edit protocols bgp group isp

[edit protocols bgp group isp]
lab@HongKong# set export add-community-65000

[edit protocols bgp group isp]
lab@HongKong# insert export add-community-65000 before to-ISP

[edit protocols bgp group isp]
lab@HongKong# show
export [ add-community-65000 to-ISP reject-all ];
neighbor 172.17.39.17 {
    description isp-a;
    peer-as 65010;
}
neighbor 172.17.55.17 {
    description isp-c;
    peer-as 65030;
}

[edit protocols bgp group isp]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode

```

Step 5.14

On the Sydney router, verify that the community is present on *isp-a*'s BGP-learned route.

```

routeserver@Sydney> show route table isp-a protocol bgp 10.14.8.0/24 detail
isp-a.inet.0: 30 destinations, 46 routes (30 active, 0 holddown, 0 hidden)
10.14.8.0/24 (3 entries, 1 announced)
    *BGP      Preference: 160/-101
                Next-hop reference count: 4
                Source: 172.17.39.18
                Next hop: 172.17.39.18 via se-3/0/0.101, selected
                State: <Active Ext>
                Local AS: 65010 Peer AS: 65108
                Age: 17:48
                Task: BGP_65108_65010.172.17.39.18+2554
                Announcement bits (2): 0-BGP RT Background 2-KRT
                AS path: 65108 I Aggregator: 65108 10.14.8.1
                Communities: 65010:444 65108:65000
                Localpref: 100
                Router ID: 10.14.8.1
    BGP      Preference: 170/-81
                Next-hop reference count: 12
                Source: 172.17.25.1

```

```

Next hop: 172.17.25.1 via lt-0/0/0.10, selected
State: <Ext>
Inactive reason: Route Preference
Local AS: 65010 Peer AS: 65020
Age: 17:48
Task: BGP_65020_65010.172.17.25.1+1293
AS path: 65020 65030 65108 I Aggregator: 65108 10.14.8.1
Communities: 65010:555 65020:444 65108:65000
Localpref: 80
Router ID: 172.17.24.1

```



Wait for your partner team to complete this part before continuing.

Part 6: Establish Connectivity to a Partner

In this part, you will work with a partner team to establish connectivity over a Frame Relay PVC. You will then configure reverse-path forwarding (RPF) on this link.

Step 6.1

Work with your partner team to establish connectivity according to the “Lab 1b: Policy” diagram.

```

lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit interfaces se-1/0/0 unit 601

[edit interfaces se-1/0/0 unit 601]
lab@HongKong# set dlci 601

[edit interfaces se-1/0/0 unit 601]
lab@HongKong# set family inet address 192.168.25.1/30

[edit interfaces se-1/0/0 unit 601]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode

lab@HongKong> ping count 4 192.168.25.2
PING 192.168.25.2 (192.168.25.2): 56 data bytes
64 bytes from 192.168.25.2: icmp_seq=0 ttl=64 time=21.835 ms
64 bytes from 192.168.25.2: icmp_seq=1 ttl=64 time=20.146 ms
64 bytes from 192.168.25.2: icmp_seq=2 ttl=64 time=20.155 ms
64 bytes from 192.168.25.2: icmp_seq=3 ttl=64 time=20.414 ms

--- 192.168.25.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 20.146/20.638/21.835/0.700 ms

```

Step 6.2

Determine if your router has a route to your partner's loopback IP address and, if so, the path it will follow.

```
lab@HongKong> show route 10.14.9.1

inet.0: 21 destinations, 31 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.14.9.0/24          *[BGP/170] 01:01:42, localpref 100
                     AS path: 65030 65109 I
                     > to 172.17.55.17 via se-1/0/0.201
                     [BGP/170] 01:01:46, localpref 100
                     AS path: 65010 65109 I
                     > to 172.17.39.17 via se-1/0/0.101

lab@HongKong> traceroute 10.14.9.1
traceroute to 10.14.9.1 (10.14.9.1), 30 hops max, 40 byte packets
 1  172.17.55.17 (172.17.55.17)  8.984 ms  8.743 ms  9.976 ms
 2  10.14.9.1 (10.14.9.1)  19.759 ms  20.251 ms  19.250 ms
```

Step 6.3

Reference the management network diagram provided by your instructor. Use Telnet to access your partner router's fe-0/0/0 management IP address. From your partner's router, ping your router's directly connected Frame Relay interface. Source the ping from the lo0 interface IP address.

```
lab@HongKong> telnet 10.210.9.178
Trying 10.210.9.178...
Connected to 10.210.9.178.
Escape character is '^]'.

Tokyo (ttyp0)

login: lab
Password:

--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC
lab@Tokyo> ping count 4 192.168.25.1 source 10.14.9.1
PING 192.168.25.1 (192.168.25.1): 56 data bytes
64 bytes from 192.168.25.1: icmp_seq=0 ttl=63 time=19.972 ms
64 bytes from 192.168.25.1: icmp_seq=1 ttl=63 time=20.635 ms
64 bytes from 192.168.25.1: icmp_seq=2 ttl=63 time=40.657 ms
64 bytes from 192.168.25.1: icmp_seq=3 ttl=63 time=40.492 ms

--- 192.168.25.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.972/30.439/40.657/10.138 ms
```

Step 6.4

On your router, configure a static route for your partner's aggregate route that directs traffic over the Frame Relay connection. Configure a route preference of 200 for the static route. Commit the configuration.

```

lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit routing-options static route 10.14.9.0/24

[edit routing-options static route 10.14.9.0/24]
lab@HongKong# set next-hop se-1/0/0.601

[edit routing-options static route 10.14.9.0/24]
lab@HongKong# set preference 200

[edit routing-options static route 10.14.9.0/24]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode

```

Step 6.5

Verify that the both the static and BGP-learned routes appear in your router's routing table.

```

lab@HongKong> show route 10.14.9.0/24

inet.0: 21 destinations, 32 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.14.9.0/24          *[BGP/170] 01:33:32, localpref 100
                     AS path: 65030 65109 I
                     > to 172.17.55.17 via se-1/0/0.201
                     [BGP/170] 01:33:36, localpref 100
                     AS path: 65010 65109 I
                     > to 172.17.39.17 via se-1/0/0.101
                     [Static/200] 00:03:56
                     > via se-1/0/0.601

```

Step 6.6

Configure your router to perform unicast RPF checks on the interface to your partner. RPF should use the default settings. Commit the configuration.

```

lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit interfaces se-1/0/0 unit 601 family inet

[edit interfaces se-1/0/0 unit 601 family inet]
lab@HongKong# set rpf-check

[edit interfaces se-1/0/0 unit 601 family inet]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode

```

Step 6.7

Reference the management network diagram provided by your instructor. Use Telnet to access your partner router's `fe-0/0/0` management IP address. From your partner's router ping your router's directly connected Frame Relay interface. Source the ping from the `lo0` interface IP address.

```
lab@HongKong> telnet 10.210.9.178
Trying 10.210.9.178...
Connected to 10.210.9.178.
Escape character is '^]'.
```

```
Tokyo (ttyp0)
```

```
login: lab
Password:
```

```
--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC
lab@Tokyo> ping count 4 192.168.25.1 source 10.14.9.1
PING 192.168.25.1 (192.168.25.1): 56 data bytes
```

```
--- 192.168.25.1 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

Question: Does the ping succeed or fail? Why or why not?

Answer: The ping fails. It is denied by the RPF filter because the packet arrived on an interface other than the active next hop to the source address.

Step 6.8

On your router, configure the unicast RPF check to consider all feasible paths. Commit the configuration.

```
lab@Tokyo> exit
```

```
Connection closed by foreign host.
```

```
lab@HongKong> configure
Entering configuration mode
```

```
[edit]
lab@HongKong# edit routing-options forwarding-table
```

```
[edit routing-options forwarding-table]
lab@HongKong# set unicast-reverse-path feasible-paths
```

```
[edit routing-options forwarding-table]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 6.9

Reference the management network diagram provided by your instructor. Use Telnet to access your partner router's fe-0/0/0 management IP address. From your partner's router, ping your router's directly connected Frame Relay interface. Source the ping from the lo0 interface IP address.

```
lab@HongKong> telnet 10.210.9.178
Trying 10.210.9.178...
Connected to 10.210.9.178.
Escape character is '^]'.
```

```
Tokyo (ttyp0)
```

```
login: lab
Password:
```

```
--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC
lab@Tokyo> ping count 4 192.168.25.1 source 10.14.9.1
PING 192.168.25.1 (192.168.25.1): 56 data bytes
64 bytes from 192.168.25.1: icmp_seq=0 ttl=63 time=14.527 ms
64 bytes from 192.168.25.1: icmp_seq=1 ttl=63 time=20.218 ms
64 bytes from 192.168.25.1: icmp_seq=2 ttl=63 time=40.216 ms
64 bytes from 192.168.25.1: icmp_seq=3 ttl=63 time=20.478 ms

--- 192.168.25.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 14.527/23.860/40.216/9.738 ms
```

Question: Does the ping succeed or fail? Why or why not?

Answer: The ping succeeds. It is permitted by the RPF filter because the packet arrived on an interface that is the next hop to a feasible route to the source address.

Part 7: Protect the Router

In this part, you will protect the router with a stateless firewall filter applied to the lo0 interface.

Step 7.1

Configure a firewall filter that denies Telnet connections from your partner's second loopback IP address and permits all other traffic.

```
lab@Tokyo> exit
```

Connection closed by foreign host.

```
lab@HongKong> configure
```

Entering configuration mode

```
[edit]
```

```
lab@HongKong# edit firewall family inet filter re-protect
```

```
[edit firewall family inet filter re-protect]
```

```
lab@HongKong# edit term deny-telnet-from-Tokyo
```

```
[edit firewall family inet filter re-protect term deny-telnet-from-Tokyo]
```

```
lab@HongKong# set from source-address 10.14.9.254/32
```

```
[edit firewall family inet filter re-protect term deny-telnet-from-Tokyo]
```

```
lab@HongKong# set from protocol tcp
```

```
[edit firewall family inet filter re-protect term deny-telnet-from-Tokyo]
```

```
lab@HongKong# set from destination-port 23
```

```
[edit firewall family inet filter re-protect term deny-telnet-from-Tokyo]
```

```
lab@HongKong# set then reject
```

```
[edit firewall family inet filter re-protect term deny-telnet-from-Tokyo]
```

```
lab@HongKong# up 1 edit term permit-all
```

```
[edit firewall family inet filter re-protect term permit-all]
```

```
lab@HongKong# set then accept
```

Step 7.2

Apply the firewall filter to the lo0 interface. Commit the configuration.

```
[edit firewall family inet filter re-protect term permit-all]
```

```
lab@HongKong# top edit interfaces lo0.0 family inet
```

```
[edit interfaces lo0 unit 0 family inet]
```

```
lab@HongKong# set filter input re-protect
```

```
[edit interfaces lo0 unit 0 family inet]
```

```
lab@HongKong# commit and-quit
```

commit complete

Exiting configuration mode

Step 7.3

Reference the management network diagram provided by your instructor. Use Telnet to access your partner router's fe-0/0/0 management IP address. Attempt to use Telnet to access your router's directly connected Frame Relay interface. Source the packets from your partner router's second loopback address.


```
lab@HongKong> telnet 10.210.9.178
Trying 10.210.9.178...
Connected to 10.210.9.178.
Escape character is '^]'.
```

Tokyo (ttyp0)

```
login: lab
Password:
```

```
--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC
lab@Tokyo> telnet 192.168.25.1 source 10.14.9.254
Trying 192.168.25.1...
telnet: connect to address 192.168.25.1: Connection refused
telnet: Unable to connect to remote host
```

Question: Does the Telnet session connect successfully? Why or why not?

Answer: No, it does not connect successfully because the traffic was denied by the firewall filter applied to the 100.0 interface.

Step 7.4

From your partner's router, attempt to use Telnet to access your router's directly connected Frame Relay interface. Source the packets from your partner router's first loopback address.

```
lab@Tokyo> telnet 192.168.25.1 source 10.14.9.1
Trying 192.168.25.1...
Connected to 192.168.25.1.
Escape character is '^]'.
```

HongKong (ttyp0)

```
login: lab
Password:
```

```
--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC
lab@HongKong> exit
```

Connection closed by foreign host.

Question: Does the Telnet session connect successfully? Why or why not?

Answer: Yes, it connects successfully because the traffic was permitted by the firewall filter applied to the 100.0 interface.

Step 7.5

Log out of your Telnet session to the partner router.

```
lab@Tokyo> exit
```

Connection closed by foreign host.

```
lab@HongKong>
```



Tell your instructor that you have completed Lab 1.

BGP Configuration (Detailed)

Overview

This lab explores the configuration of BGP in the enterprise environment. It also explores the interaction between BGP and IGP. During this lab, your team will be working with a partner team. You will begin the lab with a configuration that includes basic system settings. You will configure a virtual router on the *Sydney* router and establish internal OSPF connectivity between your router, your partner team's router, and the virtual router on the *Sydney* router. You will establish an IBGP session with your partner team's router and an EBGP session with your ISP. You will configure your AS's border routers to advertise a default router via OSPF when they have working connectivity to an ISP. You will establish primary/secondary inbound and outbound routing policies.

This lab is available in two formats: a high-level format that is designed to make you think through each step and a detailed format that offers step-by-step instructions complete with sample output from most commands.

By completing this lab, you will perform the following tasks:

- Prepare the routers.
- Establish internal connectivity.
- Configure IBGP sessions.
- Establish ISP connectivity.
- Configure intra-AS routing for BGP speakers.
- Configure intra-AS routing for non-BGP speakers.
- Establish a primary/secondary inbound routing policy.
- Establish a primary/secondary outbound routing policy.

This lab requires you to use the *Sydney* router for several purposes. You use it to view the routing table of the ISPs, each of which is implemented as a virtual router on *Sydney*. You also configure a virtual router for your own team to act as an internal router in your AS. Much of the basic configuration of this virtual router (for example, assigning the correct interfaces to your routing instance) is completed for you and is also hidden from you. You must only perform the additional configuration steps listed in these instructions.

This lab requires you to work closely with the partner team in your AS to coordinate changes to your AS's routing. You should work closely with them to simulate a real-life environment and to make the lab successful.

Key Commands

Key operational-mode commands used in this lab include the following:

```
ping
show configuration
show route
show route advertising-protocol
show route hidden
show route table
traceroute routing-instance
```

Part 1: Prepare the Routers

In this part, you will load a configuration file that includes basic system parameters. You will also configure the loopback IP address, router ID, and autonomous system. You will also configure the router to generate an aggregate route.

Step 1.1

Load the router configuration file located on the router at `/var/home/lab/ajre/lab2-reset.conf`.

```
[edit]
lab@HongKong# load override /var/home/lab/ajre/lab2-reset.conf
load complete
```

Step 1.2

View the configuration file.

```
[edit]
lab@HongKong# show
version 8.0R2.8;
system {
    host-name HongKong;
    root-authentication {
        encrypted-password "$1$KI99zGk6$MbYFuBbpLffu9tn2.sI7l1"; ## SECRET-DATA
        ssh-dsa "ssh-dss
AAAAB3NzaC1kc3MAAACBAMQrfP2bZyBXJ6PC7XXZ+MzErI8Jl6jah5L4/
O8Bsfp2hC7EvRfNoX7MqbrtCX/9gUH9gChVuBCB+ERULMdgRvM5uGhC/
gs4UX+4dBbfBgKYYwgmisM8EoT25m7qI8ybpl2YZvHNznvO8h7kr4kpYuQEpkvgsTdH/
Jle4Uqnjv7DAAAFQDZaqA6QAgbW30/
zveaLCIDj6p0dwAAAIBlil+krWrXiD8NPpY+w4dWXEqaV3bnobzPC4eyxQKBUCOr80Q5YBlWXVBHx9
elwBWZwj0SF4hLKHznExnLerVsMuTMA846RbQmSz62vM6kGM13HFonWeQvWia0TDr78+rOEgWF2KHB
SIxL51lmIDW8Gql9hJfD/Dr/
NKP97w3L0wAAAIEar3FkWU8XbYytQYEKxsIN9PlUQ1ERXB3G40YwqFO484SlyKyYCFaz+yNsaAJu2C
8UebDIR3GieyNcOAKf3inCG8jQwjLvZskuZwrvlsz/xtcxSoAh9axJcdUfSJYMW/
g+mD26JKlC1iw5rwp2nH9kUrJxeI7IREdp4egNkM4il5o= configurator@server1.he"; ##
SECRET-DATA
    }
    login {
        user lab {
            uid 2000;
```

```

        class super-user;
        authentication {
            encrypted-password "$1$84J5Maes$cni5Hrazbd/IEHr/50oY30"; ##
SECRET-DATA
        }
    }
}
services {
    ftp;
    ssh;
    telnet;
    web-management {
        http;
    }
}
syslog {
    user * {
        any emergency;
    }
    file messages {
        any any;
        authorization info;
    }
    file interactive-commands {
        interactive-commands any;
    }
}
}
interfaces {
    fe-0/0/0 {
        description "MGMT INTERFACE - DO NOT DELETE";
        unit 0 {
            family inet {
                address 10.210.0.177/28;
            }
        }
    }
}
}

```

Step 1.3

Configure the loopback interface according to the “Lab 2: BGP Routing” diagram. Also, configure this IP address to be the router ID.

```

[edit]
lab@HongKong# edit interfaces lo0

[edit interfaces lo0]
lab@HongKong# set unit 0 family inet address 10.14.243.255/32

[edit interfaces lo0]
lab@HongKong# top edit routing-options

[edit routing-options]
lab@HongKong# set router-id 10.14.243.255

```

Step 1.4

Configure the router's autonomous system (AS) number according to the diagram.

```
[edit routing-options]
lab@HongKong# set autonomous-system 65240
```

Step 1.5

Configure the router to generate an aggregate route, according to the following table:

Aggregate Routes

Router	IP Address
Hong Kong	10.14.240.0/22
Tokyo	10.14.240.0/22
London	10.14.244.0/22
Amsterdam	10.14.244.0/22
Montreal	10.14.248.0/22
San Jose	10.14.248.0/22
Denver	10.14.252.0/22
Sao Paulo	10.14.252.0/22

```
[edit routing-options]
lab@HongKong# set aggregate route 10.14.240.0/22
```

Part 2: Establish Internal Connectivity

In this part, you will configure connectivity within your AS.

Step 2.1

Configure your `fe-2/0/1` interface according to the diagram. You must enable 802.1q tagging and configure the VLAN appropriately.

```
[edit routing-options]
lab@HongKong# top edit interfaces fe-2/0/1

[edit interfaces fe-2/0/1]
lab@HongKong# set vlan-tagging

[edit interfaces fe-2/0/1]
lab@HongKong# set unit 240 vlan-id 240

[edit interfaces fe-2/0/1]
lab@HongKong# set unit 240 family inet address 10.14.243.238/28
```

Step 2.2

Configure OSPF on your router. Configure the loopback and fe-2/0/1 interfaces to be in Area 0. Commit your configuration.

```
[edit interfaces fe-2/0/1]
lab@HongKong# top edit protocols ospf

[edit protocols ospf]
lab@HongKong# set area 0 interface lo0.0
lab@HongKong# set area 0 interface fe-2/0/1.240

[edit protocols ospf]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```



Wait for your partners to complete Step 2.2 before continuing.

Step 2.3

Verify your connectivity to your partner's router by pinging its fe-2/0/1 interface.

```
lab@HongKong> ping 10.14.243.237
PING 10.14.243.237 (10.14.243.237): 56 data bytes
64 bytes from 10.14.243.237: icmp_seq=0 ttl=64 time=37.079 ms
64 bytes from 10.14.243.237: icmp_seq=1 ttl=64 time=10.383 ms
64 bytes from 10.14.243.237: icmp_seq=2 ttl=64 time=10.370 ms
64 bytes from 10.14.243.237: icmp_seq=3 ttl=64 time=10.411 ms
^C
--- 10.14.243.237 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 10.370/17.061/37.079/11.558 ms
```

Step 2.4

Reference the management network diagram provided by your instructor. Use Telnet to access the Sydney router's fe-0/0/0 interface IP address. Together with your partner, configure the Sydney virtual router assigned to your group. Log in to Sydney using the username asNNNNNN, where NNNNN is the AS number assigned to your group.

```
lab@HongKong> telnet 10.210.9.185
Trying 10.210.9.185...
Connected to 10.210.9.185.
Escape character is '^['.
```

Sydney (ttyp2)

```
login: as65240
Password:
```

```
--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC
```

NOTE: This router is divided into many virtual routers used by different teams. Please only configure your own virtual router.

You must use 'configure private' to configure this router.
as65240@Sydney>

Step 2.5

On Sydney, configure the fe-2/0/1 interface, using the VLAN ID as the unit number.

```
as65240@Sydney> configure private
warning: uncommitted changes will be discarded on exit
Entering configuration mode

[edit]
as65240@Sydney# edit interfaces fe-2/0/1

[edit interfaces fe-2/0/1]
as65240@Sydney# set unit 240 vlan-id 240

[edit interfaces fe-2/0/1]
as65240@Sydney# set unit 240 family inet address 10.14.243.236/28
```

Step 2.6

On Sydney, configure the loopback interface, using the same unit number as you did in Step 2.5.

```
[edit interfaces fe-2/0/1]
as65240@Sydney# top edit interfaces lo0

[edit interfaces lo0]
as65240@Sydney# set unit 240 family inet address 10.14.243.253/32
```

Step 2.7

On Sydney, configure OSPF in the routing instance assigned to your group. Configure the loopback and fe-2/0/1 interfaces to be in Area 0. Commit the configuration.

```
[edit interfaces lo0]
as65240@Sydney# top edit routing-instances group-a

[edit routing-instances group-a]
as65240@Sydney# edit protocols ospf

[edit routing-instances group-a protocols ospf]
as65240@Sydney# set area 0 interface fe-2/0/1.240

[edit routing-instances group-a protocols ospf]
as65240@Sydney# set area 0 interface lo0.240

[edit routing-instances group-a protocols ospf]
as65240@Sydney# top

[edit]
as65240@Sydney# commit and-quit
```


Step 2.8

From your router, ping your group's virtual router on Sydney.

```
lab@HongKong> ping 10.14.243.236
PING 10.14.243.236 (10.14.243.236): 56 data bytes
64 bytes from 10.14.243.236: icmp_seq=0 ttl=64 time=15.097 ms
64 bytes from 10.14.243.236: icmp_seq=1 ttl=64 time=20.656 ms
64 bytes from 10.14.243.236: icmp_seq=2 ttl=64 time=20.394 ms
64 bytes from 10.14.243.236: icmp_seq=3 ttl=64 time=20.410 ms
^C
--- 10.14.243.236 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 15.097/19.139/20.656/2.336 ms

lab@HongKong> ping 10.14.243.253
PING 10.14.243.253 (10.14.243.253): 56 data bytes
64 bytes from 10.14.243.253: icmp_seq=0 ttl=64 time=37.987 ms
64 bytes from 10.14.243.253: icmp_seq=1 ttl=64 time=20.420 ms
64 bytes from 10.14.243.253: icmp_seq=2 ttl=64 time=20.385 ms
64 bytes from 10.14.243.253: icmp_seq=3 ttl=64 time=20.404 ms
^C
--- 10.14.243.253 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 20.385/24.799/37.987/7.614 ms
```

Step 2.9

On your router, verify that you have formed OSPF adjacencies with the two other routers in your AS.

```
lab@HongKong> show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
10.14.243.236	fe-2/0/1.240	Full	10.14.243.253	128	38
10.14.243.237	fe-2/0/1.240	Full	10.14.243.254	128	35

Part 3: Configure IBGP Sessions

In this part, you will configure the IBGP session with your neighbor router and establish an export policy for this session.

Step 3.1

Configure a group for IBGP sessions. In this group, configure an IBGP session with your partners' router. Use the loopback addresses as the session endpoints. Commit the configuration and verify that the session is up.

```
[edit]
lab@HongKong# edit protocols bgp

[edit protocols bgp]
lab@HongKong# set group ibgp-peers type internal

[edit protocols bgp]
lab@HongKong# edit group ibgp-peers
```

```
[edit protocols bgp group ibgp-peers]
lab@HongKong# set local-address 10.14.243.255
```

```
[edit protocols bgp group ibgp-peers]
lab@HongKong# set neighbor 10.14.243.254
```

```
[edit protocols bgp group ibgp-peers]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

```
lab@HongKong> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State  Pending
inet.0          8          6          0          0          0          0
Peer           AS        InPkt    OutPkt    OutQ     Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.14.243.254  65240      12       6         0        0      1:46 6/8/0
0/0/0
```

Step 3.2

Determine which routes, if any, the router is sending to its IBGP peer.

```
lab@HongKong> show route advertising-protocol bgp 10.14.243.254
```

```
lab@HongKong>
```

Question: Are you sending any routes over the IBGP session? Why or why not?

Answer: No. The default BGP export policy accepts all BGP routes and denies all other routes. Because we have no other BGP routes, no routes match the default BGP policy.

Step 3.3

Configure a policy called *accept-aggregates* that accepts aggregate routes created by the router.

```
[edit]
lab@HongKong# edit policy-options policy-statement accept-aggregates
```

```
[edit policy-options policy-statement accept-aggregates]
lab@HongKong# set term from-aggr from protocol aggregate
```

```
[edit policy-options policy-statement accept-aggregates]
lab@HongKong# set term from-aggr then accept
```

Step 3.4

Use the *accept-aggregates* policy as the export policy for the group of IBGP sessions.
Commit your configuration.

```
[edit policy-options policy-statement accept-aggregates]
lab@HongKong# top edit protocols bgp group ibgp-peers
```

```
[edit protocols bgp group ibgp-peers]
lab@HongKong# set export accept-aggregates
```

```
[edit protocols bgp group ibgp-peers]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 3.5

Verify that the aggregate route is being advertised to your IBGP peer.

```
lab@HongKong> show route advertising-protocol bgp 10.14.243.254
```

```
inet.0: 16 destinations, 18 routes (16 active, 0 holddown, 0 hidden)
  Prefix                Nexthop        MED      Lclpref   AS path
* 10.14.240.0/22        Self          100      100       I
```

Part 4: Establish ISP Connectivity

In this part, you will configure the interface that connects to the ISP, configure an EBGP session to the ISP, and configure an export policy to control the routes the router will advertise to the ISP over the EBGP session.

Step 4.1

Configure the connection to the ISP according to the “Lab 2: BGP Routing” diagram and the following table. Use Cisco HDLC encapsulation. Commit the configuration.

IP Addresses for ISP Connectivity

Router	IP Address
Hong Kong	172.17.39.2/30
Tokyo	172.17.55.2/30
London	172.17.39.6/30
Amsterdam	172.17.55.6/30
Montreal	172.17.39.10/30
San Jose	172.17.55.10/30
Denver	172.17.39.14/30
Sao Paulo	172.17.55.14/30

```
[edit]
lab@HongKong# edit interfaces se-1/0/0

[edit interfaces se-1/0/0]
lab@HongKong# set encapsulation cisco-hdlc

[edit interfaces se-1/0/0]
lab@HongKong# set unit 0 family inet address 172.17.39.2/30

[edit interfaces se-1/0/0]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 4.2

Ping the provider's address to verify connectivity.

```
lab@HongKong> ping 172.17.39.1
PING 172.17.39.1 (172.17.39.1): 56 data bytes
64 bytes from 172.17.39.1: icmp_seq=0 ttl=64 time=6.404 ms
64 bytes from 172.17.39.1: icmp_seq=1 ttl=64 time=10.430 ms
64 bytes from 172.17.39.1: icmp_seq=2 ttl=64 time=10.360 ms
64 bytes from 172.17.39.1: icmp_seq=3 ttl=64 time=25.070 ms
^C
--- 172.17.39.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.404/13.066/25.070/7.120 ms
```

Step 4.3

Configure a group for BGP sessions to your ISPs. In this group, configure a BGP session to your provider. Commit the configuration.

```
[edit]
lab@HongKong# edit protocols bgp

[edit protocols bgp]
lab@HongKong# set group isp-peers type external

[edit protocols bgp]
lab@HongKong# edit group isp-peers

[edit protocols bgp group isp-peers]
lab@HongKong# set neighbor 172.17.39.1

[edit protocols bgp group isp-peers]
lab@HongKong# set peer-as 65010

[edit protocols bgp group isp-peers]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 4.4

Determine which routes, if any, the router is sending to your ISP.

```
lab@HongKong> show bgp summary
```

```
Groups: 2 Peers: 2 Down peers: 0
```

Table	Tot Paths	Act Paths	Suppressed	History	Damp	State	Pending
inet.0	12	6	0	0	0	0	0

Peer	AS	InPkt	OutPkt	OutQ	Flaps	Last Up/Dwn
172.17.39.1	65010	386	389	0	0	6 4/4/0
10.14.243.254	65240	53	431	0	0	22:08 2/8/0

```
lab@HongKong> show route advertising-protocol bgp 172.17.39.1
```

```
inet.0: 18 destinations, 24 routes (18 active, 0 holddown, 0 hidden)
```

Prefix	Nexthop	MED	Lclpref	AS path
* 172.17.24.0/21	Self			65030 65020 I
* 172.17.48.0/20	Self			65030 I

Question: Is the router sending any routes over the EBGp session? Why or why not?

Answer: The answer will vary. The default BGP export policy accepts all BGP routes and denies all other routes. If your partner router already established the EBGp session to its ISP, you will receive and readvertise the routes from the partner's ISP, which is shown in the previous capture. If the partner router did not yet establish the EBGp session to its ISP, you will not advertise any routes to your ISP. While your router is receiving the aggregate from your partner via BGP, the locally generated aggregate route is the active route for that prefix. Recall that you can only export active routes from the routing table.

Step 4.5

Create a prefix list called *announce-to-ISP* that includes only your aggregate prefix.

```
[edit]
```

```
lab@HongKong# edit policy-options
```

```
[edit policy-options]
```

```
lab@HongKong# set prefix-list announce-to-ISP 10.14.240.0/22
```

Step 4.6

Create a policy called *to-ISP* that accepts the routes in the *announce-to-ISP* prefix list.

```
[edit policy-options]
```

```
lab@HongKong# edit policy-statement to-ISP
```

```
[edit policy-options policy-statement to-ISP]
lab@HongKong# set term aggr-only from prefix-list announce-to-ISP
```

```
[edit policy-options policy-statement to-ISP]
lab@HongKong# set term aggr-only then accept
```

Question: If the router stops generating the aggregate route, will the *to-ISP* policy accept the aggregate that the router receives via IBGP?

Answer: Yes. The policy will accept any route that matches the prefix exactly, regardless of source.

Step 4.7

Create a policy called *reject-all* that rejects all routes.

```
[edit policy-options policy-statement reject-all]
lab@HongKong# set term all then reject
```

Step 4.8

Apply a policy chain to control the routes exported to the ISP group. The routes should be processed by the *to-ISP* and *reject-all* policies. Commit your configuration.

```
[edit protocols bgp group isp-peers]
lab@HongKong# set export [ to-ISP reject-all ]

[edit protocols bgp group isp-peers]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 4.9

Determine which routes, if any, the router is sending to your ISP.

```
lab@HongKong> show route advertising-protocol bgp 172.17.39.1
```

```
inet.0: 18 destinations, 26 routes (18 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref    AS path
* 10.14.240.0/22        Self              0         0          I
```

Question: Is the router sending any routes over the EBGp session? Why or why not?

Answer: Yes, the router is sending the aggregate, which is accepted by the *to-ISP* policy.

Part 5: Configure Intra-AS Routing for BGP Speakers

In this part, you will ensure that your IBGP peer router can use all the routes you advertise to it.

Step 5.1

Determine which routes, if any, the router is sending to its IBGP peer.

```
lab@HongKong> show route advertising-protocol bgp 10.14.243.254
```

```
inet.0: 18 destinations, 26 routes (18 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref    AS path
* 10.14.240.0/22        Self              100
* 10.14.244.0/22        172.17.39.1      100      65010 65244 I
* 10.14.248.0/22        172.17.39.1      100      65010 65248 I
* 10.14.252.0/22        172.17.39.1      100      65010 65252 I
* 172.17.24.0/21        172.17.39.1      100      65010 65020 I
* 172.17.32.0/20        172.17.39.1      100      65010 I
```

Question: Is the router sending any routes over the IBGP session? Record the routes.

Answer: Yes, the router is sending the routes it receives from its EBGp peer.

Step 5.2

Log in to your partner's router. Determine which routes, if any, the router is receiving from its IBGP peer.

```
lab@Tokyo> show route receive-protocol bgp 10.14.243.255
```

```
inet.0: 18 destinations, 24 routes (18 active, 0 holddown, 1 hidden)
  Prefix                Nexthop          MED      Lclpref    AS path
10.14.240.0/22          10.14.243.255    100      I
10.14.244.0/22          172.17.39.1      100      65010 65244 I
10.14.248.0/22          172.17.39.1      100      65010 65248 I
10.14.252.0/22          172.17.39.1      100      65010 65252 I
172.17.24.0/21          172.17.39.1      100      65010 65020 I
```

```
__juniper_private1___.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Question: Does your partner's routing table include all of the routes recorded in Step 5.1?

Answer: No, at least one of the routes should be hidden. If your group partners activated their router's BGP session to their ISP, only one of your routes should be hidden. If your group partners did not yet activate the router's BGP session to the ISP, many of your routes should be hidden.

Step 5.3

On your partner's router, determine if any of the routes your router sent it via BGP are hidden routes. If so, determine why they are hidden.

```
lab@Tokyo> show route receive-protocol bgp 10.14.243.255 hidden
```

```
inet.0: 18 destinations, 24 routes (18 active, 0 holddown, 1 hidden)
  Prefix                Nexthop              MED      Lc1pref      AS path
  172.17.32.0/20         172.17.39.1         100      65010 I

__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0
hidden)
```

```
lab@Tokyo> show route 172.17.32.0 hidden extensive
```

```
inet.0: 18 destinations, 24 routes (18 active, 0 holddown, 1 hidden)
172.17.32.0/20 (2 entries, 1 announced)
TSI:
KRT in-kernel 172.17.32.0/20 -> {172.17.55.1}
Standby generator for 0.0.0.0/0
Page 0 idx 1 Type 1 val 8716ea0
  BGP      Preference: 170/-101
           Next hop type: Unusable
           Next-hop reference count: 1
           State: <Hidden Int Ext>
           Inactive reason: Unusable path
           Local AS: 65240 Peer AS: 65240
           Age: 19:35
           Task: BGP_65240.10.14.243.255+1968
           AS path: 65010 I Aggregator: 65010 172.17.32.1
           Localpref: 100
           Router ID: 10.14.243.255
           Indirect next hops: 1
             Protocol next hop: 172.17.39.1
             Indirect next hop: 0 -
```


Question: Are there hidden routes? If so, why are they hidden?

Answer: Yes, there are hidden routes. They are hidden because the next-hop attribute of the BGP path advertisement contains a next hop to which they do not have a route. In the case of the provider's aggregate, if the router installs this route, the route is recursive because the best path to the next-hop IP address is the route itself.

Step 5.4

On your router, create a policy called *nhs* to change the next-hop attribute of all BGP announcements to the IP address your router is using as its endpoint of any BGP session.

```
[edit]
lab@HongKong# edit policy-options policy-statement nhs

[edit policy-options policy-statement nhs]
lab@HongKong# set term all-routes then next-hop self

[edit policy-options policy-statement nhs]
lab@HongKong# top edit protocols bgp group ibgp-peers
```

Step 5.5

Add the policy to the beginning of the policy chain applied to the IBGP group. Commit the configuration.

```
[edit protocols bgp group ibgp-peers]
lab@HongKong# insert export nhs before accept-aggregates

[edit protocols bgp group ibgp-peers]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 5.6

Determine which routes, if any, the router is sending to its IBGP peer.

```
lab@HongKong> show route advertising-protocol bgp 10.14.243.254
```

```
inet.0: 18 destinations, 25 routes (18 active, 0 holddown, 0 hidden)
  Prefix                Nexthop        MED      Lclpref    AS path
* 10.14.240.0/22        Self          100       100         I
* 10.14.244.0/22        Self          100       100         65010 65244 I
* 10.14.248.0/22        Self          100       100         65010 65248 I
* 10.14.252.0/22        Self          100       100         65010 65252 I
* 172.17.24.0/21         Self          100       100         65010 65020 I
* 172.17.32.0/20         Self          100       100         65010 I
```

Question: Is the router sending the same routes you recorded in Step 5.1? Record the routes.

Answer: Yes, the router should be sending the same routes it was sending in Step 5.1. (However, if your partners activated their session to their provider between Steps 5.1 and 5.6 and began sending you advertisements for paths through their ISP, it is possible that your router might have begun to prefer those routes. In that case, your router would send fewer routes over the IBGP session.)

Step 5.7

Log in to your partner's router. Determine which routes, if any, the router is receiving from its IBGP peer.

lab@Tokyo> **show route receive-protocol bgp 10.14.243.255**

```
inet.0: 18 destinations, 24 routes (18 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref    AS path
  10.14.240.0/22         10.14.243.255           100         I
  10.14.244.0/22         10.14.243.255           100        65010 65244 I
  10.14.248.0/22         10.14.243.255           100        65010 65248 I
  10.14.252.0/22         10.14.243.255           100        65010 65252 I
  172.17.24.0/21         10.14.243.255           100        65010 65020 I
  * 172.17.32.0/20       10.14.243.255           100        65010 I
```

```
__juniper_private1___.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Question: Does your partner's routing table include all of the routes recorded in Step 5.6?

Answer: Yes, it should include all these routes.

Step 5.8

On your partner's router, determine if any of the routes your router sent it via BGP are hidden routes. If so, determine why they are hidden.

```
lab@Tokyo> show route receive-protocol bgp 10.14.243.255 hidden
```

```
inet.0: 18 destinations, 24 routes (18 active, 0 holddown, 0 hidden)
```

```
__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Question: Are there hidden routes? If so, why are they hidden?

Answer: No, none of the routes your router sent it via BGP should be hidden.

Part 6: Configure Intra-AS Routing for Non-BGP Speakers

In this part, you will configure your routers to advertise a default route via OSPF when they have a working connection to their locally connected ISP.

Step 6.1

Create a policy called *match-ISP-routes* that accepts BGP routes that are received from your directly connected ISP.

```
[edit]
lab@HongKong# edit policy-options policy-statement match-ISP-routes

[edit policy-options policy-statement match-ISP-routes]
lab@HongKong# set term from-isp from protocol bgp

[edit policy-options policy-statement match-ISP-routes]
lab@HongKong# set term from-isp from neighbor 172.17.39.1

[edit policy-options policy-statement match-ISP-routes]
lab@HongKong# set term from-isp then accept

[edit policy-options policy-statement match-ISP-routes]
lab@HongKong# set term reject-all then reject
```

Step 6.2

Configure the router to generate a default route if (and only if) routes exist that match the *match-ISP-routes* policy. The router should silently discard packets that match this route. Commit the configuration

```
[edit policy-options policy-statement match-ISP-routes]
lab@HongKong# top edit routing-options

[edit routing-options]
lab@HongKong# set generate route 0/0 policy match-ISP-routes
```

```
[edit routing-options]
lab@HongKong# set generate route 0/0 discard
```

```
[edit routing-options]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Question: What is the default policy for generated routes? How does that affect the configuration in Step 6.2?

Answer: The default policy is to accept all routes. To complete Step 6.2, you must either use a policy or a policy chain that ends with a policy term that rejects all routes.

Step 6.3

Confirm that the router is generating a default route.

```
lab@HongKong> show route 0/0 exact
```

```
inet.0: 18 destinations, 19 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Aggregate/130] 03:20:44
                   Discard
                   [OSPF/150] 01:37:57, metric 1, tag 0
                   > to 10.14.243.237 via fe-2/0/1.240
                   [BGP/170] 01:04:31, localpref 100, from 10.14.243.254
                   AS path: I
                   > to 10.14.243.237 via fe-2/0/1.240
```

Question: What protocol is listed as the route's source?

Answer: The generated route is an aggregate route.

Step 6.4

Configure a policy called *default-to-OSPF* that matches the generated default route and sets it to be a Type 1 external route with a metric of 0.

```
[edit]
lab@HongKong# edit policy-options policy-statement default-to-OSPF

[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# set term default-only from protocol aggregate
```

```
[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# set term default-only from route-filter 0/0 exact
```

```
[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# set term default-only then external type 1
```

```
[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# set term default-only then metric 0
```

```
[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# set term default-only then accept
```

Question: What is significant about setting it to be a Type 1 external route?

Answer: When calculating the cost of a Type 1 external route, other routers include the cost of the path to the router advertising the route. When calculating the cost of a Type 2 external route, other routers only include the cost of the Type 2 external route. Therefore, because this is a Type 1 external route, other routers choose to use the OSPF default route that is topologically closest to them.

Step 6.5

Configure the router to use the *default-to-OSPF* policy to control exporting routes to OSPF. Commit the configuration.

```
[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# top edit protocols ospf
```

```
[edit protocols ospf]
lab@HongKong# set export default-to-OSPF
```

```
[edit protocols ospf]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Question: What is the default policy for OSPF?

Answer: The default policy for OSPF is to deny all routes. However, the router does propagate LSAs to keep a consistent link-state database.

Step 6.6

Log in to the Sydney router. Examine the routing table for your group's virtual router.

```
as65240@Sydney> show route 0/0 exact table group-a.inet.0
```

```
group-a.inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          *[OSPF/150] 00:02:43, metric 1, tag 0
                   to 10.14.243.237 via fe-2/0/1.240
```

Question: Is it receiving and using the default route your router is sending to it via OSPF?

Answer: Yes, it should be receiving and using the default route.

Step 6.7

On your router, deactivate the BGP session to your ISP. Because this is the only session in the ISP's group, you must deactivate the entire group. Commit the configuration.

```
[edit]
lab@HongKong# edit protocols bgp

[edit protocols bgp]
lab@HongKong# deactivate group isp-peers

[edit protocols bgp]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 6.8

Determine if your router is generating a default route.

```
lab@HongKong> show route 0/0 exact

inet.0: 18 destinations, 21 routes (18 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 01:54:25, metric 1, tag 0
                   > to 10.14.243.237 via fe-2/0/1.240
                   [BGP/170] 01:20:59, localpref 100, from 10.14.243.254
                     AS path: I
                     > to 10.14.243.237 via fe-2/0/1.240
```

Question: Is your router generating a default route? Why or why not?

Answer: No, it is not generating a default route. Because the session to the ISP is down, no routes in the routing table match the *match-ISP-routes* policy that the router uses to determine if it should create a default route.

Step 6.9

Log in to the Sydney router. Examine the routing table for your group's virtual router.

```
as65240@Sydney> show route 0/0 exact table group-a.inet.0
```

```
group-a.inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          *[OSPF/150] 00:00:08, metric 1, tag 0
                   > to 10.14.243.237 via fe-2/0/1.240
```

Question: Is your virtual router receiving and using a default route from your router?

Answer: No, it should not be receiving a default route from your router. It might, however, still be receiving a default route from your partner's router if your partner has not yet completed the previous step. This case is shown in the capture.

Step 6.10

Activate the BGP session to your ISP that was deactivated in Step 6.7 and commit your changes.

```
[edit protocols bgp]
lab@HongKong# activate group isp-peers
```

```
[edit protocols bgp]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```



Wait for your partners before proceeding. Coordinate the remaining parts of the lab with your partners.

Part 7: Establish Primary/Secondary Inbound Routing Policy

In this part, you will establish a primary/secondary inbound routing policy. You want all inbound traffic to use the connection through ISP A when it is available. You never want inbound traffic to use the connection through ISP C, unless the connection through ISP A is unavailable.

Your ISPs have similar routing policies. They both assign customer routes a default local preference value of 100 and assign routes received from peers or upstream ISPs a local preference value of 80. You can use the community values from the following tables to modify the local-preference value of routes you send to your ISP.

Many of the steps in this section only require modifying one router. Work with your partners to make the necessary modifications for your AS.

ISP A Communities

Community	Local Preference
65010:70	70
65010:80	80
65010:90	90
65010:100	100
65010:110	110

ISP C Communities

Community	Local Preference
65030:70	70
65030:80	80
65030:90	90
65030:100	100
65030:110	110

Step 7.1

Begin by configuring a policy called *prepend-three-times* on the router connected to ISP C. In the policy, configure the router to prepend your AS three times on all announcements.

```
[edit]
lab@Tokyo# edit policy-options policy-statement prepend-three-times

[edit policy-options policy-statement prepend-three-times]
lab@Tokyo# set term prepend-3 then as-path-prepend "65240 65240 65240"
```


Step 7.2

On the router connected to ISP C, apply the *prepend-three-times* policy to the beginning of the export policy chain in the BGP configuration for the ISP group. Commit the configuration.

```
[edit policy-options policy-statement prepend-three-times]
lab@Tokyo# top edit protocols bgp group isp-peers

[edit protocols bgp group isp-peers]
lab@Tokyo# insert export prepend-three-times before to-ISP

[edit protocols bgp group isp-peers]
lab@Tokyo# commit and-quit
commit complete
Exiting configuration mode
```

Step 7.3

Log in to the Sydney router. You will use it as a route server in this part. Examine the routing table for the *isp-c* virtual router.

```
as65240@Sydney> show route table isp-c.inet.0 10.14.240.0/22 exact

isp-c.inet.0: 18 destinations, 19 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.14.240.0/22      *[BGP/170] 00:00:29, localpref 100
                   AS path: 65240 65240 65240 65240 I
                   > to 172.17.55.2 via se-3/0/1.0
                   [BGP/170] 00:00:29, localpref 80
                   AS path: 65020 65010 65240 I
                   > to 172.17.25.5 via lt-0/0/0.12
```

Question: Does the AS path of your announcement contain three extra copies of your AS?

Answer: Yes, it should contain three extra copies.

Question: Does ISP C router prefer the primary path (via ISP A) or the secondary path (directly from your router)? Why?

Answer: It prefers the secondary path because this announcement has a higher local-preference value than the primary path.

Step 7.4

On the router connected to ISP C, configure a community called *isp-c-localpref-70* with value 65030:70.

```
[edit]
lab@Tokyo# edit policy-options

[edit policy-options]
lab@Tokyo# set community isp-c-localpref-70 members 65030:70
```

Step 7.5

On the router connected to ISP C, configure a policy called *set-isp-c-localpref* that sets the community you defined in the previous step for all routes.

```
[edit policy-options]
lab@Tokyo# edit policy-statement set-isp-c-localpref

[edit policy-options policy-statement set-isp-c-localpref]
lab@Tokyo# set term add-comm-70 then community add isp-c-localpref-70
```

Step 7.6

On the router connected to ISP C, remove the *prepend-three-times* policy from the export policy chain in the BGP configuration for the ISP group. Add the *set-isp-c-localpref* policy to the beginning of the export policy chain in the BGP configuration for the ISP group. Commit the configuration.

```
[edit policy-options policy-statement set-isp-c-localpref]
lab@Tokyo# top edit protocols bgp group isp-peers

[edit protocols bgp group isp-peers]
lab@Tokyo# delete export prepend-three-times

[edit protocols bgp group isp-peers]
lab@Tokyo# insert export set-isp-c-localpref before to-ISP

[edit protocols bgp group isp-peers]
lab@Tokyo# commit
commit complete
```

Step 7.7

Log in to the Sydney router. Examine the routing table for the *isp-c* virtual router.

```
as65240@Sydney> show route table isp-c.inet.0 10.14.240.0/22 exact

isp-c.inet.0: 18 destinations, 19 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.14.240.0/22      *[BGP/170] 00:10:43, localpref 80
                   AS path: 65020 65010 65240 I
                   > to 172.17.25.5 via lt-0/0/0.12
                   [BGP/170] 00:00:26, localpref 70
                   AS path: 65240 I
                   > to 172.17.55.2 via se-3/0/1.0
```

Question: What local preference is assigned to your announcement?

Answer: It is assigned a local-preference value of 70.

Question: Does ISP C router prefer the primary path (through ISP A) or the secondary path (directly from your router)? Why?

Answer: It prefers the primary path because this announcement has a higher local-preference value than the secondary path.

Step 7.8

Test failover by deactivating your AS's BGP session to ISP A. Commit the configuration.

```
[edit protocols bgp]
lab@HongKong# deactivate group isp-peers
```

```
[edit protocols bgp]
lab@HongKong# commit
commit complete
```

Step 7.9

Log in to the Sydney router. Examine the routing table for the *isp-a*, *isp-b*, and *isp-c* virtual routers.

```
as65240@Sydney> show route table isp-a.inet.0 10.14.240.0/22 exact
```

```
isp-a.inet.0: 18 destinations, 21 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:02:27, localpref 80
                    AS path: 65020 65030 65240 I
                    > to 172.17.25.1 via lt-0/0/0.10
```

```
as65240@Sydney> show route table isp-b.inet.0 10.14.240.0/22 exact
```

```
isp-b.inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:01:56, localpref 100
                    AS path: 65030 65240 I
                    > to 172.17.25.6 via lt-0/0/0.13
```

```
as65240@Sydney> show route table isp-c.inet.0 10.14.240.0/22 exact
```

```
isp-c.inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
```

+ = Active Route, - = Last Active, * = Both

```
10.14.240.0/22      *[BGP/170] 00:03:03, localpref 70
                   AS path: 65240 I
                   > to 172.17.55.2 via se-3/0/1.0
```

Question: Are all the ISPs preferring the secondary path? Why or why not?

Answer: Yes. That is the only available path.

Step 7.10

Test the failure recovery by activating your AS's BGP session to ISP A. Commit the configuration.

```
[edit protocols bgp]
lab@HongKong# activate group isp-peers
```

```
[edit protocols bgp]
lab@HongKong# commit
commit complete
```

Step 7.11

Log in to the Sydney router. Examine the routing table for the *isp-a*, *isp-b*, and *isp-c* virtual routers.

```
as65240@Sydney> show route table isp-a.inet.0 10.14.240.0/22 exact
```

```
isp-a.inet.0: 18 destinations, 22 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:04:28, localpref 100
                   AS path: 65240 I
                   > to 172.17.39.2 via se-3/0/0.0
                   [BGP/170] 00:08:06, localpref 80
                   AS path: 65020 65030 65240 I
                   > to 172.17.25.1 via lt-0/0/0.10
```

```
as65240@Sydney> show route table isp-b.inet.0 10.14.240.0/22 exact
```

```
isp-b.inet.0: 12 destinations, 16 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:05:35, localpref 100
                   AS path: 65030 65240 I
                   > to 172.17.25.6 via lt-0/0/0.13
                   [BGP/170] 00:01:57, localpref 100
                   AS path: 65010 65240 I
                   > to 172.17.25.2 via lt-0/0/0.11
```

```
as65240@Sydney> show route table isp-c.inet.0 10.14.240.0/22 exact
```

```
isp-c.inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:06:46, localpref 70
                   AS path: 65240 I
                   > to 172.17.55.2 via se-3/0/1.0
```

Question: Are all the ISPs preferring the primary path?
Why or why not?

Answer: No. ISP B and ISP C prefer the secondary path.
ISP B prefers the secondary path because that is the older route. ISP C prefers the secondary path because that is its only path to that prefix.

Step 7.12

On the router connected to ISP C, add the *prepend-three-times* policy to the export policy chain in the BGP configuration for the ISP group. Commit the configuration.

```
[edit protocols bgp group isp-peers]
lab@Tokyo# insert export prepend-three-times before to-ISP

[edit protocols bgp group isp-peers]
lab@Tokyo# commit
commit complete
```

Step 7.13

Log in to the Sydney router. Examine the routing table for the *isp-c* virtual router.

```
as65240@Sydney> show route table isp-c.inet.0 10.14.240.0/22 exact
```

```
isp-c.inet.0: 18 destinations, 19 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:01:13, localpref 80
                   AS path: 65020 65010 65240 I
                   > to 172.17.25.5 via lt-0/0/0.12
                   [BGP/170] 00:01:13, localpref 70
                   AS path: 65240 65240 65240 65240 I
                   > to 172.17.55.2 via se-3/0/1.0
```

Question: Does the AS path of your announcement contain three extra copies of your AS?

Answer: Yes, it should contain three extra copies.

Question: What local preference is assigned to your announcement?

Answer: It is assigned a local-preference value of 70.

Question: Does ISP C router prefer the primary path (through ISP A) or the secondary path (directly from your router)? Why?

Answer: It prefers the primary path because this announcement has a higher local-preference value than the secondary path.

Step 7.14

Test failover by deactivating your AS's BGP session to ISP A. Commit the configuration.

```
[edit protocols bgp group isp-peers]
lab@Tokyo# up

[edit protocols bgp]
lab@Tokyo# deactivate group isp-peers

[edit protocols bgp]
lab@Tokyo# commit
commit complete
```

Step 7.15

Log in to the Sydney router. Examine the routing table for the *isp-a*, *isp-b*, and *isp-c* virtual routers.

```
as65240@Sydney> show route table isp-a.inet.0 10.14.240.0/22 exact

isp-a.inet.0: 18 destinations, 21 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.14.240.0/22      *[BGP/170] 00:17:08, localpref 100
                   AS path: 65240 I
                   > to 172.17.39.2 via se-3/0/0.0

as65240@Sydney> show route table isp-b.inet.0 10.14.240.0/22 exact
```

```
isp-b.inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:17:37, localpref 100
                    AS path: 65010 65240 I
                    > to 172.17.25.2 via lt-0/0/0.11
```

```
as65240@Sydney> show route table isp-c.inet.0 10.14.240.0/22 exact
```

```
isp-c.inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:03:51, localpref 80
                    AS path: 65020 65010 65240 I
                    > to 172.17.25.5 via lt-0/0/0.12
```

Question: Are all the ISPs preferring the secondary path? Why or why not?

Answer: Yes. That is the only available path.

Step 7.16

Test the failure recovery by activating your AS's BGP session to ISP A. Commit the configuration.

```
[edit protocols bgp]
lab@Tokyo# activate group isp-peers
```

```
[edit protocols bgp]
lab@Tokyo# commit
commit complete
```

Step 7.17

Log in to the Sydney router. Examine the routing table for the *isp-a*, *isp-b*, and *isp-c* virtual routers.

```
as65240@Sydney> show route table isp-a.inet.0 10.14.240.0/22 exact
```

```
isp-a.inet.0: 18 destinations, 21 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:19:35, localpref 100
                    AS path: 65240 I
                    > to 172.17.39.2 via se-3/0/0.0
```

```
as65240@Sydney> show route table isp-b.inet.0 10.14.240.0/22 exact
```

```
isp-b.inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:20:38, localpref 100
                    AS path: 65010 65240 I
```

```
> to 172.17.25.2 via lt-0/0/0.11
```

```
as65240@Sydney> show route table isp-c.inet.0 10.14.240.0/22 exact
```

```
isp-c.inet.0: 18 destinations, 19 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.240.0/22      *[BGP/170] 00:07:10, localpref 80
                    AS path: 65020 65010 65240 I
                    > to 172.17.25.5 via lt-0/0/0.12
                    [BGP/170] 00:02:22, localpref 70
                    AS path: 65240 65240 65240 65240 I
                    > to 172.17.55.2 via se-3/0/1.0
```

Question: Are all the ISPs preferring the primary path?
Why or why not?

Answer: Yes. During the failure recovery, ISP B preferred the primary path because it has a shorter AS path length. Now, ISP B prefers the primary path because it is only receiving an announcement for the primary path. ISP C prefers the primary path because this announcement has a higher local-preference value than the secondary path.

Part 8: Establish Primary/Secondary Outbound Routing Policy

In this part, you will establish a primary/secondary outbound routing policy. You want all outbound traffic to use the connection through ISP A when it is available. You do not want outbound traffic to use the connection through ISP C, unless the connection through ISP A is unavailable. You also want traffic within your AS to follow the shortest path to ISP A. Thus, routers within the AS should always choose the OSPF default route advertised by the router directly connected to ISP A when ISP A is available.

Many of the steps in this part will only require modifying one router. Work with your partners to make the necessary modifications for your AS.

Step 8.1

Modify the *default-to-OSPF* policy so that the router advertises the default route as an external Type 2 announcement with a metric of 1 on the router attached to ISP A and a metric of 100 on the router attached to ISP C. Commit the configuration.

```
[edit]
lab@HongKong# edit policy-options policy-statement default-to-OSPF

[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# set term default-only then external type 2
```



```
[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# set term default-only then metric 1
```

```
[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# commit
commit complete
```

Step 8.2

Log in to the Sydney router. Examine the OSPF database for your group's virtual router.

```
as65240@Sydney> show ospf database extensive instance group-a
```

```
OSPF link state database, Area 0.0.0.0
```

Type	ID	Adv Rtr	Seq	Age	Opt	Cksum	Len
Router	*10.14.243.253	10.14.243.253	0x80000009	2808	0x22	0x208	48
bits 0x0, link count 2							
id 10.14.243.237, data 10.14.243.236, Type Transit (2)							
TOS count 0, TOS 0 metric 1							
id 10.14.243.253, data 255.255.255.255, Type Stub (3)							
TOS count 0, TOS 0 metric 0							
Gen timer 00:03:11							
Aging timer 00:13:11							
Installed 00:46:48 ago, expires in 00:13:12, sent 00:46:46 ago							
Last changed 02:37:00 ago, Change count: 3, Ours							
Router	10.14.243.254	10.14.243.254	0x80000019	321	0x22	0xfbf7	48
bits 0x2, link count 2							
id 10.14.243.237, data 10.14.243.237, Type Transit (2)							
TOS count 0, TOS 0 metric 1							
id 10.14.243.254, data 255.255.255.255, Type Stub (3)							
TOS count 0, TOS 0 metric 0							
Aging timer 00:54:39							
Installed 00:05:18 ago, expires in 00:54:39							
Last changed 02:37:10 ago, Change count: 1							
Router	10.14.243.255	10.14.243.255	0x80000017	564	0x22	0x14dd	48
bits 0x2, link count 2							
id 10.14.243.237, data 10.14.243.238, Type Transit (2)							
TOS count 0, TOS 0 metric 1							
id 10.14.243.255, data 255.255.255.255, Type Stub (3)							
TOS count 0, TOS 0 metric 0							
Aging timer 00:50:36							
Installed 00:09:21 ago, expires in 00:50:36							
Last changed 01:07:13 ago, Change count: 2							
Network	10.14.243.237	10.14.243.254	0x8000000d	784	0x22	0x1cf6	36
mask 255.255.255.240							
attached router 10.14.243.254							
attached router 10.14.243.253							
attached router 10.14.243.255							
Aging timer 00:46:55							
Installed 00:13:01 ago, expires in 00:46:56							
Last changed 02:37:04 ago, Change count: 2							
OSPF AS SCOPE link state database							
Type	ID	Adv Rtr	Seq	Age	Opt	Cksum	Len
Extern	0.0.0.0	10.14.243.254	0x80000002	321	0x22	0x7ac7	36

```

mask 0.0.0.0
Type 2, TOS 0x0, metric 100, fwd addr 0.0.0.0, tag 0.0.0.0
Aging timer 00:54:39
Installed 00:05:20 ago, expires in 00:54:39
Last changed 00:05:20 ago, Change count: 2
Extern 0.0.0.0 10.14.243.255 0x80000002 564 0x22 0x9212 36
mask 0.0.0.0
Type 2, TOS 0x0, metric 1, fwd addr 0.0.0.0, tag 0.0.0.0
Aging timer 00:50:36
Installed 00:09:23 ago, expires in 00:50:36
Last changed 00:09:23 ago, Change count: 2

```

```
as65240@Sydney> show route 0/0 exact table group-a.inet.0
```

```

group-a.inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

```

```

0.0.0.0/0          *[OSPF/150] 00:06:53, metric 1, tag 0
                   > to 10.14.243.238 via fe-2/0/1.240

```

Question: Is the virtual router receiving the default routes with the correct metrics?

Answer: Yes, it should be receiving two external Type 2 announcements for the default route, one from each router. It should use the default route it is receiving from the router attached to ISP A.

Step 8.3

On the router attached to ISP A, create a policy called *localpref150* that matches all routes and sets a local-preference value of 150.

```

[edit policy-options policy-statement default-to-OSPF]
lab@HongKong# up

```

```

[edit policy-options]
lab@HongKong# edit policy-statement localpref150

```

```

[edit policy-options policy-statement localpref150]
lab@HongKong# set term all then local-preference 150

```

Step 8.4

Configure the router attached to ISP A to use the *localpref150* policy as the import policy for the ISP group. Commit the configuration.

```

[edit policy-options policy-statement localpref150]
lab@HongKong# top edit protocols bgp group isp-peers

```

```

[edit protocols bgp group isp-peers]
lab@HongKong# set import localpref150

```

```
[edit protocols bgp group isp-peers]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 8.5

Examine the routing tables on both routers.

```
lab@HongKong> show route 172.17.48.0/20
```

```
inet.0: 18 destinations, 19 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
172.17.48.0/20      *[BGP/170] 00:39:12, localpref 150
                   AS path: 65010 65020 65030 I
                   > to 172.17.39.1 via se-1/0/0.0
```

```
lab@Tokyo> show route 172.17.48.0/20
```

```
inet.0: 18 destinations, 27 routes (18 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
172.17.48.0/20      *[BGP/170] 00:02:26, localpref 150, from 10.14.243.255
                   AS path: 65010 65020 65030 I
                   > to 10.14.243.238 via fe-2/0/1.240
                   [BGP/170] 00:20:50, localpref 100
                   AS path: 65030 I
                   > to 172.17.55.1 via se-1/0/1.0
172.17.55.0/30      *[Direct/0] 03:24:15
                   > via se-1/0/1.0
172.17.55.2/32      *[Local/0] 03:24:19
                   Local via se-1/0/1.0
```

Question: What path do the routers prefer to ISP C's aggregate (172.17.48.0/20)? Why?

Answer: Both routers should prefer the path through ISP A because this path has the highest local-preference value.

Step 8.6

Log in to the Sydney router. From your group's virtual router, trace the route to 172.17.48.1, which is an IP address on ISP C's network.

```
as65240@Sydney> traceroute 172.17.48.1 routing-instance group-a
traceroute to 172.17.48.1 (172.17.48.1), 30 hops max, 40 byte packets
 1  10.14.243.238 (10.14.243.238)  7.267 ms  11.451 ms  9.930 ms
 2  172.17.39.1 (172.17.39.1)  18.888 ms  19.987 ms  9.913 ms
 3  172.17.25.1 (172.17.25.1)  19.700 ms  9.363 ms  39.809 ms
 4  172.17.48.1 (172.17.48.1)  9.883 ms  39.272 ms  9.944 ms
```

Question: Does this traceroute follow the primary path or the secondary path?

Answer: The traceroute should follow the primary path.

Step 8.7

Test failover by deactivating your AS's BGP session to ISP A. Commit the configuration.

```
[edit]
lab@HongKong# edit protocols bgp

[edit protocols bgp]
lab@HongKong# deactivate group isp-peers

[edit protocols bgp]
lab@HongKong# commit
commit complete
```

Step 8.8

Log in to the Sydney router. From your virtual router, trace the route to 172.17.48.1, which is an IP address on ISP C's network.

```
as65240@Sydney> traceroute 172.17.48.1 routing-instance group-a
traceroute to 172.17.48.1 (172.17.48.1), 30 hops max, 40 byte packets
 1  10.14.243.237 (10.14.243.237)  7.310 ms  8.828 ms  9.807 ms
 2  172.17.48.1 (172.17.48.1)  9.900 ms  18.248 ms  20.963 ms
```

Question: Does this traceroute follow the primary path or the secondary path?

Answer: The traceroute should follow the secondary path.

Step 8.9

Test the failure recovery by activating your AS's BGP session to ISP A. Commit the configuration.

```
[edit protocols bgp]
lab@HongKong# activate group isp-peers

[edit protocols bgp]
lab@HongKong# commit
commit complete
```

Step 8.10

Log in to the Sydney router. From your group's virtual router, trace the route to 172.17.48.1, which is an IP address on ISP C's network.

```
as65240@Sydney> traceroute 172.17.48.1 routing-instance group-a
traceroute to 172.17.48.1 (172.17.48.1), 30 hops max, 40 byte packets
 1  10.14.243.238 (10.14.243.238)  7.085 ms  8.643 ms  10.101 ms
 2  172.17.39.1 (172.17.39.1)  4.915 ms  13.972 ms  19.941 ms
 3  172.17.25.1 (172.17.25.1)  20.235 ms  2.531 ms  17.029 ms
 4  172.17.48.1 (172.17.48.1)  19.205 ms  11.680 ms  29.264 ms
```

Question: Does this traceroute follow the primary path or the secondary path?

Answer: The traceroute should follow the primary path.



Tell your instructor that you have completed Lab 2.

Not For Reproduction

Lab 3

IGP Conversion (Detailed)

Overview

This lab demonstrates a successful IGP transition from RIP to OSPF using the overlay method. It is available in two formats: a high-level format that is designed to make you think through each step and a detailed format that offers step-by-step instructions complete with sample output from most commands.

By completing this lab, you will perform the following tasks:

- Prepare the routers.
- Configure RIP.
- Prepare for the overlay transition.
- Deactivate RIP.
- Perform additional OSPF configuration.

Note

This lab is designed to simulate an actual IGP transition in which changes are coordinated throughout a network. Be aware that you must coordinate your changes closely with the other teams to ensure the success of the lab.

Key Commands

Key operational-mode commands used in this lab include the following:

```
ping
show route
```

Part 1: Prepare the Routers

In this lab part, you will prepare the routers for the lab.

Step 1.1

Load the router configuration file located on the router at `/var/home/lab/ajre/lab3-reset.conf`.

```
lab@SanJose> configure
Entering configuration mode

[edit]
lab@SanJose# load override ajre/lab3-reset.conf
load complete
```

Step 1.2

View the configuration file.

```
[edit]
lab@SanJose# show
version 8.0R2.8;
system {
    host-name SanJose;
    root-authentication {
        encrypted-password "$1$KI99zGk6$MbYFuBbpLffu9tn2.sI7l1"; ## SECRET-DATA
        ssh-dsa "ssh-dss
AAAAB3NzaC1kc3MAAACBAMQrfP2bZyBXJ6PC7XXZ+MzErI8Jl6jah5L4/
O8Bsfp2hC7EvRfNoX7MqbrtCX/9gUH9gChVuBCB+ERULMdgrVm5uGhC/
gs4UX+4dBbfBgKYYwgmisM8EoT25m7qI8ybpl2YZvHNznvO8h7kr4kpYuQEpkvgstDh/
Jle4Uqnjv7DAAAFQDZaqA6QAgbW30/
zveaLCIDj6p0dwAAAIBliL+krWrXiD8NPpY+w4dWxEqaV3bnobzPC4eyxQKBUCOr80Q5YBlWXVBHx9
elwBWZwj0SF4hLKHznExnLerVsMutMA846RbQmSz62vM6kGM13HFonWeQvWia0TDrr78+rOEgWF2KHB
SIxL51lmIDW8Gql9hJfD/Dr/
NKP97w3L0wAAAIEAr3FkWU8XbYytQYEKxsIN9PlUQ1ERXB3G40YwqFO484SlyKyYCFaz+yNsaAJu2C
8UebDIR3GieyNcoAKf3inCG8jQwjLvZskuZwrVlsz/xtcxSoAh9axJcdUfSJYMW/
g+mD26JK1C1iw5rwp2nH9kUrJxeI7IREdp4egNkM4il5o= configurator@server1.he"; ##
SECRET-DATA
    }
    login {
        user lab {
            uid 2000;
            class super-user;
            authentication {
                encrypted-password "$1$84J5Maes$cni5Hrazbd/IEHr/50oY30"; ##
SECRET-DATA
            }
        }
    }
}
```



```

    }
  }
}
services {
  ftp;
  ssh;
  telnet;
  web-management {
    http;
  }
}
syslog {
  user * {
    any emergency;
  }
  file messages {
    any any;
    authorization info;
  }
  file interactive-commands {
    interactive-commands any;
  }
}
}
interfaces {
  fe-0/0/0 {
    description "MGMT INTERFACE - DO NOT DELETE";
    unit 0 {
      family inet {
        address 10.210.0.182/28;
      }
    }
  }
  fe-0/0/1 {
    unit 0 {
      family inet {
        address 10.14.8.5/30;
      }
    }
  }
  se-1/0/0 {
    serial-options {
      clocking-mode internal;
    }
    unit 0 {
      family inet {
        address 10.14.8.13/30;
      }
    }
  }
  fe-2/0/0 {
    unit 0 {
      family inet {
        address 10.14.8.1/30;
      }
    }
  }
}

```

```

    }
  }
}
fe-2/0/1 {
  vlan-tagging;
  unit 16 {
    vlan-id 16;
    family inet {
      address 172.17.38.20/29;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.25.6/32;
    }
  }
}
}
routing-options {
  static {
    route 0.0.0.0/0 next-hop 172.17.38.17;
  }
}
}

```

Step 1.3

Commit your changes.

```

[edit]
lab@SanJose# commit

```

Part 2: Configure RIP

In this part, you will configure RIP to distribute complete internal routing information.

Step 2.1

Configure a RIP group called *peer-routers*. Using the “Lab 3a: IGP Conversion” diagram, configure your router to accept RIP updates on interfaces that attach to other routers within your internal network, and configure those interfaces to be part of this group.

```

[edit]
lab@SanJose# edit protocols rip group peer-routers

```

```

[edit protocols rip group peer-routers]
lab@SanJose# set neighbor se-1/0/0.0

```

```

[edit protocols rip group peer-routers]
lab@SanJose# set neighbor fe-2/0/0.0

```

```

[edit protocols rip group peer-routers]
lab@SanJose# set neighbor fe-0/0/1.0

```

Step 2.2

Configure a policy called *rip-export-policy*, which you will use to control the routes that are exported via RIP. This policy should export all RIP routes and all directly connected subnets except the subnet assigned to the *fe-0/0/0.0* interface. On the SanJose and Montreal routers, it should also export the default static route to the ISP.

```
[edit protocols rip group peer-routers]
lab@SanJose# top edit policy-options policy-statement rip-export-policy

[edit policy-options policy-statement rip-export-policy]
lab@SanJose# set term deny-fe-0-0-0 from interface fe-0/0/0.0

[edit policy-options policy-statement rip-export-policy]
lab@SanJose# set term deny-fe-0-0-0 then reject

[edit policy-options policy-statement rip-export-policy]
lab@SanJose# set term rip-direct-out from protocol [ direct rip ]

[edit policy-options policy-statement rip-export-policy]
lab@SanJose# set term rip-direct-out then accept

[edit policy-options policy-statement rip-export-policy]
lab@SanJose# set term def-stat-out from route-filter 0.0.0.0/0 exact

[edit policy-options policy-statement rip-export-policy]
lab@SanJose# set term def-stat-out from protocol static

[edit policy-options policy-statement rip-export-policy]
lab@SanJose# set term def-stat-out then accept

[edit policy-options policy-statement rip-export-policy]
lab@SanJose# show
term deny-fe-0-0-0 {
    from interface fe-0/0/0.0;
    then reject;
}
term rip-direct-out {
    from protocol [ direct rip ];
    then accept;
}
term def-stat-out {
    from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
    }
    then accept;
}
```

Question: What is the default export policy for RIP? How does that change the way you write this policy?

Answer: The default export policy for RIP rejects all routes. Therefore, your export policy must explicitly allow any routes (including other RIP routes) that you want the router to send to RIP neighbors.

Step 2.3

Configure the router to use the *rip-export-policy* as the export policy for the RIP group *peer-routers*.

```
[edit policy-options policy-statement rip-export-policy]
lab@SanJose# top edit protocols rip group peer-routers
```

```
[edit protocols rip group peer-routers]
lab@SanJose# set export rip-export-policy
```

Step 2.4

Commit your changes and verify that you are sending information via RIP.

```
[edit protocols rip group peer-routers]
lab@SanJose# commit and-quit
commit complete
Exiting configuration mode
```

```
lab@SanJose> show route advertising-protocol rip 10.14.8.1
```

```
inet.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 02:39:13
                   > to 172.17.38.17 via fe-2/0/1.16
10.14.8.4/30       *[Direct/0] 02:39:13
                   > via fe-0/0/1.0
10.14.8.12/30      *[Direct/0] 02:39:08
                   > via se-1/0/0.0
172.17.38.16/29    *[Direct/0] 02:39:13
                   > via fe-2/0/1.16
192.168.25.6/32    *[Direct/0] 02:39:13
                   > via lo0.0
```

Step 2.5

Monitor the RIP routes you are receiving. Once all teams complete their RIP configuration, you should have complete internal routing information.

```
lab@SanJose> show route
```

```
inet.0: 32 destinations, 33 routes (32 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

0.0.0.0/0      *[Static/5] 00:16:47
                > to 172.17.38.17 via fe-2/0/1.16
                [RIP/100] 00:02:05, metric 2, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
10.14.8.0/30   *[Direct/0] 00:16:47
                > via fe-2/0/0.0
10.14.8.1/32   *[Local/0] 00:16:47
                Local via fe-2/0/0.0
10.14.8.4/30   *[Direct/0] 00:16:47
                > via fe-0/0/1.0
10.14.8.5/32   *[Local/0] 00:16:47
                Local via fe-0/0/1.0
10.14.8.8/30   *[RIP/100] 00:06:32, metric 2, tag 0
                > to 10.14.8.6 via fe-0/0/1.0
                to 10.14.8.2 via fe-2/0/0.0
10.14.8.12/30  *[Direct/0] 00:06:33
                > via se-1/0/0.0
10.14.8.13/32  *[Local/0] 00:16:47
                Local via se-1/0/0.0
10.14.8.16/30  *[RIP/100] 00:06:30, metric 2, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
10.14.8.20/30  *[RIP/100] 00:06:25, metric 2, tag 0
                > to 10.14.8.6 via fe-0/0/1.0
10.14.8.24/30  *[RIP/100] 00:06:33, metric 2, tag 0
                > to 10.14.8.14 via se-1/0/0.0
10.14.8.28/30  *[RIP/100] 00:06:30, metric 3, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
10.14.9.0/27   *[RIP/100] 00:06:26, metric 3, tag 0
                > to 10.14.8.14 via se-1/0/0.0
10.14.9.64/27  *[RIP/100] 00:06:33, metric 2, tag 0
                > to 10.14.8.14 via se-1/0/0.0
10.14.10.0/26  *[RIP/100] 00:06:25, metric 3, tag 0
                > to 10.14.8.6 via fe-0/0/1.0
10.14.10.64/27 *[RIP/100] 00:06:32, metric 2, tag 0
                > to 10.14.8.6 via fe-0/0/1.0
10.14.11.0/27  *[RIP/100] 00:06:30, metric 4, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
10.14.11.64/27 *[RIP/100] 00:06:30, metric 3, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
10.210.0.176/28 *[Direct/0] 00:22:25
                > via fe-0/0/0.0
10.210.0.182/32 *[Local/0] 00:22:27
                Local via fe-0/0/0.0
172.17.37.16/29 *[RIP/100] 00:02:05, metric 2, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
172.17.38.16/29 *[Direct/0] 00:16:47
                > via fe-2/0/1.16
172.17.38.20/32 *[Local/0] 00:16:47
                Local via fe-2/0/1.16
192.168.25.1/32 *[RIP/100] 00:06:33, metric 2, tag 0
                > to 10.14.8.14 via se-1/0/0.0
192.168.25.2/32 *[RIP/100] 00:06:26, metric 3, tag 0
                > to 10.14.8.14 via se-1/0/0.0

```

```

192.168.25.3/32      *[RIP/100] 00:06:30, metric 4, tag 0
                    > to 10.14.8.2 via fe-2/0/0.0
192.168.25.4/32      *[RIP/100] 00:06:30, metric 3, tag 0
                    > to 10.14.8.2 via fe-2/0/0.0
192.168.25.5/32      *[RIP/100] 00:06:30, metric 2, tag 0
                    > to 10.14.8.2 via fe-2/0/0.0
192.168.25.6/32      *[Direct/0] 00:16:47
                    > via lo0.0
192.168.25.7/32      *[RIP/100] 00:06:32, metric 2, tag 0
                    > to 10.14.8.6 via fe-0/0/1.0
192.168.25.8/32      *[RIP/100] 00:06:25, metric 3, tag 0
                    > to 10.14.8.6 via fe-0/0/1.0
224.0.0.9/32         *[RIP/100] 00:11:49, metric 1
                    MultiRecv

```

__juniper_private1___.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

10.0.0.1/32          *[Direct/0] 00:23:02
                    > via lo0.16385
10.0.0.16/32         *[Direct/0] 00:23:02
                    > via lo0.16385

```



Wait until all teams have completed this part before continuing.

Part 3: Prepare for the Overlay Transition

We will now begin the overlay transition by assigning RIP a more preferred route preference value than OSPF and by configuring OSPF.

Step 3.1

Configure RIP to use a better route preference than OSPF. In this case, configure RIP to use a route preference of 7. Commit the configuration.

```

lab@SanJose> configure
Entering configuration mode

[edit]
lab@SanJose# set protocols rip group peer-routers preference 7

[edit]
lab@SanJose# commit
commit complete

```

Step 3.2

Confirm that RIP routes are now assigned the new route preference.

```
[edit]
```

```
lab@SanJose# run show route protocol rip
```

```
inet.0: 32 destinations, 33 routes (32 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          [RIP/7] 00:50:45, metric 2, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.8.8/30       *[RIP/7] 00:55:12, metric 2, tag 0
                   > to 10.14.8.6 via fe-0/0/1.0
                   to 10.14.8.2 via fe-2/0/0.0
10.14.8.16/30      *[RIP/7] 00:55:10, metric 2, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.8.20/30      *[RIP/7] 00:55:05, metric 2, tag 0
                   > to 10.14.8.6 via fe-0/0/1.0
10.14.8.24/30      *[RIP/7] 00:08:16, metric 4, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.8.28/30      *[RIP/7] 00:55:10, metric 3, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.9.64/27      *[RIP/7] 00:55:13, metric 2, tag 0
                   > to 10.14.8.14 via se-1/0/0.0
10.14.10.0/26      *[RIP/7] 00:55:05, metric 3, tag 0
                   > to 10.14.8.6 via fe-0/0/1.0
10.14.10.64/27     *[RIP/7] 00:55:12, metric 2, tag 0
                   > to 10.14.8.6 via fe-0/0/1.0
10.14.11.0/27      *[RIP/7] 00:55:10, metric 4, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.11.64/27     *[RIP/7] 00:55:10, metric 3, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.11.128/27    *[RIP/7] 00:00:10, metric 5, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
172.17.37.16/29    *[RIP/7] 00:50:45, metric 2, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.1/32    *[RIP/7] 00:55:13, metric 2, tag 0
                   > to 10.14.8.14 via se-1/0/0.0
192.168.25.2/32    *[RIP/7] 00:00:10, metric 5, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.3/32    *[RIP/7] 00:55:10, metric 4, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.4/32    *[RIP/7] 00:55:10, metric 3, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.5/32    *[RIP/7] 00:55:10, metric 2, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.7/32    *[RIP/7] 00:55:12, metric 2, tag 0
                   > to 10.14.8.6 via fe-0/0/1.0
192.168.25.8/32    *[RIP/7] 00:55:05, metric 3, tag 0
                   > to 10.14.8.6 via fe-0/0/1.0
224.0.0.9/32      *[RIP/100] 00:46:40, metric 1
                   MultiRecv
```

```
__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Step 3.3

Configure OSPF according to the “Lab 3b: IGP Conversion” diagram. Configure OSPF to run on all the interfaces shown on the diagram except the interfaces to the ISP. Configure OSPF to run on the loopback interface. Configure OSPF to run in passive mode on all interfaces that do not connect to other routers on the diagram. Configure all areas as normal areas, but do not configure any summarization.

Note

Do not place the `fe-0/0/0.0` interface in OSPF.

```
[edit]
lab@SanJose# edit protocols ospf

[edit protocols ospf]
lab@SanJose# set area 0 interface fe-2/0/0.0

[edit protocols ospf]
lab@SanJose# set area 0 interface fe-0/0/1.0

[edit protocols ospf]
lab@SanJose# set area 1 interface se-1/0/0.0

[edit protocols ospf]
lab@SanJose# set area 0 interface lo0.0

[edit protocols ospf]
lab@SanJose# show
area 0.0.0.0 {
    interface fe-2/0/0.0;
    interface fe-0/0/1.0;
    interface lo0.0;
}
area 0.0.0.1 {
    interface se-1/0/0.0;
}
```

Question: What does passive mode cause the router to do?

Answer: The router does not send OSPF hellos or form OSPF adjacencies on passive interfaces, but it does advertise the interface's directly connected subnets in its router LSAs.

Step 3.4 (SanJose and Montreal Routers Only)

Configure an export policy to export the static default route into OSPF. The router should advertise it as a Type 1 external route with a metric of 0.


```

[edit protocols ospf]
lab@SanJose# top edit policy-options policy-statement default-to-ospf

[edit policy-options policy-statement default-to-ospf]
lab@SanJose# set term send-default from route-filter 0.0.0.0/0 exact

[edit policy-options policy-statement default-to-ospf]
lab@SanJose# set term send-default from protocol static

[edit policy-options policy-statement default-to-ospf]
lab@SanJose# set term send-default then accept

[edit policy-options policy-statement default-to-ospf]
lab@SanJose# set term send-default then external type 1

[edit policy-options policy-statement default-to-ospf]
lab@SanJose# set term send-default then metric 0

[edit policy-options policy-statement default-to-ospf]
lab@SanJose# show
term send-default {
    from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
    }
    then {
        metric 0;
        external {
            type 1;
        }
        accept;
    }
}

[edit policy-options policy-statement default-to-ospf]
lab@SanJose# top

[edit]
lab@SanJose# set protocols ospf export default-to-ospf

```

Step 3.5

Commit the configuration. Monitor your routing table to ensure that you are receiving the same routes via OSPF and RIP.

```

[edit]
lab@SanJose# commit and-quit
commit complete
Exiting configuration mode

```

```
lab@SanJose> show route
```

```

inet.0: 33 destinations, 54 routes (33 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

```

```

0.0.0.0/0      *[Static/5] 01:13:45
                > to 172.17.38.17 via fe-2/0/1.16
                [RIP/7] 00:59:03, metric 2, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
                [OSPF/150] 00:00:47, metric 1, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
10.14.8.0/30   *[Direct/0] 01:13:45
                > via fe-2/0/0.0
10.14.8.1/32   *[Local/0] 01:13:45
                Local via fe-2/0/0.0
10.14.8.4/30   *[Direct/0] 01:13:45
                > via fe-0/0/1.0
10.14.8.5/32   *[Local/0] 01:13:45
                Local via fe-0/0/1.0
10.14.8.8/30   *[RIP/7] 01:03:30, metric 2, tag 0
                > to 10.14.8.6 via fe-0/0/1.0
                to 10.14.8.2 via fe-2/0/0.0
                [OSPF/10] 00:00:47, metric 2
                > to 10.14.8.6 via fe-0/0/1.0
                to 10.14.8.2 via fe-2/0/0.0
10.14.8.12/30  *[Direct/0] 01:03:31
                > via se-1/0/0.0
                [OSPF/10] 00:00:57, metric 6
                > via se-1/0/0.0
10.14.8.13/32  *[Local/0] 01:13:45
                Local via se-1/0/0.0
10.14.8.16/30  *[RIP/7] 01:03:28, metric 2, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
                [OSPF/10] 00:00:47, metric 7
                > to 10.14.8.2 via fe-2/0/0.0
10.14.8.20/30  *[RIP/7] 01:03:23, metric 2, tag 0
                > to 10.14.8.6 via fe-0/0/1.0
                [OSPF/10] 00:00:47, metric 7
                > to 10.14.8.6 via fe-0/0/1.0
10.14.8.24/30  *[RIP/7] 00:16:34, metric 4, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
                [OSPF/10] 00:00:47, metric 14
                > to 10.14.8.2 via fe-2/0/0.0
10.14.8.28/30  *[RIP/7] 01:03:28, metric 3, tag 0
                > to 10.14.8.2 via fe-2/0/0.0
                [OSPF/10] 00:00:47, metric 8
                > to 10.14.8.2 via fe-2/0/0.0
10.14.9.64/27  *[RIP/7] 01:03:31, metric 2, tag 0
                > to 10.14.8.14 via se-1/0/0.0
                [OSPF/10] 00:00:57, metric 7
                > via se-1/0/0.0
10.14.10.0/26  *[RIP/7] 01:03:23, metric 3, tag 0
                > to 10.14.8.6 via fe-0/0/1.0
                [OSPF/10] 00:00:47, metric 8
                > to 10.14.8.6 via fe-0/0/1.0
10.14.10.64/27 *[RIP/7] 01:03:30, metric 2, tag 0
                > to 10.14.8.6 via fe-0/0/1.0
                [OSPF/10] 00:00:47, metric 2
                > to 10.14.8.6 via fe-0/0/1.0

```

```

10.14.11.0/27      *[RIP/7] 01:03:28, metric 4, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
                   [OSPF/10] 00:00:47, metric 9
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.11.64/27     *[RIP/7] 01:03:28, metric 3, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
                   [OSPF/10] 00:00:47, metric 8
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.11.128/27    *[RIP/7] 00:08:28, metric 5, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
                   [OSPF/10] 00:00:47, metric 15
                   > to 10.14.8.2 via fe-2/0/0.0
10.210.0.176/28    *[Direct/0] 01:19:23
                   > via fe-0/0/0.0
10.210.0.182/32    *[Local/0] 01:19:25
                   Local via fe-0/0/0.0
172.17.37.16/29    *[RIP/7] 00:59:03, metric 2, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
172.17.38.16/29    *[Direct/0] 01:13:45
                   > via fe-2/0/1.16
172.17.38.20/32    *[Local/0] 01:13:45
                   Local via fe-2/0/1.16
192.168.25.1/32    *[RIP/7] 01:03:31, metric 2, tag 0
                   > to 10.14.8.14 via se-1/0/0.0
                   [OSPF/10] 00:00:57, metric 6
                   > via se-1/0/0.0
192.168.25.2/32    *[RIP/7] 00:08:28, metric 5, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
                   [OSPF/10] 00:00:47, metric 14
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.3/32    *[RIP/7] 01:03:28, metric 4, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
                   [OSPF/10] 00:00:47, metric 8
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.4/32    *[RIP/7] 01:03:28, metric 3, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
                   [OSPF/10] 00:00:47, metric 7
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.5/32    *[RIP/7] 01:03:28, metric 2, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
                   [OSPF/10] 00:00:47, metric 1
                   > to 10.14.8.2 via fe-2/0/0.0
192.168.25.6/32    *[Direct/0] 01:13:45
                   > via lo0.0
192.168.25.7/32    *[RIP/7] 01:03:30, metric 2, tag 0
                   > to 10.14.8.6 via fe-0/0/1.0
                   [OSPF/10] 00:00:47, metric 1
                   > to 10.14.8.6 via fe-0/0/1.0
192.168.25.8/32    *[RIP/7] 01:03:23, metric 3, tag 0
                   > to 10.14.8.6 via fe-0/0/1.0
                   [OSPF/10] 00:00:47, metric 7
                   > to 10.14.8.6 via fe-0/0/1.0
224.0.0.5/32      *[OSPF/10] 00:01:02, metric 1
                   MultiRecv

```

```
224.0.0.9/32      *[RIP/100] 00:01:02, metric 1
                  MultiRecv
```

```
__juniper_private1__inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0
hidden)
```

+ = Active Route, - = Last Active, * = Both

```
10.0.0.1/32      *[Direct/0] 01:20:00
                  > via lo0.16385
10.0.0.16/32     *[Direct/0] 01:20:00
                  > via lo0.16385
```

Question: Which routes is your router using?

Answer: The router should prefer the RIP routes because they are configured with a better route preference.

Question: Are any routes known via RIP, but not via OSPF? How do you explain these discrepancies?

Answer: The ISP routes (172.17.37.16/29 and 172.17.38.16/29) might be known via RIP and not via OSPF. The RIP export policy on `SanJose` and `Montreal` causes these routers to advertise these routes via RIP; however, OSPF is not configured to run on these interfaces. In this case, this is not a problem because you do not necessarily need routes for these subnets internally; rather, all routers must know about the default route.



Wait until all teams have completed this part before continuing.

Part 4: Deactivate RIP

Now that you have activated the parallel OSPF infrastructure, you will deactivate the RIP configuration. You should conduct this transition in a controlled manner, proceeding from the edges of the network toward the center. You should be able to maintain a continuous ping to a service provider address (172.17.24.1) if you want.

Step 4.1

Deactivate RIP on your router. Commit the configuration in such a way that the router will automatically revert to the previous configuration in 15 minutes if you lose contact with it.

```
lab@SanJose> configure
Entering configuration mode

[edit]
lab@SanJose# deactivate protocols rip

[edit]
lab@SanJose# commit confirmed 15
commit confirmed will be automatically rolled back in 15 minutes unless
confirmed
commit complete

# commit confirmed will be rolled back in 15 minutes
```

Step 4.2

Monitor the routing table to ensure you maintain full routing information.

```
[edit]
lab@SanJose# run show route

inet.0: 31 destinations, 33 routes (31 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 01:28:30
                   > to 172.17.38.17 via fe-2/0/1.16
                   [OSPF/150] 00:15:32, metric 1, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.8.0/30      *[Direct/0] 01:28:30
                   > via fe-2/0/0.0
10.14.8.1/32      *[Local/0] 01:28:30
                   Local via fe-2/0/0.0
10.14.8.4/30      *[Direct/0] 01:28:30
                   > via fe-0/0/1.0
10.14.8.5/32      *[Local/0] 01:28:30
                   Local via fe-0/0/1.0
10.14.8.8/30      *[OSPF/10] 00:15:32, metric 2
                   > to 10.14.8.6 via fe-0/0/1.0
                   to 10.14.8.2 via fe-2/0/0.0
10.14.8.12/30     *[Direct/0] 01:18:16
                   > via se-1/0/0.0
                   [OSPF/10] 00:15:42, metric 6
                   > via se-1/0/0.0
10.14.8.13/32     *[Local/0] 01:28:30
                   Local via se-1/0/0.0
10.14.8.16/30     *[OSPF/10] 00:15:32, metric 7
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.8.20/30     *[OSPF/10] 00:15:32, metric 7
                   > to 10.14.8.6 via fe-0/0/1.0
10.14.8.24/30     *[OSPF/10] 00:15:32, metric 14
                   > to 10.14.8.2 via fe-2/0/0.0
```

```

10.14.8.28/30      *[OSPF/10] 00:15:32, metric 8
                  > to 10.14.8.2 via fe-2/0/0.0
10.14.9.64/27     *[OSPF/10] 00:15:42, metric 7
                  > via se-1/0/0.0
10.14.10.0/26     *[OSPF/10] 00:15:32, metric 8
                  > to 10.14.8.6 via fe-0/0/1.0
10.14.10.64/27    *[OSPF/10] 00:15:32, metric 2
                  > to 10.14.8.6 via fe-0/0/1.0
10.14.11.0/27     *[OSPF/10] 00:15:32, metric 9
                  > to 10.14.8.2 via fe-2/0/0.0
10.14.11.64/27    *[OSPF/10] 00:15:32, metric 8
                  > to 10.14.8.2 via fe-2/0/0.0
10.14.11.128/27   *[OSPF/10] 00:15:32, metric 15
                  > to 10.14.8.2 via fe-2/0/0.0
10.210.0.176/28   *[Direct/0] 01:34:08
                  > via fe-0/0/0.0
10.210.0.182/32   *[Local/0] 01:34:10
                  Local via fe-0/0/0.0
172.17.38.16/29   *[Direct/0] 00:09:06
                  > via fe-2/0/1.16
172.17.38.20/32   *[Local/0] 00:09:06
                  Local via fe-2/0/1.16
192.168.25.1/32   *[OSPF/10] 00:15:42, metric 6
                  > via se-1/0/0.0
192.168.25.2/32   *[OSPF/10] 00:15:32, metric 14
                  > to 10.14.8.2 via fe-2/0/0.0
192.168.25.3/32   *[OSPF/10] 00:15:32, metric 8
                  > to 10.14.8.2 via fe-2/0/0.0
192.168.25.4/32   *[OSPF/10] 00:15:32, metric 7
                  > to 10.14.8.2 via fe-2/0/0.0
192.168.25.5/32   *[OSPF/10] 00:15:32, metric 1
                  > to 10.14.8.2 via fe-2/0/0.0
192.168.25.6/32   *[Direct/0] 01:28:30
                  > via lo0.0
192.168.25.7/32   *[OSPF/10] 00:15:32, metric 1
                  > to 10.14.8.6 via fe-0/0/1.0
192.168.25.8/32   *[OSPF/10] 00:15:32, metric 7
                  > to 10.14.8.6 via fe-0/0/1.0
224.0.0.5/32      *[OSPF/10] 00:15:47, metric 1
                  MultiRecv

__juniper_privatel__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.1/32       *[Direct/0] 01:34:44
                  > via lo0.16385
10.0.0.16/32      *[Direct/0] 01:34:44
                  > via lo0.16385

# commit confirmed will be rolled back in 13 minutes

```

Step 4.3

Once RIP is completely removed from the network and you have verified connectivity, confirm your changes to prevent the router from automatically reverting to the previous configuration.

```
[edit]
lab@SanJose# commit
commit complete
```

Step 4.4

Now, delete the RIP configuration and commit your changes.

```
[edit]
lab@SanJose# delete protocols rip
```

```
[edit]
lab@SanJose# commit
commit complete
```

Part 5: Perform Additional OSPF Configuration

In this part, we will configure area summarization and different area types. You should again coordinate all these changes with all other teams.

Step 5.1

Configure the area summarization shown on the “Lab 3b: IGP Conversion” diagram. Commit the configuration.

```
[edit]
lab@SanJose# set protocols ospf area 1 area-range 10.14.9.0/24
```

```
[edit]
lab@SanJose# commit
commit complete
```

Step 5.2

Examine the routing table on a router in Area 0 and a router without an interface in Area 0.

```
[edit]
lab@SanJose# run show route

inet.0: 29 destinations, 31 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 01:45:52
                   > to 172.17.38.17 via fe-2/0/1.16
                   [OSPF/150] 00:32:54, metric 1, tag 0
                   > to 10.14.8.2 via fe-2/0/0.0
10.14.8.0/30       *[Direct/0] 01:45:52
                   > via fe-2/0/0.0
10.14.8.1/32       *[Local/0] 01:45:52
                   Local via fe-2/0/0.0
10.14.8.4/30       *[Direct/0] 01:45:52
                   > via fe-0/0/1.0
```

```

10.14.8.5/32      *[Local/0] 01:45:52
                  Local via fe-0/0/1.0
10.14.8.8/30      *[OSPF/10] 00:32:54, metric 2
                  > to 10.14.8.6 via fe-0/0/1.0
                  to 10.14.8.2 via fe-2/0/0.0
10.14.8.12/30     *[Direct/0] 01:35:38
                  > via se-1/0/0.0
                  [OSPF/10] 00:33:04, metric 6
                  > via se-1/0/0.0
10.14.8.13/32     *[Local/0] 01:45:52
                  Local via se-1/0/0.0
10.14.8.16/30     *[OSPF/10] 00:32:54, metric 7
                  > to 10.14.8.2 via fe-2/0/0.0
10.14.8.20/30     *[OSPF/10] 00:32:54, metric 7
                  > to 10.14.8.6 via fe-0/0/1.0
10.14.8.24/30     *[OSPF/10] 00:32:54, metric 14
                  > to 10.14.8.2 via fe-2/0/0.0
10.14.8.28/30     *[OSPF/10] 00:32:54, metric 8
                  > to 10.14.8.2 via fe-2/0/0.0
10.14.9.0/24      *[OSPF/10] 00:04:06, metric 16777215
                  Discard
10.14.9.64/27     *[OSPF/10] 00:33:04, metric 7
                  > via se-1/0/0.0
10.14.10.0/24     *[OSPF/10] 00:03:49, metric 8
                  > to 10.14.8.6 via fe-0/0/1.0
10.14.11.0/24     *[OSPF/10] 00:03:55, metric 15
                  > to 10.14.8.2 via fe-2/0/0.0
10.210.0.176/28   *[Direct/0] 01:51:30
                  > via fe-0/0/0.0
10.210.0.182/32   *[Local/0] 01:51:32
                  Local via fe-0/0/0.0
172.17.38.16/29   *[Direct/0] 00:26:28
                  > via fe-2/0/1.16
172.17.38.20/32   *[Local/0] 00:26:28
                  Local via fe-2/0/1.16
192.168.25.1/32   *[OSPF/10] 00:33:04, metric 6
                  > via se-1/0/0.0
192.168.25.2/32   *[OSPF/10] 00:32:54, metric 14
                  > to 10.14.8.2 via fe-2/0/0.0
192.168.25.3/32   *[OSPF/10] 00:32:54, metric 8
                  > to 10.14.8.2 via fe-2/0/0.0
192.168.25.4/32   *[OSPF/10] 00:32:54, metric 7
                  > to 10.14.8.2 via fe-2/0/0.0
192.168.25.5/32   *[OSPF/10] 00:32:54, metric 1
                  > to 10.14.8.2 via fe-2/0/0.0
192.168.25.6/32   *[Direct/0] 01:45:52
                  > via lo0.0
192.168.25.7/32   *[OSPF/10] 00:32:54, metric 1
                  > to 10.14.8.6 via fe-0/0/1.0
192.168.25.8/32   *[OSPF/10] 00:32:54, metric 7
                  > to 10.14.8.6 via fe-0/0/1.0
224.0.0.5/32      *[OSPF/10] 00:33:09, metric 1
                  MultiRecv

```



```
__juniper_privatel__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0
hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
10.0.0.1/32      *[Direct/0] 01:52:06
                  > via lo0.16385
10.0.0.16/32     *[Direct/0] 01:52:06
                  > via lo0.16385
```

```
lab@HongKong> show route
```

```
inet.0: 25 destinations, 26 routes (25 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0        *[OSPF/150] 00:33:49, metric 6, tag 0
                  > via se-1/0/1.0
10.14.8.0/30     *[OSPF/10] 00:33:49, metric 7
                  > via se-1/0/1.0
10.14.8.4/30     *[OSPF/10] 00:33:49, metric 7
                  > via se-1/0/1.0
10.14.8.8/30     *[OSPF/10] 00:33:39, metric 8
                  > via se-1/0/1.0
10.14.8.12/30    *[Direct/0] 01:36:25
                  > via se-1/0/1.0
                  [OSPF/10] 01:36:23, metric 6
                  > via se-1/0/1.0
10.14.8.14/32    *[Local/0] 01:36:28
                  Local via se-1/0/1.0
10.14.8.16/30    *[OSPF/10] 00:33:39, metric 13
                  > via se-1/0/1.0
10.14.8.20/30    *[OSPF/10] 00:33:39, metric 13
                  > via se-1/0/1.0
10.14.8.24/30    *[OSPF/10] 00:33:39, metric 20
                  > via se-1/0/1.0
10.14.8.28/30    *[OSPF/10] 00:33:39, metric 14
                  > via se-1/0/1.0
10.14.9.64/27    *[Direct/0] 01:36:28
                  > via fe-2/0/1.101
10.14.9.65/32    *[Local/0] 01:36:28
                  Local via fe-2/0/1.101
10.14.10.0/24    *[OSPF/10] 00:04:36, metric 14
                  > via se-1/0/1.0
10.14.11.0/24    *[OSPF/10] 00:04:42, metric 21
                  > via se-1/0/1.0
10.210.0.176/28  *[Direct/0] 01:52:38
                  > via fe-0/0/0.0
10.210.0.177/32  *[Local/0] 01:52:40
                  Local via fe-0/0/0.0
192.168.25.1/32  *[Direct/0] 01:36:28
                  > via lo0.0
192.168.25.2/32  *[OSPF/10] 00:33:39, metric 20
                  > via se-1/0/1.0
192.168.25.3/32  *[OSPF/10] 00:33:39, metric 14
```

```

> via se-1/0/1.0
192.168.25.4/32    *[OSPF/10] 00:33:39, metric 13
> via se-1/0/1.0
192.168.25.5/32    *[OSPF/10] 00:33:39, metric 7
> via se-1/0/1.0
192.168.25.6/32    *[OSPF/10] 00:33:49, metric 6
> via se-1/0/1.0
192.168.25.7/32    *[OSPF/10] 00:33:39, metric 7
> via se-1/0/1.0
192.168.25.8/32    *[OSPF/10] 00:33:39, metric 13
> via se-1/0/1.0
224.0.0.5/32      *[OSPF/10] 01:36:28, metric 1
MultiRecv

```

__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

10.0.0.1/32      *[Direct/0] 01:53:12
> via lo0.16385
10.0.0.16/32     *[Direct/0] 01:53:12
> via lo0.16385

```

Question: What routes are now available on the router in Area 0?

Answer: The router has all routes for the areas in which it has interfaces, routes for the loopbacks from all areas (which are not within the summary range of the area), and summary routes for the other areas. It does not have more-specific routes within the summary routes of other areas.

Question: What routes are now available on the router not in Area 0?

Answer: The router has all routes for its own area, routes for the loopbacks from all areas, and summary routes for the other areas. It does not have more-specific routes within the summary routes of other areas.

Step 5.3

Configure all areas other than Area 0 to be stub areas. Commit your changes.

```
[edit]
lab@SanJose# set protocols ospf area 1 stub
```

```
[edit]
lab@SanJose# commit
commit complete
```



Wait until all teams have completed Step 5.3 before continuing.

Step 5.4

Examine the routing table on a router without an interface in Area 0.

```
lab@HongKong> show route
```

```
inet.0: 24 destinations, 25 routes (24 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.14.8.0/30      *[OSPF/10] 00:04:52, metric 7
                  > via se-1/0/1.0
10.14.8.4/30      *[OSPF/10] 00:04:52, metric 7
                  > via se-1/0/1.0
10.14.8.8/30      *[OSPF/10] 00:04:52, metric 8
                  > via se-1/0/1.0
10.14.8.12/30     *[Direct/0] 01:47:09
                  > via se-1/0/1.0
                  [OSPF/10] 01:47:07, metric 6
                  > via se-1/0/1.0
10.14.8.14/32     *[Local/0] 01:47:12
                  Local via se-1/0/1.0
10.14.8.16/30     *[OSPF/10] 00:04:52, metric 13
                  > via se-1/0/1.0
10.14.8.20/30     *[OSPF/10] 00:04:52, metric 13
                  > via se-1/0/1.0
10.14.8.24/30     *[OSPF/10] 00:01:23, metric 20
                  > via se-1/0/1.0
10.14.8.28/30     *[OSPF/10] 00:02:35, metric 14
                  > via se-1/0/1.0
10.14.9.64/27     *[Direct/0] 01:47:12
                  > via fe-2/0/1.101
10.14.9.65/32     *[Local/0] 01:47:12
                  Local via fe-2/0/1.101
10.14.10.0/24     *[OSPF/10] 00:04:52, metric 14
                  > via se-1/0/1.0
10.14.11.0/24     *[OSPF/10] 00:01:23, metric 21
                  > via se-1/0/1.0
10.210.0.176/28   *[Direct/0] 02:03:22
                  > via fe-0/0/0.0
10.210.0.177/32   *[Local/0] 02:03:24
                  Local via fe-0/0/0.0
192.168.25.1/32   *[Direct/0] 01:47:12
                  > via lo0.0
```

```

192.168.25.2/32      *[OSPF/10] 00:01:23, metric 20
                    > via se-1/0/1.0
192.168.25.3/32      *[OSPF/10] 00:01:23, metric 14
                    > via se-1/0/1.0
192.168.25.4/32      *[OSPF/10] 00:02:35, metric 13
                    > via se-1/0/1.0
192.168.25.5/32      *[OSPF/10] 00:04:52, metric 7
                    > via se-1/0/1.0
192.168.25.6/32      *[OSPF/10] 00:04:52, metric 6
                    > via se-1/0/1.0
192.168.25.7/32      *[OSPF/10] 00:04:52, metric 7
                    > via se-1/0/1.0
192.168.25.8/32      *[OSPF/10] 00:04:52, metric 13
                    > via se-1/0/1.0
224.0.0.5/32         *[OSPF/10] 01:47:12, metric 1
                    MultiRecv

```

__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

10.0.0.1/32          *[Direct/0] 02:03:56
                    > via lo0.16385
10.0.0.16/32         *[Direct/0] 02:03:56
                    > via lo0.16385

```

Question: What routes are available? If these routes changed, why?

Answer: The same routes observed in Step 5.2 should still be available except for the default route. External LSAs are not allowed in stub areas. Because the default route is advertised as an external LSA, it is not sent into stub areas.

Step 5.5

On the routers attached to Area 0, configure the router to advertise a default route into the directly attached stub areas with a metric of 5. Commit your changes.

```

[edit]
lab@SanJose# set protocols ospf area 1 stub default-metric 5

[edit]
lab@SanJose# commit
commit complete

```

Step 5.6

Examine the routing table on a router without an interface in Area 0.

lab@HongKong> **show route**

inet.0: 25 destinations, 26 routes (25 active, 0 holddown, 0 hidden)
 + = Active Route, - = Last Active, * = Both

```

0.0.0.0/0          *[OSPF/10] 00:01:14, metric 11
                   > via se-1/0/1.0
10.14.8.0/30       *[OSPF/10] 00:07:43, metric 7
                   > via se-1/0/1.0
10.14.8.4/30       *[OSPF/10] 00:07:43, metric 7
                   > via se-1/0/1.0
10.14.8.8/30       *[OSPF/10] 00:07:43, metric 8
                   > via se-1/0/1.0
10.14.8.12/30      *[Direct/0] 01:50:00
                   > via se-1/0/1.0
                   [OSPF/10] 01:49:58, metric 6
                   > via se-1/0/1.0
10.14.8.14/32      *[Local/0] 01:50:03
                   Local via se-1/0/1.0
10.14.8.16/30      *[OSPF/10] 00:07:43, metric 13
                   > via se-1/0/1.0
10.14.8.20/30      *[OSPF/10] 00:07:43, metric 13
                   > via se-1/0/1.0
10.14.8.24/30      *[OSPF/10] 00:04:14, metric 20
                   > via se-1/0/1.0
10.14.8.28/30      *[OSPF/10] 00:05:26, metric 14
                   > via se-1/0/1.0
10.14.9.64/27      *[Direct/0] 01:50:03
                   > via fe-2/0/1.101
10.14.9.65/32      *[Local/0] 01:50:03
                   Local via fe-2/0/1.101
10.14.10.0/24      *[OSPF/10] 00:07:43, metric 14
                   > via se-1/0/1.0
10.14.11.0/24      *[OSPF/10] 00:04:14, metric 21
                   > via se-1/0/1.0
10.210.0.176/28    *[Direct/0] 02:06:13
                   > via fe-0/0/0.0
10.210.0.177/32    *[Local/0] 02:06:15
                   Local via fe-0/0/0.0
192.168.25.1/32    *[Direct/0] 01:50:03
                   > via lo0.0
192.168.25.2/32    *[OSPF/10] 00:04:14, metric 20
                   > via se-1/0/1.0
192.168.25.3/32    *[OSPF/10] 00:04:14, metric 14
                   > via se-1/0/1.0
192.168.25.4/32    *[OSPF/10] 00:05:26, metric 13
                   > via se-1/0/1.0
192.168.25.5/32    *[OSPF/10] 00:07:43, metric 7
                   > via se-1/0/1.0
192.168.25.6/32    *[OSPF/10] 00:07:43, metric 6
                   > via se-1/0/1.0
192.168.25.7/32    *[OSPF/10] 00:07:43, metric 7
                   > via se-1/0/1.0
192.168.25.8/32    *[OSPF/10] 00:07:43, metric 13
  
```

```

                > via se-1/0/1.0
224.0.0.5/32    *[OSPF/10] 01:50:03, metric 1
                MultiRecv

```

__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

10.0.0.1/32    *[Direct/0] 02:06:47
                > via lo0.16385
10.0.0.16/32   *[Direct/0] 02:06:47
                > via lo0.16385

```

Question: What routes are available? If these routes changed, why?

Answer: The same routes observed in Step 5.4 should still be available, except the default route should also now be available.

Step 5.7

On the routers attached to Area 0, configure the router to stop advertising summary LSAs to the directly attached stub areas. Commit your changes.

```

[edit]
lab@SanJose# set protocols ospf area 1 stub no-summaries

```

```

[edit]
lab@SanJose# commit
commit complete

```

Step 5.8

Examine the routing table on a router without an interface in Area 0.

```

lab@HongKong> show route

inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0      *[OSPF/10] 00:00:32, metric 11
                > via se-1/0/1.0
10.14.8.12/30  *[Direct/0] 01:51:21
                > via se-1/0/1.0
                [OSPF/10] 01:51:19, metric 6
                > via se-1/0/1.0
10.14.8.14/32  *[Local/0] 01:51:24
                Local via se-1/0/1.0
10.14.9.64/27  *[Direct/0] 01:51:24

```

```

> via fe-2/0/1.101
10.14.9.65/32    *[Local/0] 01:51:24
                  Local via fe-2/0/1.101
10.210.0.176/28 *[Direct/0] 02:07:34
> via fe-0/0/0.0
10.210.0.177/32 *[Local/0] 02:07:36
                  Local via fe-0/0/0.0
192.168.25.1/32 *[Direct/0] 01:51:24
> via lo0.0
224.0.0.5/32    *[OSPF/10] 01:51:24, metric 1
                  MultiRecv

```

__juniper_private1__ .inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

10.0.0.1/32      *[Direct/0] 02:08:08
> via lo0.16385
10.0.0.16/32     *[Direct/0] 02:08:08
> via lo0.16385

```

Question: What routes are available?

Answer: No interarea routes should be available except for the default route.



Tell your instructor that you have completed Lab 3.

Not For Reproduction

Lab 4

Layer 2 Services (Detailed)

Overview

This lab explores Layer 2 services configuration. It is available in two formats: a high-level format that is designed to make you think through each step and a detailed format that offers step-by-step instructions complete with sample output from most commands.

By completing this lab, you will perform the following tasks:

- Configure the Compressed Real-Time Transport Protocol (CRTP).
- Verify the operation of the link services interface.
- Configure Multilink PPP (MLPPP).
- Configure the MLPPP interface to optimize voice traffic.

Note

You will coordinate many of your changes with a team working on another router. Look at the “Lab 4a: CRTP” diagram and find the serial connection attached to your router. The router that is connected to yours using a serial connection is your partner router for this lab.

Key Commands

Key operational-mode commands used in this lab include the following:

```
ping
show ospf neighbor
show services crtp
show services crtp extensive
show services crtp flows
show interfaces
show interfaces extensive
show interfaces terse
```

Part 1: Configure CRTP

In this part, you will configure the J-series routers to use CRTP to compress RTP communications over the serial links shown in the “Lab 4a: CRTP” diagram. This lab begins with the same configurations that you had at the end of the previous lab.

Step 1.1

Prepare the link services interface to perform CRTP compression. Configure Unit 0 to perform CRTP for RTP sessions on UDP ports 2000 through 64009.

```
lab@SanJose> configure
Entering configuration mode

[edit]
lab@SanJose# edit interfaces ls-0/0/0 unit 0

[edit interfaces ls-0/0/0 unit 0]
lab@SanJose# set compression rtp port minimum 2000 maximum 64009
```

Step 1.2

Configure the link services interface with the IP address assigned to the serial interface.

```
[edit interfaces ls-0/0/0 unit 0]
lab@SanJose# top show interfaces se-1/0/0
serial-options {
    clocking-mode internal;
}
unit 0 {
    family inet {
        address 10.14.8.13/30;
    }
}

[edit interfaces ls-0/0/0 unit 0]
lab@SanJose# set family inet address 10.14.8.13/30
```

Step 1.3

Delete the IP configuration from the serial interface. Configure the router to use the ls-0/0/0.0 interface for compression on this serial interface.

```
[edit interfaces ls-0/0/0 unit 0]
lab@SanJose# top edit interfaces se-1/0/0 unit 0
```

```
[edit interfaces se-1/0/0 unit 0]
lab@SanJose# delete family inet
```

```
[edit interfaces se-1/0/0 unit 0]
lab@SanJose# set compression-device ls-0/0/0.0
```

Step 1.4

Update the OSPF configuration to reference the `ls-` interface instead of the `se-` interface.
Commit your configuration.

```
[edit interfaces se-1/0/0 unit 0]
lab@SanJose# top edit protocols ospf
```

```
[edit protocols ospf]
lab@SanJose# rename area 1 interface se-1/0/0.0 to interface ls-0/0/0.0
```

```
[edit protocols ospf]
lab@SanJose# commit and-quit
commit complete
Exiting configuration mode
```



Wait for the team configuring your partner router to complete this part before continuing.

Part 2: Verify the Operation of the Link Services Interface

In this part, you will verify that the link services interface is functioning correctly.

Step 2.1

Display the status of all interfaces on the router.

```
lab@SanJose> show interfaces terse
```

Interface	Admin	Link	Proto	Local	Remote
fe-0/0/0	up	up			
fe-0/0/0.0	up	up	inet	10.210.0.182/28	
gr-0/0/0	up	up			
ip-0/0/0	up	up			
ls-0/0/0	up	up			
ls-0/0/0.0	up	up	inet	10.14.8.13/30	
lt-0/0/0	up	up			
mt-0/0/0	up	up			
pd-0/0/0	up	up			
pe-0/0/0	up	up			
sp-0/0/0	up	up			
sp-0/0/0.16383	up	up	inet		
fe-0/0/1	up	up			
fe-0/0/1.0	up	up	inet	10.14.8.5/30	

```

se-1/0/0          up    up
se-1/0/0.0        up    up    comp-dev ls-0/0/0.0
se-1/0/1          up    down
fe-2/0/0          up    up
fe-2/0/0.0        up    up    inet    10.14.8.1/30
fe-2/0/1          up    up
fe-2/0/1.16       up    up    inet    172.17.38.20/29
dsc               up    up
gre               up    up
ipip              up    up
lo0               up    up
lo0.0             up    up    inet    192.168.25.6    --> 0/0
lo0.16385         up    up    inet    10.0.0.1        --> 0/0
                  10.0.0.16    --> 0/0

lsi               up    up
mtun              up    up
pimd              up    up
pime              up    up
pp0               up    up
tap               up    up

```

Step 2.2

Display extensive statistics for the serial and link services interfaces.

```

lab@SanJose> show interfaces extensive se-1/0/0
Physical interface: se-1/0/0, Enabled, Physical link is Up
  Interface index: 139, SNMP ifIndex: 35, Generation: 140
  Type: Serial, Link-level type: Singlelink-PPP, MTU: 1504, Maximum speed:
16384kbps
  Device flags      : Present Running
  Interface flags: Point-To-Point Internal: 0x4000
  Link flags        : None
  Hold-times        : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 20 (last seen 00:00:07 ago)
    Output: 21 (last sent 00:00:03 ago)
  LCP state: Opened
  CHAP state: Closed
  CoS queues      : 8 supported, 8 maximum usable queues
  Last flapped    : 2006-11-27 16:38:17 UTC (17:49:30 ago)
  Statistics last cleared: Never
  Traffic statistics:
    Input bytes      :                682360                272 bps
    Output bytes     :                764802                 0 bps
    Input packets    :                20351                 0 pps
    Output packets   :                20155                 0 pps
  Input errors:
    Errors: 6, Drops: 0, Framing errors: 6, Runts: 0, Giants: 0, Policed
discards: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 1, Errors: 0, Drops: 0, MTU errors: 0, Resource
errors: 0
  Serial media information:

```

```

Line protocol: v.35
Resync history:
  Sync loss count: 0
Data signal:
  Rx Clock: OK
Control signals:
  Local mode: DCE
  To DTE: CTS: up, DCD: up, DSR: up
  From DTE: DTR: up, RTS: up
DCE loopback override: Off
Clocking mode: internal
Clock rate: 8.0 MHz
Loopback: none
Tx clock: non-invert
Line encoding: nrz
Packet Forwarding Engine configuration:
  Destination slot: 1, PLP byte: 1 (0x00)
  CoS transmit queue          Bandwidth          Buffer Priority
Limit
                                %          bps          %          usec
0 best-effort          95          15564800      95          0          low
none
3 network-control      5           819200        5           0          low
none

Logical interface se-1/0/0.0 (Index 72) (SNMP ifIndex 40) (Generation 139)
  Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
  Compression device: ls-0/0/0.0, Generation: 144, Route table: 0

lab@SanJose> show interfaces extensive ls-0/0/0
Physical interface: ls-0/0/0, Enabled, Physical link is Up
  Interface index: 134, SNMP ifIndex: 31, Generation: 135
  Link-level type: LinkService, MTU: 1504
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
  Last flapped   : 2006-11-27 16:32:36 UTC (17:55:21 ago)
  Statistics last cleared: Never
  Traffic statistics:
    Input bytes   :                2668                264 bps
    Output bytes  :                2580                 0 bps
    Input packets :                 37                 0 pps
    Output packets:                 34                 0 pps
  Frame exceptions:
    Oversized frames                0
    Errored input frames             0
    Input on disabled link/bundle    0
    Output for disabled link/bundle  0
    Queuing drops                   0
  Buffering exceptions:
    Packet data buffer overflow      0
    Fragment data buffer overflow    0
  Assembly exceptions:
    Fragment timeout                 0
    Missing sequence number          0

```

```

Out-of-order sequence number      0
Out-of-range sequence number      0
Hardware errors (sticky):
  Data memory error                0
  Control memory error             0
Egress queues: 8 supported, 8 in use
Queue counters:
  Queued packets  Transmitted packets  Dropped packets
0 best-effort      0                  0                  0
1 expedited-fo     0                  0                  0
2 assured-forw     0                  0                  0
3 network-cont     34                 34                 0

Logical interface ls-0/0/0.0 (Index 71) (SNMP ifIndex 43) (Generation 138)
Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
Bandwidth: 16384kbps
Bundle options:
  MRRU                1504
  Remote MRRU         N/A
  Drop timer period   0
  Sequence number format long (24 bits)
  Fragmentation threshold 0
  Links needed to sustain bundle 1
  Interleave fragments Disabled
Bundle errors:
  Packet drops        0 (0 bytes)
  Fragment drops      0 (0 bytes)
  MRRU exceeded       0
  Exception events    0
Statistics           Frames      fps      Bytes      bps
Bundle:
  Fragments:
    Input :           37          0        2705      272
    Output:           34          0        2444       0
  Packets:
    Input :           37          0        2668       0
    Output:           34          0        2580       0
Link:
  se-1/0/0.0
    Input :           37          0        2705      272
    Output:           34          0        2444       0
NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
Not-configured
Protocol inet, MTU: 1500, Generation: 143, Route table: 0
Flags: None
Addresses, Flags: Is-Preferred Is-Primary
  Destination: 10.14.8.12/30, Local: 10.14.8.13, Broadcast: Unspecified,
Generation: 148

```

Step 2.3

Ensure that you can ping your partner router successfully.

```
lab@SanJose> ping 10.14.8.14
PING 10.14.8.14 (10.14.8.14): 56 data bytes
64 bytes from 10.14.8.14: icmp_seq=0 ttl=64 time=10.630 ms
64 bytes from 10.14.8.14: icmp_seq=1 ttl=64 time=30.431 ms
64 bytes from 10.14.8.14: icmp_seq=2 ttl=64 time=10.424 ms
64 bytes from 10.14.8.14: icmp_seq=3 ttl=64 time=10.427 ms
64 bytes from 10.14.8.14: icmp_seq=4 ttl=64 time=10.418 ms
^C
--- 10.14.8.14 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 10.418/14.466/30.431/7.983 ms
```

Step 2.4

Determine whether your OSPF adjacency established correctly.

```
lab@SanJose> show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
10.14.8.6	fe-0/0/1.0	Full	192.168.25.7	128	33
10.14.8.2	fe-2/0/0.0	Full	192.168.25.5	128	31
10.14.8.14	ls-0/0/0.0	Full	192.168.25.1	128	31

Step 2.5

Display CRTP statistics for the interface.

```
lab@SanJose> show services crtp extensive
Interface: ls-0/0/0.0
Port minimum: 2000, Port maximum: 64009
Maximum UDP compressed sessions: 256
CRTP maximum period: 256, CRTP maximum time: 5
Compression ratio: 0, Decompression ratio: 0, Discards: 0
```

CRTP stats	Receive	Transmit
Sessions	0	0
IP bytes	0	0
Compressed bytes	0	0
CRTP packets	0	0
CUDP/CNTCP packets	0	0
Full header packets	0	0
Context state packets	0	0
IP packets	0	0
Compressed packets	0	0

Step 2.6

Display the CRTP flow table.

```
lab@SanJose> show services crtp flows
```

```
lab@SanJose>
```



Wait for the team configuring your partner router to complete this part before continuing.

Part 3: Configure MLPPP

In this part, you will configure the J-series routers to use Multilink PPP (MLPPP) to connect two routers. See the “Lab 4b: Multilink PPP” diagram.

Step 3.1

Delete the `ls-` interface configuration.

```
lab@SanJose> configure  
Entering configuration mode
```

```
[edit]  
lab@SanJose# delete interfaces ls-0/0/0
```

Step 3.2

Prepare the link services interface to perform MLPPP. Configure Unit 1 with Multilink PPP encapsulation.

```
[edit]  
lab@SanJose# edit interfaces ls-0/0/0 unit 1  
  
[edit interfaces ls-0/0/0 unit 1]  
lab@SanJose# set encapsulation multilink-ppp
```

Step 3.3

Configure the link service interface's Unit 1 with the IP address shown on the diagram.

```
[edit interfaces ls-0/0/0 unit 1]  
lab@SanJose# set family inet address 10.14.8.13/30
```

Step 3.4

Delete the compression configuration from the currently configured `se-` interface.

```
[edit interfaces ls-0/0/0 unit 1]  
lab@SanJose# top edit interfaces se-1/0/0 unit 0  
  
[edit interfaces se-1/0/0 unit 0]  
lab@SanJose# delete compression-device
```

Step 3.5

Configure the existing `se-` interface to run MLPPP as part of the bundle configured on the link service interface's Unit 1.

```
[edit interfaces se-1/0/0 unit 0]  
lab@SanJose# set family mlppp bundle ls-0/0/0.1
```

Step 3.6

Update the OSPF configuration by removing the `ls-0/0/0.0` interface and replacing it with the `ls-0/0/0.1` interface. Commit the configuration.

```
[edit interfaces se-1/0/0 unit 0]  
lab@SanJose# top edit protocols ospf
```



```
[edit protocols ospf]
lab@SanJose# rename area 1 interface ls-0/0/0.0 to interface ls-0/0/0.1

[edit protocols ospf]
lab@SanJose# commit and-quit
commit complete
Exiting configuration mode
```



Wait for the team configuring your partner router to complete the previous step before continuing.

Step 3.7

Check the status of all interfaces.

```
lab@SanJose> show interfaces terse
```

Interface	Admin	Link	Proto	Local	Remote
fe-0/0/0	up	up			
fe-0/0/0.0	up	up	inet	10.210.0.182/28	
gr-0/0/0	up	up			
ip-0/0/0	up	up			
ls-0/0/0	up	up			
ls-0/0/0.1	up	up	inet	10.14.8.13/30	
lt-0/0/0	up	up			
mt-0/0/0	up	up			
pd-0/0/0	up	up			
pe-0/0/0	up	up			
sp-0/0/0	up	up			
sp-0/0/0.16383	up	up	inet		
fe-0/0/1	up	up			
fe-0/0/1.0	up	up	inet	10.14.8.5/30	
se-1/0/0	up	up			
se-1/0/0.0	up	up	mlppp	ls-0/0/0.1	
se-1/0/1	up	down			
fe-2/0/0	up	up			
fe-2/0/0.0	up	up	inet	10.14.8.1/30	
fe-2/0/1	up	up			
fe-2/0/1.16	up	up	inet	172.17.38.20/29	
dsc	up	up			
gre	up	up			
ipip	up	up			
lo0	up	up			
lo0.0	up	up	inet	192.168.25.6	--> 0/0
lo0.16385	up	up	inet	10.0.0.1	--> 0/0
				10.0.0.16	--> 0/0
lsi	up	up			
mtun	up	up			
pimd	up	up			
pime	up	up			
pp0	up	up			
tap	up	up			

Question: How does the constituent link (se-) appear in the output of the **show interfaces terse** command?

Answer: The designation mlppp appears in the Protocol field, and the associated link services interface appears in the Local field.

Step 3.8

Check the status of the serial interface.

```
lab@SanJose> show interfaces se-1/0/0
Physical interface: se-1/0/0, Enabled, Physical link is Up
  Interface index: 139, SNMP ifIndex: 35
  Type: Serial, Link-level type: Multilink-PPP, MTU: 1510, Maximum speed:
16384kbps
  Device flags      : Present Running
  Interface flags: Point-To-Point Internal: 0x4000
  Link flags        : None
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive: Input: 24 (00:00:08 ago), Output: 23 (00:00:08 ago)
  LCP state: Opened
  CHAP state: Closed
  CoS queues        : 8 supported, 8 maximum usable queues
  Last flapped      : 2006-11-27 16:38:17 UTC (18:03:55 ago)
  Input rate        : 0 bps (0 pps)
  Output rate       : 288 bps (0 pps)

Logical interface se-1/0/0.0 (Index 73) (SNMP ifIndex 40)
  Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
  Protocol mlppp, Multilink bundle: ls-0/0/0.1, MTU: 1506
```

Question: What information appears for the serial interface's logical interface in the output of the **show interfaces** command?

Answer: It is identified as using Protocol mlppp and the link services interface for the multilink bundle is listed.

Step 3.9

Check the status of the link services interface.

```

lab@SanJose> show interfaces ls-0/0/0
Physical interface: ls-0/0/0, Enabled, Physical link is Up
  Interface index: 134, SNMP ifIndex: 31
  Link-level type: LinkService, MTU: 1504
  Device flags      : Present Running
  Interface flags: Point-To-Point SNMP-Traps
  Last flapped     : 2006-11-27 16:32:36 UTC (18:17:46 ago)
  Input rate       : 0 bps (0 pps)
  Output rate      : 0 bps (0 pps)

Logical interface ls-0/0/0.1 (Index 72) (SNMP ifIndex 44)
  Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: Multilink-PPP
  Bandwidth: 16384kbps
  Statistics          Frames          fps          Bytes          bps
Bundle:
  Fragments:
    Input :           97             0          7355           0
    Output:           95             0          7038           0
  Packets:
    Input :          194             0         13352           0
    Output:           95             0          7038           0
  Link:
    se-1/0/0.0
    Input :           97             0          7355           0
    Output:           95             0          7038           0
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
Not-configured
  Protocol inet, MTU: 1500
  Flags: None
  Addresses, Flags: Is-Preferred Is-Primary
    Destination: 10.14.8.12/30, Local: 10.14.8.13

```

Step 3.10

Confirm that you can ping the router on the other side of the service interface.

```

lab@SanJose> ping 10.14.8.14
PING 10.14.8.14 (10.14.8.14): 56 data bytes
64 bytes from 10.14.8.14: icmp_seq=0 ttl=64 time=6.727 ms
64 bytes from 10.14.8.14: icmp_seq=1 ttl=64 time=20.462 ms
64 bytes from 10.14.8.14: icmp_seq=2 ttl=64 time=20.366 ms
64 bytes from 10.14.8.14: icmp_seq=3 ttl=64 time=40.490 ms
64 bytes from 10.14.8.14: icmp_seq=4 ttl=64 time=20.515 ms
^C
--- 10.14.8.14 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.727/21.712/40.490/10.789 ms

```

Step 3.11

Confirm that you formed an OSPF adjacency over the MLPPP interface.

```
lab@SanJose> show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
10.14.8.6	fe-0/0/1.0	Full	192.168.25.7	128	36
10.14.8.2	fe-2/0/0.0	Full	192.168.25.5	128	39
10.14.8.14	ls-0/0/0.1	Full	192.168.25.1	128	36

Step 3.12

Copy the configuration from the existing serial interface to the new serial interface shown on the “Lab 4b:Multilink PPP” diagram. Commit the configuration.

```
lab@SanJose> configure
```

```
Entering configuration mode
```

```
[edit]
```

```
lab@SanJose# copy interfaces se-1/0/0 to se-1/0/1
```

```
[edit]
```

```
lab@SanJose# commit and-quit
```

```
commit complete
```

```
Exiting configuration mode
```



Wait for the team configuring your partner router to complete the previous step before continuing.

Step 3.13

Check the status of the new serial interface.

```
lab@SanJose> show interfaces se-1/0/1
```

```
Physical interface: se-1/0/1, Enabled, Physical link is Up
```

```
Interface index: 140, SNMP ifIndex: 36
```

```
Type: Serial, Link-level type: Multilink-PPP, MTU: 1510, Maximum speed: 16384kbps
```

```
Device flags : Present Running
```

```
Interface flags: Point-To-Point Internal: 0x4000
```

```
Link flags : None
```

```
Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
```

```
Keepalive: Input: 6 (00:00:05 ago), Output: 6 (00:00:07 ago)
```

```
LCP state: Opened
```

```
CHAP state: Closed
```

```
CoS queues : 8 supported, 8 maximum usable queues
```

```
Last flapped : 2006-11-28 10:53:07 UTC (00:01:08 ago)
```

```
Input rate : 0 bps (0 pps)
```

```
Output rate : 0 bps (0 pps)
```

```
Logical interface se-1/0/1.0 (Index 71) (SNMP ifIndex 45)
```

```
Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
```

```
Protocol mlppp, Multilink bundle: ls-0/0/0.1, MTU: 1506
```

Step 3.14

Check the status of the link services interface.

```
lab@SanJose> show interfaces ls-0/0/0.1
```

```
Logical interface ls-0/0/0.1 (Index 72) (SNMP ifIndex 44)
```

```
Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: Multilink-PPP
```

```
Bandwidth: 32768kbps
```

```
Statistics          Frames          fps          Bytes          bps
```

```
Bundle:
```

```
Fragments:
```

```
Input :           138           0          10606           0
```

```
Output:           137           0          10274          288
```

```
Packets:
```

```
Input :           276           0          19280           0
```

```
Output:           137           0          10274           0
```

```
Link:
```

```
se-1/0/0.0
```

```
Input :           132           0          10160           0
```

```
Output:           130           0           9760           0
```

```
se-1/0/1.0
```

```
Input :             6           0           446           0
```

```
Output:             7           0           514          288
```

```
NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls: Not-configured
```

```
Protocol inet, MTU: 1500
```

```
Flags: None
```

```
Addresses, Flags: Is-Preferred Is-Primary
```

```
Destination: 10.14.8.12/30, Local: 10.14.8.13
```

Question: What kind of statistics are available in the output of the **show interfaces ls-0/0/0.1** command?

Answer: The output contains statistics for the bundle as well as the individual constituent interfaces.

Step 3.15

Confirm that you can ping the router on the other side of the service interface.

```
lab@SanJose> ping 10.14.8.14
```

```
PING 10.14.8.14 (10.14.8.14): 56 data bytes
```

```
64 bytes from 10.14.8.14: icmp_seq=0 ttl=64 time=35.612 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=1 ttl=64 time=10.482 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=2 ttl=64 time=10.388 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=3 ttl=64 time=10.388 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=4 ttl=64 time=10.395 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=5 ttl=64 time=20.465 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=6 ttl=64 time=10.401 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=7 ttl=64 time=10.389 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=8 ttl=64 time=30.193 ms
```

```
64 bytes from 10.14.8.14: icmp_seq=9 ttl=64 time=10.410 ms
```

```
^C
```

```
--- 10.14.8.14 ping statistics ---
```

10 packets transmitted, 10 packets received, 0% packet loss
 round-trip min/avg/max/stddev = 10.388/15.912/35.612/9.082 ms

Step 3.16

Confirm that you formed an OSPF adjacency over the MLPPP interface.

```
lab@SanJose> show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
10.14.8.6	fe-0/0/1.0	Full	192.168.25.7	128	33
10.14.8.2	fe-2/0/0.0	Full	192.168.25.5	128	36
10.14.8.14	ls-0/0/0.1	Full	192.168.25.1	128	30

Part 4: Configure the MLPPP Interface to Optimize Voice Traffic

In this part, you will add CRTP to the MLPPP interface and configure it to support link fragmentation and interleaving (LFI).

Step 4.1

Confirm the current fragmentation configuration with the command **show interfaces ls-0/0/0.1 extensive**.

```
lab@SanJose> show interfaces ls-0/0/0.1 extensive
```

Logical interface ls-0/0/0.1 (Index 72) (SNMP ifIndex 44) (Generation 140)

Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: Multilink-PPP

Bandwidth: 32768kbps

Bundle options:

MRRU	1504
Remote MRRU	1504
Drop timer period	0
Sequence number format	long (24 bits)
Fragmentation threshold	0
Links needed to sustain bundle	1
Interleave fragments	Disabled

Bundle errors:

Packet drops	0 (0 bytes)
Fragment drops	0 (0 bytes)
MRRU exceeded	0
Exception events	0

Statistics	Frames	fps	Bytes	bps
------------	--------	-----	-------	-----

Bundle:

Fragments:

Input :	161	0	12491	0
Output:	160	0	12136	0

Packets:

Input :	322	0	22728	0
Output:	160	0	12136	0

Link:

se-1/0/0.0

Input :	143	0	11049	0
Output:	142	0	10744	0

se-1/0/1.0

Input :	18	0	1442	0
Output:	18	0	1392	0

```

NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
Not-configured
  Protocol inet, MTU: 1500, Generation: 145, Route table: 0
    Flags: None
    Addresses, Flags: Is-Preferred Is-Primary
      Destination: 10.14.8.12/30, Local: 10.14.8.13, Broadcast: Unspecified,
Generation: 150

```

Step 4.2

Configure the link services interface to perform CRTP for RTP sessions on UDP ports 2000 through 64009.

```

lab@SanJose> configure
Entering configuration mode

[edit]
lab@SanJose# edit interfaces ls-0/0/0 unit 1

[edit interfaces ls-0/0/0 unit 1]
lab@SanJose# set compression rtp port minimum 2000 maximum 64009

```

Step 4.3

Configure the link services interface to fragment packets larger than 256 bytes.

```

[edit interfaces ls-0/0/0 unit 1]
lab@SanJose# set fragment-threshold 256

```

Step 4.4

Configure the link services interface to interleave fragments. Commit the configuration.

```

[edit interfaces ls-0/0/0 unit 1]
lab@SanJose# set interleave-fragments

[edit interfaces ls-0/0/0 unit 1]
lab@SanJose# commit and-quit
commit complete
Exiting configuration mode

```

Step 4.5

View the current fragmentation configuration with the command **show interfaces ls-0/0/0.1 extensive**.

```

lab@SanJose> show interfaces ls-0/0/0.1 extensive
Logical interface ls-0/0/0.1 (Index 72) (SNMP ifIndex 44) (Generation 140)
Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: Multilink-PPP
Bandwidth: 32768kbps
Bundle options:
  MRRU                                1504
  Remote MRRU                         1504
  Drop timer period                   0
  Sequence number format              long (24 bits)
  Fragmentation threshold             256
  Links needed to sustain bundle      1
  Interleave fragments                Enabled

```

Bundle errors:

Packet drops	0 (0 bytes)
Fragment drops	0 (0 bytes)
MRRU exceeded	0
Exception events	0

Statistics	Frames	fps	Bytes	bps
------------	--------	-----	-------	-----

Bundle:

Fragments:

Input :	198	0	15386	0
Output:	194	0	14700	0

Packets:

Input :	396	0	28000	0
Output:	195	0	14774	0

Link:

se-1/0/0.0

Input :	162	0	12518	0
Output:	159	0	12018	0

se-1/0/1.0

Input :	36	0	2868	0
Output:	35	0	2682	0

NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls: Not-configured

Protocol inet, MTU: 1500, Generation: 145, Route table: 0

Flags: None

Addresses, Flags: Is-Preferred Is-Primary

Destination: 10.14.8.12/30, Local: 10.14.8.13, Broadcast: Unspecified, Generation: 150

Question: Why will these changes help voice traffic?

Answer: CRTP compresses the IP, UDP, and RTP headers for RTP traffic, reducing the transmission time for RTP packets. LFI reduces serialization delay by breaking larger packets into smaller fragments and allowing higher-priority traffic to be transmitted between fragments of lower-priority traffic.



Tell your instructor that you have completed Lab 4.

Stateful Firewall and NAT (Detailed)

Overview

This lab introduces you to stateful firewall and Network Address Translation (NAT) configuration. You will configure a virtual router on the Sydney router to act as a host inside your network. You will then configure your router to use a next-hop-style service set to perform NAT to provide your internal network with connectivity to the Internet. You will also configure stateful firewall rules to protect your network. Finally, you will migrate this configuration to use an interface-style service set.

This lab is available in two formats: a high-level format that is designed to make you think through each step and a detailed format that offers step-by-step instructions complete with sample output from most commands.

By completing this lab, you will perform the following tasks:

- Prepare the routers.
- Configure a virtual router.
- Configure a next-hop-style service set.
- Configure NAT rules.
- Verify NAT operation.
- Configure stateful firewall rules.
- Verify stateful firewall operation.
- Configure an application-level gateway (ALG).
- Configure an interface-style service set.

Key Commands

Key operational-mode commands used in this lab include the following:

```
ping
show route table
show services stateful-firewall flows
show services stateful-firewall statistics
show services stateful-firewall statistics extensive
start shell
```

Part 1: Prepare the Routers

This lab uses a configuration similar to the one used in Lab 1. To aid you, a starting configuration is available in the lab user's home directory in the file *ajre/lab5-reset.conf*. In this part, you will load the configuration and configure a virtual router on Sydney. This time, each team will have its own virtual router.

The JUNOS software sometimes processes transit traffic slightly differently than local traffic. We will use the virtual routers to produce transit traffic and simulate a real user workstation.

Step 1.1

Load the configuration file located in the lab user's home directory called *ajre/lab5-reset.conf*. Commit the configuration.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# load override ajre/lab5-reset.conf
load complete

[edit]
lab@HongKong# commit
commit complete
```

Step 1.2

View the configuration.

```
[edit]
lab@HongKong# show
version 8.0R2.8;
system {
    host-name HongKong;
    root-authentication {
        encrypted-password "$1$KI99zGk6$MbYFuBbpLffu9tn2.sI7l1"; ## SECRET-DATA
        ssh-dsa "ssh-dss
AAAAB3NzaC1kc3MAAACBAMQrfP2bZyBXJ6PC7XXZ+MzErI8Jl6jah5L4/
O8BsfP2hC7EvRfNoX7MqbrtCX/9gUH9gChVuBCB+ERULMdgrVm5uGhC/
gs4UX+4dBbfBgKYYwgmisM8EoT25m7qI8ybpl2YZvHNznvO8h7kr4kpYuQEpkvgsTdH/
Jle4Uqnjv7DAAAFQDZaqA6QAgbW30/
zveaLCIDj6p0dwAAAIbLiL+krWrXiD8NPpY+w4dWxEqaV3bnobzPC4eyxQKBUCOr80Q5YBlWXVBHx9
elwBWZwj0SF4hLKHznExnLerVsMuTMA846RbQmSz62vM6kGM13HFonWeQvWia0TD78+rOEgWF2KHB
```

```

SIXL511mIDW8Gql9hJfD/Dr/
NKP97w3L0wAAAIEAr3FkWU8XbYytQYEkxsIN9P1UQ1ERXB3G40YwqFO484SlyKyYCFaz+yNsaAJu2C
8UebDIR3GieyNcOAKf3inCG8jQwjLvZskuZwrvlsz/xtcxSoAh9axJcdUfSJYMW/
g+mD26JKl1Cliw5rwp2nH9kUrJxeI7IREdp4egNkM4i15o= configurator@server1.he"; ##
SECRET-DATA
}
login {
    user lab {
        uid 2000;
        class super-user;
        authentication {
            encrypted-password "$1$84J5Maes$cni5Hrazbd/IEHr/50oY30"; ##
SECRET-DATA
        }
    }
}
services {
    ftp;
    ssh;
    telnet;
    web-management {
        http;
    }
}
syslog {
    user * {
        any emergency;
    }
    file messages {
        any any;
        authorization info;
    }
    file interactive-commands {
        interactive-commands any;
    }
}
}
interfaces {
    fe-0/0/0 {
        description "MGMT INTERFACE - DO NOT DELETE";
        unit 0 {
            family inet {
                address 10.210.0.177/28;
            }
        }
    }
    se-1/0/0 {
        encapsulation frame-relay;
        unit 101 {
            dlci 101;
            family inet {
                address 172.17.39.18/30;
            }
        }
    }
}

```

```

        unit 201 {
            dlci 201;
            family inet {
                address 172.17.55.18/30;
            }
        }
    }
    fe-2/0/1 {
        vlan-tagging;
        unit 200 {
            vlan-id 200;
            family inet {
                address 192.168.200.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.14.8.1/32 {
                    primary;
                    preferred;
                }
                address 10.14.8.254/32;
            }
        }
    }
}
routing-options {
    aggregate {
        route 10.14.8.0/24;
    }
    generate {
        route 0.0.0.0/0 {
            policy [ match-ISP reject-all ];
            discard;
        }
    }
    router-id 10.14.8.1;
    autonomous-system 65108;
}
protocols {
    bgp {
        group isp {
            export [ to-ISP reject-all ];
            neighbor 172.17.39.17 {
                description isp-a;
                peer-as 65010;
            }
            neighbor 172.17.55.17 {
                description isp-c;
                peer-as 65030;
            }
        }
    }
}

```

```

}
ospf {
    export default-to-ospf;
    area 0.0.0.0 {
        interface fe-2/0/1.200;
        interface lo0.0 {
            passive;
        }
    }
}
}
}
policy-options {
    prefix-list announce-to-ISP {
        10.14.8.0/24;
    }
    policy-statement default-to-ospf {
        term match-default {
            from {
                route-filter 0.0.0.0/0 exact;
            }
            then {
                metric 0;
                external {
                    type 1;
                }
                accept;
            }
        }
    }
}
policy-statement match-ISP {
    term received-from-isp-a {
        from {
            protocol bgp;
            neighbor 172.17.39.17;
        }
        then accept;
    }
    term received-from-isp-c {
        from {
            protocol bgp;
            neighbor 172.17.55.17;
        }
        then accept;
    }
}
policy-statement reject-all {
    then reject;
}
policy-statement to-ISP {
    term match-routes {
        from {
            prefix-list announce-to-ISP;
        }
        then accept;
    }
}

```

```

    }
  }
}

```

Step 1.3

Ensure that the router has Internet connectivity from your aggregate prefix by attempting to ping the IP address 172.17.24.1 (an IP address on ISP B's network), sourcing the ping from your loopback address.

```

[edit]
lab@HongKong# run ping 172.17.24.1 source 10.14.8.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=63 time=14.592 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=63 time=10.391 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=63 time=10.367 ms
64 bytes from 172.17.24.1: icmp_seq=3 ttl=63 time=10.439 ms
^C
--- 172.17.24.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 10.367/11.447/14.592/1.816 ms

```

Step 1.4

Examine the “Lab 5b: Firewall Policy and NAT” diagram and locate the information regarding the virtual router configuration. The virtual router is again located on Sydney. Log in to the Sydney router using your router's name (all lower-case letters) as the username, and use the password provided by your instructor. Much of the basic configuration of your virtual router, including assigning interfaces to the routing instance, is completed for you and is also hidden from you. Begin by configuring the fe-2/0/1 interface as shown on the diagram. Use the VLAN number as the unit number for the fe-2/0/1 interface.

Sydney (ttypl)

login: hongkong
Password:

--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC

NOTE: This router is divided into many virtual routers used by different teams. Please only configure your own virtual router.

You must use 'configure private' to configure this router.

```

hongkong@Sydney> show configuration
interfaces {
    apply-groups;
    fe-2/0/1 {
    }
    lo0 {
    }
}
policy-options {
}
routing-instances {
    HongKong-vr {

```

```

        instance-type virtual-router;
        interface fe-2/0/1.200; ## 'fe-2/0/1.200' is not defined
    }
}

```

```

hongkong@Sydney> configure private
warning: uncommitted changes will be discarded on exit
Entering configuration mode

```

```

[edit]
hongkong@Sydney# edit interfaces fe-2/0/1 unit 200

[edit interfaces fe-2/0/1 unit 200]
hongkong@Sydney# set vlan-id 200

[edit interfaces fe-2/0/1 unit 200]
hongkong@Sydney# set family inet address 192.168.200.2/24

```

Step 1.5

Configure OSPF in your routing instance. Commit the configuration.

```

[edit interfaces fe-2/0/1 unit 200]
hongkong@Sydney# top edit routing-instances HongKong-vr protocols ospf

[edit routing-instances HongKong-vr protocols ospf]
hongkong@Sydney# set area 0 interface fe-2/0/1.200

[edit routing-instances HongKong-vr protocols ospf]
hongkong@Sydney# top

[edit]
hongkong@Sydney# commit and-quit
commit complete
Exiting configuration mode

```

Step 1.6

View the OSPF neighbor state and routing table for your virtual router on Sydney. Ensure that you are receiving a default route via OSPF.

```

hongkong@Sydney> show ospf neighbor instance HongKong-vr

```

Address	Interface	State	ID	Pri	Dead
192.168.200.1	fe-2/0/1.200	Full	10.14.8.1	128	33

```

hongkong@Sydney> show route table HongKong-vr

HongKong-vr.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 00:03:24, metric 1, tag 0
                   > to 192.168.200.1 via fe-2/0/1.200
10.14.8.1/32       *[OSPF/10] 00:03:24, metric 1
                   > to 192.168.200.1 via fe-2/0/1.200
10.14.8.254/32     *[OSPF/10] 00:03:24, metric 1
                   > to 192.168.200.1 via fe-2/0/1.200

```

```
192.168.200.0/24    *[Direct/0] 00:03:29
                  > via fe-2/0/1.200
192.168.200.2/32  *[Local/0] 00:03:29
                  Local via fe-2/0/1.200
224.0.0.5/32      *[OSPF/10] 00:07:16, metric 1
                  MultiRecv
```

Part 2: Configure a Virtual Router

You will configure a next-hop-style service set to provide stateful firewall and NAT services. In this part, you will configure a virtual router to hold the internal hosts. You will perform all these steps on your router unless otherwise indicated.

Step 2.1

Configure a routing instance called *trusted-vr* of type *virtual-router*.

```
[edit]
lab@HongKong# edit routing-instances trusted-vr

[edit routing-instances trusted-vr]
lab@HongKong# set instance-type virtual-router
```

Step 2.2

Assign the appropriate *fe-2/0/1* logical interface to the routing instance.

```
[edit routing-instances trusted-vr]
lab@HongKong# set interface fe-2/0/1.200
```

Step 2.3

Deactivate the OSPF configuration in the main router.

```
[edit routing-instances trusted-vr]
lab@HongKong# top deactivate protocols ospf
```

Step 2.4

Configure OSPF in the routing instance. Assign the appropriate *fe-2/0/1* logical interface to Area 0.

```
[edit routing-instances trusted-vr]
lab@HongKong# set protocols ospf area 0 interface fe-2/0/1.200
```

Step 2.5

Configure the router to export default routes to the routing instance's OSPF process.

```
[edit routing-instances trusted-vr]
lab@HongKong# set protocols ospf export default-to-ospf
```

Part 3: Configure a Next-Hop-Style Service Set

In this part, you will configure a next-hop-style service set to secure your internal network. Unless otherwise indicated, you will perform all these steps on your own router.

Step 3.1

Begin by configuring the services interfaces. Configure family `inet` on Unit 0.

```
[edit routing-instances trusted-vr]
lab@HongKong# top edit interfaces sp-0/0/0

[edit interfaces sp-0/0/0]
lab@HongKong# set unit 0 family inet
```

Step 3.2

Configure Unit 1 to be an outside services interface. Configure family `inet` on this unit.

```
[edit interfaces sp-0/0/0]
lab@HongKong# set unit 1 service-domain outside

[edit interfaces sp-0/0/0]
lab@HongKong# set unit 1 family inet
```

Step 3.3

Configure Unit 2 to be an inside services interface. Configure family `inet` on this unit.

```
[edit interfaces sp-0/0/0]
lab@HongKong# set unit 2 service-domain inside

[edit interfaces sp-0/0/0]
lab@HongKong# set unit 2 family inet
```

Step 3.4

Configure a next-hop-style service set called `trust-untrust` using the inside and outside services interfaces you just configured.

```
[edit interfaces sp-0/0/0]
lab@HongKong# top edit services service-set trust-untrust

[edit services service-set trust-untrust]
lab@HongKong# set next-hop-service outside-service-interface sp-0/0/0.1

[edit services service-set trust-untrust]
lab@HongKong# set next-hop-service inside-service-interface sp-0/0/0.2
```

Step 3.5

Assign the inside services interface for this service set to the `trusted-vr` routing instance.

```
[edit services service-set trust-untrust]
lab@HongKong# top edit routing-instances trusted-vr

[edit routing-instances trusted-vr]
lab@HongKong# set interface sp-0/0/0.2
```

Question: To what routing instance is the outside services interface assigned?

Answer: All interfaces are assigned to the master routing instance by default. Therefore, the outside services interface is assigned to the master routing instance because you did not assign it to a different routing instance.

Step 3.6

In the *trusted-vr* routing instance, configure a static default route that directs traffic to the inside services interface.

```
[edit routing-instances trusted-vr]
lab@HongKong# set routing-options static route 0.0.0.0/0 next-hop sp-0/0/0.2
```

Part 4: Configure NAT Rules

In this part, you will configure NAT rules and apply them to the service set.

Step 4.1

Configure a NAT pool called *dot2patpool*. Configure the pool to consist of the .2 address in your aggregate. Configure the router to perform dynamic Port Address Translation (PAT) when using the NAT pool.

```
[edit routing-instances trusted-vr]
lab@HongKong# top edit services nat pool dot2patpool

[edit services nat pool dot2patpool]
lab@HongKong# set address 10.14.8.2/32

[edit services nat pool dot2patpool]
lab@HongKong# set port automatic
```

Step 4.2

Configure a NAT rule called *PAT-internal* with a single term that matches all traffic with a source address from the subnet assigned to the appropriate *fe-2/0/1* logical interface. Configure the router to perform dynamic source translation for this traffic using the *dot2patpool* NAT pool.

```
[edit services nat pool dot2patpool]
lab@HongKong# up 1 edit rule PAT-internal term match-internal

[edit services nat rule PAT-internal term match-internal]
lab@HongKong# set from source-address 192.168.200.0/24

[edit services nat rule PAT-internal term match-internal]
lab@HongKong# set then translated translation-type source dynamic

[edit services nat rule PAT-internal term match-internal]
lab@HongKong# set then translated source-pool dot2patpool
```

Step 4.3

Configure the NAT rule to match traffic in the input direction.

```
[edit services nat rule PAT-internal term match-internal]
lab@HongKong# up

[edit services nat rule PAT-internal]
lab@HongKong# set match-direction input
```

Question: Why should this rule match traffic in the input direction?

Answer: Because this will be used in a next-hop-style service set, the router considers traffic that is inbound on the inside interface to be input traffic.

Step 4.4

Configure the *trust-untrust* service set to use the *PAT-internal* NAT rule.

```
[edit services nat rule PAT-internal]
lab@HongKong# top edit services service-set trust-untrust

[edit services service-set trust-untrust]
lab@HongKong# set nat-rules PAT-internal
```

Step 4.5

Commit your configuration.

```
[edit services service-set trust-untrust]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 4.6

Verify that the master routing instance has a route for the NAT pool IP address with a next hop of the outside services interface.

```
lab@HongKong> show route 10.14.8.2/32

inet.0: 21 destinations, 31 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.14.8.2/32          *[Static/1] 00:00:07
                    > via sp-0/0/0.1
```

Step 4.7

Verify that the *trusted-vr* routing instance has a default route with a next hop of the inside services interface.

```
lab@HongKong> show route table trusted-vr 0.0.0.0/0 exact
```

```
trusted-vr.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          *[Static/5] 00:03:20
                   > via sp-0/0/0.2
```

Step 4.8

Verify that the *trusted-vr* routing instance is exporting the default route to OSPF.

```
lab@HongKong> show route table trusted-vr 0.0.0.0/0 exact extensive
```

```
trusted-vr.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
0.0.0.0/0 (1 entry, 1 announced)
```

TSI:

OSPF area : 0.0.0.0, LSA ID : 0.0.0.0, LSA type : Extern

KRT in-kernel 0.0.0.0/0 -> {sp-0/0/0.2}

*Static Preference: 5

Next-hop reference count: 3

Next hop: via sp-0/0/0.2, selected

State: <Active Int Ext>

Age: 6:41

Task: RT

Announcement bits (2): 0-trusted-vr-OSPFv2 2-KRT

AS path: I

```
lab@HongKong> show ospf database instance trusted-vr
```

OSPF link state database, Area 0.0.0.0

Type	ID	Adv Rtr	Seq	Age	Opt	Cksum	Len
Router	10.14.8.1	10.14.8.1	0x80000009	720	0x22	0xf02f	60
Router	*192.168.200.1	192.168.200.1	0x80000002	399	0x22	0x2240	36
Router	192.168.200.2	192.168.200.2	0x80000005	405	0x22	0x144a	36
Network	192.168.200.1	10.14.8.1	0x80000004	420	0x22	0xbbd1	32
Network	192.168.200.2	192.168.200.2	0x80000002	405	0x22	0xbfa9	32

OSPF AS SCOPE link state database

Type	ID	Adv Rtr	Seq	Age	Opt	Cksum	Len
Extern	0.0.0.0	10.14.8.1	0x80000004	1020	0x22	0x6ea0	36
Extern	*0.0.0.0	192.168.200.1	0x80000001	404	0x22	0xed12	36

Step 4.9

Log in to the Sydney router. Verify that your virtual router on the Sydney router is receiving a default route via OSPF.

```
hongkong@Sydney> show route table HongKong-vr
```

```
HongKong-vr.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          *[OSPF/150] 00:07:43, metric 1, tag 0
                   > to 192.168.200.1 via fe-2/0/1.200
192.168.200.0/24   *[Direct/0] 00:50:45
                   > via fe-2/0/1.200
```

```

192.168.200.2/32    *[Local/0] 00:50:45
                   Local via fe-2/0/1.200
224.0.0.5/32      *[OSPF/10] 00:54:32, metric 1
                   MultiRecv

```

Part 5: Verify NAT Operation

In this part, you will verify that the NAT rule you just configured is working correctly.

Step 5.1

From Sydney, attempt to ping the IP address 172.17.24.1 (an IP address on ISP B's network) from your virtual router. Allow the router to continue to ping so as to provide a steady flow of traffic for you to examine.

```

hongkong@Sydney> ping routing-instance HongKong-vr 172.17.24.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=62 time=13.208 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=62 time=20.489 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=62 time=20.429 ms
64 bytes from 172.17.24.1: icmp_seq=3 ttl=62 time=132.850 ms
64 bytes from 172.17.24.1: icmp_seq=4 ttl=62 time=20.741 ms
[...]
```

Step 5.2

On your router, examine the flow table.

```

lab@HongKong> show services stateful-firewall flows
Interface: sp-0/0/0, Service set: trust-untrust
Flow                                     State   Dir      Frm count
ICMP      192.168.200.2:33567 -> 172.17.24.1   Watch   I          9
      NAT source 192.168.200.2:33567 -> 10.14.8.2:1025
ICMP      172.17.24.1:1025 -> 10.14.8.2     Watch   O          9
      NAT dest   10.14.8.2:1025 -> 192.168.200.2:33567

```

Question: How many flows do you see?

Answer: You should see two flows, one for each direction of traffic.

Question: In what state are the flows? What does that indicate?

Answer: The flows are in the `watch` state. This state indicates that the router is accepting the traffic and applying an ALG. To perform PAT with pings, the router must use an ALG to examine ICMP fields to correctly match incoming and outgoing packets.

Question: Examine the direction of the flows. Do they match your understanding of flow direction?

Answer: The flow from the virtual router to the Internet is marked as an input flow (I), while the flow from the Internet to the virtual router is marked as an output flow (O).

Step 5.3

Examine the NAT traffic statistics using the **show services stateful-firewall statistics extensive** command. Notice the types of statistics that are available.

```
lab@HongKong> show services stateful-firewall statistics extensive
Interface: sp-0/0/0
Service set: trust-untrust
New flows:
  Accepts: 0, Discards: 0, Rejects: 0
Existing flows:
  Accepts: 1014, Discards: 0, Rejects: 0
Drops:
  IP option: 0, TCP SYN defense: 0
  NAT ports exhausted: 0
Errors:
  IP: 0, TCP: 0
  UDP: 0, ICMP: 0
  Non-IP packets: 0, ALG: 0
IP errors:
  IP packet length inconsistencies: 0
  Minimum IP header length check failures: 0
  Reassembled packet exceeds maximum IP length: 0
  Illegal source address: 0
  Illegal destination address: 0
  TTL zero errors: 0, Illegal IP protocol number (0 or 255): 0
  Land attack: 0
  Non-IPv4 packets: 0, Bad checksum: 0
  Illegal IP fragment length: 0
  IP fragment overlap: 0
  IP fragment reassembly timeout: 0
  Unknown: 0
TCP errors:
  TCP header length inconsistencies: 0
  Source or destination port number is zero: 0
  Illegal sequence number and flags combinations: 0
  SYN attack (multiple SYN messages seen for the same flow): 0
  First packet not a SYN message: 0
  TCP port scan (TCP handshake, RST seen from server for SYN): 0
  Bad SYN cookie response: 0
UDP errors:
  IP data length less than minimum UDP header length (8 bytes): 0
  Source or destination port number is zero: 0
```

```

    UDP port scan (ICMP error seen for UDP flow): 0
  ICMP errors:
    IP data length less than minimum ICMP header length (8 bytes): 0
    ICMP error length inconsistencies: 0
    Duplicate ping sequence number: 0
    Mismatched ping sequence number: 0
  ALG errors:
    BOOTP: 0, DCE-RPC: 0, DCE-RPC portmap: 0
    DNS: 0, Exec: 0, FTP: 0
    H323: 0, ICMP: 0, IIOP: 0
    Login: 0, NetBIOS: 0, NetShow: 0
    Real Audio: 0, RPC: 0, RPC portmap: 0
    RTSP: 0, Shell: 0, SIP: 0
    SNMP: 0, SQLNet: 0, TFTP: 0
    Traceroute: 0

```

Step 5.4

On Sydney, stop the ping you started in Step 5.1.

```

[... ]
64 bytes from 172.17.24.1: icmp_seq=361 ttl=62 time=20.223 ms
64 bytes from 172.17.24.1: icmp_seq=362 ttl=62 time=20.427 ms
^C
--- 172.17.24.1 ping statistics ---
363 packets transmitted, 363 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.471/27.640/229.907/25.852 ms

```

Part 6: Configure Stateful Firewall Rules

In this part, you will configure stateful firewall rules and apply them to the service set.

Step 6.1

Configure a stateful firewall rule called *allow-internal-outbound* with a single term that accepts all traffic.

```

lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit services stateful-firewall rule allow-internal-outbound

[edit services stateful-firewall rule allow-internal-outbound]
lab@HongKong# set term allow-all then accept

```

Step 6.2

Configure the stateful firewall rule to match all traffic in the input direction.

```

[edit services stateful-firewall rule allow-internal-outbound]
lab@HongKong# set match-direction input

```

Step 6.3

Configure the *trust-untrust* service set to use the *allow-internal-outbound* stateful firewall rule.

```
[edit services stateful-firewall rule allow-internal-outbound]
lab@HongKong# up 2 edit service-set trust-untrust

[edit services service-set trust-untrust]
lab@HongKong# set stateful-firewall-rules allow-internal-outbound
```

Step 6.4

Commit your configuration.

```
[edit services service-set trust-untrust]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Part 7: Verify Stateful Firewall Operation

In this part, you will verify that the stateful firewall configuration you just configured is working correctly.

Step 7.1

From Sydney, attempt to ping the IP address 172.17.24.1 (an IP address on ISP B's network) from your virtual router. Allow the router to continue to ping so as to provide a steady flow of traffic for you to examine.

```
hongkong@Sydney> ping routing-instance HongKong-vr 172.17.24.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=62 time=5.069 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=62 time=20.620 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=62 time=20.499 ms
64 bytes from 172.17.24.1: icmp_seq=3 ttl=62 time=20.241 ms
64 bytes from 172.17.24.1: icmp_seq=4 ttl=62 time=20.238 ms
```

Step 7.2

On your router, examine the flow table.

```
lab@HongKong> show services stateful-firewall flows
Interface: sp-0/0/0, Service set: trust-untrust
```

Flow	State	Dir	Frm count
ICMP 192.168.200.2:47135 -> 172.17.24.1	Watch	I	7
NAT source 192.168.200.2:47135 -> 10.14.8.2:1025			
ICMP 172.17.24.1:1025 -> 10.14.8.2	Watch	O	7
NAT dest 10.14.8.2:1025 -> 192.168.200.2:47135			

Step 7.3

On Sydney, stop the ping you started in Step 7.1.


```
[...]
64 bytes from 172.17.24.1: icmp_seq=200 ttl=62 time=20.682 ms
64 bytes from 172.17.24.1: icmp_seq=201 ttl=62 time=40.586 ms
64 bytes from 172.17.24.1: icmp_seq=202 ttl=62 time=20.661 ms
64 bytes from 172.17.24.1: icmp_seq=203 ttl=62 time=20.489 ms
64 bytes from 172.17.24.1: icmp_seq=204 ttl=62 time=20.236 ms
^C
--- 172.17.24.1 ping statistics ---
205 packets transmitted, 205 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.470/25.738/206.318/24.054 ms
```

Part 8: Configure an ALG

In this part, we will demonstrate the operation and configuration of ALGs.

Step 8.1

Open an FTP session to 172.17.24.1 using the command **ftp routing-instance routing-instance 172.17.24.1**. Log in using your router's hostname (in all lower-case letters) as the username. Use the password provided by your instructor.

```
hongkong@Sydney> ftp routing-instance HongKong-vr 172.17.24.1
Connected to 172.17.24.1.
220 Sydney FTP server (Version 6.00LS) ready.
Name (172.17.24.1:hongkong): hongkong
331 Password required for hongkong.
Password:
230 User hongkong logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Step 8.2

Attempt to retrieve a directory listing using the **ls** command.

```
ftp> ls
500 Illegal PORT range rejected.
```

Question: Did the transfer succeed?

Answer: No, the transfer failed.

Step 8.3

On your router, examine the flow table.

```
lab@HongKong> show services stateful-firewall flows
Interface: sp-0/0/0, Service set: trust-untrust
Flow
```

Flow	State	Dir	Frm count
TCP 172.17.24.1:21 -> 10.14.8.2:1027	Forward	O	13
NAT dest 10.14.8.2:1027 -> 192.168.200.2:3459			
TCP 192.168.200.2:3459 -> 172.17.24.1:21	Forward	I	16
NAT source 192.168.200.2:3459 -> 10.14.8.2:1027			

Question: In what state are the flows? What does that indicate?

Answer: The flows are in the `Forward` state. This state indicates that no ALG is active, which explains why your FTP transfer failed.

Step 8.4

On your router, examine the stateful firewall statistics.

```
lab@HongKong> show services stateful-firewall statistics
```

Interface	Service set	Accept	Discard	Reject	Errors
sp-0/0/0	trust-untrust	2378	15	0	0

Question: Do you see any discards? If so, what do you think is the source of these discards?

Answer: Yes, you should see discards. The FTP packets were discarded.

Step 8.5

Add a term to the `allow-internal-outbound` stateful firewall rule that accepts all traffic using the `junos-ftp` ALG. Insert the term at the beginning of the rule. Commit your changes.

```
lab@HongKong> configure
```

```
Entering configuration mode
```

```
[edit]
```

```
lab@HongKong# edit services stateful-firewall rule allow-internal-outbound
```

```
[edit services stateful-firewall rule allow-internal-outbound]
```

```
lab@HongKong# set term allow-ftp from applications junos-ftp
```

```
[edit services stateful-firewall rule allow-internal-outbound]
```

```
lab@HongKong# set term allow-ftp then accept
```

```
[edit services stateful-firewall rule allow-internal-outbound]
```

```
lab@HongKong# insert term allow-ftp before term allow-all
```

```
[edit services stateful-firewall rule allow-internal-outbound]
```

```
lab@HongKong# commit and-quit
```

```
commit complete
```

```
Exiting configuration mode
```

Step 8.6

From Sydney, using the existing FTP session you opened in Step 8.1, attempt to retrieve a directory listing using the **ls** command.

```
ftp> ls
500 Illegal PORT range rejected.
```

Question: Did the transfer succeed? Why or why not?

Answer: No, the transfer failed because NAT rule changes do not affect existing sessions.

Step 8.7

Close the FTP session and reestablish the FTP session to session to 172.17.24.1 using the command **ftp routing-instance routing-instance 172.17.24.1**. Log in using your router's hostname (in all lower-case letters) as the username. Use the password provided by your instructor.

```
ftp> bye
221 Goodbye.
```

```
hongkong@Sydney> ftp routing-instance HongKong-vr 172.17.24.1
Connected to 172.17.24.1.
220 Sydney FTP server (Version 6.00LS) ready.
Name (172.17.24.1:hongkong): hongkong
331 Password required for hongkong.
Password:
230 User hongkong logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Step 8.8

Attempt to retrieve a directory listing using the **ls** command.

```
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for '/bin/ls'.
total 2
drwxr-xr-x  2 hongkong  staff  512 Nov  6 18:28 .ssh
226 Transfer complete.
```

Question: Did the transfer succeed?

Answer: Yes, the transfer should succeed.

Step 8.9

On your router, examine the flow table.

```
lab@HongKong> show services stateful-firewall flows
Interface: sp-0/0/0, Service set: trust-untrust
Flow                                     State   Dir      Frm count
TCP      172.17.24.1:21      ->      10.14.8.2:1025  Watch   O          16
      NAT dest      10.14.8.2:1025      ->      192.168.200.2:2940
TCP      192.168.200.2:2940  ->      172.17.24.1:21  Watch   I          19
      NAT source    192.168.200.2:2940  ->      10.14.8.2:1025
```

Question: In what state are the flows? What does that state indicate?

Answer: The flows are in the `Watch` state, which indicates that an ALG is active.

Step 8.10

On Sydney, close the FTP session and return to the router CLI.

```
ftp> bye
221 Goodbye.
```

```
hongkong@Sydney>
```

Part 9: Configure an Interface-Style Service Set

In this part, you will convert the next-hop-style service set to an interface-style service set. You will perform all this configuration on your router, unless otherwise indicated.

Step 9.1

On your router, delete the `trusted-vr` routing instance.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# delete routing-instances trusted-vr
```

Step 9.2

Delete Units 1 and 2 on the `sp-0/0/0` interface.

```
[edit]
lab@HongKong# delete interfaces sp-0/0/0 unit 1

[edit]
lab@HongKong# delete interfaces sp-0/0/0 unit 2
```

Step 9.3

Activate the OSPF configuration.

```
[edit]
lab@HongKong# activate protocols ospf
```

Step 9.4

Change the *trust-untrust* service set to be an interface-style service set that uses the *sp-0/0/0* services interface.

```
[edit]
lab@HongKong# edit services service-set trust-untrust

[edit services service-set trust-untrust]
lab@HongKong# delete next-hop-service

[edit services service-set trust-untrust]
lab@HongKong# set interface-service service-interface sp-0/0/0
```

Step 9.5

Reverse the match direction used for the *PAT-internal* NAT rule and the *allow-internal-outbound* stateful firewall rule.

```
[edit services service-set trust-untrust]
lab@HongKong# up 1 edit stateful-firewall rule allow-internal-outbound

[edit services stateful-firewall rule allow-internal-outbound]
lab@HongKong# set match-direction output

[edit services stateful-firewall rule allow-internal-outbound]
lab@HongKong# up 2 edit nat rule PAT-internal

[edit services nat rule PAT-internal]
lab@HongKong# set match-direction output
```

Question: Why is reversing the match direction necessary?

Answer: The next-hop-style and interface-style service sets view packet flow differently. Therefore, it is necessary to reverse the match direction used by all rules referenced in the service set.

Step 9.6

Apply the *trust-untrust* service set to both the logical interfaces that you use to connect to your ISPs. Commit the configuration.

```
[edit services nat rule PAT-internal]
lab@HongKong# top edit interfaces se-1/0/0 unit 101 family inet

[edit interfaces se-1/0/0 unit 101 family inet]
lab@HongKong# set service input service-set trust-untrust

[edit interfaces se-1/0/0 unit 101 family inet]
lab@HongKong# set service output service-set trust-untrust

[edit interfaces se-1/0/0 unit 101 family inet]
lab@HongKong# up 2 edit unit 201 family inet

[edit interfaces se-1/0/0 unit 201 family inet]
lab@HongKong# set service input service-set trust-untrust

[edit interfaces se-1/0/0 unit 201 family inet]
lab@HongKong# set service output service-set trust-untrust

[edit interfaces se-1/0/0 unit 201 family inet]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 9.7

On your router, examine the flow table and the extensive firewall statistics.

```
lab@HongKong> show services stateful-firewall flows

lab@HongKong> show services stateful-firewall statistics extensive
Interface: sp-0/0/0
Service set: trust-untrust
New flows:
  Accepts: 0, Discards: 0, Rejects: 0
Existing flows:
  Accepts: 0, Discards: 30, Rejects: 0
Drops:
  IP option: 0, TCP SYN defense: 0
  NAT ports exhausted: 0
Errors:
  IP: 0, TCP: 30
  UDP: 0, ICMP: 0
  Non-IP packets: 0, ALG: 0
IP errors:
  IP packet length inconsistencies: 0
  Minimum IP header length check failures: 0
  Reassembled packet exceeds maximum IP length: 0
  Illegal source address: 0
  Illegal destination address: 0
  TTL zero errors: 0, Illegal IP protocol number (0 or 255): 0
  Land attack: 0
  Non-IPv4 packets: 0, Bad checksum: 0
  Illegal IP fragment length: 0
  IP fragment overlap: 0
  IP fragment reassembly timeout: 0
```

```

Unknown: 0
TCP errors:
  TCP header length inconsistencies: 0
  Source or destination port number is zero: 0
  Illegal sequence number and flags combinations: 0
  SYN attack (multiple SYN messages seen for the same flow): 0
  First packet not a SYN message: 30
  TCP port scan (TCP handshake, RST seen from server for SYN): 0
  Bad SYN cookie response: 0
UDP errors:
  IP data length less than minimum UDP header length (8 bytes): 0
  Source or destination port number is zero: 0
  UDP port scan (ICMP error seen for UDP flow): 0
ICMP errors:
  IP data length less than minimum ICMP header length (8 bytes): 0
  ICMP error length inconsistencies: 0
  Duplicate ping sequence number: 0
  Mismatched ping sequence number: 0
ALG errors:
  BOOTP: 0, DCE-RPC: 0, DCE-RPC portmap: 0
  DNS: 0, Exec: 0, FTP: 0
  H323: 0, ICMP: 0, IIOP: 0
  Login: 0, NetBIOS: 0, NetShow: 0
  Real Audio: 0, RPC: 0, RPC portmap: 0
  RTSP: 0, Shell: 0, SIP: 0
  SNMP: 0, SQLNet: 0, TFTP: 0
  Traceroute: 0

```

Question: Are your BGP sessions listed? In what state are these sessions?

Answer: If they are listed, they might be in a Discard state because the stateful firewall did not observe the session startup. If you wait for the BGP session to reset, you will see it establish correctly.

Question: Why did the firewall discard packets?

Answer: In this example, it discarded 30 TCP packets because they were the first packets seen in a flow, but were not SYN packets. These packets are the BGP packets from the existing BGP session.

Step 9.8

Examine the BGP neighbor status. Notice the state and whether the InPkt counter is incrementing.

```
lab@HongKong> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State  Pending
inet.0         20        10        0          0        0        0        0
Peer           AS         InPkt    OutPkt    OutQ     Flaps Last Up/Dwn
State|#Active/Received/Damped...
172.17.39.17   65010     1468     1460      0        0    12:08:37 9/10/0
0/0/0
172.17.55.17   65030     1469     1459      0        0    12:08:33 1/10/0
0/0/0
```

Step 9.9

Wait until the flow table shows that the stateful firewall is accepting both your BGP sessions.

```
lab@HongKong> show services stateful-firewall flows
Interface: sp-0/0/0, Service set: trust-untrust
Flow                                     State  Dir  Frm count
TCP      172.17.39.17:179      ->    172.17.39.18:4385 Forward I      45
TCP      172.17.39.18:4385     ->    172.17.39.17:179 Forward O      42
TCP      172.17.55.17:179      ->    172.17.55.18:4749 Forward I      45
TCP      172.17.55.18:4749     ->    172.17.55.17:179 Forward O      47
```

Step 9.10

View your BGP session status. Ensure that your BGP sessions are reestablished.

```
lab@HongKong> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State  Pending
inet.0         20        10        0          0        0        0        0
Peer           AS         InPkt    OutPkt    OutQ     Flaps Last Up/Dwn
State|#Active/Received/Damped...
172.17.39.17   65010     1514     1500      0        2    11:10 9/10/0
0/0/0
172.17.55.17   65030     1519     1497      0        2    11:03 1/10/0
0/0/0
```

Step 9.11

From Sydney, attempt to ping the IP address 172.17.24.1 (an IP address on ISP B's network) from your virtual router. Allow the router to continue to ping so as to provide a steady flow of traffic for you to examine.

```
hongkong@Sydney> ping routing-instance HongKong-vr 172.17.24.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=62 time=106.532 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=62 time=11.715 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=62 time=20.495 ms
64 bytes from 172.17.24.1: icmp_seq=3 ttl=62 time=20.232 ms
64 bytes from 172.17.24.1: icmp_seq=4 ttl=62 time=20.369 ms
[...]
```

Step 9.12

On your router, examine the flow table.


```
lab@HongKong> show services stateful-firewall flows
```

```
Interface: sp-0/0/0, Service set: trust-untrust
```

Flow	State	Dir	Frm count
TCP 172.17.39.17:179 -> 172.17.39.18:4385	Forward	I	53
TCP 172.17.39.18:4385 -> 172.17.39.17:179	Forward	O	50
TCP 172.17.55.17:179 -> 172.17.55.18:4749	Forward	I	53
ICMP 192.168.200.2:42784 -> 172.17.24.1	Watch	O	20
NAT source 192.168.200.2:42784 -> 10.14.8.2:1028			
TCP 172.17.55.18:4749 -> 172.17.55.17:179	Forward	O	55
ICMP 172.17.24.1:1028 -> 10.14.8.2	Watch	I	20
NAT dest 10.14.8.2:1028 -> 192.168.200.2:42784			

Question: Does your router appear to be performing address translation correctly?

Answer: Yes, it should be performing address translation correctly.

Step 9.13

Purposely create asymmetric traffic flow. Configure the *reject-all* policy as the import policy for ISP A and the export policy for ISP C. Commit the configuration.

```
lab@HongKong> configure
```

```
Entering configuration mode
```

```
[edit]
```

```
lab@HongKong# edit protocols bgp group isp
```

```
[edit protocols bgp group isp]
```

```
lab@HongKong# set neighbor 172.17.39.17 import reject-all
```

```
[edit protocols bgp group isp]
```

```
lab@HongKong# set neighbor 172.17.55.17 export reject-all
```

```
[edit protocols bgp group isp]
```

```
lab@HongKong# commit and-quit
```

```
commit complete
```

```
Exiting configuration mode
```

Step 9.14

All incoming traffic should now use ISP A, and all outgoing traffic should now use ISP C. Verify this by examining the routing table on your router and on the *isp-b* routing instance on the Sydney router.

```
lab@HongKong> show route
```

```
inet.0: 23 destinations, 33 routes (23 active, 0 holddown, 10 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          *[Aggregate/130] 12:30:30
```

```

Discard
10.14.8.0/24      *[Aggregate/130] 12:30:30
                  Reject
10.14.8.1/32      *[Direct/0] 12:30:30
                  > via lo0.0
10.14.8.254/32    *[Direct/0] 12:30:30
                  > via lo0.0
10.14.9.0/24      *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65109 I
                  > to 172.17.55.17 via se-1/0/0.201
10.14.10.0/24     *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65110 I
                  > to 172.17.55.17 via se-1/0/0.201
10.14.11.0/24     *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65111 I
                  > to 172.17.55.17 via se-1/0/0.201
10.14.12.0/24     *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65112 I
                  > to 172.17.55.17 via se-1/0/0.201
10.14.13.0/24     *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65113 I
                  > to 172.17.55.17 via se-1/0/0.201
10.14.14.0/24     *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65114 I
                  > to 172.17.55.17 via se-1/0/0.201
10.14.15.0/24     *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65115 I
                  > to 172.17.55.17 via se-1/0/0.201
10.210.0.176/28   *[Direct/0] 2w2d 14:44:59
                  > via fe-0/0/0.0
10.210.0.177/32   *[Local/0] 2w2d 14:45:00
                  Local via fe-0/0/0.0
172.17.24.0/21    *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65020 I
                  > to 172.17.55.17 via se-1/0/0.201
172.17.32.0/20    *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 65020 65010 I
                  > to 172.17.55.17 via se-1/0/0.201
172.17.39.16/30   *[Direct/0] 12:30:30
                  > via se-1/0/0.101
172.17.39.18/32   *[Local/0] 12:30:30
                  Local via se-1/0/0.101
172.17.48.0/20    *[BGP/170] 00:00:29, localpref 100
                  AS path: 65030 I
                  > to 172.17.55.17 via se-1/0/0.201
172.17.55.16/30   *[Direct/0] 12:30:30
                  > via se-1/0/0.201
172.17.55.18/32   *[Local/0] 12:30:30
                  Local via se-1/0/0.201
192.168.200.0/24  *[Direct/0] 00:18:09
                  > via fe-2/0/1.200
192.168.200.1/32  *[Local/0] 00:18:09
                  Local via fe-2/0/1.200
224.0.0.5/32      *[OSPF/10] 00:18:09, metric 1

```

MultiRecv

```
__juniper_private1__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
10.0.0.1/32      *[Direct/0] 2w2d 14:45:35
                  > via lo0.16385
10.0.0.16/32     *[Direct/0] 2w2d 14:45:35
                  > via lo0.16385
```

(From Sydney:)

```
hongkong@Sydney> show route table isp-b 10.14.8.0
```

```
isp-b.inet.0: 16 destinations, 23 routes (16 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
10.14.8.0/24     *[BGP/160] 00:17:32, localpref 100
                  AS path: 65010 65108 I
                  > to 172.17.25.2 via lt-0/0/0.11
```

Step 9.15

From Sydney, verify that you can continue to ping the IP address 172.17.24.1 (an IP address on ISP B's network) from your virtual router.

```
hongkong@Sydney> ping routing-instance HongKong-vr 172.17.24.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=62 time=13.686 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=62 time=20.241 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=62 time=20.348 ms
^C
--- 172.17.24.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 13.686/18.092/20.348/3.116 ms
```

Step 9.16

On your router, remove the reject-all policies from the ISP A and ISP C neighbor configurations. Commit your configuration.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit protocols bgp group isp

[edit protocols bgp group isp]
lab@HongKong# delete neighbor 172.17.39.17 import

[edit protocols bgp group isp]
lab@HongKong# delete neighbor 172.17.55.17 export

[edit protocols bgp group isp]
lab@HongKong# commit and-quit
```

```
commit complete  
Exiting configuration mode
```



Tell your instructor that you have completed Lab 5.

Not For Reproduction

Lab 6

IPSec VPN (Detailed)

Overview

This lab explores IPSec VPN configuration. In this lab, you will continue using the virtual routers on the *Sydney* router to simulate hosts inside your network. You will establish VPN connectivity with a partner router to connect your two internal networks. You will first establish an IPSec-over-GRE tunnel using interface-style service sets. Then, you will configure an IPSec VPN (without GRE) using a next-hop-style service set.

This lab is available in two formats: a high-level format that is designed to make you think through each step and a detailed format that offers step-by-step instructions complete with sample output from most commands.

By completing this lab, you will perform the following tasks:

- Prepare the router.
- Configure an IPSec-over-GRE tunnel.
- Configure a service filter.
- Configure a stateful firewall filter.
- Configure a next-hop-style VPN.

Note

This lab requires you to work closely with a partner team to establish VPN tunnels. View the “Lab 6: IPSec VPNS” diagram to determine the team with which you will be forming VPN connections. You should coordinate your activities with this team to ensure the success of the lab.

Key Commands

```
clear bgp neighbor
clear services ipsec-vpn ipsec security-associations
ping
show bgp summary
show services ipsec-vpn ike security-associations
show services ipsec-vpn ipsec security-associations
show services ipsec-vpn ipsec statistics
show services stateful-firewall flows
```

Part 1: Prepare the Router

This lab begins where the last lab ended. In this part, you will deactivate your connection to ISP C and verify that you still have connectivity.

Step 1.1

Delete the unit on the serial interface that connects to ISP C.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# delete interfaces se-1/0/0 unit 201
```

Step 1.2

Delete the BGP neighbor configuration for ISP C. Commit your changes.

```
[edit]
lab@HongKong# delete protocols bgp group isp neighbor 172.17.55.17

[edit]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 1.3

View the routing table.

```
lab@HongKong> show route

inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Aggregate/130] 06:10:53
                   Discard
10.14.8.0/24       *[Aggregate/130] 06:10:53
                   Reject
10.14.8.1/32       *[Direct/0] 06:10:53
                   > via lo0.0
10.14.8.254/32     *[Direct/0] 06:10:53
                   > via lo0.0
```

```

10.14.9.0/24      *[BGP/170] 05:05:30, localpref 100
                  AS path: 65010 65109 I
                  > to 172.17.39.17 via se-1/0/0.101
10.210.8.176/28  *[Direct/0] 06:23:13
                  > via fe-0/0/0.0
10.210.8.177/32  *[Local/0] 06:23:15
                  Local via fe-0/0/0.0
172.17.24.0/21   *[BGP/170] 06:03:01, localpref 100
                  AS path: 65010 65020 I
                  > to 172.17.39.17 via se-1/0/0.101
172.17.32.0/20   *[BGP/170] 06:03:01, localpref 100
                  AS path: 65010 I
                  > to 172.17.39.17 via se-1/0/0.101
172.17.39.16/30  *[Direct/0] 06:03:09
                  > via se-1/0/0.101
172.17.39.18/32  *[Local/0] 06:10:53
                  Local via se-1/0/0.101
172.17.48.0/20   *[BGP/170] 06:03:01, localpref 100
                  AS path: 65010 65020 65030 I
                  > to 172.17.39.17 via se-1/0/0.101
172.17.55.16/30  *[Direct/0] 04:43:53
                  > via se-1/0/0.201
172.17.55.18/32  *[Local/0] 04:43:53
                  Local via se-1/0/0.201
192.168.200.0/24 *[Direct/0] 06:10:53
                  > via fe-2/0/1.200
192.168.200.1/32 *[Local/0] 06:10:53
                  Local via fe-2/0/1.200
224.0.0.5/32     *[OSPF/10] 06:10:53, metric 1
                  MultiRecv

__juniper_private1__inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.1/32      *[Direct/0] 06:23:48
                  > via lo0.16385
10.0.0.16/32     *[Direct/0] 06:23:48
                  > via lo0.16385

```

Step 1.4

Log in to the Sydney router using your router's name (all lower-case letters) as the username. Use the password provided by your instructor. This lab again uses the virtual router you configured on Sydney in the previous lab to simulate user traffic. Confirm that you can reach the Internet by attempting to ping the IP address 172.17.24.1 (an IP address on ISP B's network) from your routing instance.

Sydney (ttyp0)

login: hongkong

Password:

--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC

NOTE: This router is divided into many virtual routers used by different teams. Please only configure your own virtual router.

You must use 'configure private' to configure this router.

```
hongkong@Sydney> ping routing-instance HongKong-vr 172.17.24.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=62 time=25.753 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=62 time=20.207 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=62 time=20.424 ms
^C
--- 172.17.24.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 20.207/22.128/25.753/2.565 ms
```

Part 2: Configure an IPSec-over-GRE Tunnel

In this part, you will configure an IPSec-over-GRE tunnel to your partner router.

Step 2.1

Begin by configuring a GRE tunnel between the two routers. Use your router's interface to ISP A as the source address of the tunnel and use your partner router's interface to ISP A as the destination address of the tunnel. Configure an IP address of 192.168.25.X/24. Use the X value from the "Lab 5b: Firewall Policy and NAT" diagram.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit interfaces gr-0/0/0 unit 0

[edit interfaces gr-0/0/0 unit 0]
lab@HongKong# set tunnel source 172.17.39.18

[edit interfaces gr-0/0/0 unit 0]
lab@HongKong# set tunnel destination 172.17.39.22

[edit interfaces gr-0/0/0 unit 0]
lab@HongKong# set family inet address 192.168.25.200/24
```

Step 2.2

Delete the Layer 3 services configuration on the interface to the service provider.

```
[edit interfaces gr-0/0/0 unit 0]
lab@HongKong# top

[edit]
lab@HongKong# delete interfaces se-1/0/0 unit 101 family inet service
```

Step 2.3

Configure a static route for your partner's 192.168.X.0/24 network with a next hop of the tunnel interface. Commit the configuration.


```
[edit]
lab@HongKong# edit routing-options static

[edit routing-options static]
lab@HongKong# set route 192.168.201.0/24 next-hop gr-0/0/0.0

[edit routing-options static]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 2.4

Once your partner team has configured the GRE tunnel, try to ping your partner's GRE tunnel address.

```
lab@HongKong> ping 192.168.25.201
PING 192.168.25.201 (192.168.25.201): 56 data bytes
64 bytes from 192.168.25.201: icmp_seq=0 ttl=64 time=11.637 ms
64 bytes from 192.168.25.201: icmp_seq=1 ttl=64 time=10.365 ms
64 bytes from 192.168.25.201: icmp_seq=2 ttl=64 time=10.373 ms
64 bytes from 192.168.25.201: icmp_seq=3 ttl=64 time=10.393 ms
^C
--- 192.168.25.201 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 10.365/10.692/11.637/0.546 ms
```

Question: Is the ping successful?

Answer: Yes, it should be successful once your partner configures the tunnel.

Question: Do you need to assign an IP address to the tunnel interface to enable the processing of IP traffic on the tunnel?

Answer: No. You configure an IP address in this lab to aid in confirming the tunnel's operation; however, it is not necessary to configure an actual IP address. Simply configuring `family inet` enables the router to process IP traffic on the tunnel interface.

Step 2.5

From the Sydney virtual router, try to ping your partner's virtual router (192.168.X.2).

```
lab@HongKong> ping 192.168.201.2
PING 192.168.201.2 (192.168.201.2): 56 data bytes
64 bytes from 192.168.201.2: icmp_seq=0 ttl=63 time=13.324 ms
64 bytes from 192.168.201.2: icmp_seq=1 ttl=63 time=10.374 ms
64 bytes from 192.168.201.2: icmp_seq=2 ttl=63 time=10.378 ms
64 bytes from 192.168.201.2: icmp_seq=3 ttl=63 time=30.182 ms
```

```
^C
--- 192.168.201.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 10.374/16.064/30.182/8.239 ms
```

Question: Is the ping successful?

Answer: Yes, it should be successful.

Step 2.6

Next, you will configure an IPSec tunnel to secure the GRE traffic. Configure an IKE proposal called *psk_sha1_3des_ike_proposal* that specifies preshared keys, the SHA1 algorithm for authentication, and the 3DES encryption algorithm.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit services ipsec-vpn ike proposal psk_sha1_3des_ike_proposal

[edit services ipsec-vpn ike proposal psk_sha1_3des_ike_proposal]
lab@HongKong# set authentication-method pre-shared-keys

[edit services ipsec-vpn ike proposal psk_sha1_3des_ike_proposal]
lab@HongKong# set authentication-algorithm sha1

[edit services ipsec-vpn ike proposal psk_sha1_3des_ike_proposal]
lab@HongKong# set encryption-algorithm 3des-cbc
```

Step 2.7

Configure an IKE policy called *main_mode_ike_policy* that specifies IKE should use main mode, the *psk_sha1_3des_ike_proposal* proposal, and a preshared key of *test*.

```
[edit services ipsec-vpn ike proposal psk_sha1_3des_ike_proposal]
lab@HongKong# up 1 edit policy main_mode_ike_policy

[edit services ipsec-vpn ike policy main_mode_ike_policy]
lab@HongKong# set mode main

[edit services ipsec-vpn ike policy main_mode_ike_policy]
lab@HongKong# set proposals psk_sha1_3des_ike_proposal

[edit services ipsec-vpn ike policy main_mode_ike_policy]
lab@HongKong# set pre-shared-key ascii-text test
```

Step 2.8

Configure an IPSec proposal called *esp_sha1_3des_ipsec_proposal* that specifies the ESP protocol, the SHA1 algorithm for authentication, and the 3DES encryption algorithm.

```
[edit services ipsec-vpn ike policy main_mode_ike_policy]
lab@HongKong# up 2 edit ipsec proposal esp_sha1_3des_ipsec_proposal

[edit services ipsec-vpn ipsec proposal esp_sha1_3des_ipsec_proposal]
lab@HongKong# set protocol esp

[edit services ipsec-vpn ipsec proposal esp_sha1_3des_ipsec_proposal]
lab@HongKong# set authentication-algorithm hmac-sha1-96

[edit services ipsec-vpn ipsec proposal esp_sha1_3des_ipsec_proposal]
lab@HongKong# set encryption-algorithm 3des-cbc
```

Step 2.9

Configure an IPSec policy called *dynamic_ipsec_policy* that specifies that IPSec tunnels should use perfect forward secrecy with Diffie-Hellman Group 2 keys and that specifies the *esp_sha1_3des_ipsec_proposal* as the only IPSec proposal.

```
[edit services ipsec-vpn ipsec proposal esp_sha1_3des_ipsec_proposal]
lab@HongKong# up 1 edit policy dynamic_ipsec_policy

[edit services ipsec-vpn ipsec policy dynamic_ipsec_policy]
lab@HongKong# set perfect-forward-secrecy keys group2

[edit services ipsec-vpn ipsec policy dynamic_ipsec_policy]
lab@HongKong# set proposals esp_sha1_3des_ipsec_proposal
```

Step 2.10

Configure an IPSec VPN rule called *GRE-VPN* that encrypts traffic from the GRE tunnel's source address to the GRE tunnel's destination address. The rule should match traffic in the output direction. Traffic should be sent over a dynamic VPN using the *main_mode_ike_policy* IKE policy and the *dynamic_ipsec_policy* IPSec policy. The VPN endpoints will be the same ones used for the GRE tunnel.

```
[edit services ipsec-vpn ipsec policy dynamic_ipsec_policy]
lab@HongKong# up 2 edit rule GRE-VPN

[edit services ipsec-vpn rule GRE-VPN]
lab@HongKong# set match-direction output

[edit services ipsec-vpn rule GRE-VPN]
lab@HongKong# edit term gre-tunnel

[edit services ipsec-vpn rule GRE-VPN term gre-tunnel]
lab@HongKong# set from source-address 172.17.39.18

[edit services ipsec-vpn rule GRE-VPN term gre-tunnel]
lab@HongKong# set from destination-address 172.17.39.22

[edit services ipsec-vpn rule GRE-VPN term gre-tunnel]
lab@HongKong# set then dynamic ike-policy main_mode_ike_policy

[edit services ipsec-vpn rule GRE-VPN term gre-tunnel]
lab@HongKong# set then dynamic ipsec-policy dynamic_ipsec_policy
```

```
[edit services ipsec-vpn rule GRE-VPN term gre-tunnel]
lab@HongKong# set then remote-gateway 172.17.39.22
```

Step 2.11

Configure an interface-style service-set called *partner-vpn* that uses the *GRE-VPN* rule and specifies that the local VPN endpoint will be the same as the GRE tunnel's source address.

```
[edit services ipsec-vpn rule GRE-VPN term gre-tunnel]
lab@HongKong# up 3 edit service-set partner-vpn

[edit services service-set partner-vpn]
lab@HongKong# set interface-service service-interface sp-0/0/0

[edit services service-set partner-vpn]
lab@HongKong# set ipsec-vpn-rules GRE-VPN

[edit services service-set partner-vpn]
lab@HongKong# set ipsec-vpn-options local-gateway 172.17.39.18
```

Step 2.12

Configure the router to attempt to establish IPSec VPNs immediately, whether or not traffic is flowing over the VPN.

```
[edit services service-set partner-vpn]
lab@HongKong# up

[edit services]
lab@HongKong# set ipsec-vpn establish-tunnels immediately
```

Step 2.13

Configure the router to use the *partner-vpn* service set for traffic entering and exiting the service provider interface.

```
[edit services]
lab@HongKong# top edit interfaces se-1/0/0 unit 101 family inet

[edit interfaces se-1/0/0 unit 101 family inet]
lab@HongKong# set service input service-set partner-vpn

[edit interfaces se-1/0/0 unit 101 family inet]
lab@HongKong# set service output service-set partner-vpn
```



Wait for your partner team to complete this part before proceeding.

Part 3: Configure a Service Filter

In this part, you will determine whether you must write a service filter. You will then apply one, if necessary.

Step 3.1

Now, commit your configuration. Monitor the status of your router's BGP session to the ISP for a few minutes.

```
[edit interfaces se-1/0/0 unit 101 family inet]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

```
lab@HongKong> show bgp summary
Groups: 1 Peers: 1 Down peers: 1
Table          Tot Paths  Act Paths  Suppressed    History Damp State   Pending
inet.0          0          0          0             0       0      0        0
Peer           AS        InPkt      OutPkt      OutQ     Flaps  Last Up/Dwn
State|#Active/Received/Damped...
172.17.39.17    65010      1327       1330        0        1      1:08 Connect
```

Question: What happens to the BGP session?

Answer: Eventually, the hold timer expires and the BGP session drops.

Question: Why is this happening?

Answer: The service set contains no rule to process the BGP packets, so it is dropping the packets.

Step 3.2

Configure a service filter called *vpn-input-sf* that services all traffic from your partner's VPN endpoint to your VPN endpoint. All other traffic should be skipped. Create a service filter called *vpn-output-sf* that services all traffic from the GRE tunnel source to the GRE tunnel destination. All other traffic should be skipped.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit firewall family inet service-filter vpn-input-sf

[edit firewall family inet service-filter vpn-input-sf]
lab@HongKong# set term service-vpn from source-address 172.17.39.22

[edit firewall family inet service-filter vpn-input-sf]
lab@HongKong# set term service-vpn from destination-address 172.17.39.18
```

```
[edit firewall family inet service-filter vpn-input-sf]
lab@HongKong# set term service-vpn then service

[edit firewall family inet service-filter vpn-input-sf]
lab@HongKong# set term skip-all then skip

[edit firewall family inet service-filter vpn-input-sf]
lab@HongKong# up 1 edit service-filter vpn-output-sf

[edit firewall family inet service-filter vpn-output-sf]
lab@HongKong# set term service-vpn from source-address 172.17.39.18

[edit firewall family inet service-filter vpn-output-sf]
lab@HongKong# set term service-vpn from destination-address 172.17.39.22

[edit firewall family inet service-filter vpn-output-sf]
lab@HongKong# set term service-vpn then service

[edit firewall family inet service-filter vpn-output-sf]
lab@HongKong# set term skip-all then skip
```

Step 3.3

Apply the *vpn-input-sf* to the *partner-vpn* service set on input on the interface to the provider. Apply the *vpn-output-sf* to the *partner-vpn* service set on output on the interface to the provider. Commit the configuration.

```
[edit firewall family inet service-filter vpn-output-sf]
lab@HongKong# top edit interfaces se-1/0/0 unit 101 family inet service

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# set input service-set partner-vpn service-filter vpn-input-sf

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# set output service-set partner-vpn service-filter vpn-output-sf

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 3.4

Monitor the status of your router's BGP session to the ISP for a few minutes.

```
lab@HongKong> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 3 3 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Damped...
172.17.39.17 65010 1337 1340 0 1 3:06 3/3/0
0/0/0
```

Question: What happens to the BGP session?

Answer: Eventually, the BGP session reestablishes.

Step 3.5

Once both you and your partner's BGP sessions to ISP A reestablish, view the status of IKE negotiations.

```
lab@HongKong> show services ipsec-vpn ike security-associations
Remote Address  State          Initiator cookie  Responder cookie  Exchange type
172.17.39.22    Matured         ce18f897435ab5d1 ddcc9ffa2c161bb5  Main
```

Step 3.6

View the status of IPSec security associations.

```
lab@HongKong> show services ipsec-vpn ipsec security-associations
Service set: partner-vpn

Rule: GRE-VPN, Term: gre-tunnel, Tunnel index: 1
Local gateway: 172.17.39.18, Remote gateway: 172.17.39.22
Tunnel MTU: 1500
Direction SPI      AUX-SPI  Mode      Type      Protocol
inbound  2079167286  0        tunnel    dynamic   ESP
outbound 1903903392  0        tunnel    dynamic   ESP
```

Question: Are there IPSec security associations?

Answer: There should be IPSec security associations. If necessary, use the **clear services ipsec-vpn ipsec security-associations** command to cause the router to attempt to reestablish IPSec SAs.

Step 3.7

View the IPSec VPN traffic statistics. Record these statistics as a baseline to use in future steps.

```
lab@HongKong> show services ipsec-vpn ipsec statistics
```

PIC: sp-0/0/0, Service set: partner-vpn

```
ESP Statistics:
  Encrypted bytes:      224
  Decrypted bytes:      224
  Encrypted packets:    2
  Decrypted packets:    2
AH Statistics:
  Input bytes:          0
  Output bytes:         0
```

```

Input packets:          0
Output packets:         0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0

```

Step 3.8

From the Sydney virtual router, try to ping your partner's virtual router (192.168.X.2).

```

hongkong@Sydney> ping routing-instance HongKong-vr 192.168.201.2
PING 192.168.201.2 (192.168.201.2): 56 data bytes
64 bytes from 192.168.201.2: icmp_seq=0 ttl=62 time=15.266 ms
64 bytes from 192.168.201.2: icmp_seq=1 ttl=62 time=20.174 ms
64 bytes from 192.168.201.2: icmp_seq=2 ttl=62 time=20.173 ms
64 bytes from 192.168.201.2: icmp_seq=3 ttl=62 time=20.270 ms
64 bytes from 192.168.201.2: icmp_seq=4 ttl=62 time=20.170 ms
^C
--- 192.168.201.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 15.266/19.211/20.270/1.973 ms

```

Question: Is the ping successful?

Answer: Yes, it should be successful.

Step 3.9

On your router, view the IPsec VPN statistics, and compare these statistics to the values previously recorded.

```
lab@HongKong> show services ipsec-vpn ipsec statistics
```

```
PIC: sp-0/0/0, Service set: partner-vpn
```

```

ESP Statistics:
  Encrypted bytes:          784
  Decrypted bytes:          784
  Encrypted packets:        7
  Decrypted packets:        7
AH Statistics:
  Input bytes:              0
  Output bytes:             0
  Input packets:            0
  Output packets:           0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0

```


Question: Was any traffic encrypted?

Answer: Yes, traffic was encrypted. In fact, the virtual router on Sydney sent five packets and the IPSec statistics incremented by five packets.



Wait for your partner team to complete this part before continuing.

Part 4: Configure a Stateful Firewall Filter

In this part, you will configure a stateful firewall to run on the same interface where the VPN service set is configured. You will use the existing *trust-untrust* service set to perform stateful firewall and NAT services.

Step 4.1

Configure the *trust-untrust* service set on the interface to the provider. Because service sets are processed in the order they are listed in the configuration and the *partner-vpn* service set is listed first, VPN traffic should not be subjected to stateful firewall processing. Commit your configuration.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit interfaces se-1/0/0 unit 101 family inet service

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# set input service-set trust-untrust

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# set output service-set trust-untrust

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 4.2

Reset the BGP session to your provider. Wait for it to reestablish.

```
lab@HongKong> clear bgp neighbor
Cleared 1 connections
```

```
lab@HongKong> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State  Pending
inet.0         4          4          0          0        0        0
Peer           AS        InPkt    OutPkt    OutQ     Flaps Last Up/Dwn
State|#Active/Received/Damped...
172.17.39.17   65010     1395     1403      0        2      8:28 4/4/0
0/0/0
```

Question: When will your BGP session be disrupted like this?

Answer: Your BGP session will be disrupted in this manner only when adding a new stateful firewall service set. Because the stateful firewall did not see the TCP session start up, it blocks the packets of the existing session.

Question: What configuration change could you make to prevent the need to do this? What are the implications of that change?

Answer: You could use a service filter to block traffic for this BGP session from being processed by the stateful firewall service set. However, using a service filter allows traffic for your BGP session to pass unprotected, without the benefits the stateful firewall can provide.

Step 4.3

On your router, view the IPSec VPN traffic statistics. Record these statistics as a baseline to use in future steps.

```
lab@HongKong> show services ipsec-vpn ipsec statistics
```

```
PIC: sp-0/0/0, Service set: partner-vpn
```

```
ESP Statistics:
  Encrypted bytes:          784
  Decrypted bytes:          784
  Encrypted packets:        7
  Decrypted packets:        7
AH Statistics:
  Input bytes:              0
```

```

Output bytes:                0
Input packets:               0
Output packets:              0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0

```

Step 4.4

From the Sydney virtual router, try to ping your partner's virtual router (192.168.X.2).

```

hongkong@Sydney> ping routing-instance HongKong-vr 192.168.201.2
PING 192.168.201.2 (192.168.201.2): 56 data bytes
^C
--- 192.168.201.2 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss

```

Question: Is the ping successful?

Answer: No, it should not be successful.

Step 4.5

Examine the stateful firewall session table on your router.

```

lab@HongKong> show services stateful-firewall flows
Interface: sp-0/0/0, Service set: trust-untrust
Flow                                     State   Dir      Frm count
TCP      172.17.39.18:3321 -> 172.17.39.17:179 Forward  O          44
TCP      172.17.39.17:179 -> 172.17.39.18:3321 Forward  I          47
GRE      172.17.39.18:0   -> 172.17.39.22:0   Drop     I           3

```

Question: Do you see the GRE packets listed?

Answer: Yes, they should be listed. (If not, attempt pinging across the GRE tunnel again and then re-examine the session table.)

Question: Examine the source and destination addresses. Are these the incoming or outgoing GRE packets?

Answer: The source and destination IP addresses should indicate that these are the outgoing GRE packets.

Question: In what direction did the packet arrive at the stateful firewall?

Answer: It was receiving in the input direction.

Question: What action is the firewall applying to the GRE packets?

Answer: It is discarding them. In fact, it is discarding them before they even reach the output service set that encrypts them. (You can verify this by looking at the IPSec encryption statistics, which will not be incrementing.)

Step 4.6

Determine which stateful firewall rules are configured in the *trust-untrust* service set. View those stateful firewall rules.

```
lab@HongKong> show configuration services service-set trust-untrust
stateful-firewall-rules allow-internal-outbound;
nat-rules PAT-internal;
interface-service {
    service-interface sp-0/0/0;
}

lab@HongKong> show configuration services stateful-firewall rule
allow-internal-outbound
match-direction output;
term allow-ftp {
    from {
        applications junos-ftp;
    }
    then {
        accept;
    }
}
term allow-all {
    then {
        accept;
    }
}
```

Question: Does viewing the stateful firewall rules help explain the behavior you are seeing?

Answer: Yes. The firewall filter does not contain any rules to accept packets from the input direction.

Step 4.7

Configure a service filter called *no-gre* that skips outbound GRE traffic (GRE traffic from the tunnel's source address to the tunnel's destination address). All other traffic should be serviced.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit firewall family inet service-filter no-gre

[edit firewall family inet service-filter no-gre]
lab@HongKong# set term block-gre from source-address 172.17.39.18

[edit firewall family inet service-filter no-gre]
lab@HongKong# set term block-gre from destination-address 172.17.39.22

[edit firewall family inet service-filter no-gre]
lab@HongKong# set term block-gre from protocol gre

[edit firewall family inet service-filter no-gre]
lab@HongKong# set term block-gre then skip

[edit firewall family inet service-filter no-gre]
lab@HongKong# set term service-others then service
```

Step 4.8

Apply the *no-gre* service filter to the *trust-untrust* service set on input. Commit the configuration.

```
[edit firewall family inet service-filter no-gre]
lab@HongKong# top edit interfaces se-1/0/0 unit 101 family inet service

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# set input service-set trust-untrust service-filter no-gre

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Question: What effect will this configuration have?

Answer: It will prevent the GRE packets from being processed by the stateful firewall.

Question: Why is this change necessary?

Answer: As explained in the course, outbound GRE packets are processed through input filters on the next-hop output interface.



Wait for your partner team to complete the previous step before continuing.

Step 4.9

From the Sydney virtual router, try to ping your partner's virtual router (192.168.X.2).

```
hongkong@Sydney> ping routing-instance HongKong-vr 192.168.201.2
PING 192.168.201.2 (192.168.201.2): 56 data bytes
64 bytes from 192.168.201.2: icmp_seq=0 ttl=62 time=15.961 ms
64 bytes from 192.168.201.2: icmp_seq=1 ttl=62 time=20.194 ms
64 bytes from 192.168.201.2: icmp_seq=2 ttl=62 time=40.238 ms
^C
--- 192.168.201.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 15.961/25.464/40.238/10.589 ms
```

Question: Is the ping successful?

Answer: Yes, it should be successful.

Step 4.10

On your router, view the IPsec VPN statistics and compare these to the values previously recorded.

```
lab@HongKong> show services ipsec-vpn ipsec statistics
```

```
PIC: sp-0/0/0, Service set: partner-vpn
```

```
ESP Statistics:
```

Encrypted bytes:	1120
Decrypted bytes:	1120

```

Encrypted packets:          10
Decrypted packets:          10
AH Statistics:
  Input bytes:              0
  Output bytes:             0
  Input packets:            0
  Output packets:           0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0

```

Question: Was any traffic encrypted?

Answer: Yes, traffic was encrypted.



Wait for your partner team to complete this part before continuing.

Part 5: Configure a Next-Hop-Style VPN

In this part, you will delete the IPsec-over-GRE tunnel and configure a next-hop-style VPN to your partner.

Step 5.1

Delete the `gr-0/0/0` interface. Delete the services configuration on the interface to ISP A. Delete the `partner-vpn` service set. Delete the `GRE-VPN` IPsec VPN rule.

```

lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# delete interfaces gr-0/0/0

[edit]
lab@HongKong# delete interfaces se-1/0/0 unit 101 family inet service

[edit]
lab@HongKong# delete services service-set partner-vpn

[edit]
lab@HongKong# delete services ipsec-vpn rule GRE-VPN

```

Step 5.2

Configure the `sp-0/0/0` Unit 1 to be an outside services interface. Configure family `inet` on this unit. Configure Unit 2 to be an inside services interface. Configure family `inet` on this unit.

```
[edit]
lab@HongKong# set interfaces sp-0/0/0 unit 1 service-domain outside

[edit]
lab@HongKong# set interfaces sp-0/0/0 unit 1 family inet

[edit]
lab@HongKong# set interfaces sp-0/0/0 unit 2 service-domain inside

[edit]
lab@HongKong# set interfaces sp-0/0/0 unit 2 family inet
```

Step 5.3

Configure a new VPN rule called *next-hop-VPN* that encrypts traffic from any source address to any destination address. The rule should match traffic in the input direction. Traffic should be sent over a dynamic VPN using the *main_mode_ike_policy* IKE policy and the *dynamic_ipsec_policy* IPsec policy. The remote VPN endpoint will be your partner router's interface to ISP A.

```
[edit]
lab@HongKong# edit services ipsec-vpn rule next-hop-VPN

[edit services ipsec-vpn rule next-hop-VPN]
lab@HongKong# set match-direction input

[edit services ipsec-vpn rule next-hop-VPN]
lab@HongKong# edit term match-all

[edit services ipsec-vpn rule next-hop-VPN term match-all]
lab@HongKong# set then dynamic ike-policy main_mode_ike_policy

[edit services ipsec-vpn rule next-hop-VPN term match-all]
lab@HongKong# set then dynamic ipsec-policy dynamic_ipsec_policy

[edit services ipsec-vpn rule next-hop-VPN term match-all]
lab@HongKong# set then remote-gateway 172.17.39.22
```

Question: Why is the input match direction used?

Answer: Because this VPN rule will be used in a next-hop-style service set, traffic outbound from the local router will be considered input traffic.

Step 5.4

Configure a next-hop-style service set called *NH-partner-vpn* that uses the *next-hop-VPN* rule and specifies that the local VPN endpoint will be the router's interface to ISP A. The service set should use the inside and outside services interfaces defined earlier in this part.


```
[edit services ipsec-vpn rule next-hop-VPN term match-all]
lab@HongKong# up 3 edit service-set NH-partner-vpn

[edit services service-set NH-partner-vpn]
lab@HongKong# set ipsec-vpn-rules next-hop-VPN

[edit services service-set NH-partner-vpn]
lab@HongKong# set ipsec-vpn-options local-gateway 172.17.39.18

[edit services service-set NH-partner-vpn]
lab@HongKong# set next-hop-service outside-service-interface sp-0/0/0.1

[edit services service-set NH-partner-vpn]
lab@HongKong# set next-hop-service inside-service-interface sp-0/0/0.2
```

Step 5.5

Modify the next hop for the static route to your partner's 192.168.X.0/24 network. The next hop should now be the inside services interface used in the previous step.

```
[edit services service-set NH-partner-vpn]
lab@HongKong# top edit routing-options static route 192.168.201.0/24

[edit routing-options static route 192.168.201.0/24]
lab@HongKong# delete next-hop gr-0/0/0.0

[edit routing-options static route 192.168.201.0/24]
lab@HongKong# set next-hop sp-0/0/0.2
```

Step 5.6

Configure IKE traceoptions. Commit the configuration.

```
[edit routing-options static route 192.168.201.0/24]
lab@HongKong# top

[edit]
lab@HongKong# set services ipsec-vpn traceoptions flag ike

[edit]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```



Wait for your partner team to complete the previous step before continuing.

Step 5.7

On your router, view the IKE security association status.

```
lab@HongKong> show services ipsec-vpn ike security-associations
Remote Address  State          Initiator cookie  Responder cookie  Exchange type
172.17.39.22    Matured          e033890c1b62e0a1 d2d08f552e56999b Main
```

Step 5.8

On your router, view the IPsec security association status.

```
lab@HongKong> show services ipsec-vpn ipsec security-associations
Service set: NH-partner-vpn
```

```

Rule: next-hop-VPN, Term: match-all, Tunnel index: 1
Local gateway: 172.17.39.18, Remote gateway: 172.17.39.22
IPsec inside interface: sp-0/0/0.2, Tunnel MTU: 1500
  Direction SPI          AUX-SPI      Mode      Type      Protocol
  inbound   3936317276      0          tunnel   dynamic   ESP
  outbound  1836857081      0          tunnel   dynamic   ESP

```

Step 5.9

From the Sydney virtual router, try to ping your partner's virtual router (192.168.X.2).

```

hongkong@Sydney> ping routing-instance HongKong-vr 192.168.201.2
PING 192.168.201.2 (192.168.201.2): 56 data bytes
64 bytes from 192.168.201.2: icmp_seq=0 ttl=62 time=15.999 ms
64 bytes from 192.168.201.2: icmp_seq=1 ttl=62 time=20.632 ms
64 bytes from 192.168.201.2: icmp_seq=2 ttl=62 time=20.431 ms
64 bytes from 192.168.201.2: icmp_seq=3 ttl=62 time=20.152 ms
^C
--- 192.168.201.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 15.999/19.303/20.632/1.915 ms

```

Question: Is the ping successful?

Answer: Yes, it should be successful.

Step 5.10

On your router, view the IPsec VPN statistics. Because these statistics are tracked per service set, these statistics were reset when you activated the new service set.

```
lab@HongKong> show services ipsec-vpn ipsec statistics
```

```
PIC: sp-0/0/0, Service set: NH-partner-vpn
```

```

ESP Statistics:
  Encrypted bytes:          352
  Decrypted bytes:          352
  Encrypted packets:         4
  Decrypted packets:         4
AH Statistics:
  Input bytes:               0
  Output bytes:              0
  Input packets:             0
  Output packets:            0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0

```

Bad headers: 0, Bad trailers: 0

Step 5.11

View the IKE traceoptions output in the kmd log.

```
lab@HongKong> show log kmd | last
Nov  8 03:03:06 ike_qm_sa_reply: Selected proposal 0, and transform 0 for
protocol 0
Nov  8 03:03:06 ike_st_i_private: Start
Nov  8 03:03:06 ike_st_o_qm_hash_2: Start
Nov  8 03:03:06 ike_st_o_qm_sa_values: Start
Nov  8 03:03:06 ike_st_o_qm_nonce: Start
Nov  8 03:03:06 ike_policy_reply_qm_nonce_data_len: Start
Nov  8 03:03:06 ike_st_o_qm_optional_ke: Start
Nov  8 03:03:06 ike_st_o_qm_optional_ids: Start
Nov  8 03:03:06 ike_policy_reply_qm_local_id: Start
Nov  8 03:03:06 ike_policy_reply_qm_remote_id: Start
Nov  8 03:03:06 ike_st_qm_optional_id: Start
Nov  8 03:03:06 ike_st_qm_optional_id: Start
Nov  8 03:03:06 ike_st_o_qm_optional_responder_lifetime_n: Start
Nov  8 03:03:06 ike_st_o_private: Start
Nov  8 03:03:06 ike_st_o_encrypt: Marking encryption for packet
Nov  8 03:03:06 ike_encode_packet: Start, SA = { 0xe033890c 1b62e0a1 - d2d08f55
2e56999b } / a767ea65, nego = 0
Nov  8 03:03:06 ike_send_packet: Start, send SA = { e033890c 1b62e0a1 -
d2d08f55 2e56999b}, nego = 0, src = 172.17.39.18:500, dst = 172.17.39.22:500
Nov  8 03:03:06 ike_udp_callback: Packet ready in source :
Nov  8 03:03:06 ike_get_sa: Start, SA = { e033890c 1b62e0a1 - d2d08f55 2e56999b
} / a767ea65, remote = 172.17.39.22:500
Nov  8 03:03:06 ike_sa_find: Found SA = { e033890c 1b62e0a1 - d2d08f55 2e56999b
}
Nov  8 03:03:06 ike_decode_packet: Start
Nov  8 03:03:06 ike_decode_packet: Start, SA = { e033890c 1b62e0a1 - d2d08f55
2e56999b} / a767ea65, nego = 0
Nov  8 03:03:06 ike_st_i_encrypt: Check that packet was encrypted succeeded
Nov  8 03:03:06 ike_st_i_qm_hash_3: Start, hash[0..20] = b6798992 dee050e5 ...
Nov  8 03:03:06 ike_st_i_private: Start
Nov  8 03:03:06 172.17.39.18:500 (Responder) <-> 172.17.39.22:500 { e033890c
1b62e0a1 - d2d08f55 2e56999b [0] / 0xa767ea65 } QM; MESSAGE: Phase 2 connection
succeeded, Using PFS, group = 2
Nov  8 03:03:06 ike_qm_call_callback: MESSAGE: Phase 2 connection succeeded,
Using PFS, group = 2
Nov  8 03:03:06 172.17.39.18:500 (Responder) <-> 172.17.39.22:500 { e033890c
1b62e0a1 - d2d08f55 2e56999b [0] / 0xa767ea65 } QM; MESSAGE: SA[0][0] = ESP
3des, life = 0 kB/28800 sec, group = 2, tunnel, hmac-sha1-96, key len = 0, key
rounds = 0
Nov  8 03:03:06 ike_qm_call_callback: MESSAGE: SA[0][0] = ESP 3des, life = 0
kB/28800 sec, group = 2, tunnel, hmac-sha1-96, key len = 0, key rounds = 0
Nov  8 03:03:06 ike_st_o_qm_wait_done: Marking for waiting for done
Nov  8 03:03:06 ike_send_notify: Connected, SA = { e033890c 1b62e0a1 - d2d08f55
2e56999b}, nego = 0
Nov  8 03:03:56 ike_state_restart_packet: Start, restart packet SA = { e033890c
1b62e0a1 - d2d08f55 2e56999b}, nego = 0
Nov  8 03:03:56 ike_st_o_qm_done: Quick Mode negotiation done
```

```
Nov  8 03:03:56 ike_send_notify: Connected, SA = { e033890c 1b62e0a1 - d2d08f55
2e56999b}, nego = 0
Nov  8 03:03:56 ike_delete_negotiation: Start, SA = { e033890c 1b62e0a1 -
d2d08f55 2e56999b}, nego = 0
Nov  8 03:03:56 ike_free_negotiation_qm: Start, nego = 0
Nov  8 03:03:56 ike_free_negotiation: Start, nego = 0
Nov  8 03:03:56 ike_free_id_payload: Start, id type = 4
Nov  8 03:03:56 ike_free_id_payload: Start, id type = 4
Nov  8 03:03:56 ike_free_id_payload: Start, id type = 4
Nov  8 03:03:56 ike_free_id_payload: Start, id type = 4
```



Tell your instructor that you have completed Lab 6. If time allows, you may proceed with the optional step.

Step 5.12 (Optional)

With your partner teams, you can create various failure scenarios and view the output in the `kmd` log. Some suggestions include using stateless firewall filters to block IKE traffic in one direction only, mismatching the source and destination networks in the IPsec VPN rules, mismatching the IPsec proposals, and mismatching the IKE proposals.

Class of Service (Detailed)

Overview

This lab explores CoS configuration on JUNOS routers. It is available in two formats: a high-level format that is designed to make you think through each step and a detailed format that offers step-by-step instructions complete with sample output from most commands.

By completing this lab, you will perform the following tasks:

- Prepare the router.
- Configure queues and scheduler maps.
- Configure multifield classification.
- Verify the operation of the multifield classifier.
- Configure BA rewrite rules.
- Configure virtual channels.

Key Commands

```
ping
show class-of-service classifiers
show interfaces queues
show firewall counter
```

Part 1: Prepare the Router

You will be beginning this lab where the last lab ended. You will restore the NAT configuration to enable communication between the internal (192.168.X.X) addresses and the Internet. Additionally, you will add an extra address to your Sydney virtual router's Fast Ethernet interface. In later parts of this lab, you will configure multifield classifiers to place traffic from different source addresses into different forwarding classes. You will use the multiple addresses on the Fast Ethernet interface to test this classification.

Step 1.1

Modify the *trust-untrust* service set by removing the *allow-internal-outbound* stateful firewall rule.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit services service-set trust-untrust

[edit services service-set trust-untrust]
lab@HongKong# delete stateful-firewall-rules allow-internal-outbound
```

Step 1.2

Verify that the *trust-untrust* service set configuration includes only the *PAT-internal* NAT rule.

```
[edit services service-set trust-untrust]
lab@HongKong# show
nat-rules PAT-internal;
interface-service {
    service-interface sp-0/0/0;
}
```

Step 1.3

Configure the router to use the *trust-untrust* interface-style service set on the connection to ISP A.

```
[edit services service-set trust-untrust]
lab@HongKong# top edit interfaces se-1/0/0 unit 101 family inet service

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# set input service-set trust-untrust

[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# set output service-set trust-untrust
```

Step 1.4

Log in to the Sydney router using your router's name (all lower-case letters) as the username and password. This lab will again use the virtual router you configured on Sydney in the previous lab(s) to simulate user traffic.

Sydney (tty0)

login: hongkong

Password:

--- JUNOS 8.0R2.8 built 2006-09-29 09:22:36 UTC

NOTE: This router is divided into many virtual routers used by different teams. Please only configure your own virtual router.

You must use 'configure private' to configure this router.

hongkong@Sydney>

Step 1.5

On the Sydney router, modify the configuration of your fe-2/0/1 interface by adding the extra IP address shown on the "Lab 7: CoS" diagram. Also configure your fe-2/0/1 interface to use the *verify-rewrite* input firewall filter. Commit your changes.

hongkong@Sydney> **configure private**

warning: uncommitted changes will be discarded on exit

Entering configuration mode

[edit]

hongkong@Sydney# **edit interfaces fe-2/0/1 unit 200 family inet**

[edit interfaces fe-2/0/1 unit 200 family inet]

hongkong@Sydney# **set address 192.168.200.254/24**

[edit interfaces fe-2/0/1 unit 200 family inet]

hongkong@Sydney# **set filter input verify-rewrite**

[edit interfaces fe-2/0/1 unit 200 family inet]

hongkong@Sydney# **top**

[edit]

hongkong@Sydney# **commit and-quit**

commit complete

Exiting configuration mode

Part 2: Configure Queues and Scheduler Maps

By default, the router assigns all traffic to the best-effort or network-control forwarding classes. Before you can assign traffic to other forwarding classes, you must configure a scheduler map for each interface with schedulers for those forwarding classes. In this part, you will associate queues with forwarding classes and configure schedulers and a scheduler map that you can apply to all interfaces.

Use the following table to assist you in this part:.

Table 7-1: Forwarding Class Configuration

Queue	Forwarding Class	Bandwidth and Buffers Allocation (%)	Priority
0	<i>best-effort</i>	40	Low
1	<i>admin</i>	45	Medium-low
2	<i>voip</i>	10	High
3	<i>network-control</i>	5	Medium-high

Step 2.1

Configure the forwarding classes to map to queues as shown in Table 7-1.

```
[edit interfaces se-1/0/0 unit 101 family inet service]
lab@HongKong# top

[edit]
lab@HongKong# edit class-of-service forwarding-classes

[edit class-of-service forwarding-classes]
lab@HongKong# set queue 1 admin

[edit class-of-service forwarding-classes]
lab@HongKong# set queue 2 voip
```

Question: Must you define the best-effort and network-control forwarding classes or assign them to Queues 0 and 3?

Answer: No. These are the default assignments.

Step 2.2

Configure a scheduler for each forwarding class using the parameters shown in Table 7-1.

```
[edit class-of-service forwarding-classes]
lab@HongKong# up 1 edit schedulers best-effort-sched

[edit class-of-service schedulers best-effort-sched]
lab@HongKong# set buffer-size percent 40

[edit class-of-service schedulers best-effort-sched]
lab@HongKong# set transmit-rate percent 40

[edit class-of-service schedulers best-effort-sched]
lab@HongKong# set priority low
```



```
[edit class-of-service schedulers best-effort-sched]
lab@HongKong# up 1 edit admin-sched

[edit class-of-service schedulers admin-sched]
lab@HongKong# set buffer-size percent 45

[edit class-of-service schedulers admin-sched]
lab@HongKong# set transmit-rate percent 45

[edit class-of-service schedulers admin-sched]
lab@HongKong# set priority medium-low

[edit class-of-service schedulers admin-sched]
lab@HongKong# up 1 edit voip-sched

[edit class-of-service schedulers voip-sched]
lab@HongKong# set buffer-size percent 10

[edit class-of-service schedulers voip-sched]
lab@HongKong# set transmit-rate percent 10

[edit class-of-service schedulers voip-sched]
lab@HongKong# set priority high

[edit class-of-service schedulers voip-sched]
lab@HongKong# up 1 edit network-control-sched

[edit class-of-service schedulers network-control-sched]
lab@HongKong# set buffer-size percent 5

[edit class-of-service schedulers network-control-sched]
lab@HongKong# set transmit-rate percent 5

[edit class-of-service schedulers network-control-sched]
lab@HongKong# set priority medium-high
```

Step 2.3

Configure a scheduler map that associates each forwarding class with its scheduler.

```
[edit class-of-service schedulers network-control-sched]
lab@HongKong# up 2 edit scheduler-maps classroom-sched-map

[edit class-of-service scheduler-maps classroom-sched-map]
lab@HongKong# set forwarding-class best-effort scheduler best-effort-sched

[edit class-of-service scheduler-maps classroom-sched-map]
lab@HongKong# set forwarding-class admin scheduler admin-sched

[edit class-of-service scheduler-maps classroom-sched-map]
lab@HongKong# set forwarding-class voip scheduler voip-sched

[edit class-of-service scheduler-maps classroom-sched-map]
lab@HongKong# set forwarding-class network-control scheduler network-control-sched
```

Step 2.4

Configure the fe-2/0/1 and serial interface that connects to your ISP for per-unit scheduling.

```
[edit class-of-service scheduler-maps classroom-sched-map]
lab@HongKong# top edit interfaces
```

```
[edit interfaces]
lab@HongKong# set se-1/0/0 per-unit-scheduler
```

```
[edit interfaces]
lab@HongKong# set fe-2/0/1 per-unit-scheduler
```

Question: What are the limitations on using per-unit scheduling?

Answer: You can only configure per-unit schedulers on hardware that supports this feature. J-series routers support per-unit scheduling on all interfaces. Support varies by hardware on M-series routers.

Step 2.5

Assign the scheduler map to all configured interfaces. Do not forget to assign the scheduler map to the sp-0/0/0 interface.

```
[edit interfaces]
lab@HongKong# top edit class-of-service interfaces
```

```
[edit class-of-service interfaces]
lab@HongKong# set se-1/0/0 unit 101 scheduler-map classroom-sched-map
```

```
[edit class-of-service interfaces]
lab@HongKong# set fe-2/0/1 unit 200 scheduler-map classroom-sched-map
```

```
[edit class-of-service interfaces]
lab@HongKong# set sp-0/0/0 scheduler-map classroom-sched-map
```

Question: What negative results might you experience if you failed to assign a scheduler map to all interfaces (even logical interfaces, such as *sp-*, *gr-*, and *lt-* interfaces)?

Answer: The router would use the default scheduler for this traffic. The default scheduler contains no buffers for traffic in queues other than Queues 0 and 3. Therefore, traffic in queues other than Queues 0 and 3 might be dropped.

Step 2.6

Configure the router to shape traffic on the logical interface to ISP A to 768 kbps.

```
[edit class-of-service interfaces]
lab@HongKong# set se-1/0/0 unit 101 shaping-rate 768k
```

Part 3: Configure Multifield Classification

In this part, you will configure the router to place traffic in a forwarding class using a multifield classifier.

Step 3.1

Configure a firewall filter called *classify-traffic* to perform multifield classification. Create a term that places SIP traffic (5060/UDP and 5060/TCP) in the *voip* forwarding class.

```
[edit class-of-service interfaces]
lab@HongKong# top edit firewall filter classify-traffic

[edit firewall filter classify-traffic]
lab@HongKong# set term sip from protocol [ tcp udp ] port 5060

[edit firewall filter classify-traffic]
lab@HongKong# set term sip then forwarding-class voip

[edit firewall filter classify-traffic]
lab@HongKong# set term sip then accept
```

Step 3.2

Add a term to the *classify-traffic* firewall filter that places RTP traffic (16384–32767/UDP) in the *voip* forwarding class.

```
[edit firewall filter classify-traffic]
lab@HongKong# set term rtp from protocol udp port 16384-32767
```

```
[edit firewall filter classify-traffic]
lab@HongKong# set term rtp then forwarding-class voip
```

```
[edit firewall filter classify-traffic]
lab@HongKong# set term rtp then accept
```

Step 3.3

Add a term to the *classify-traffic* firewall filter that places traffic with a source address of 192.168.X.0/25 (where X is the value from the “Lab 5b: Firewall Policy and NAT” diagram) in the *admin* forwarding class.

```
[edit firewall filter classify-traffic]
lab@HongKong# set term admin from source-address 192.168.200.0/25
```

```
[edit firewall filter classify-traffic]
lab@HongKong# set term admin then forwarding-class admin
```

```
[edit firewall filter classify-traffic]
lab@HongKong# set term admin then accept
```

Step 3.4

Add a term to the *classify-traffic* firewall filter that accepts all traffic and places it in the default forwarding class.

```
[edit firewall filter classify-traffic]
lab@HongKong# set term accept-all then accept
```

Step 3.5

Configure the router to use the *classify-traffic* firewall filter to process traffic inbound on the fe-2/0/1 VLAN interface. Commit the configuration.

```
[edit firewall filter classify-traffic]
lab@HongKong# top edit interfaces fe-2/0/1 unit 200 family inet
```

```
[edit interfaces fe-2/0/1 unit 200 family inet]
lab@HongKong# set filter input classify-traffic
```

```
[edit interfaces fe-2/0/1 unit 200 family inet]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Part 4: Verify the Operation of the Multifield Classifier

In this part, you will generate traffic from your Sydney virtual router and ensure that it is being placed in the correct forwarding classes.

Step 4.1

View the statistics on the interface to the service provider using the **show interfaces queue** command. You should see per-queue traffic statistics. Record the output as baseline statistics.

```
lab@HongKong> show interfaces queue se-1/0/0.101
Logical interface se-1/0/0.101 (Index 74) (SNMP ifIndex 40)
Forwarding classes: 8 supported, 8 in use
Egress queues: 8 supported, 8 in use
Burst size: 0
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets      :                10      0 pps
    Bytes        :               845      0 bps
  Transmitted:
    Packets      :                10      0 pps
    Bytes        :               845      0 bps
    Tail-dropped packets :            0      0 pps
    RED-dropped packets :            0      0 pps
      Low        :            0      0 pps
      Medium-low :            0      0 pps
      Medium-high:            0      0 pps
      High       :            0      0 pps
    RED-dropped bytes :            0      0 bps
      Low        :            0      0 bps
      Medium-low :            0      0 bps
      Medium-high:            0      0 bps
      High       :            0      0 bps
Queue: 1, Forwarding classes: admin
  Queued:
    Packets      :                0      0 pps
    Bytes        :                0      0 bps
  Transmitted:
    Packets      :                0      0 pps
    Bytes        :                0      0 bps
    Tail-dropped packets :            0      0 pps
    RED-dropped packets :            0      0 pps
      Low        :            0      0 pps
      Medium-low :            0      0 pps
      Medium-high:            0      0 pps
      High       :            0      0 pps
    RED-dropped bytes :            0      0 bps
      Low        :            0      0 bps
      Medium-low :            0      0 bps
      Medium-high:            0      0 bps
      High       :            0      0 bps
Queue: 2, Forwarding classes: voip
  Queued:
    Packets      :                0      0 pps
    Bytes        :                0      0 bps
  Transmitted:
    Packets      :                0      0 pps
    Bytes        :                0      0 bps
```

```

Tail-dropped packets : 0 0 pps
RED-dropped packets : 0 0 pps
  Low : 0 0 pps
  Medium-low : 0 0 pps
  Medium-high : 0 0 pps
  High : 0 0 pps
RED-dropped bytes : 0 0 bps
  Low : 0 0 bps
  Medium-low : 0 0 bps
  Medium-high : 0 0 bps
  High : 0 0 bps
Queue: 3, Forwarding classes: network-control
Queued:
  Packets : 2 0 pps
  Bytes : 30 0 bps
Transmitted:
  Packets : 2 0 pps
  Bytes : 30 0 bps
Tail-dropped packets : 0 0 pps
RED-dropped packets : 0 0 pps
  Low : 0 0 pps
  Medium-low : 0 0 pps
  Medium-high : 0 0 pps
  High : 0 0 pps
RED-dropped bytes : 0 0 bps
  Low : 0 0 bps
  Medium-low : 0 0 bps
  Medium-high : 0 0 bps
  High : 0 0 bps

```

Step 4.2

Log in to the Sydney router and ping the IP address 172.17.24.1 (an IP address on ISP B's network), sourcing the ping from the 192.168.X.2 address.

```

hongkong@Sydney> ping routing-instance HongKong-vr source 192.168.200.2
172.17.24.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=62 time=19.029 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=62 time=20.661 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=62 time=20.442 ms
64 bytes from 172.17.24.1: icmp_seq=3 ttl=62 time=20.440 ms
64 bytes from 172.17.24.1: icmp_seq=4 ttl=62 time=20.446 ms
64 bytes from 172.17.24.1: icmp_seq=5 ttl=62 time=20.604 ms
64 bytes from 172.17.24.1: icmp_seq=6 ttl=62 time=20.447 ms
^C
--- 172.17.24.1 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.029/20.296/20.661/0.524 ms

```

Question: To what forwarding class should your router assign this traffic?

Answer: You router should assign it to the *admin* forwarding class.

Step 4.3

On your router, view the statistics on the interface to the service provider using the **show interfaces queue** command and compare it to the baseline statistics you recorded earlier. You should see that the statistics for Queue 1 incremented.

```
lab@HongKong> show interfaces queue se-1/0/0.101
Logical interface se-1/0/0.101 (Index 74) (SNMP ifIndex 40)
Forwarding classes: 8 supported, 8 in use
Egress queues: 8 supported, 8 in use
Burst size: 0
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets      :           16          0 pps
    Bytes        :        1244          0 bps
  Transmitted:
    Packets      :           16          0 pps
    Bytes        :        1244          0 bps
    Tail-dropped packets :           0          0 pps
    RED-dropped packets :           0          0 pps
      Low        :           0          0 pps
      Medium-low :           0          0 pps
      Medium-high:           0          0 pps
      High       :           0          0 pps
    RED-dropped bytes :           0          0 bps
      Low        :           0          0 bps
      Medium-low :           0          0 bps
      Medium-high:           0          0 bps
      High       :           0          0 bps
Queue: 1, Forwarding classes: admin
  Queued:
    Packets      :              7          0 pps
    Bytes        :         623          0 bps
  Transmitted:
    Packets      :              7          0 pps
    Bytes        :         623          0 bps
    Tail-dropped packets :           0          0 pps
    RED-dropped packets :           0          0 pps
      Low        :           0          0 pps
      Medium-low :           0          0 pps
      Medium-high:           0          0 pps
      High       :           0          0 pps
    RED-dropped bytes :           0          0 bps
      Low        :           0          0 bps
      Medium-low :           0          0 bps
      Medium-high:           0          0 bps
      High       :           0          0 bps
```

Queue: 2, Forwarding classes: voip

Queued:

Packets	:	0	0 pps
Bytes	:	0	0 bps

Transmitted:

Packets	:	0	0 pps
Bytes	:	0	0 bps
Tail-dropped packets	:	0	0 pps
RED-dropped packets	:	0	0 pps
Low	:	0	0 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	0	0 bps
Low	:	0	0 bps
Medium-low	:	0	0 bps
Medium-high	:	0	0 bps
High	:	0	0 bps

Queue: 3, Forwarding classes: network-control

Queued:

Packets	:	9	0 pps
Bytes	:	135	0 bps

Transmitted:

Packets	:	9	0 pps
Bytes	:	135	0 bps
Tail-dropped packets	:	0	0 pps
RED-dropped packets	:	0	0 pps
Low	:	0	0 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	0	0 bps
Low	:	0	0 bps
Medium-low	:	0	0 bps
Medium-high	:	0	0 bps
High	:	0	0 bps

Step 4.4

Log in to the Sydney router and simulate SIP traffic by opening a Telnet session to the IP address 172.17.24.1 on port 5060, sourcing the traffic from your virtual router.

```
hongkong@Sydney> telnet routing-instance HongKong-vr 172.17.24.1 port 5060
Trying 172.17.24.1...
telnet: connect to address 172.17.24.1: Connection refused
telnet: Unable to connect to remote host
```

Step 4.5

On your router, view the statistics on the interface to the service provider using the **show interfaces queue** command and compare it to the baseline statistics you recorded earlier. You should see that the statistics for Queue 2 incremented.

lab@HongKong> **show interfaces queue se-1/0/0.101**

Logical interface se-1/0/0.101 (Index 74) (SNMP ifIndex 40)

Forwarding classes: 8 supported, 8 in use

Egress queues: 8 supported, 8 in use

Burst size: 0

Queue: 0, Forwarding classes: best-effort

Queued:

Packets	:	20	0 pps
Bytes	:	1510	0 bps

Transmitted:

Packets	:	20	0 pps
Bytes	:	1510	0 bps
Tail-dropped packets	:	0	0 pps
RED-dropped packets	:	0	0 pps
Low	:	0	0 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	0	0 bps
Low	:	0	0 bps
Medium-low	:	0	0 bps
Medium-high	:	0	0 bps
High	:	0	0 bps

Queue: 1, Forwarding classes: admin

Queued:

Packets	:	7	0 pps
Bytes	:	623	0 bps

Transmitted:

Packets	:	7	0 pps
Bytes	:	623	0 bps
Tail-dropped packets	:	0	0 pps
RED-dropped packets	:	0	0 pps
Low	:	0	0 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	0	0 bps
Low	:	0	0 bps
Medium-low	:	0	0 bps
Medium-high	:	0	0 bps
High	:	0	0 bps

Queue: 2, Forwarding classes: voip

Queued:

Packets	:	1	0 pps
Bytes	:	65	0 bps

Transmitted:

Packets	:	1	0 pps
Bytes	:	65	0 bps
Tail-dropped packets	:	0	0 pps
RED-dropped packets	:	0	0 pps
Low	:	0	0 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps

```

RED-dropped bytes      :                0                0 bps
  Low                   :                0                0 bps
  Medium-low            :                0                0 bps
  Medium-high           :                0                0 bps
  High                  :                0                0 bps
Queue: 3, Forwarding classes: network-control
Queued:
  Packets               :                16                0 pps
  Bytes                 :               240                0 bps
Transmitted:
  Packets               :                16                0 pps
  Bytes                 :               240                0 bps
Tail-dropped packets   :                0                0 pps
RED-dropped packets    :                0                0 pps
  Low                   :                0                0 pps
  Medium-low            :                0                0 pps
  Medium-high           :                0                0 pps
  High                  :                0                0 pps
RED-dropped bytes      :                0                0 bps
  Low                   :                0                0 bps
  Medium-low            :                0                0 bps
  Medium-high           :                0                0 bps
  High                  :                0                0 bps

```

Part 5: Configure BA Rewrite Rules

In this part, you will configure your router to rewrite a BA marker based on the forwarding class. You will verify this configuration by sending traffic from your Sydney virtual router to your partner's Sydney virtual router and monitoring that traffic.

Step 5.1

Configure your router to use the default IP precedence rewrite rule for traffic outbound to your service provider. Also, configure it to apply this rule for traffic outbound on the inside services interface of your VPN (sp-0/0/0.2). Commit the configuration.

```

lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit class-of-service interfaces

[edit class-of-service interfaces]
lab@HongKong# set se-1/0/0 unit 101 rewrite-rules inet-precedence default

[edit class-of-service interfaces]
lab@HongKong# set sp-0/0/0 unit 2 rewrite-rules inet-precedence default

[edit class-of-service interfaces]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode

```

Question: What is the significance of applying this rewrite rule to the `sp-0/0/0.2` interface?

Answer: This rewrite rule will cause the router to rewrite the ToS bits in the IP header before traffic is encrypted.

Question: Will the BA marking be overwritten in transit?

Answer: In IPsec's tunnel mode (which is the mode Juniper Networks routers use), the IP ToS bits are preserved in transit. The IPsec RFC states that the de-encapsulating device should preserve the ToS bits from the encapsulated packet.

Step 5.2

From your virtual router on Sydney, ping `192.168.X.254` (where `X` is your partner router's `X` value from the "Lab 5b: Firewall Policy and NAT" diagram). Source your pings from the `fe-2/0/1` VLAN interface's `.2` address. These pings should not succeed because they are being discarded by a firewall filter on the other router.

```
hongkong@Sydney> ping routing-instance HongKong-vr source 192.168.200.2
192.168.201.254
PING 192.168.201.254 (192.168.201.254): 56 data bytes
^C
--- 192.168.201.254 ping statistics ---
7 packets transmitted, 0 packets received, 100% packet loss
```

Step 5.3

On Sydney, show the counters for the firewall filter `verify-rewrite`. This firewall filter includes filter-specific counters, so the router automatically generates a different identifier for each interface to which the filter is applied. The pattern is `filtername-interface-direction`. In the case of the Tokyo router (HongKong's partner router), the identifier is `verify-rewrite-fe-2/0/1.201-i` (The Tokyo-`vr` uses the `fe-2/0/1.201` interface, and the filter is applied in the inbound direction on that interface).

This firewall filter counts traffic from your Sydney virtual router to your partner's virtual router and keeps one counter per forwarding class. You should see the `queue1` counter increment.

```
hongkong@Sydney> show firewall filter verify-rewrite-fe-2/0/1.201-i
Filter: verify-rewrite-fe-2/0/1.201-i
Counters:

```

Name	Bytes	Packets
queue0-fe-2/0/1.201-i	0	0
queue1-fe-2/0/1.201-i	588	7
queue2-fe-2/0/1.201-i	0	0
queue3-fe-2/0/1.201-i	0	0

Step 5.4

From your virtual router on *Sydney*, simulate SIP traffic by opening a Telnet session to 192.168.X.254 (where X is your partner router's X value from the "Lab 5b: Firewall Policy and NAT" diagram). Your Telnet session should use port 5060. These packets are being discarded by a firewall filter on the other router, so the session will not succeed. After a few seconds, you can cancel the Telnet connection.

```
hongkong@Sydney> telnet routing-instance HongKong-vr 192.168.201.254 port 5060
Trying 192.168.201.254...
^C
```

Step 5.5

On *Sydney*, show the counters for the firewall filter *verify-rewrite*. This firewall filter counts traffic from your *Sydney* virtual router to your partner's virtual router and keeps one counter per forwarding class. You should see the *queue2* counter increment.

```
hongkong@Sydney> show firewall filter verify-rewrite-fe-2/0/1.201-i
Filter: verify-rewrite-fe-2/0/1.201-i
Counters:
Name                               Bytes      Packets
queue0-fe-2/0/1.201-i              0           0
queue1-fe-2/0/1.201-i             588         7
queue2-fe-2/0/1.201-i             180         3
queue3-fe-2/0/1.201-i              0           0
```

Part 6: Configure BA Classifiers

In this part, you will configure your router to classify packets based on a BA marker. You will verify this by sending traffic from your *Sydney* virtual router with different ToS settings.

Step 6.1

On your router, deactivate the multifield classifier on the *fe-2/0/1* VLAN interface.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# deactivate interfaces fe-2/0/1.200 family inet filter input
```

Step 6.2

Configure the router to use the default IP precedence classifier for traffic it receives on the *fe-2/0/1* VLAN interface. Commit the configuration.

```
[edit]
lab@HongKong# edit class-of-service interfaces

[edit class-of-service interfaces]
lab@HongKong# set fe-2/0/1 unit 200 classifiers inet-precedence default

[edit class-of-service interfaces]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```

Step 6.3

Use the command **show class-of-service classifier type inet-precedence** to view the mapping between ToS bits and forwarding classes.

```
lab@HongKong> show class-of-service classifier type inet-precedence
```

Classifier: ipprec-default, Code point type: inet-precedence, Index: 11

Code point	Forwarding class	Loss priority
000	best-effort	low
001	voip	low
010	best-effort	low
011	best-effort	low
100	best-effort	low
101	admin	low
110	network-control	low
111	network-control	high

Classifier: ipprec-compatibility, Code point type: inet-precedence, Index: 12

Code point	Forwarding class	Loss priority
000	best-effort	low
001	best-effort	high
010	best-effort	low
011	best-effort	high
100	best-effort	low
101	best-effort	high
110	network-control	low
111	network-control	high

Question: What is the difference between the default IP precedence classifier and the default classifier that the router uses when no other BA classifier is configured?

Answer: The default IP precedence classifier (ipprec-default) assigns traffic to the forwarding classes associated with Queues 0-3, using the default IP precedence values. The default classifier that the router uses when no other BA classifier is configured (ipprec-compatibility) assigns all traffic to the forwarding classes associated with Queues 0 or 3.

Step 6.4

View the statistics on the interface to the service provider using the **show interfaces queue** command. You should see per-queue traffic statistics. Record the output as baseline statistics.

```
lab@HongKong> show interfaces queue se-1/0/0.101
```

```
Logical interface se-1/0/0.101 (Index 74) (SNMP ifIndex 40)
```

```
Forwarding classes: 8 supported, 8 in use
```

Egress queues: 8 supported, 8 in use

Burst size: 0

Queue: 0, Forwarding classes: best-effort

Queued:

Packets	:	138	0 pps
Bytes	:	9357	520 bps

Transmitted:

Packets	:	138	0 pps
Bytes	:	9357	520 bps
Tail-dropped packets	:	0	0 pps
RED-dropped packets	:	0	0 pps
Low	:	0	0 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	0	0 bps
Low	:	0	0 bps
Medium-low	:	0	0 bps
Medium-high	:	0	0 bps
High	:	0	0 bps

Queue: 1, Forwarding classes: admin

Queued:

Packets	:	298	0 pps
Bytes	:	41654	0 bps

Transmitted:

Packets	:	298	0 pps
Bytes	:	41654	0 bps
Tail-dropped packets	:	0	0 pps
RED-dropped packets	:	0	0 pps
Low	:	0	0 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	0	0 bps
Low	:	0	0 bps
Medium-low	:	0	0 bps
Medium-high	:	0	0 bps
High	:	0	0 bps

Queue: 2, Forwarding classes: voip

Queued:

Packets	:	4	0 pps
Bytes	:	416	0 bps

Transmitted:

Packets	:	4	0 pps
Bytes	:	416	0 bps
Tail-dropped packets	:	0	0 pps
RED-dropped packets	:	0	0 pps
Low	:	0	0 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	0	0 bps
Low	:	0	0 bps
Medium-low	:	0	0 bps

```

      Medium-high      :                0                0 bps
      High             :                0                0 bps
Queue: 3, Forwarding classes: network-control
Queued:
  Packets             :                193                0 pps
  Bytes               :               2895               56 bps
Transmitted:
  Packets             :                193                0 pps
  Bytes               :               2895               56 bps
  Tail-dropped packets :                0                0 pps
  RED-dropped packets :                0                0 pps
    Low               :                0                0 pps
    Medium-low        :                0                0 pps
    Medium-high       :                0                0 pps
    High              :                0                0 pps
  RED-dropped bytes   :                0                0 bps
    Low               :                0                0 bps
    Medium-low        :                0                0 bps
    Medium-high       :                0                0 bps
    High              :                0                0 bps

```

Step 6.5

Log in to the Sydney router and ping the IP address 172.17.24.1 (an IP address on ISP B's network), sourcing the ping from your virtual router. You should use the **tos** argument to manually set the ToS field to 160. (This argument sets the IP precedence bits 101, which are associated with Queue 1 in the default IP precedence classifier.)

```

hongkong@Sydney> ping routing-instance HongKong-vr tos 160 172.17.24.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=62 time=18.797 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=62 time=11.753 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=62 time=20.175 ms
64 bytes from 172.17.24.1: icmp_seq=3 ttl=62 time=20.597 ms
64 bytes from 172.17.24.1: icmp_seq=4 ttl=62 time=20.435 ms
64 bytes from 172.17.24.1: icmp_seq=5 ttl=62 time=40.271 ms
^C
--- 172.17.24.1 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 11.753/22.005/40.271/8.726 ms

```

Step 6.6

On your router, view the statistics on the interface to the service provider using the **show interfaces queue** command and compare it to the baseline statistics you recorded earlier. You should see that the statistics for Queue 1 incremented.

```

lab@HongKong> show interfaces queue se-1/0/0.101
Logical interface se-1/0/0.101 (Index 74) (SNMP ifIndex 40)
Forwarding classes: 8 supported, 8 in use
Egress queues: 8 supported, 8 in use
Burst size: 0
Queue: 0, Forwarding classes: best-effort
Queued:
  Packets             :                147                0 pps
  Bytes               :               10006               0 bps

```

```

Transmitted:
  Packets      :          147          0 pps
  Bytes        :        10006          0 bps
  Tail-dropped packets :          0          0 pps
  RED-dropped packets :          0          0 pps
    Low        :          0          0 pps
    Medium-low  :          0          0 pps
    Medium-high :          0          0 pps
    High        :          0          0 pps
  RED-dropped bytes :          0          0 bps
    Low        :          0          0 bps
    Medium-low  :          0          0 bps
    Medium-high :          0          0 bps
    High        :          0          0 bps

```

Queue: 1, Forwarding classes: admin

```

Queued:
  Packets      :          304          0 pps
  Bytes        :        42188          0 bps
Transmitted:
  Packets      :          304          0 pps
  Bytes        :        42188          0 bps
  Tail-dropped packets :          0          0 pps
  RED-dropped packets :          0          0 pps
    Low        :          0          0 pps
    Medium-low  :          0          0 pps
    Medium-high :          0          0 pps
    High        :          0          0 pps
  RED-dropped bytes :          0          0 bps
    Low        :          0          0 bps
    Medium-low  :          0          0 bps
    Medium-high :          0          0 bps
    High        :          0          0 bps

```

Queue: 2, Forwarding classes: voip

```

Queued:
  Packets      :           4          0 pps
  Bytes        :         416          0 bps
Transmitted:
  Packets      :           4          0 pps
  Bytes        :         416          0 bps
  Tail-dropped packets :          0          0 pps
  RED-dropped packets :          0          0 pps
    Low        :          0          0 pps
    Medium-low  :          0          0 pps
    Medium-high :          0          0 pps
    High        :          0          0 pps
  RED-dropped bytes :          0          0 bps
    Low        :          0          0 bps
    Medium-low  :          0          0 bps
    Medium-high :          0          0 bps
    High        :          0          0 bps

```

Queue: 3, Forwarding classes: network-control

```

Queued:
  Packets      :          206          0 pps
  Bytes        :         3090          0 bps

```



```

Transmitted:
Packets           :                206                0 pps
Bytes             :               3090                0 bps
Tail-dropped packets :                0                0 pps
RED-dropped packets :                0                0 pps
  Low              :                0                0 pps
  Medium-low       :                0                0 pps
  Medium-high      :                0                0 pps
  High             :                0                0 pps
RED-dropped bytes  :                0                0 bps
  Low              :                0                0 bps
  Medium-low       :                0                0 bps
  Medium-high      :                0                0 bps
  High             :                0                0 bps

```

Step 6.7

Log in to the Sydney router and ping the IP address 172.17.24.1 (an IP address on ISP B's network), sourcing the ping from your virtual router. You should use the **tos** argument to manually set the ToS field to 32. (This argument sets the IP precedence bits 001, which are associated with Queue 2 in the default IP precedence classifier.)

```

hongkong@Sydney> ping routing-instance HongKong-vr tos 32 172.17.24.1
PING 172.17.24.1 (172.17.24.1): 56 data bytes
64 bytes from 172.17.24.1: icmp_seq=0 ttl=62 time=19.183 ms
64 bytes from 172.17.24.1: icmp_seq=1 ttl=62 time=20.709 ms
64 bytes from 172.17.24.1: icmp_seq=2 ttl=62 time=20.179 ms
64 bytes from 172.17.24.1: icmp_seq=3 ttl=62 time=20.182 ms
64 bytes from 172.17.24.1: icmp_seq=4 ttl=62 time=20.180 ms
^C
--- 172.17.24.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.183/20.087/20.709/0.496 ms

```

Step 6.8

On your router, view the statistics on the interface to the service provider using the **show interfaces queue** command and compare it to the baseline statistics you recorded earlier. You should see that the statistics for Queue 2 incremented.

```

lab@HongKong> show interfaces queue se-1/0/0.101
Logical interface se-1/0/0.101 (Index 74) (SNMP ifIndex 40)
Forwarding classes: 8 supported, 8 in use
Egress queues: 8 supported, 8 in use
Burst size: 0
Queue: 0, Forwarding classes: best-effort
Queued:
Packets           :                149                0 pps
Bytes             :               10139                0 bps
Transmitted:
Packets           :                149                0 pps
Bytes             :               10139                0 bps
Tail-dropped packets :                0                0 pps
RED-dropped packets :                0                0 pps
  Low              :                0                0 pps
  Medium-low       :                0                0 pps

```

```

    Medium-high      :                0                0 pps
    High             :                0                0 pps
    RED-dropped bytes :                0                0 bps
    Low              :                0                0 bps
    Medium-low       :                0                0 bps
    Medium-high      :                0                0 bps
    High             :                0                0 bps
Queue: 1, Forwarding classes: admin
  Queued:
    Packets          :                304                0 pps
    Bytes            :            42188                0 bps
  Transmitted:
    Packets          :                304                0 pps
    Bytes            :            42188                0 bps
    Tail-dropped packets :                0                0 pps
    RED-dropped packets :                0                0 pps
    Low              :                0                0 pps
    Medium-low       :                0                0 pps
    Medium-high      :                0                0 pps
    High             :                0                0 pps
    RED-dropped bytes :                0                0 bps
    Low              :                0                0 bps
    Medium-low       :                0                0 bps
    Medium-high      :                0                0 bps
    High             :                0                0 bps
Queue: 2, Forwarding classes: voip
  Queued:
    Packets          :                9                0 pps
    Bytes            :             861                0 bps
  Transmitted:
    Packets          :                9                0 pps
    Bytes            :             861                0 bps
    Tail-dropped packets :                0                0 pps
    RED-dropped packets :                0                0 pps
    Low              :                0                0 pps
    Medium-low       :                0                0 pps
    Medium-high      :                0                0 pps
    High             :                0                0 pps
    RED-dropped bytes :                0                0 bps
    Low              :                0                0 bps
    Medium-low       :                0                0 bps
    Medium-high      :                0                0 bps
    High             :                0                0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets          :                210                0 pps
    Bytes            :            3150                0 bps
  Transmitted:
    Packets          :                210                0 pps
    Bytes            :            3150                0 bps
    Tail-dropped packets :                0                0 pps
    RED-dropped packets :                0                0 pps
    Low              :                0                0 pps
    Medium-low       :                0                0 pps

```

Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	0	0 bps
Low	:	0	0 bps
Medium-low	:	0	0 bps
Medium-high	:	0	0 bps
High	:	0	0 bps

Part 7: Configure Virtual Channels

In this part, you will configure virtual channels to control the traffic you send to the Internet. You will configure one virtual channel for your VPN to Sydney and one for all remaining traffic.

Step 7.1

Define a virtual channel named *VPN-vc*. Define a virtual channel named *default-vc*.

```
lab@HongKong> configure
Entering configuration mode

[edit]
lab@HongKong# edit class-of-service

[edit class-of-service]
lab@HongKong# set virtual-channels VPN-vc

[edit class-of-service]
lab@HongKong# set virtual-channels default-vc
```

Step 7.2

Define a virtual channel group named *wan-vc-group*. It should use the scheduler map configured in Step 2.3 for both the virtual channels. It should shape the *VPN-vc* virtual channel to 384 kbps. It should use the *default-vc* virtual channel as the default virtual channel.

```
[edit class-of-service]
lab@HongKong# edit virtual-channel-groups wan-vc-group

[edit class-of-service virtual-channel-groups wan-vc-group]
lab@HongKong# set VPN-vc scheduler-map classroom-sched-map

[edit class-of-service virtual-channel-groups wan-vc-group]
lab@HongKong# set VPN-vc shaping-rate 384k

[edit class-of-service virtual-channel-groups wan-vc-group]
lab@HongKong# set default-vc scheduler-map classroom-sched-map

[edit class-of-service virtual-channel-groups wan-vc-group]
lab@HongKong# set default-vc default
```

Step 7.3

Remove the scheduler map from the interface to your provider. Configure the router to use the *wan-vc-group* for this interface.

```
[edit class-of-service virtual-channel-groups wan-vc-group]
lab@HongKong# top edit class-of-service interfaces

[edit class-of-service interfaces]
lab@HongKong# delete se-1/0/0 unit 101 scheduler-map

[edit class-of-service interfaces]
lab@HongKong# set se-1/0/0 unit 101 virtual-channel-group wan-vc-group
```

Step 7.4

Configure a firewall filter called *choose-vc*. Create a term that matches ESP and AH traffic with a source address of your router's interface to ISP A and a destination address of your partner router's interface to ISP A. This term should accept this traffic and place it in the *VPN-vc* virtual channel.

```
[edit class-of-service interfaces]
lab@HongKong# top edit firewall family inet filter choose-vc

[edit firewall family inet filter choose-vc]
lab@HongKong# set term VPN-traffic from source-address 172.17.39.18/32

[edit firewall family inet filter choose-vc]
lab@HongKong# set term VPN-traffic from destination-address 172.17.39.22/32

[edit firewall family inet filter choose-vc]
lab@HongKong# set term VPN-traffic from protocol [ esp ah ]

[edit firewall family inet filter choose-vc]
lab@HongKong# set term VPN-traffic then accept

[edit firewall family inet filter choose-vc]
lab@HongKong# set term VPN-traffic then virtual-channel VPN-vc
```

Step 7.5

Add a term to the *choose-vc* firewall filter that accepts all remaining traffic.

```
[edit firewall family inet filter choose-vc]
lab@HongKong# set term accept-all then accept
```

Step 7.6

Configure the router to use the *choose-vc* firewall filter as an output filter on the interface to ISP A. Commit your changes.

```
[edit firewall family inet filter choose-vc]
lab@HongKong# top edit interfaces se-1/0/0 unit 101

[edit interfaces se-1/0/0 unit 101]
lab@HongKong# set family inet filter output choose-vc

[edit interfaces se-1/0/0 unit 101]
lab@HongKong# commit and-quit
commit complete
Exiting configuration mode
```



Tell your instructor that you have completed Lab 7.

Not For Reproduction

Not For Reproduction

Branch Office (Optional) (Detailed)

Overview

This optional lab will incorporate topics examined in the past several labs into a single, comprehensive lab that incorporates Layer 3 services and CoS in ways they can be used for branch-office connectivity. You will need to work with your partner group to complete this lab. You can choose to do part, or all, of this lab. If you choose to do multiple portions of this lab, you should plan transitions between the topologies and attempt to perform them with minimum down time.

The *Lab 8a* diagram provides a diagram of branch-office connectivity using Frame Relay. You should configure an IGP for internal connectivity and should configure the central site to provide all Internet connectivity via NAT (using the router's ISP address). You should also configure CoS within the network.

The *Lab 8b* diagram provides a diagram of branch-office connectivity using VPNs. You should configure an IGP for internal connectivity. You could initially configure all traffic between the branch office and Internet to flow over the VPN through the central site and then transition that traffic to flow directly to the branch office (split tunneling). You will again need to configure NAT (using the router's ISP address).

The *Lab 8c* diagram provides a diagram of branch-office connectivity using Layer 2 VPNs backed up by IPSec VPNs. You should configure an IGP for internal connectivity and ensure that the routers prefer the Layer 2 VPN when it is available. You should configure split tunneling from each site. You should also configure each site so that it can provide Internet connectivity (via the Layer 2 VPN) to the other site, in case its local Internet connection becomes unavailable. You will again need to configure NAT (using the router's ISP address).

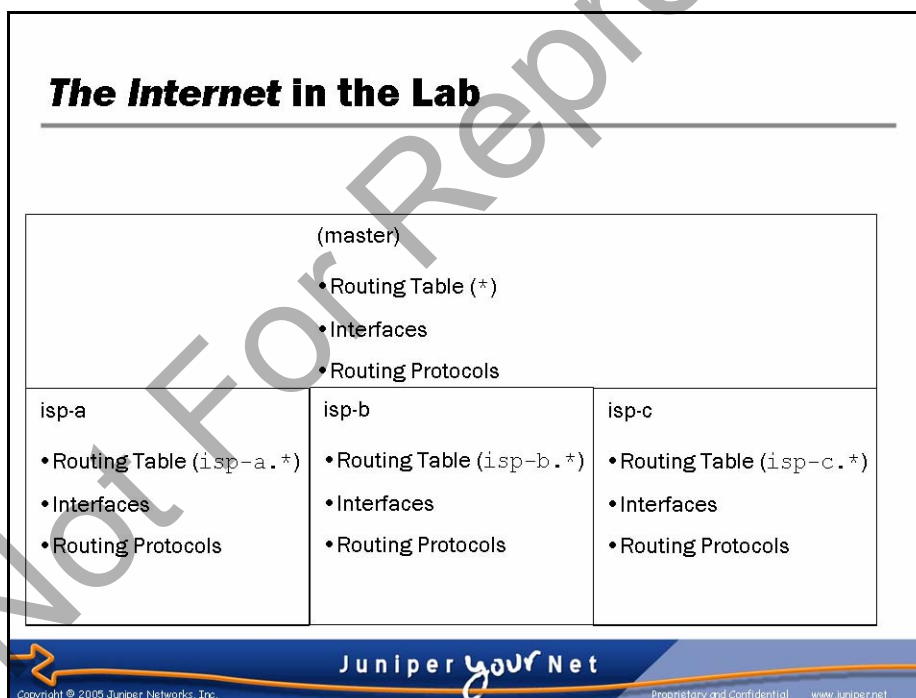
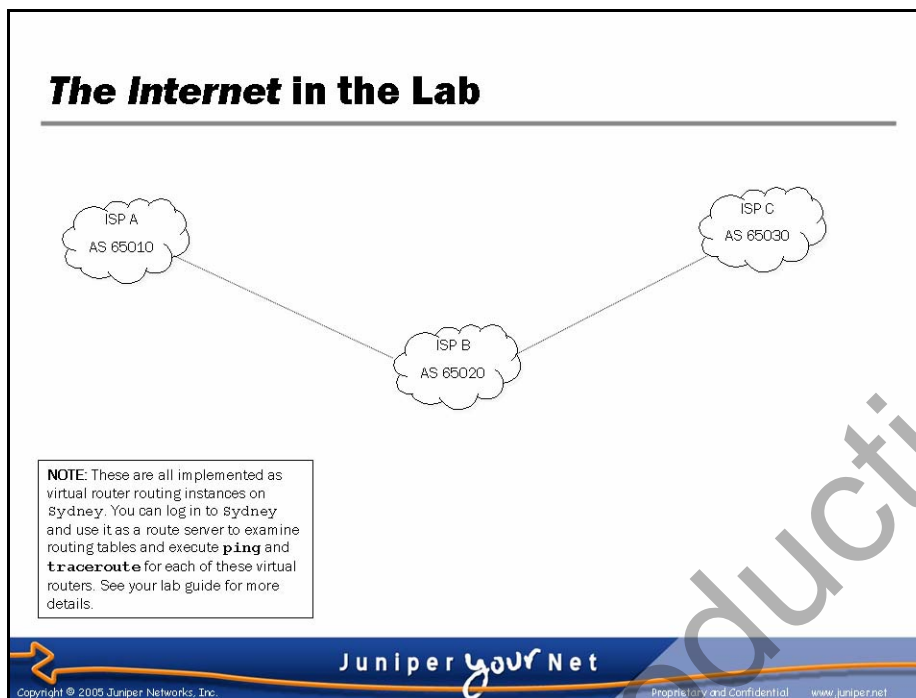
Not For Reproduction



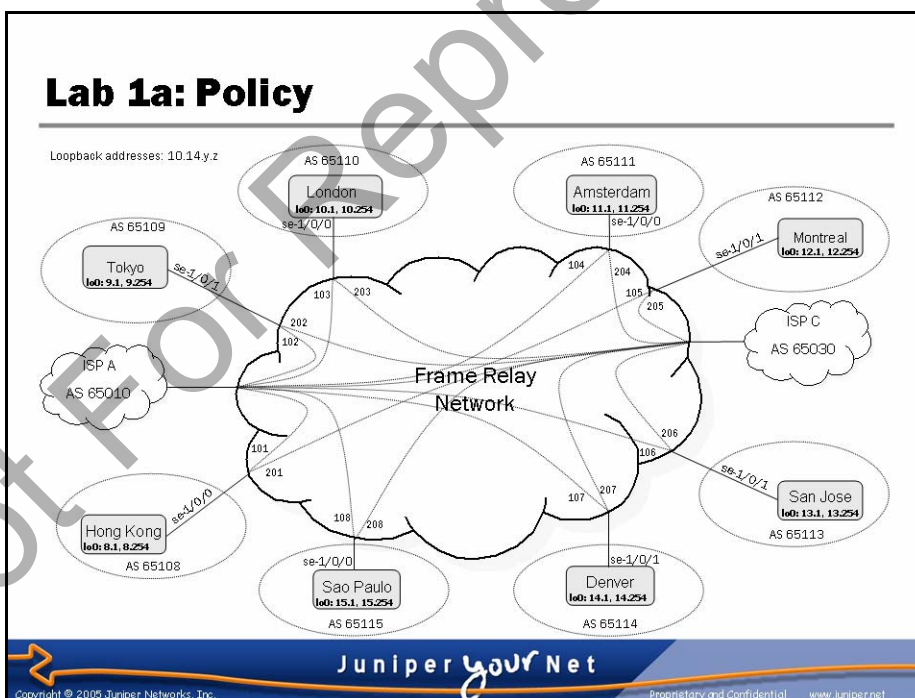
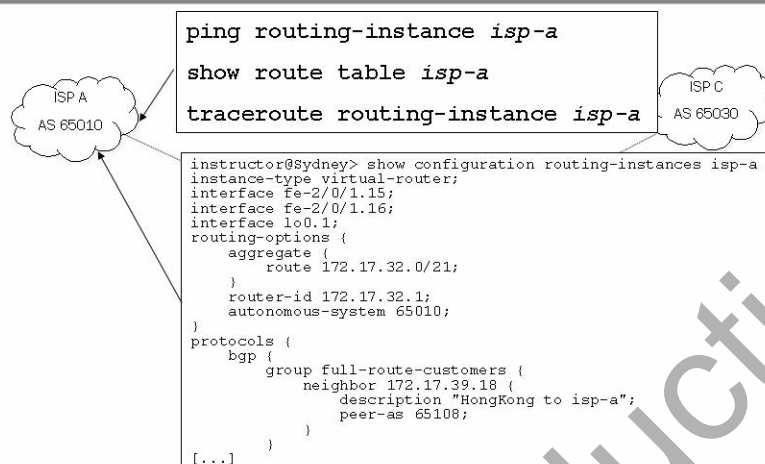
Advanced Juniper Networks Routing in the Enterprise

Appendix A: Lab Diagrams

Not For Reproduction

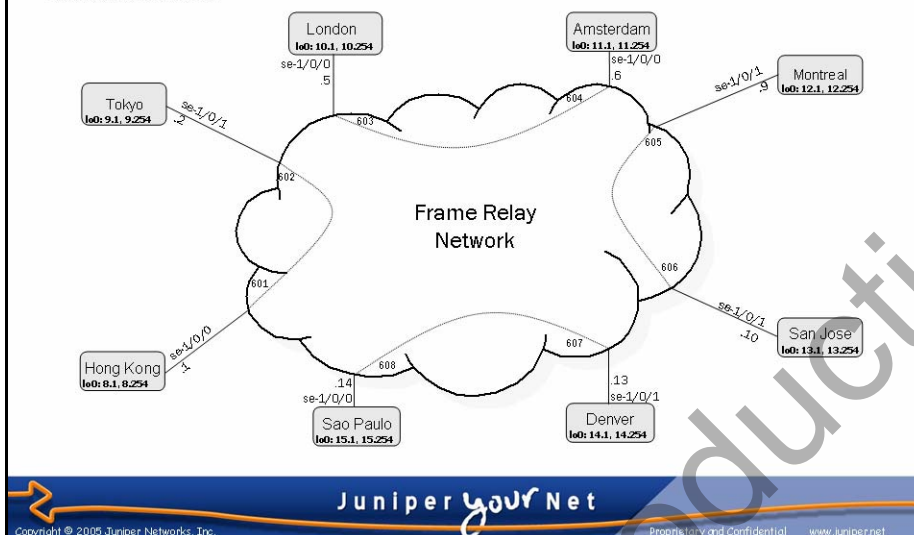


The Internet in the Lab

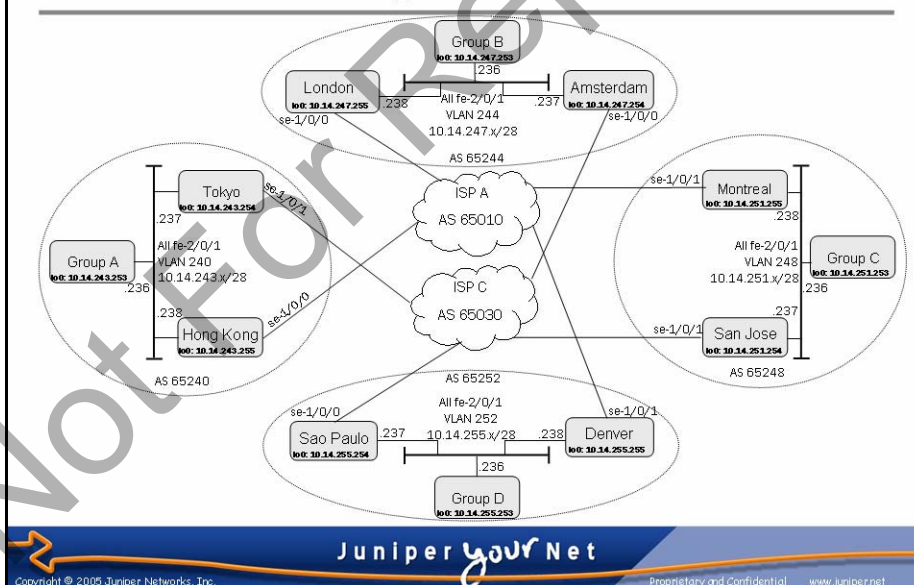


Lab 1b: Policy

Serial addresses: 192.168.25.x/30
Loopback addresses: 10.14.y.z

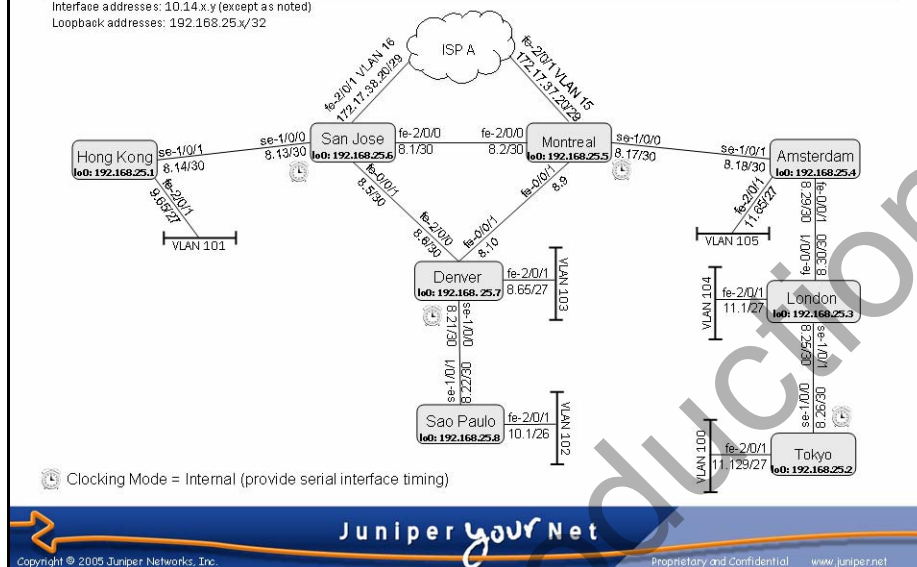


Lab 2: BGP Routing



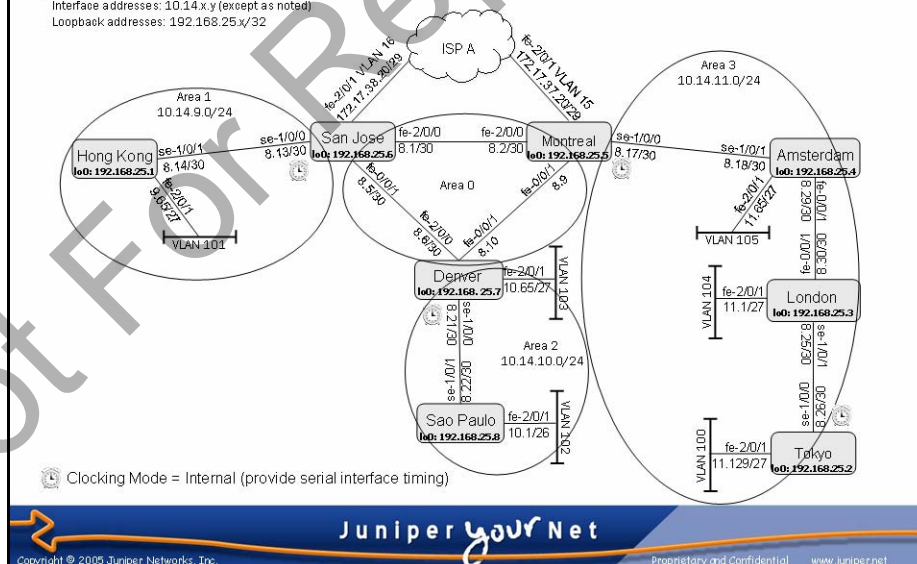
Lab 3a: IGP Conversion

Interface addresses: 10.14.x.y (except as noted)
Loopback addresses: 192.168.25.x/32



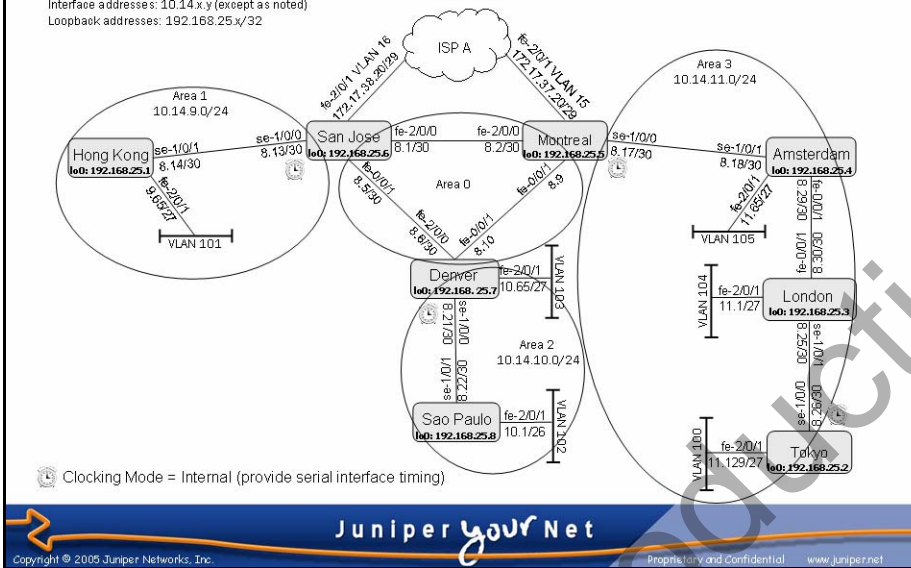
Lab 3b: IGP Conversion

Interface addresses: 10.14.x.y (except as noted)
Loopback addresses: 192.168.25.x/32



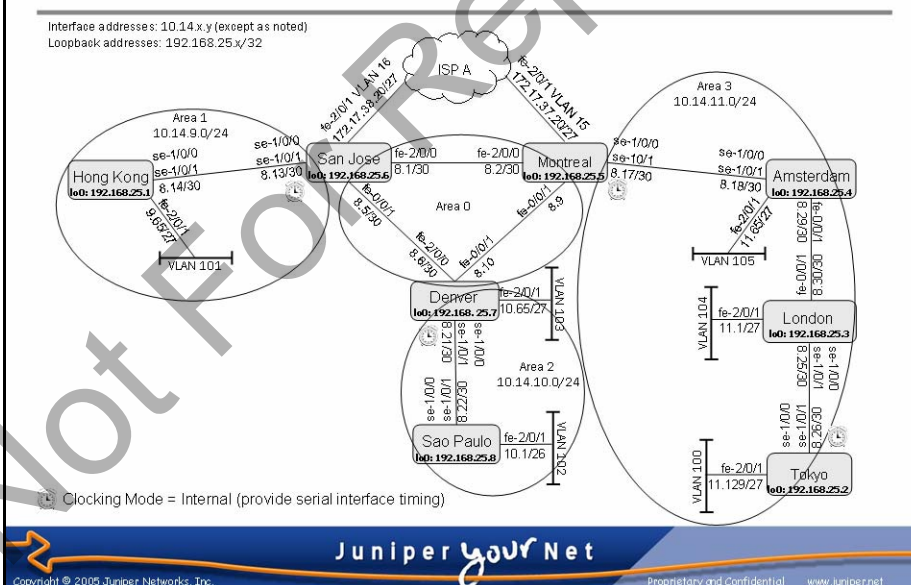
Lab 4a: CRTP

Interface addresses: 10.14.x.y (except as noted)
Loopback addresses: 192.168.25.x/32



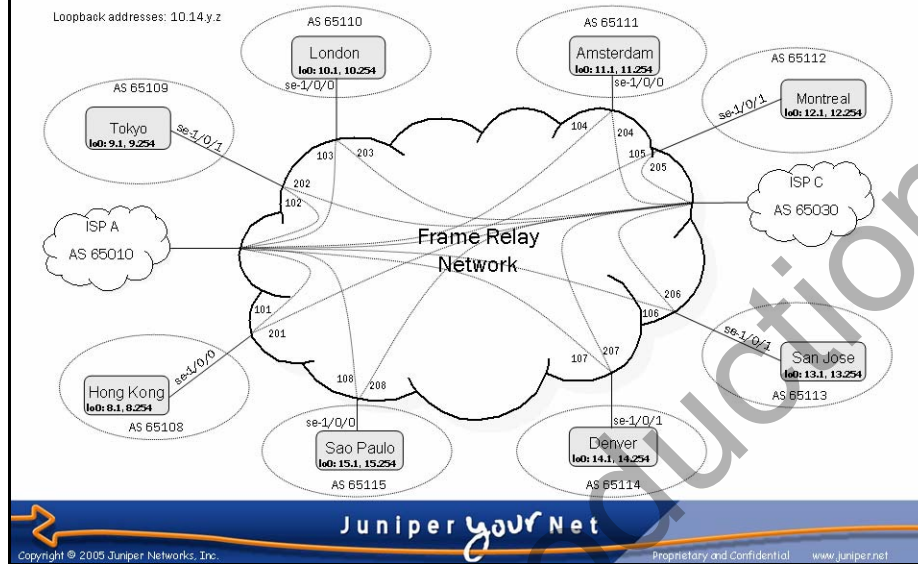
Lab 4b: Multilink PPP

Interface addresses: 10.14.x.y (except as noted)
Loopback addresses: 192.168.25.x/32



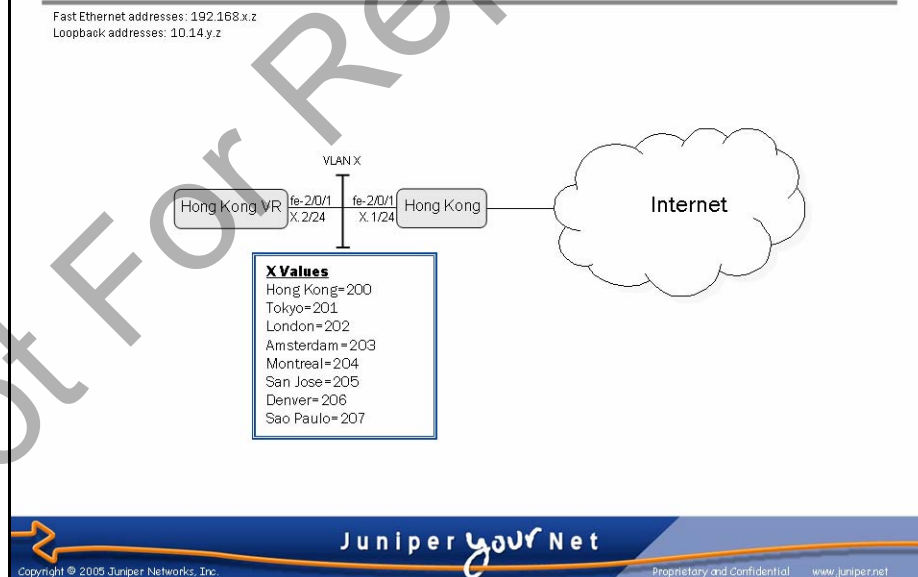
Lab 5a: Firewall Policy and NAT

Loopback addresses: 10.14.y.z



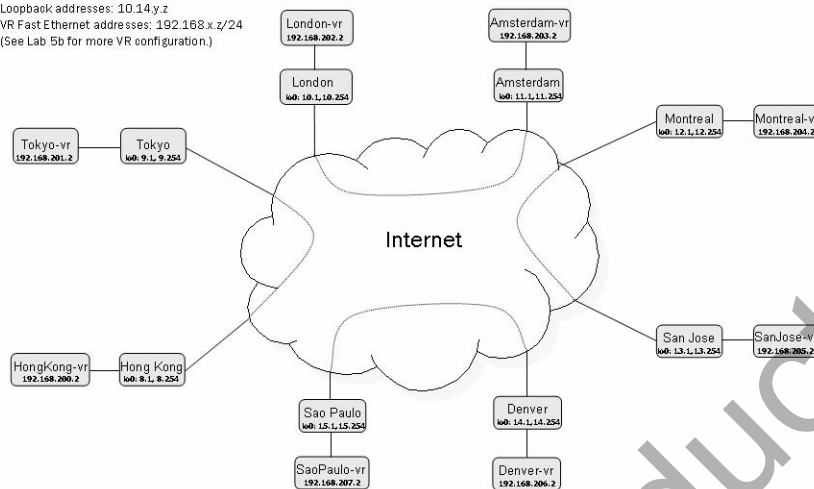
Lab 5b: Firewall Policy and NAT

Fast Ethernet addresses: 192.168.x.z
Loopback addresses: 10.14.y.z



Lab 6: IPSec VPNs

Loopback addresses: 10.14.y.z
VR Fast Ethernet addresses: 192.168.x.z/24
(See Lab 5b for more VR configuration.)



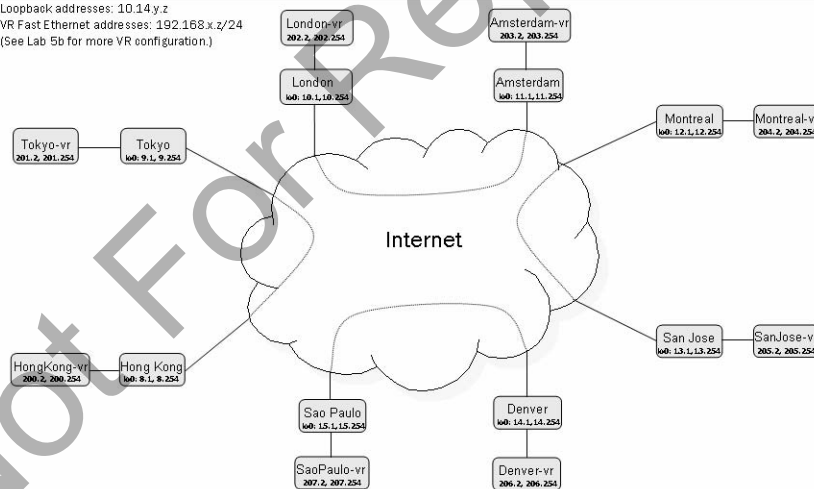
Juniper your Net

Copyright © 2005 Juniper Networks, Inc.

Proprietary and Confidential www.juniper.net

Lab 7: CoS

Loopback addresses: 10.14.y.z
VR Fast Ethernet addresses: 192.168.x.z/24
(See Lab 5b for more VR configuration.)



Juniper your Net

Copyright © 2005 Juniper Networks, Inc.

Proprietary and Confidential www.juniper.net

Lab 8a: Branch Office Connectivity

Fast Ethernet Addresses: 172.17.x.y/29
Provider Gateway: 172.17.x.z

Serial Interface and DLCI
(See Lab 1b diagram for interface, DLCI, and IP address assignments.)

Router	lo0	VLAN (w)	fe-2/0/1 (x.y)	Provider GW (x.z)
Hong Kong	192.168.200.1, /24	11	37.4	37.1
Tokyo	192.168.201.1, /24			
London	192.168.202.1, /24	13	37.12	37.9
Amsterdam	192.168.203.1, /24			
Montreal	192.168.204.1, /24	15	37.20	37.17
San Jose	192.168.205.1, /24			
Denver	192.168.206.1, /24	17	37.28	37.25
Sao Paulo	192.168.207.1, /24			

Juniper your Net

Copyright © 2005 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Not For Reproduction