Junos® Networking Technologies

DAY ONE: ADVANCED OSPF IN THE ENTERPRISE

Get ready for future growth and expansion by utilizing OSPF in your enterprise network. This book shows you how with rock-solid Junos OS techniques.

By Timothy Beaumont

DAY ONE:
ADVANCED OSPF IN THE ENTERPRISE

As your network grows, as your business takes on new employees, as more and more services are shared within your network environment, and as all the other forces at work in today's networking environment come into play, OSPF's ability to scale is key to the success of your enterprise environment.

As one of the most widely used routing protocols, OSPF has many advantages over other IGPs (Interior Gateway Protocols), with its ability to scale in a manageable way being one of the most important. OSPF also has the advantage of converging much faster than other enterprise level IGPs, such as EIGRP or RIP. This book shows you how to avoid common pitfalls when applying OSPF in Junos devices in an enterprise environment and how to prepare your network for future growth and expansion.

"*Day One: Advanced OSPF in the Enterprise* provides an easy to understand deep dive into the implementation and configuration of OSPF in an enterprise environment. It is an excellent resource for reference material."

Chris Jones, Senior Consultant, Accuvant Inc.

IT'S DAY ONE AND YOU HAVE A JOB TO DO, SO LEARN HOW TO:

- Configure and verify the use of areas within a Junos network.
- Use, configure, and verify the use of various types of stub areas within a Junos network.
- Connect Multiple Homogeneous OSPF domains within a network and verify that the network is operational.
- Connect Multiple Autonomous systems within an enterprise network using OSPF and BGP and verify its operational status.
- Configure and verify OSPF route redistribution within an enterprise network.
- Use and configure OSPF virtual links within the network.

Juniper Networks Books are singularly focused on network productivity and efficiency. Peruse the complete library at www.juniper.net/books.

Published by Juniper Networks Books

ISBN 978-1936779444

51600

9 781936 779444

07100148

JUNIPer NETWORKS

Junos® Networking Technologies

Day One: Advanced OSPF in the Enterprise

By Timothy Beaumont

JUNIPER NETWORKS

Published by Juniper Networks Books
Authors: Timothy Beaumont
Technical Reviewers: Jamie Panagos, Brandon Bennet, Trey Campbell
Editor in Chief: Patrick Ames
Editor and Proofer: Nancy Koerbel
J-Net Community Manager: Julie Wider

About the Author
Timothy Beaumont is a Senior Consulting Engineer with Juniper Networks. He has been with Juniper for over six years working with large enterprise customers.

Author's Acknowledgments
I would like to thank my family for giving me the time to write this book. And to Nancy Koerbel, my editor, for all the time she devoted to this project.

ISBN: 978-1-936779-44-4 (print)
Printed in the USA by Vervante Corporation.

ISBN: 978-1-936779-45-1 (ebook)

Version History: v1 January 2012
2 3 4 5 6 7 8 9 10 #7100148-en

This book is available in a variety of formats at: www.juniper.net/dayone.

Send your suggestions, comments, and critiques by email to dayone@juniper.net.

Welcome to Day One

This book is part of a growing library of *Day One* books, produced and published by Juniper Networks Books.

Day One books were conceived to help you get just the information that you need on day one. The series covers Junos OS and Juniper Networks networking essentials with straightforward explanations, step-by-step instructions, and practical examples that are easy to follow.

The *Day One* library also includes a slightly larger and longer suite of *This Week* books, whose concepts and test bed examples are more similar to a weeklong seminar.

You can obtain either series, in multiple formats:

- Download a free PDF edition at http://www.juniper.net/dayone.
- Get the ebook edition for iPhones and iPads from the iTunes Store. Search for Juniper Networks Books.
- Get the ebook edition for any device that runs the Kindle app (Android, Kindle, iPad, PC, or Mac) by opening your device's Kindle app and going to the Kindle Store. Search for Juniper Networks Books.
- Purchase the paper edition at either Vervante Corporation (www.vervante.com) or Amazon (www.amazon.com) for between $12-$28, depending on page length.
- Note that Nook, iPad, and various Android apps can also view PDF files.
- If your device or ebook app uses .epub files, but isn't an Apple product, open iTunes and download the .epub file from the iTunes Store. You can now drag and drop the file out of iTunes onto your desktop and sync with your .epub device.

What You Need to Know Before Reading This Book

Before reading this book, you should be familiar with the basic administrative functions of the Junos operating system, including the ability to work with operational commands and to read, understand, and change the Junos configuration.

This book makes a few assumptions about your network knowledge and understanding of the OSPF protocol and working with it in Junos. If you do not meet the following assumptions, portions of this book and its tutorials, may be difficult to comprehend:

- You should have experience with the configuration and operation of medium to large enterprise networks.
- You should have an understanding of IPv4 addressing and how to summarize IPv4 networks.
- You should also have a basic understanding of how to configure OSPF using Junos within a network.
- And finally, you should also have basic knowledge of how OSPF discovers neighbors, the different states of neighbor discovery, and the different LSAs used.

After Reading This Book, You'll Be Able To...

- Configure and verify the use of areas within a Junos network.
- Use, configure, and verify the use of various types of stub areas within a Junos network.
- Connect Multiple Homogeneous OSPF domains within a network and verify that the network is operational.
- Connect Multiple Autonomous systems within an enterprise network using OSPF and BGP and verify its operational status.
- Configure and verify OSPF route redistribution within an enterprise network.
- Use and configure OSPF virtual links within the network.

Chapter 1

OSPF in a Nutshell

The OSPF (Open Shortest Path First) algorithm is a link-state routing mechanism designed for medium to large enterprise networks. As one of the most widely used routing protocols, OSPF has many advantages over other IGPs (Interior Gateway Protocols),with its ability to scale in a manageable way being one of the most important. OSPF also has the advantage of converging much faster than other enterprise level IGPs, such as EIGRP or RIP. As your network grows, as your business takes on new employees, as more and more services are shared within your network environment, and as all the other forces at work in today's networking environment come into play, OSPF's ability to scale is key to the success of your enterprise environment.

Even though OSPF is widely used and well documented, operating it can be riddled with shortcomings if it's been implemented improperly. The pitfalls that occur are usually centered around the improper use of areas, suboptimal summarization, and large changes in network topology, such as additions of new networks within the enterprise network, going unchecked.

This book addresses pitfalls that arise when applying OSPF in Junos devices in an enterprise environment. Before you roll up your sleeves and get to work, however, you might want to read the overview of the basic elements of OSPF in an enterprise environment provided in this chapter. Subsequent chapters reference these elements extensively and a quick sync with OSPF basics should, in the long run, save you time and eliminate uncertainty.

If you're coming from IOS or the operating system of another networking vendor, such as Cisco IOS, the quick review in this chapter should help you grasp a Junos-centric view of OSPF.

Let's begin with the basics. OSPF is a link-state routing protocol based on the router's full knowledge of the entire network, which means the router must have a complete understanding of the network topology within the area. Each router keeps track of its own link-states and notifies all other connected routers if any change is made. Each router also has a database of connections or links from every other router in the network, from a link perspective. In large networks, these databases can become unmanageable if every router has information about the database of every other router, and it also makes the shortest path first (SPF) calculations inefficient when the router builds the routing table. To alleviate this problem, OSPF networks can be grouped into logical network areas, which are discussed in further detail in the next section.

Areas and Area Types

OSPF areas are independent entities, and all routers within an area must have identical link-state databases. The link-state databases of separate OSPF areas will differ from each other. OSPF has a rule requiring all areas to connect to a single contiguous backbone Area 0. Therefore the OSPF Area 0 will always exist in every OSPF network with more than one area.

There are three types of OSPF areas:

1. OSPF Backbone Area 0: The one constant of OSPF is there must always be a backbone Area 0.

2. OSPF Regular area: An area that is not Area 0, but is attached to Area 0. All routers within this area are aware of each other's link-state database.

3. Stub Area: An area that is not Area 0, and is not flooded with AS-external LSAs making a smaller OSPF LSA database, and therefore, a smaller routing table.

Adjacency

OSPF, as stated before, is a link-state protocol that sends information about its links to other routers in the network. In order for OSPF to send this information, it first must have an adjacency. Adjacency is a bidirectional communication that is kept alive by OSPF hello packets. The hello packets are sent at predetermined intervals. The intervals are configurable, but must agree with both sides of the adjaceny. The default intervals vary depending on the type of network the hello packets are communicating on. If the hellos are not seen for a period of time, the peering relationship will be removed. The defaults are listed in Table 1.1 below:

Table 1.1 Default Intervals for Adjacencies

Network type	Hello frequency	Dead timer
Broadcast	10 sec	40 sec
NBMA	30 sec	120 sec
Point-to-point	10 sec	40 sec
Point-to-multipoint	30 sec	120 sec

An adjacency can be in one of a number of different states listed in Table 1.2 during its life cycle.

Table 1.2 Adjacency States

Neighbor state	Description
Down	Beginning state, no peering
Attempt	NMBA only, designates a sent hello
Init	Hello packet was received
2-way	Bi-directional communication with neighbor
Exstart	Starting the ability to exchange database information
Exchange	Sharing of databases
Loading	Exchanging of the LSAs
Full	Complete adjacency

LSAs and LSAs Types

After OSPF has full adjacency with its neighbors, routers exchange their knowledge of the network as each one sees it. This process begins when a router receives a *link-state update packet* (LSU), or a *link-state request packet* (LSR). The router begins flooding *link-state advertisements* (LSAs) to its neighbors, and the neighbors also flood these LSAs to their other neighbors until the entire area is flooded. This synchronizes the LSAs database, allowing the routing table to be populated. The LSAs types and descriptions are listed in Table 1.3.

Table 1.3 LSAs Types and Descriptions

LSAs Type	Name	Description
Type-1	Router-LSA	States of the routers interfaces and information on them.
Type-2	Network-LSA	The set of routers attached to the network, originated by the DR(explained later).

Type-3	Network-summary-LSA	The inter-area routes from other areas. These are originated by the ABR's at the area's border.
Type-4	ASBR-summary-LSA	Describes the inter-area routes to the AS boundary routers; originated by the ABR's at the area borders. They validate reachability to an ASBR.
Type-5	AS-external-LSA	Routes from outside the OSPF routing domain; originated by the AS boundary routers.
Type-6	Group membership	Used by multicast, or MOSPF, not used by Juniper routers.
Type-7	NSSA-LSA	External routes information inside a NSSA area.
Type-8	Link-LSA	Used for IPV6 prefixes.

OSPF Router Types

Router types are an attribute of the OSPF process. They determine what types of LSAs the router generates, and their functions within the network. The OSPF router types are listed below and are also used in coming chapters.

1. Area Border Router (ABR) – A router that connects one or more areas to the backbone Area 0. It is considered a member of all areas it's connected to. The ABR maintains multiple copies of the link-state database, one for each area it is in.

2. Autonomous System Boundary Router (ASBR) – This router connects to more than one routing protocol. It is used to distribute routes received from other external ASs throughout its own AS. It creates external LSAs for these external routes for OSPF.

3. Internal Router (IR) – A router that only connects to a single area.

4. Backbone Router (BR) – A router that is connected to the backbone Area 0.

With this in mind, it's useful to note that an individual router can have multiple router classifications. An ABR is always a BR, for example. An ABR can also be an ASBR. The classification of a router type within OSPF is important when describing connectivity and LSA generation.

Summary

That's it for a general review of the basic elements of OSPF in an enterprise environment. If you want or need more OSPF-centric details, look for the *MORE?* paragraphs throughout this book.

If you're following along in your lab, a testbed, or Junosphere, the next chapter begins with a basic topology and complexity is added throughout the remaining chapters.

Let's get started with a Single Domain Intra Area.

Chapter 2

Single Domain Intra Area

This chapter details the fundamentals of a basic enterprise deployment of an OSPF network using a *multi-area* network within a single instance of OSPF. The areas used will be Area 0, Area 1, and Area 2 as shown in Figure 2.1.

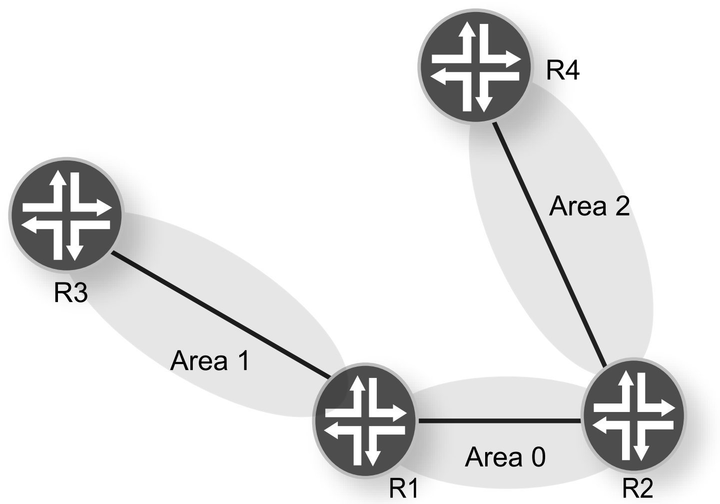

Figure 2.1 OSPF Topology for Chapter 2

As you can see in Figure 2.1, R1 and R2 are in Area 0 and are using Ethernet interfaces. The LSA types that the two Area 0 devices are sending are type-1, type-2, and type-3, and the router types are ABR routers for Area 0.

There are many LSA types, but only type-1, type-2, and type-3 are discussed in this chapter.

MORE? ABR LSA types can vary depending on what they are connected to and the type of area they are communicating with. For more information on LSA types see http://users.lmi.net/canepa/subdir/ospf_fundamentals.html. For more information on the LSA types that OSPF can use, see the book *Juniper Enterprise Routing, 2nd Edition*, by Peter Southwick, Doug Marshcke, and Harry Reynolds, at http://www.juniper.net/books.

The interface IP list and connection types used in the topology of this chapter are listed in Table 2.1.

Table 2.1 Interface and Connections Listed in the OSPF Topology of Chapter 2

Link	Interface	IP	Interface	IP
R1 - R2	ge-0/0/0.0	192.168.1.1/30	ge-0/0/0.0	192.168.1.2/30
R1 - R3	ge-0/0/1.0	192.168.2.1/0	ge-0/0/0.0	192.168.2.2/30
R2 - R4	ge-0/0/1.0	192.168.3.1/30	ge-0/0/0.0	192.168.3.2/30
R1 loopback	lo0.0	10.1.1.1/32		
R2 loopback	lo0.0	10.1.1.2/32		
R3 loopback	lo0.0	10.1.1.3/32		
R4 loopback	lo0.0	10.1.1.4/32		

Configuration of Backbone Area 0

The backbone area (also known as Area 0 or Area 0.0.0.0) forms the core of an OSPF network. All other areas are connected to it, and inter-area routing is supported via routers connected to the backbone area and to their own associated areas. It is the logical and physical structure for the *OSPF domain* and is attached to all non-zero areas in the OSPF domain.

The backbone area is responsible for distributing routing information between non-backbone areas. Therefore, if Area 0 were to go down, the OSPF network would fail and each area would become its own network "island." This situation leads some engineers to create multiple instances of OSPF to limit the failure domains of networks, a scenario covered later in this book.

A common best practice is manually setting the router ID for the OSPF process on each router. This helps traffic to navigate through large networks and to find certain routers within an advertising or peering relationship. Router IDs also enable SSH-ing to routers by using the router ID found in the database list as an easy guide to where routes originate.

Another advantage to manually defining the router ID is that it provides the ability to convey configuration information within the router ID. For example, let's say your large, international company has several subsidiaries located around the world. The router ID can still be consistent and convey the location of a router or its use, which can be useful when viewing OSPF relationships.

The configuration of the router ID is as follows:

```
set routing-options router-id 10.1.1.1
```

The best interface to use for this is the loopback interface address, because as long as the router is functioning it's always up, even if all of the links are down.

MORE? For an in depth discussion on how router ID's affect OSPF networks, see *Juniper Enterprise Routing, 2nd Edition*, by Peter Southwick, Doug Marshcke, and Harry Reynolds, at http://www.juniper.net/books.

Ethernet topologies require a *designated router* (DR). R1 is the router chosen for this function, and to ensure that it is designated appropriately the interface priority is set manually. This is necessary because a multi-access network type like Ethernet is used in the backbone. The DR is responsible for advertising the LAN and advertising LSA's to the *backup designated router* (BDR) and any router that is *neither the DR nor the BDR* (DROTHER).

The router types used in this example are *Area Border Routers* (ABR) and *Internal Routers* (IR). The ABR router connects areas to the backbone area and keeps multiple copies of the link-state database for each area. The IR is a router with all of its interfaces in a single area. The designations for these router types are as follows:

- R1: ABR
- R2: ABR
- R3: IR
- R4: IR

MORE? For more information on OSPF router types please see http://en.wikipedia.org/wiki/Open_Shortest_Path_First.

The initial OSPF configuration of R1 is as follows:

```
[edit protocols]
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0 priority 255
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.1 interface ge-0/0/1.0 interface-type p2p
```

Use the `interface-type p2p` command on the inter area links to force a WAN-like connectivity type (although with the advent of carrier provided Ethernet networks, the use of Ethernet is becoming a more prevalent technology). The command displays how to determine the type of network an interface is running as it pertains to OSPF:

```
root@R1# run show ospf interface ge-0/0/1.0 extensive
Interface           State   Area            DR ID           BDR ID          Nbrs
ge-0/0/1.0          PtToPt  0.0.0.1         0.0.0.0         0.0.0.0            1
  Type: P2P, Address: 192.168.2.1, Mask: 255.255.255.252, MTU: 1500, Cost: 1
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Protection type: None
  Topology default (ID 0) -> Cost: 0
```

As you can see, the network type is P2P, which is expected since it was configured this way on the ge-0/0/1.0 interface. The other interface, ge-0/0/1.0 on R1, is in OSPF and it was not manually configured. Again, the command below shows how to determine the type of network an interface is running in OSPF:

```
root@R1# run show ospf interface ge-0/0/0.0 extensive
Interface           State   Area            DR ID           BDR ID          Nbrs
ge-0/0/0.0          DR      0.0.0.0         10.1.1.1        10.1.1.2           1
  Type: LAN, Address: 192.168.1.1, Mask: 255.255.255.252, MTU: 1500, Cost: 1
  DR addr: 192.168.1.1, BDR addr: 192.168.1.2, Priority: 255
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Protection type: None
  Topology default (ID 0) -> Cost: 0
```

As shown, the network type displayed is LAN, which is a broadcast network as far as OSPF is concerned. This means a DR and BDR (which are backup designated routers) will be elected.

NOTE Whether to use broadcast (LAN), or P2P OSPF network types on Ethernet links, is often debated, since most of the time the routers are connected directly to one another. Configuring these links as P2P prevents using DR and BDR and simplifies the SPF calulations needed, as they are found on legacy WAN types.

MORE? For more information on OSPF interface types see: http://www.juniper.net/techpubs/software/junos/junos94/swconfig-routing/interface-type.html.

Designated Router

The function of a designated router within OSPF is forming adjacencies to all other routers on a multi-access network such as a LAN. It uses the multicast address 224.0.0.6 to elect a DR within this network. Multicast is used by OSPF as a communication mechanism to get to all routers within the LAN. OSPF will also elect a backup designated router and these two devices represent the multi-access network to outlying routers. All of the other routers within the multi-access network will be coded as *DRother*, which means they will not be used as DR or BDR unless one or the other fails. If a failure occurs the BDR becomes the DR and a new election process begins electing a BDR.

With the DR elected, the DR will form a neighboring relationship with the BDR. All other routers will neighbor with the DR and the BDR.

You also can manually configure the designated router (DR) for the Area 0 network, by issuing the `priority 255` command under the ge-0/0/0 interface within the OSPF configuration. The highest value for priority one can set is 255. The higher the priority the higher the chance of being the DR. The default for Juniper is a priority of 128 but is configurable with a range of 0 - 255. A priority of 0 will assure that it does not become DR. In order to do this, set the priority to 255, thus assuring that the router is the DR for Area 0 within your OSPF network. If there is a tie, the highest Router ID wins.

Step 1

Configure the priority:

```
set protocols ospf Area 0.0.0.0 interface ge-0/0/0.0 priority 255
```

Here is the output to verify that R1 is the DR in this network:

```
root@R1# run show ospf interface
Interface           State    Area          DR ID         BDR ID          Nbrs
ge-0/0/0.0          DR       0.0.0.0       10.1.1.1     10.1.1.2           1
lo0.0                DRother 0.0.0.0       0.0.0.0       0.0.0.0            0
ge-0/0/1.0          PtToPt  0.0.0.1       0.0.0.0       0.0.0.0            1
```

As shown, the DR for the LAN connected to ge-0/0/0 is10.1.1.1, which is the loopback address of R1. This means that it is the DR for the Area 0 network as it is connected to R2. Now, here is the output for R2:

```
root@R2# run show ospf interface
Interface           State   Area            DR ID         BDR ID        Nbrs
ge-0/0/0.0          BDR      0.0.0.0        10.1.1.1      10.1.1.2         1
lo0.0               DRother 0.0.0.0         0.0.0.0       0.0.0.0          0
ge-0/0/1.0          PtToPt  0.0.0.2         0.0.0.0       0.0.0.0          1
```

R2 is showing that R1 is the DR for its connection as well. In a larger network it would be wise to designate a single router as the DR for Area 0 with the priority command shown. This helps if any troubleshooting for OSPF is needed if a fault occurrs.

MORE? For more information on OSPF and the designated router, check out http://en.wikipedia.org/wiki/Open_Shortest_Path_First.

Loopback Interface

Another piece of the configuration that needs to be set is the loopback interface. It should be set as *passive*. This is a generally accepted practice that allows the loopback to be in the OSPF database but not to establish adjacencies. The main advantage of this practice is that it limits the convergence on an OSPF network by not adding another peer.

The configuration of R2 is very similar to that of R1 but with some minor differences. Here are the configuration pieces important to this discussion:

```
set routing-options router-id 10.1.1.2
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0 priority 100
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.2 interface ge-0/0/1.0 interface-type p2p
```

The major differences between the two routers are the *priority settings* for the multi-access interface, which are done to make sure that R1 is the DR, and to ensure that if there are more devices in the multi-access network, R2 would be the BDR. The other difference is the router ID address. This needs to be unique for every router in the network – if they were duplicated, unexpected results would be seen in the network.

Verification that Area 0 is functioning as expected is paramount within OSPF networking. It's already been determined that R1 is the DR and that R2 is the BDR within our OSPF multi-access Area 0, as expected.

Both of these routers have an adjacency displayed here:

```
root@R1# run show ospf neighbor area 0
Address          Interface              State     ID               Pri  Dead
192.168.1.2      ge-0/0/0.0             Full      10.1.1.2         100    37

root@R2# run show ospf neighbor area 0
Address          Interface              State     ID               Pri  Dead
192.168.1.1      ge-0/0/0.0             Full      10.1.1.1         255    37
```

As you can see, the adjacency state between R1 and R2 is full with the proper router ID for each device. If the adjacency state is not full it means there is an issue with the OSPF hello packets not being sent or received on one or both devices, that the adjacency is still forming, or there is a disagreement in the configuration between the devices. This can be caused by a multitude of issues, but the most common cause is is a misconfiguration between the devices.

OSPF neighboring states are how you determine if the peering relationship is up, or not, and their names describe the state fairly well. There are eight possible OSPF states:

- down
- attempt
- init
- 2-way
- exstart
- exchange
- loading
- full

MORE? For complete information on the different adjacency states in OSPF, you might want to read the networking classic, *OSPF: Anatomy of an Internet Routing Protocol*, by John T. Moy, available wherever books are sold.

Configuration of the Outlying Areas

According to our chapter topology in Figure 2.1, two more devices need to connect to the network from separate areas: Area 1 and Area 2. Router R3 will represent Area 1, and R4 will represent Area 2.

This might raise the question: *Why do we even have multiple areas in OSPF?* The answer is, in order to decrease the size of the link-state database within the area of a network. And this is why OSPF can scale so well with your network and your network's future.

With too many routers and links within large networks, LSA flooding could cripple your network and cause routers to continually conduct SPF calculations, thus leading to suboptimal convergence. As your network gets larger, you can limit this activity to an individual area and lessen the impact on the SPF calculations for the entire network by using multiple areas.

Configuration of R3

The first step in configuration of OSPF for R3 is to add the router ID for this device. Keep in mind that this is now the router ID for the router in all routing protocols going forward.

Step 1

Configure the router ID:

```
set routing-options router-id 10.1.1.3
```

Step 2

Next configure the OSPF protocol itself. This is very similar to the backbone routers cited earlier, but you will note that the area is different:

```
set protocols ospf Area 0.0.0.1 interface ge-0/0/0.0 interface-type p2p
set protocols ospf Area 0.0.0.1 interface lo0.0 passive
```

The configuration makes this router basically a non Area 0 router, or essentially a *stub router*, which is described in more detail in Chapter 3.

Step 3

Once the configuration is commited and everything is connected between R1 and R3, verification of a OSPF adjacency is needed to validate that the neighbor relationship is completed:

```
root@R3> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
192.168.2.1      ge-0/0/0.0             Full      10.1.1.1         128    39
```

This operational command shows that the adjacency is formed and the neighboring state between the two routers is full, validating that OSPF is running properly.

Step 4

There is a similar output on R1 as well, with the exception on the peering between R1 and R2:

```
root@R1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
192.168.1.2      ge-0/0/0.0             Full      10.1.1.2         100    34
192.168.2.2      ge-0/0/1.0             Full      10.1.1.3         128    32
```

Configuration of R4

Configuration of R4 is almost exactly like the configuration of R3, with the exception of the area. For this example, let's make R4 part of Area 3. Since the configurations are virtually the same between R3 and R4 it's not necessary to explain the process, just the output from the device:

```
set routing-options router-id 10.1.1.4
set protocols ospf Area 0.0.0.2 interface ge-0/0/0.0 interface-type p2p
set protocols ospf Area 0.0.0.2 interface lo0.0 passive

root@R4> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
192.168.3.1      ge-0/0/0.0             Full      10.1.1.2         128    36
```

This verifies that an adjacency is made between R4 and R2 and completes the initial configuration of our simple OSPF network.

Use and Configuration of Summarization

Our simple example network is of limited use without end points to route to. End points need to be represented by routes through the network.

Let's get these routes through the network, and then limit the link-state database by taking the diverse end point routes and representing all of them as a single route for the table. This action is know as route summarization.

Route summarization requires careful planning of the addresses within a network. In large networks it can be troublesome, since companies often acquire or merge with other businesses and want to integrate their systems into an existing network. The most common practice to combat poor address allocation is to assign a contiguous set of addresses to an OSPF area, so the whole area can be represented by a single address range. Summarization is typically done on the ABRs.

In this example, R1 and R2 perform the summarization for Areas 2 and 3 respectively, as shown in Figure 2.2. This way Area 1 and Area 2 can see each other's routes, but as a single route entry, as opposed to three different routes.

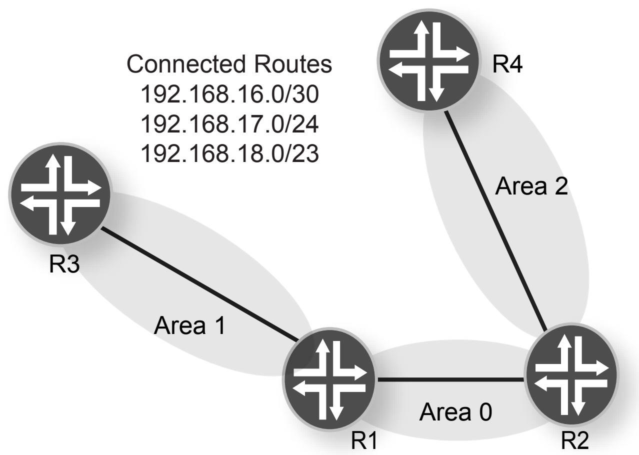

Figure 2.2 Example Network

Without summarization the routing table and OSPF database look like this:

```
root@R1> show route
inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.1.1.1/32        *[Direct/0] 4d 07:02:04
                    > via lo0.0
10.1.1.2/32        *[OSPF/10] 05:26:48, metric 1
                    > to 192.168.1.2 via ge-0/0/0.0
10.1.1.3/32        *[OSPF/10] 01:44:58, metric 1
                    > to 192.168.2.2 via ge-0/0/1.0
```

```
10.1.1.4/32        *[OSPF/10] 01:14:50, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
10.10.2.99/32      *[Local/0] 4d 07:35:34
                      Reject
192.168.1.0/30     *[Direct/0] 4d 02:37:43
                    > via ge-0/0/0.0
192.168.1.1/32     *[Local/0] 4d 07:20:26
                      Local via ge-0/0/0.0
192.168.2.0/30     *[Direct/0] 01:44:59
                    > via ge-0/0/1.0
192.168.2.1/32     *[Local/0] 4d 06:28:00
                      Local via ge-0/0/1.0
192.168.3.0/30     *[OSPF/10] 01:55:34, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.16.0/30    *[OSPF/10] 00:01:03, metric 3
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.17.0/24    *[OSPF/10] 00:01:03, metric 4
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.18.0/23    *[OSPF/10] 00:01:03, metric 3
                    > to 192.168.1.2 via ge-0/0/0.0
224.0.0.5/32       *[OSPF/10] 4d 07:15:36, metric 1
                      MultiRecv

root@R1> show ospf database area 0
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr          Seq      Age  Opt  Cksum  Len
Router  *10.1.1.1         10.1.1.1         0x800000ad   235  0x22 0xd89e  48
Router   10.1.1.2         10.1.1.2         0x800000ad   208  0x22 0xec86  48
Network *192.168.1.1      10.1.1.1         0x8000007f  2982  0x22 0x5fcb  32
Summary *10.1.1.3         10.1.1.1         0x80000003  1484  0x22 0x68b5  28
Summary  10.1.1.4         10.1.1.2         0x80000005  2876  0x22 0x54c5  28
Summary *192.168.2.0      10.1.1.1         0x80000004  1234  0x22 0x447f  28
Summary  192.168.3.0      10.1.1.2         0x80000007   223  0x22 0x2d91  28
Summary  192.168.16.0     10.1.1.2         0x80000003   208  0x22 0xaf05  28
Summary  192.168.17.0     10.1.1.2         0x80000001   208  0x22 0xc4ec  28
Summary  192.168.18.0     10.1.1.2         0x80000001   208  0x22 0xaa08  28
ASBRSum  10.1.1.4         10.1.1.2         0x8000000d  1846  0x22 0x36da  28
```

Routes listed in Figure 2.2 are shown on R1; these routes are contiguous and can be combined into a single summary route to be advertised to all other areas. As stated before, summarizing benefits the network administrator, and the network itself, by keeping the route table and the link-state database smaller. Convergence assists in troubleshooting, although it requires some planning to keep addressing localized to OSPF areas. The routes above can be summarized into a single route of 192.168.16.0/21 and still have room for growth.

The configuration to summarize the routes is completed on R2 since it is the ABR for Area 2. R2 then advertises the summary route to Area 0. Here is the configuration:

Step 1

First the configuration to summarize the routes:

```
set protocols ospf area 0.0.0.2 area-range 192.168.16.0/21
```

You can see the configuration is very simple.Verification of the summary being sent can be seen on R1 and R3:

```
root@R1> show route
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.1.1.1/32        *[Direct/0] 4d 06:54:59
                    > via lo0.0
10.1.1.2/32        *[OSPF/10] 05:19:43, metric 1
                    > to 192.168.1.2 via ge-0/0/0.0
10.1.1.3/32        *[OSPF/10] 01:37:53, metric 1
                    > to 192.168.2.2 via ge-0/0/1.0
10.1.1.4/32        *[OSPF/10] 01:07:45, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
10.10.2.99/32      *[Local/0] 4d 07:28:29
                      Reject
192.168.1.0/30     *[Direct/0] 4d 02:30:38
                    > via ge-0/0/0.0
192.168.1.1/32     *[Local/0] 4d 07:13:21
                      Local via ge-0/0/0.0
192.168.2.0/30     *[Direct/0] 01:37:54
                    > via ge-0/0/1.0
192.168.2.1/32     *[Local/0] 4d 06:20:55
                      Local via ge-0/0/1.0
192.168.3.0/30     *[OSPF/10] 01:48:29, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
                      192.168.16.0/21  *[OSPF/10] 01:06:57, metric 4

                              > to 192.168.1.2 via ge-0/0/0.0

224.0.0.5/32       *[OSPF/10] 4d 07:08:31, metric 1
                      MultiRecv

root@R1> show ospf database area 0
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.1         10.1.1.1         0x800000ac  2873  0x22 0xda9d  48
Router   10.1.1.2         10.1.1.2         0x800000ac  2259  0x22 0xee85  48
Network *192.168.1.1      10.1.1.1         0x8000007f  2623  0x22 0x5fcb  32
Summary *10.1.1.3         10.1.1.1         0x80000003  1125  0x22 0x68b5  28
```

```
Summary  10.1.1.4        10.1.1.2        0x80000005  2517  0x22 0x54c5  28
Summary *192.168.2.0     10.1.1.1        0x80000004   875  0x22 0x447f  28
Summary  192.168.3.0     10.1.1.2        0x80000006  2774  0x22 0x2f90  28
                      Summary 192.168.16.0  10.1.1.2   0x80000002 1229 0x22 0xaa0e
                      28
ASBRSum  10.1.1.4        10.1.1.2        0x8000000d  1487  0x22 0x36da  28

lab@R3> show route
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.1.1.1/32        *[OSPF/10] 01:57:09, metric 1
                    > to 192.168.2.1 via ge-0/0/0.0
10.1.1.2/32        *[OSPF/10] 01:57:09, metric 2
                    > to 192.168.2.1 via ge-0/0/0.0
10.1.1.3/32        *[Direct/0] 04:08:25
                    > via lo0.0
10.1.1.4/32        *[OSPF/10] 01:27:00, metric 3
                    > to 192.168.2.1 via ge-0/0/0.0
192.168.1.0/30     *[OSPF/10] 01:57:09, metric 2
                    > to 192.168.2.1 via ge-0/0/0.0
192.168.2.0/30     *[Direct/0] 01:57:10
                    > via ge-0/0/0.0
192.168.2.2/32     *[Local/0] 04:08:09
                      Local via ge-0/0/0.0
192.168.3.0/30     *[OSPF/10] 01:57:09, metric 3
                    > to 192.168.2.1 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/10] 00:02:29, metric 5
                    > to 192.168.2.1 via ge-0/0/0.0
224.0.0.2/32       *[PIM/0] 04:08:28
                      MultiRecv
224.0.0.5/32       *[OSPF/10] 04:08:28, metric 1
                      MultiRecv
224.0.0.13/32      *[PIM/0] 04:08:28
                      MultiRecv

lab@R3> show ospf database area 1
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.1        10.1.1.1        0x80000019  1635  0x22 0x4fbc  48
Router  *10.1.1.3        10.1.1.3        0x80000019  1060  0x22 0x9852  60
Summary  10.1.1.1        10.1.1.1        0x80000082  1385  0x22 0x732e  28
Summary  10.1.1.2        10.1.1.1        0x8000007d   422  0x22 0x7d27  28
Summary  10.1.1.4        10.1.1.1        0x80000006   208  0x22 0x62b6  28
Summary  192.168.1.0     10.1.1.1        0x800000bd  2634  0x22 0xdb2f  28
Summary  192.168.3.0     10.1.1.1        0x80000004  1136  0x22 0x437e  28
Summary  192.168.23.255  10.1.1.1        0x80000001   213  0x22 0x6f43  28
ASBRSum  10.1.1.4        10.1.1.1        0x80000005  2883  0x22 0x56c2  28
```

In a small network, like the one in this example, it is hard to see how summarization would be a benefit, but in a larger network, with many end points and hundreds of routers, the use of summarization is paramount to a stable OSPF and therefore to a stable network.

The final step in this chapter is verifying the LSA types from the configured routers. Let's take a look at how to do that from two different perspectives, those of R2 and R4.

Step 1

First let's look at R2:

```
[edit protocols ospf]
lab@R2# run show ospf database lsa-id 10.1.1.4 detail
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr          Seq      Age  Opt  Cksum  Len
Summary *10.1.1.4        10.1.1.2         0x80000002    41  0x22 0x5ac2  28
  mask 255.255.255.255
  Topology default (ID 0) -> Metric: 1
Summary  10.1.1.4        10.1.1.10        0x80000001   143  0x22 0x2ce9  28
  mask 255.255.255.255
  Topology default (ID 0) -> Metric: 1
ASBRSum *10.1.1.4        10.1.1.2         0x80000004    41  0x22 0x48d1  28
  mask 0.0.0.0
  Topology default (ID 0) -> Metric: 1
ASBRSum  10.1.1.4        10.1.1.10        0x80000004   133  0x22 0x18f9  28
  mask 0.0.0.0
  Topology default (ID 0) -> Metric: 1
    OSPF database, Area 0.0.0.2
 Type       ID               Adv Rtr          Seq      Age  Opt  Cksum  Len
Router   10.1.1.4        10.1.1.4         0x80000012    42  0x22 0xf864  84
  bits 0x2, link count 5
  id 10.1.1.2, data 192.168.3.2, Type PointToPoint (1)
    Topology count: 0, Default metric: 1
  id 192.168.3.0, data 255.255.255.252, Type Stub (3)
    Topology count: 0, Default metric: 1
  id 192.168.16.2, data 192.168.16.1, Type PointToPoint (1)
    Topology count: 0, Default metric: 1
  id 192.168.16.0, data 255.255.255.252, Type Stub (3)
    Topology count: 0, Default metric: 1
  id 10.1.1.4, data 255.255.255.255, Type Stub (3)
    Topology count: 0, Default metric: 0
  Topology default (ID 0)
    Type: PointToPoint, Node ID: 192.168.16.2
      Metric: 1, Bidirectional
    Type: PointToPoint, Node ID: 10.1.1.2
      Metric: 1, Bidirectional
```

The above output shows that R2 is receiving type-1 and type-3 LSAs from R4. This is the expected behavior that you should see from this router to the backbone Area 0.

Step 2

Next is to check the LSAs on R4:

```
root@R4# run show ospf database lsa-id 10.1.1.2 detail
    OSPF database, Area 0.0.0.2
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.2         10.1.1.2         0x80000006   310  0x22 0x9780  48
  bits 0x3, link count 2
  id 10.1.1.4, data 192.168.3.1, Type PointToPoint (1)
    Topology count: 0, Default metric: 1
  id 192.168.3.0, data 255.255.255.252, Type Stub (3)
    Topology count: 0, Default metric: 1
  Topology default (ID 0)
    Type: PointToPoint, Node ID: 10.1.1.4
      Metric: 1, Bidirectional
Summary  10.1.1.2         10.1.1.2         0x80000002    66  0x22 0x64bb  28
  mask 255.255.255.255
  Topology default (ID 0) -> Metric: 0
```

As shown here, there are type-1 and type-3 LSAs received from R2. These LSAs would be seen in a normal scenario for this device.

Summary

OSPF, in this example network, is fairly simple and small. The fundamentals of a single routing domain inter area network do not change by network size. In every large OSPF network there will be an Area 0, as well as other outlying areas to contend with. Planning is one of the most important steps one can make in designing a network, although many times network engineers do not get to design the OSPF network, they just inherit it.

The next chapter focuses on more advanced OSPF topics that a network engineer may see due to mergers and acquisitions with other enterprises or simply because of aggressive growth of the enterprise and therefore the network.

Chapter 3

Stubbiness

This chapter focuses on OSPF stub areas. Stub areas do not receive external routes from outside the OSPF autonomous system, and are represented by a default 0.0.0.0 route.

OSPF can support several types of stub areas:

- Stub Area: no external routes, internal summaries only.
- Totally Stubby Area: receive only a default route.
- NSSA Not So Stubby Area: advertise external routes, receive externals that orginated from the OSPF routing domain.
- NSSA Totally Stubby Area: same as NSSA except receives a default only.

MORE? For more information on stub area types within OSPF please see *Juniper Enterprise Routing, 2nd Edition*, by Peter Southwick, Doug Marshcke, and Harry Reynolds, at http://www.juniper.net/books

In this chapter you will learn how to configure two different, but common, types of OSPF stub areas using Juniper Networks routers running Junos. The types of OSPF stub areas that will be used are *Totally Stubby Areas* and *NSSA Areas*. These are generally the most widely used stub area types and by chapter's end you will be able to verify that the configuration is performing as desired, and that the LSA types seen, are correct.

Without Stubs

This chapter's network example uses the previous design from Chapter 1, but with static routes in Area 2 to represent external routes coming from outside of the OSPF autonomous system. The focus is on Area 1, which will be the stub area. Initially the static routes are shown in Area 1 as they would appear without any stubbiness configuration. This will be the control and baseline for how stub networks can assist network engineers in managing their outlying area link-state databases and routing tables. Figure 3.1 diagrams the use case for this chapter.

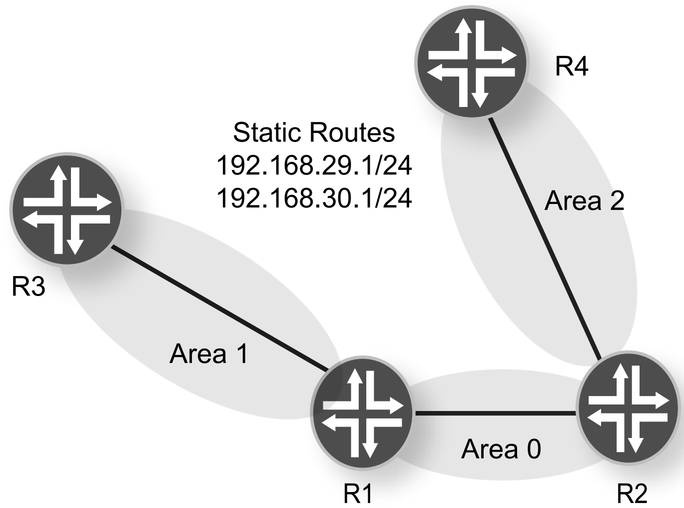

Figure 3.1 Example Network with Stubs

The configuration changes that are different from Chapter 1 occur on R4.

Step 1

Configuration of the static routes on R4:

```
set routing-options static route 192.168.29.0/24 next-hop 192.168.16.2
set routing-options static route 192.168.30.0/24 next-hop 192.168.16.2
```

Next, distribute these routes into the OSPF process, which is needed to show these routes as external LSA's.

MORE? OSPF route redistribution is covered in more detail in Chapter 4.

Step 2

Configuration of redistribution of the static routes into OSPF through a routing policy:

```
set policy-options policy-statement redistribute-static term 1 from protocol static
set policy-options policy-statement redistribute-static term 1 then accept
```

Lastly, configuration to allow the distribution to happen within OSPF, allowing the router to advertise these static routes into the OSPF process.

Step 3

Configuration of the redistribution process into OSPF on R4:

```
set protocols ospf export redistribute-static
```

Once these entries are committed to the configuration, the OSPF process starts advertising these static routes into Area 0 of this OSPF process. Proof can be seen on R4 as shown here:

```
root@R4> show ospf database advertising-router self
    OSPF database, Area 0.0.0.2
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.4         10.1.1.4         0x80000005   274  0x22 0x2db1  96
    OSPF AS SCOPE link-state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *192.168.29.0     10.1.1.4         0x80000001   274  0x22 0x8d8e  36
Extern  *192.168.30.0     10.1.1.4         0x80000001   274  0x22 0x8298  36
```

As displayed, the routes 192.168.29.0 and 192.168.30.0 are being advertised into OSPF as external routes. Next, it's necessary to see how R1 is interpreting these routes. R1 is in the backbone Area 0, and should see these routes as external. Let's check:

```
root@R1> show ospf route
Topology default Route Table:
Prefix              Path  Route      NH       Metric NextHop       Nexthop
                    Type  Type       Type            Interface     Address/LSP
10.1.1.2            Intra Area BR    IP            1 ge-0/0/0.0    192.168.1.2
10.1.1.3            Intra Router     IP            1 ge-0/0/1.0    192.168.2.2
10.1.1.4            Inter AS BR      IP            2 ge-0/0/0.0    192.168.1.2
10.1.1.1/32         Intra Network    IP            0 lo0.0
10.1.1.2/32         Intra Network    IP            1 ge-0/0/0.0    192.168.1.2
10.1.1.3/32         Intra Network    IP            1 ge-0/0/1.0    192.168.2.2
10.1.1.4/32         Inter Network    IP            2 ge-0/0/0.0    192.168.1.2
192.168.1.0/30      Intra Network    IP            1 ge-0/0/0.0
192.168.2.0/30      Intra Network    IP            1 ge-0/0/1.0
192.168.3.0/30      Inter Network    IP            2 ge-0/0/0.0    192.168.1.2
192.168.16.0/21     Inter Network    IP            4 ge-0/0/0.0    192.168.1.2
192.168.29.0/24     Ext2  Network    IP            0 ge-0/0/0.0    192.168.1.2
192.168.30.0/24     Ext2  Network    IP            0 ge-0/0/0.0    192.168.1.2
```

These routes are being learned as type-5 LSA's, which means they are external to the OSPF network and that OSPF has no knowledge of these routes beyond the OSPF autonomous system.

Lastly, R3, which is in another outlying OSPF area, must see these routes, along with all of the other routes within the OSPF autonomous system, as external. So let's check R3:

```
root@R3> show ospf route
Topology default Route Table:
Prefix              Path  Route       NH      Metric NextHop        Nexthop
                    Type  Type        Type           Interface      Address/LSP
10.1.1.1            Intra Area BR     IP           1 ge-0/0/0.0     192.168.2.1
10.1.1.4            Inter AS BR       IP           3 ge-0/0/0.0     192.168.2.1
10.1.1.1/32         Inter Network     IP           1 ge-0/0/0.0     192.168.2.1
10.1.1.2/32         Inter Network     IP           2 ge-0/0/0.0     192.168.2.1
10.1.1.3/32         Intra Network     IP           0 lo0.0
10.1.1.4/32         Inter Network     IP           3 ge-0/0/0.0     192.168.2.1
192.168.1.0/30      Inter Network     IP           2 ge-0/0/0.0     192.168.2.1
192.168.2.0/30      Intra Network     IP           1 ge-0/0/0.0
192.168.3.0/30      Inter Network     IP           3 ge-0/0/0.0     192.168.2.1
192.168.16.0/21     Inter Network     IP           5 ge-0/0/0.0     192.168.2.1
192.168.29.0/24     Ext2  Network     IP           0 ge-0/0/0.0     192.168.2.1
192.168.30.0/24     Ext2  Network     IP           0 ge-0/0/0.0     192.168.2.1
```

Again, you can see these static routes show up in the OSPF routing table as external routes after traversing the backbone area, even on R3, which is in Area 1. Remember this routing table, as this chapter refers to it again.

Totally Stubby Area

If your goal is to achieve the absolute minimal link-state database and routing table, totally stubby areas are the best choice. A totally stubby area is similar to a stub area, however, it does not allow external routes or summary routes. Inter-area (LSA3 or IA) routes are not summarized into totally stubby areas. The only way for traffic to get routed outside of the area is a default route. This is represented by a type-7 LSA which, as stated above, is a totally stubby default LSA route. When there is only one route out of the area, fewer routing decisions have to be made by the route processor, which lowers system resource utilization, and memory usage.

The configuration for a totally stubby area starts at the area border router (ABR) and the configuration for R1 needs to be changed to support a totally stubby area.

Step 1

The configuration of R1 is as follows:

```
set protocols ospf area 0.0.0.1 stub
set protocols ospf area 0.0.0.1 stub default-metric 5
set protocols ospf area 0.0.0.1 stub no-summaries
```

The no-summaries part of the statement is what makes this a totally stubby area within the Junos configuration.

Step 2

Now let's check the complete OSPF configuration on R1:

```
[edit protocols ospf]
root@R1# show
area 0.0.0.0 {
    interface ge-0/0/0.0 {
        priority 255;
    }
    interface lo0.0 {
        passive;
    }
}
area 0.0.0.1 {
    stub default-metric 5 no-summaries;
    interface ge-0/0/1.0 {
        interface-type p2p;
    }
}
```

The commands added to the configuration are centered around the `stub` command. The portion that has `no-summaries` tells the configuration within OSPF not to send summary routes to the stub Area 1. There is also the addition of `default-metric 5` to the existing configuration, this command forces the R1 to send a type-3 summary LSA with a default route to the outlying area R3 with an initial metric of 5.

The configuration of R3 is the next step in configuring a totally stubby area within OSPF.

Step 1

Adding this set is very simple, as it is really just one command added to the router:

```
set protocols ospf area 0.0.0.1 stub
```

This forces the router into a stub area mode like in the previous configuration. If the routers on both ends of a neighboring relationship do not have this command, the routers will not agree, and so will not peer.

Router R3's complete OSPF configuration looks like this:

```
{master:0}[edit protocols ospf]
root@R3# show
Area 0.0.0.1 {
    stub;
    interface ge-0/0/0.0 {
        interface-type p2p;
    }
    interface lo0.0 {
        passive;
    }
}
```

When these configuration changes are committed to the routers, the route table of R3 should show a single route learned from the backbone Area 0:

```
{master:0}[edit protocols ospf]
root@R3# run show route
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[OSPF/10] 01:41:25, metric 6
                    > to 192.168.2.1 via ge-0/0/0.0
10.1.1.3/32        *[Direct/0] 02:54:16
                    > via lo0.0
192.168.2.0/30     *[Direct/0] 02:53:54
                    > via ge-0/0/0.0
192.168.2.2/32     *[Local/0] 02:53:57
                      Local via ge-0/0/0.0
224.0.0.2/32       *[PIM/0] 02:54:19
                      MultiRecv
224.0.0.5/32       *[OSPF/10] 02:54:20, metric 1
                      MultiRecv
224.0.0.13/32      *[PIM/0] 02:54:19
                      MultiRecv
```

The route table shown here has a default route to the upstream backbone router at 192.168.2.1, which happens to be R1. This keeps the routing table for R3 at a minimum, and allows network engineers to utilize a much smaller device within the outlying areas. Another option for verification is looking at what R1 is sending to R3 using the following command:

```
root@R1# run show ospf database advertising-router self Area 1
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.1         10.1.1.1         0x80000002   357  0x20 0x9b89  48
Summary *0.0.0.0          10.1.1.1         0x80000015   723  0x20 0x42d6  28
```

As shown here, R1 advertises only its router LSA and the default route to Area 1.

Sometimes a network engineer may want to send more than just a default route, they may want import external LSA's to the route table. Let's review that option in the next section.

Not-So-Stubby Area

The purpose of a not-so-stubby area is to allow type-7 LSA's to be converted to type-5 LSA routes, basically causing the ABR to act like an ASBR for external routes. This means that a router in the NSSA router can connect to a non OSPF routing domain and advertise those routes to the backbone area, as well as to all other routers within its local area, but the NSSA area will not receive any external routes from the backbone area itself. The visual description in Figure 3.2 should help clarify how this works.

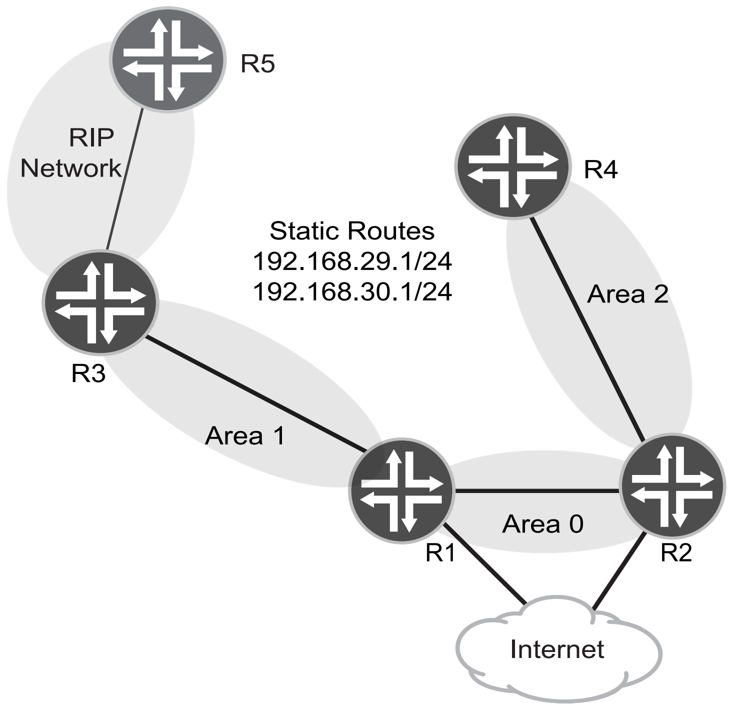

Figure 3.2 Network Example With an NSSA Area

In Figure 3.2, R5 advertises routes to R3 via routing protocol RIP, and these routes will be added into the OSPF network through OSPF Area 1 as an NSSA LSA.

MORE? For more information on the RIP routing protocol see *Juniper Enterprise Routing, 2nd Edition*, by Peter Southwick, Doug Marshcke, and Harry Reynolds, at http://www.juniper.net/books.

R3 won't receive any external LSA's from R1 that didn't originate from the OSPF network. Those routes are summarized into a default route in much the same way as a totally stubby area, as described in the previous section.

The most common deployment for a not-so-stubby area occurs when a newly acquired or merged company needs connectivity into an existing network. Connecting it directly to the core of the OSPF network is not commonplace, so using a outlying router to connect this *new network* to the *existing network* is the best practice.

For this exercise, as stated, a new router is added to the existing design. Table 3.1 lists the IP address table updated with the new router, R5.

Table 3.1 Updated Address Table for R5

Link	Interface	IP	Interface	IP
R1 - R2	ge-0/0/0.0	192.168.1.1/30	ge-0/0/0.0	192.168.1.2/30
R1 - R3	ge-0/0/1.0	192.168.2.1/30	ge-0/0/0.0	192.168.2.2/30
R2 - R4	ge-0/0/1.0	192.168.3.1/30	ge-0/0/0.0	192.168.3.2/30
R1 loopback	lo0.0	10.1.1.1/32		
R2 loopback	lo0.0	10.1.1.2/32		
R3 loopback	lo0.0	10.1.1.3/32		
R4 loopback	lo0.0	10.1.1.4/32		
R5 – R3	lo0.0	192.168.70.2/30	ge-0/0/1.0	192.168.70.1/30

The configuration of R5, and the RIP configuration of R3, are not shown here, as they are outside the scope of this book. RIP is used only as a means to show external routes.

So, armed with this information, your logical starting point would be with R1. Just as in the previous section, the additions to the configuration are fairly minimal.

Step 1

First let's configure R1:

```
set protocols ospf Area 0.0.0.1 nssa
set protocols ospf Area 0.0.0.1 nssa default-lsa default-metric 1
set protocols ospf Area 0.0.0.1 nssa default-lsa type-7
set protocols ospf Area 0.0.0.1 nssa summaries
```

And with these additions the total OSP configuration for R1 looks like this:

```
[edit protocols ospf]
root@R1# show
Area 0.0.0.0 {
    interface ge-0/0/0.0 {
        priority 255;
    }
    interface lo0.0 {
        passive;
    }
}
Area 0.0.0.1 {
    nssa {
        default-lsa {
            default-metric 1;
            type-7;
        }
        summaries;
    }
    interface ge-0/0/1.0 {
        interface-type p2p;
    }
}
```

Note the configuration additions that are around the NSSA statement within Area 1 of OSPF. Similar to the stub configuration, the `default-lsa default-metric` commands force R1 into sending a default route to Area 1's downstream R3. The next command you need includes the `default-lsa type-7` command that forces the use of a type-7 LSA. If you recall, the NSSA area converts type-7 LSA's into type-5 LSA's. The default route is advertised to R3 for the external routes from R4 (from the previous section). In the end, R1 advertises a default route, which is all of the routes that are generated from the OSPF autonomous system.

Step 2

Next, the configuration of R3 is needed to see an LSA exchange completed between R1 and R3:

```
set protocols ospf Area 0.0.0.1 nssa
```

The completed configuration for R3 looks like this:

```
[edit protocols ospf]
root@R3# show
export redistribute RIP;
Area 0.0.0.1 {
    nssa;
    interface ge-0/0/0.0 {
        interface-type p2p;
    }
    interface lo0.0 {
        passive;
    }
}
```

Quick observation reveals that there is not much change in the configuration for R3 from the previous sections. The only real change is adding the NSSA command for Area 1, which makes the area a not-so-stubby area. The other part of the configuration to note is the added `export redistribute RIP` command used to redistribute the created RIP routes to show external routes coming into the area through a policy.

MORE? For more information on route policies within Junos please see http://www.juniper.net/techpubs/en_US/junos9.6/information-products/topic-collections/config-guide-policy/frameset.html.

Let's verify that R3 is receiving the correct LSDB and, in turn, the correct routing table, using the following commands:

```
root@R3# run show route protocol ospf
inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[OSPF/150] 01:16:05, metric 2, tag 0
                    > to 192.168.2.1 via ge-0/0/0.0
10.1.1.1/32        *[OSPF/10] 01:51:30, metric 1
                    > to 192.168.2.1 via ge-0/0/0.0
10.1.1.2/32        *[OSPF/10] 01:16:05, metric 2
                    > to 192.168.2.1 via ge-0/0/0.0
10.1.1.4/32        *[OSPF/10] 01:16:05, metric 3
                    > to 192.168.2.1 via ge-0/0/0.0
192.168.1.0/30     *[OSPF/10] 01:51:30, metric 2
```

```
                    > to 192.168.2.1 via ge-0/0/0.0
192.168.3.0/30     *[OSPF/10] 01:16:05, metric 3
                    > to 192.168.2.1 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/10] 01:16:05, metric 5
                    > to 192.168.2.1 via ge-0/0/0.0
224.0.0.5/32       *[OSPF/10] 01:56:31, metric 1
                      MultiRecv
root@R3# run show ospf database
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.1         10.1.1.1         0x80000011   395  0x20 0x8390  48
Router  *10.1.1.3         10.1.1.3         0x80000010  2112  0x20 0xce25  60
Summary  10.1.1.1         10.1.1.1         0x80000005   995  0x20 0x8c94  28
Summary  10.1.1.2         10.1.1.1         0x80000005  2343  0x20 0x8c92  28
Summary  10.1.1.4         10.1.1.1         0x80000005  2156  0x20 0x8299  28
Summary  192.168.1.0      10.1.1.1         0x8000000e  1596  0x20 0x5963  28
Summary  192.168.3.0      10.1.1.1         0x80000005  1969  0x20 0x5f63  28
Summary  192.168.16.0     10.1.1.1         0x80000005  1783  0x20 0xd2e4  28
NSSA     0.0.0.0          10.1.1.1         0x80000003  2530  0x20 0x160d  36
NSSA    *192.168.71.0     10.1.1.3         0x80000003  1350  0x28 0xfc8   36
NSSA    *192.168.72.0     10.1.1.3         0x80000003   514  0x28 0x4d2   36
root@R3# run show ospf database summary
Area 0.0.0.1:
   2 Router LSAs
   12 Summary LSAs
   4 NSSA LSAs
Externals:
Interface ge-0/0/0.0:
Area 0.0.0.1:
Interface lo0.0:
Area 0.0.0.1:
```

You can see from the output here that R3 is receiving routes from within the OSPF autonomous system, as well as a default route for the networks that are listed as external from the outlying Area 2. You should note that R3 is also advertising the routes that have been redistributed from RIP. These routes are 192.168.71.0 and 192.168.72.0. This is one of the reasons to use a NSSA OSPF area – with a stub area, no routes are seen.

Verification that the backbone R1 is correctly receiving the external routes from the upstream router is necessary to show the OSPF route table and OSPF link-state database on R1. Let’s check:

```
root@R1# run show route protocol ospf
inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.1.1.2/32        *[OSPF/10] 01:28:47, metric 1
                    > to 192.168.1.2 via ge-0/0/0.0
```

```
10.1.1.3/32        *[OSPF/10] 02:04:12, metric 1
                    > to 192.168.2.2 via ge-0/0/1.0
10.1.1.4/32        *[OSPF/10] 01:28:47, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.3.0/30     *[OSPF/10] 01:28:47, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/10] 01:28:47, metric 4
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.29.0/24    *[OSPF/150] 01:28:47, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.30.0/24    *[OSPF/150] 01:28:47, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.71.0/24    *[OSPF/150] 01:31:02, metric 2, tag 0
                    > to 192.168.2.2 via ge-0/0/1.0
192.168.72.0/24    *[OSPF/150] 01:31:02, metric 2, tag 0
                    > to 192.168.2.2 via ge-0/0/1.0
224.0.0.5/32       *[OSPF/10] 1w3d 06:20:28, metric 1
                      MultiRecv
root@R1# run show ospf database
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.1         10.1.1.1         0x80000162   772  0x22 0x724d  48
Router   10.1.1.2         10.1.1.2         0x8000015d   988  0x22 0x8a38  48
Network *192.168.1.1      10.1.1.1         0x80000130   572  0x22 0xfa7e  32
Summary *10.1.1.3         10.1.1.1         0x80000004  2171  0x22 0x66b6  28
Summary  10.1.1.4         10.1.1.2         0x80000004  1449  0x22 0x56c4  28
Summary *192.168.2.0      10.1.1.1         0x80000007  1971  0x22 0x3e82  28
Summary  192.168.3.0      10.1.1.2         0x80000007  1911  0x22 0x2d91  28
Summary  192.168.16.0     10.1.1.2         0x80000004  1219  0x22 0xa610  28
ASBRSum *10.1.1.3         10.1.1.1         0x80000006   972  0x22 0x54c5  28
ASBRSum  10.1.1.4         10.1.1.2         0x8000000b  2373  0x22 0x3ad8  28
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.1         10.1.1.1         0x80000011  1172  0x20 0x8390  48
Router   10.1.1.3         10.1.1.3         0x80000011   502  0x20 0xcc26  60
Summary *10.1.1.1         10.1.1.1         0x80000005  1772  0x20 0x8c94  28
Summary *10.1.1.2         10.1.1.1         0x80000006   172  0x20 0x8a93  28
Summary *10.1.1.4         10.1.1.1         0x80000005  2932  0x20 0x8299  28
Summary *192.168.1.0      10.1.1.1         0x8000000e  2372  0x20 0x5963  28
Summary *192.168.3.0      10.1.1.1         0x80000005  2746  0x20 0x5f63  28
Summary *192.168.16.0     10.1.1.1         0x80000005  2559  0x20 0xd2e4  28
NSSA    *0.0.0.0          10.1.1.1         0x80000004   372  0x20 0x140e  36
NSSA     192.168.71.0     10.1.1.3         0x80000003  2129  0x28 0xfc8   36
NSSA     192.168.72.0     10.1.1.3         0x80000003  1292  0x28 0x4d2   36
    OSPF AS SCOPE link-state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   192.168.29.0     10.1.1.4         0x80000004  1882  0x22 0x8791  36
Extern   192.168.30.0     10.1.1.4         0x80000004  1104  0x22 0x7c9b  36
Extern  *192.168.71.0     10.1.1.1         0x80000004  1572  0x22 0x8f51  36
Extern  *192.168.72.0     10.1.1.1         0x80000004  1372  0x22 0x845b  36
```

Note that R1 can see both routes advertised by R3 as both external and NSSA routes, and that they are inserted into the routing table correctly. The reason it shows up in both sections is that it is a network route to Area 1 and a NSSA external route being advertised out. The backbone Area 0, router R2, will advertise these routes to the outlying Area 2 as external routes. Area 2 will see these routes as external routes since Area 2 is not a stub or NSSA area. This can be seen in output from R4 shown here:

```
root@R4> show ospf database
    OSPF database, Area 0.0.0.2
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.2         10.1.1.2         0x80000007  1884  0x22 0x8f89  48
Router  *10.1.1.4         10.1.1.4         0x80000007   563  0x22 0x29b3  96
Router   192.168.16.2    192.168.16.2     0x80000006  1860  0x22 0xd9bf  60
Summary  10.1.1.1         10.1.1.2         0x800000b2   961  0x22 0x1758  28
Summary  10.1.1.2         10.1.1.2         0x800000b1  2345  0x22 0x56b   28
Summary  10.1.1.3         10.1.1.2         0x80000004   730  0x22 0x6ab0  28
Summary  192.168.1.0      10.1.1.2         0x800000d6    38  0x22 0xa34d  28
Summary  192.168.2.0      10.1.1.2         0x80000005   500  0x22 0x467a  28
ASBRSum  10.1.1.1         10.1.1.2         0x80000007  2796  0x22 0x60b9  28
ASBRSum  10.1.1.3         10.1.1.2         0x80000004   269  0x22 0x5cbd  28
    OSPF AS SCOPE link-state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *192.168.29.0     10.1.1.4         0x80000004  2083  0x22 0x8791  36
Extern  *192.168.30.0     10.1.1.4         0x80000004  1306  0x22 0x7c9b  36
Extern   192.168.71.0     10.1.1.1         0x80000004  1778  0x22 0x8f51  36
Extern   192.168.72.0     10.1.1.1         0x80000004  1578  0x22 0x845b  36
```

NOTE It's also easy to switch this network from a NSSA area to a totally stubby network – just add the no-summaries statement to R1 as shown here:

```
[edit protocols ospf Area 0.0.0.1]
root@R1# show
nssa {
    default-lsa {
        default-metric 1;
        type-7;
    }
    no-summaries;
}
interface ge-0/0/1.0 {
    interface-type p2p;
}
```

This command forces the backbone R1 to send only a default route to the upstream NSSA R3, but still allows the Area 1 router R3 to be a

ASBR and send the external RIP routes into the backbone Area 0 and the rest of the network.

Looking at the interface, you can see that it is now classified as NSSA, – by the way, if the configurations on both ends do not match as NSSA the peering relationship will not happen.

```
root@R1# run show ospf interface ge-0/0/1.0 detail
Interface           State   Area            DR ID           BDR ID          Nbrs
ge-0/0/1.0          PtToPt  0.0.0.1         0.0.0.0         0.0.0.0            1
  Type: P2P, Address: 192.168.2.1, Mask: 255.255.255.252, MTU: 1500, Cost: 1
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Stub NSSA
  Auth type: None
  Protection type: None
  Topology default (ID 0) -> Cost: 0
```

Step 3

Lastly, let's check the LSA types from the perspective of R1 and R3, to see the LSA types being used:

```
root@R1# run show ospf database lsa-id 10.1.1.3 detail
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Summary *10.1.1.3        10.1.1.1         0x80000001   821  0x22 0x6cb3  28
  mask 255.255.255.255
  Topology default (ID 0) -> Metric: 1
ASBRSum *10.1.1.3        10.1.1.1         0x80000006   744  0x22 0x54c5  28
  mask 0.0.0.0
  Topology default (ID 0) -> Metric: 1
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.3        10.1.1.3         0x80000006   826  0x20 0xe21b  60
  bits 0x2, link count 3
  id 10.1.1.1, data 192.168.2.2, Type PointToPoint (1)
    Topology count: 0, Default metric: 1
  id 192.168.2.0, data 255.255.255.252, Type Stub (3)
    Topology count: 0, Default metric: 1
  id 10.1.1.3, data 255.255.255.255, Type Stub (3)
    Topology count: 0, Default metric: 0
  Topology default (ID 0)
    Type: PointToPoint, Node ID: 10.1.1.1
      Metric: 1, Bidirectional
```

From the output, you can see that R3 is sending type-3, type-4, and type-1 LSA's. And as stated in the explanation found in Chapter 1, this is expected behavior. From R3's perspective, a different verification approach is taken. Let's look at the LSA database to see the differences in the LSA's received:

```
root@R3> show ospf database brief
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.1         10.1.1.1         0x80000003  1210  0x20 0x9f82  48
Router  *10.1.1.3         10.1.1.3         0x80000006  1961  0x20 0xe21b  60
Summary  10.1.1.1         10.1.1.1         0x80000002  1543  0x20 0x9291  28
Summary  10.1.1.2         10.1.1.1         0x80000001  1882  0x20 0x948e  28
Summary  10.1.1.4         10.1.1.1         0x80000002  1719  0x20 0x8896  28
Summary  192.168.1.0      10.1.1.1         0x8000000b  1237  0x20 0x5f60  28
Summary  192.168.3.0      10.1.1.1         0x80000001  1882  0x20 0x675f  28
Summary  192.168.16.0     10.1.1.1         0x80000003  1714  0x20 0xd6e2  28
Summary  192.168.16.3     10.1.1.1         0x80000001  1237  0x20 0xd7db  28
Summary  192.168.17.0     10.1.1.1         0x80000001  1237  0x20 0x7aa   28
Summary  192.168.18.0     10.1.1.1         0x80000001  1237  0x20 0xecc5  28
Summary  192.168.75.0     10.1.1.1         0x80000001  1243  0x20 0x5627  28
Summary  192.168.76.0     10.1.1.1         0x80000001  1237  0x20 0x6711  28
Summary  192.168.77.0     10.1.1.1         0x80000001  1237  0x20 0x5c1b  28
NSSA     0.0.0.0          10.1.1.1         0x80000004   305  0x20 0x140e  36
NSSA     10.1.1.6         10.1.1.1         0x80000002   428  0x20 0x7a0d  36
NSSA    *192.168.71.0     10.1.1.3         0x80000002   877  0x28 0x11c7  36
NSSA    *192.168.72.0     10.1.1.3         0x80000002   219  0x28 0x6d1   36
```

And from the bolded output you can see that R3 is receiving NSSA LSA's from its neighbors.

Summary

Stubbiness has great benefits for the network engineer, as this chapter has attempted to illustrate. For example, routers can be expensive. And typically the larger routers with increased memory and faster processors are even more expensive. Using OSPF stub areas can substantially decrease the size of a network's remote area routing tables and link-state databases inside remote areas, which allows these areas to be supported with more scalable routers, such as the MX 5, 10, or 20 Series.

The use of stub areas can also ease troubleshooting of routing issues within networks for remote devices that the network engineer may not have physical access to – and having a single route leaving an OSPF area makes troubleshooting routing issues within those devices much easier.

Chapter 4

Redistribution

An enterprise network cannot survive with only OSPF to maintain all of its routes. There are many devices that will not support a routing protocol or connectivity to Internet Service Providers via BGP or static routes. In most cases, OSPF will be the protocol of choice for the IGP of enterprises, and these other route types will need to be added into the OSPF database via a redistribution process.

There are other situations where OSPF would need to be entered into another routing process and redistribution of OSPF would need to be accomplished. For this to happen you need to redistribute these other processes into OSPF.

Redistribution using Junos is accomplished via *route policies*. These are policies that can match on a protocol, route filter, or route type, as well as on numerous other values that then can be choosen to accept these types for redistribution into another routing protocol.

MORE? For more information on Junos route policies see http://www.juniper.net/techpubs/en_US/junos9.6/information-products/topic-collections/config-guide-policy/frameset.html.

Types of Redistribution

You can use different types of route redistribution, each with benefits and, of course, limitations. The three types of route redistribution are:

- Basic Mutual Redistribution
- Mutual Redistribution
- Hierarchical Redistribution.

This section explores the three different types of redistribution, how they are used in various environments, and the benefits and limitations of each.

Basic Mutual Redistribution

As you can see from Figure 4.1, the router in the middle is used as a single device to connect two separate routing domains.

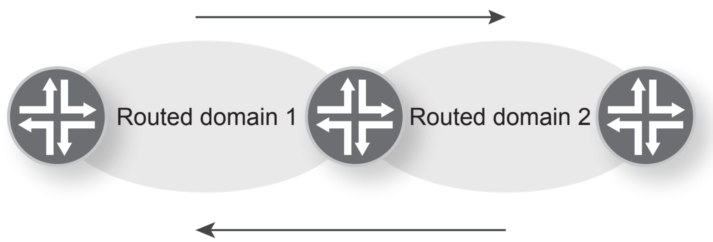

Figure 4.1 Basic Mutual Redistribution

Basic mutual redistribution means using a single device for this redistribution process. In Figure 4.1 the router in the middle limits control of the process to only the distributing device. The main benefit of a single device is that all filters to prevent routes learned from domain 1, that have been put into domain 2, will not be re-sent to domain 1. This would cause routing loops and prevent communication between routers from taking place. As stated previously, filters are applied to the router in the middle to prevent this.

Basic Mutual Redistribution is commonly used when merging the networks of two companies. It is the easiest type of redestribution to maintain because a single device manages the redistribution process. The major downfalls of Basic Mutual Redistribution are scaling and redundancy, as it will only scale as large as the device route table, memory, or bandwidth allows. Also, if the router were to fail, the process would fail with it, and no communication would occur between the two domains.

Multiple Mutual Redistribution

Multiple mutual redistribution is very similar to basic mutual redistribution. The major difference between the two is that multiple mutual distribution scales across multiple routers to connect two or more routed domains. Figure 4.2 shows two routers performing a redistribution process between the two routed domains.

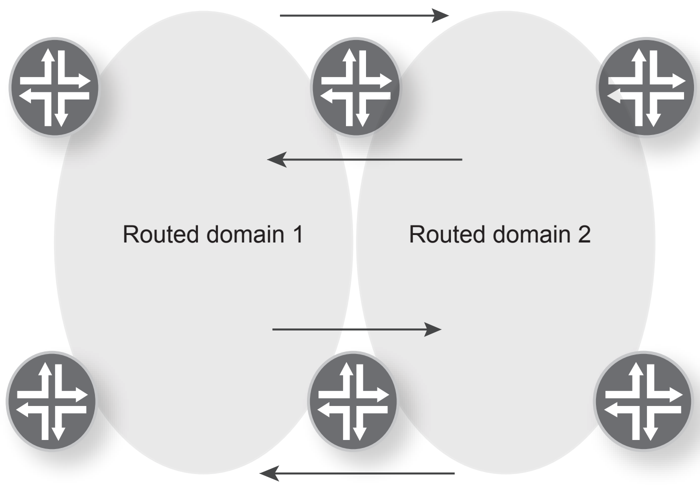

Figure 4.2 Multiple Mutual Redistribution

With this technique, scaling is no longer an issue and you can easily connect more routers to scale even larger. The other benefit of multiple mutual distribution is that there is no longer a single point of failure, and redundancy is available. A large drawback of multiple mutual redistribution is that route filtering is now much more complex, and management of the filtering is needed on more devices than it is with basic mutual redistribution.

Hierarchical Redistribution

Hierarchical Resistribution is very different from the previous two redistribution methods, because the idea is to take other routing domains and send them a default from the main routed domain. In fact, Figure 4.3 looks a lot like the OSPF stub areas covered earlier in this book.

The main benefit of hierarchical redistribution is that it removes the need for route filtering between routing domains on the redistribution routers, so the complexity is greatly reduced. Redundancy can easily be added, if necessary since there is no requirement for filtering, as there is with the multiple ,mutual redistribution example. The one large downfall of hiereachical redistribution is that there can only be one default gateway within all routed domains, but in cases where enterprises need to merge two diverse networks together with their own independent route tables, hiereachical redistribution would surely suffice.

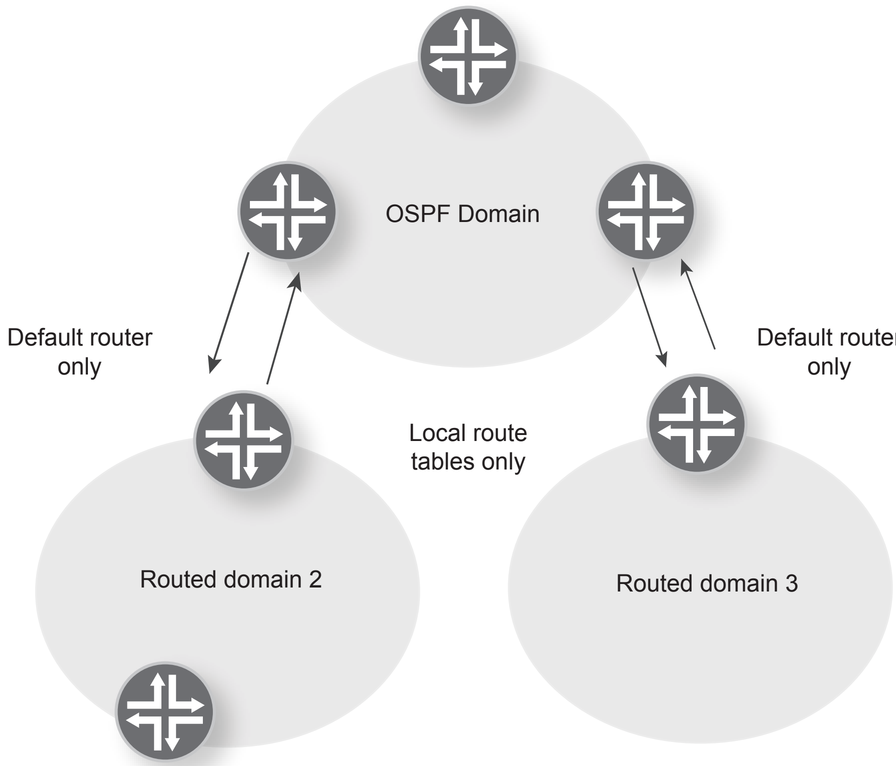

Figure 4.3 Diagram of Hierarchical Redistribution

Redistribution Design

This chapter shows you how to configure a redistribution policy to redistribute a default route, representing an internet feed, using a redistribution policy from the RIP protocol used in Chapter 3. It also examines how these routes are shown in the routing table and in the OSPF database, using the network digram shown in Figure 4.4.

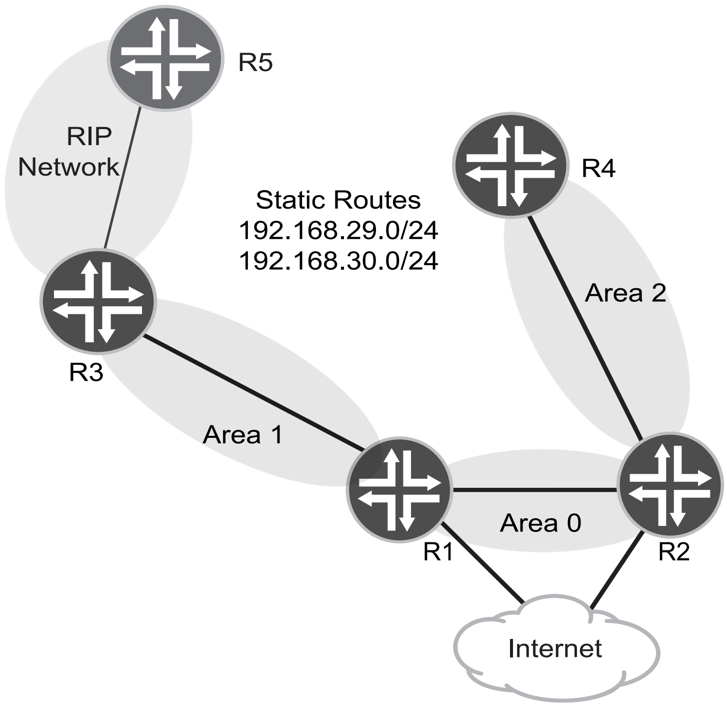

Figure 4.4 Example Network with Redistribution for Chapter 4

The internet feeds will be represented as a default route on both backbone Area 0 routers, R1 and R2. This could easily be a BGP feed from two different service providers, but that is a example for a different book.

MORE? To get more information on BGP please see *Juniper Enterprise Routing, 2nd Edition*, by Peter Southwick, Doug Marshcke, and Harry Reynolds, at http://www.juniper.net/books.

Dual Default Route

As mentioned previously, most enterprise organizations have connections to internet service providers (ISPs). They use these connections to support internet connectivity for corporate resources and end users. Many times they will use an EGP (external gateway protocol) like BGP (border gateway protocol) to receive routes from ISPs as a way to provide redundancy for internet connections.

This exercise uses a *dual static route* scenario to simulate how this scenario would work, in addition to, providing the resulting OSPF database and routing table.

Table 4.1 **Static Routes Table**

Link	Interface	IP	Interface	IP
R1 - R2	ge-0/0/0.0	192.168.1.1/30	ge-0/0/0.0	192.168.1.2/30
R1 - R3	ge-0/0/1.0	192.168.2.1/30	ge-0/0/0.0	192.168.2.2/30
R2 - R4	ge-0/0/1.0	192.168.3.1/30	ge-0/0/0.0	192.168.3.2/30
R1 loopback	lo0.0	10.1.1.1/32		
R2 loopback	lo0.0	10.1.1.2/32		
R3 loopback	lo0.0	10.1.1.3/32		
R4 loopback	lo0.0	10.1.1.4/32		
R5 – R3	ge-0/0/0.0	192.168.70.2/30	ge-0/0/1.0	192.168.70.1/30
R1 - internet	fe-0/0/2.0	16.23.15.1/30		
R2 - internet	fe-0/0/2.0	16.23.15.5/30		

To do this you need to add the static routes to the configuration and point them to a new interface within the device using a few configuration steps.

Step 1 Router R1

First configure R1:

```
set routing-options static route 0.0.0.0/0 next-hop 16.23.15.2
```

Step 2 Router R2

Then configure R2:

```
set routing-options static route 0.0.0.0/0 next-hop 16.23.15.6
```

With the static routes in the configuration, the route tables should now look like this:

```
lab@R1# run show route
inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 00:06:32
```

```
                     > to 16.23.15.2 via fe-0/0/2.0
10.1.1.1/32        *[Direct/0] 3d 23:59:37
                     > via lo0.0
10.1.1.2/32        *[OSPF/10] 3d 23:57:29, metric 1
                     > to 192.168.1.2 via ge-0/0/0.0
10.1.1.3/32        *[OSPF/10] 00:01:12, metric 1
                     > to 192.168.2.2 via ge-0/0/1.0
10.10.2.99/32      *[Local/0] 3d 23:58:58
                       Reject
16.23.15.0/30      *[Direct/0] 00:06:32
                     > via fe-0/0/2.0
16.23.15.1/32      *[Local/0] 00:06:32
                       Local via fe-0/0/2.0
192.168.1.0/30     *[Direct/0] 3d 23:58:19
                     > via ge-0/0/0.0
192.168.1.1/32     *[Local/0] 3d 23:58:59
                       Local via ge-0/0/0.0
192.168.2.0/30     *[Direct/0] 00:01:19
                     > via ge-0/0/1.0
192.168.2.1/32     *[Local/0] 3d 23:58:59
                       Local via ge-0/0/1.0
192.168.71.0/24    *[OSPF/150] 00:01:03, metric 2, tag 0
                     > to 192.168.2.2 via ge-0/0/1.0
192.168.72.0/24    *[OSPF/150] 00:01:03, metric 2, tag 0
                     > to 192.168.2.2 via ge-0/0/1.0
224.0.0.5/32       *[OSPF/10] 3d 23:59:40, metric 1
                       MultiRecv
```

As shown here, there is now a default route present in the routing table.

Now that the default, or 0/0, route is in the table you need to get this into OSPF. In order to do so you need to create a policy to add the external LSA into OSPF.

Step 3 Both R1 and R2

Create the following policy:

```
set policy-options policy-statement redistribution term 1 from protocol static
set policy-options policy-statement redistribution term 1 from route-
filter 0.0.0.0/0 exact
set policy-options policy-statement redistribution term 1 then accept
```

As shown here, the statement `from protocol static` simply means that the statement is matching on the protocol static. The `route-filter 0.0.0.0/0 exact` command will match specifically for the default route and not select other static routes that may exist. And the `then accept` command accepts the configuration into the policy

MORE? For more information on route policies within Junos please see: http://www.juniper.net/techpubs/en_US/junos9.6/information-products/topic-collections/config-guide-policy/frameset.html.

Now these policies need to be applied to OSPF to redistribute the routes into the OSPF database.

Step 4 Both R1 and R2

Use the `set protocols ospf export redistribution` command to force OSPF to export the policy into the OSPF database.

Now let's check it:

```
lab@R2# run show ospf database external
    OSPF AS SCOPE link-state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   0.0.0.0          10.1.1.1         0x80000001   313  0x22 0x9114  36
Extern  *0.0.0.0          10.1.1.2         0x80000001   344  0x22 0x8b19  36
Extern   192.168.29.0     10.1.1.4         0x80000002  1948  0x22 0x8b8f  36
Extern   192.168.30.0     10.1.1.4         0x80000002  1342  0x22 0x8099  36
Extern   192.168.71.0     10.1.1.1         0x80000002  1528  0x22 0x934f  36
Extern   192.168.72.0     10.1.1.1         0x80000002  1328  0x22 0x8859  36
```

The highlighted entry is the external route added via the policy statement for the default route. It's also important to note that the asterisk denotes local origination. One might further note that there is another default route entered into the database from R1 as well, since it also has a default route advertised as a type-5 LSA into the OSPF process.

Now that R1 and R2 have the correct entries in the LSDB, let's focus on the outlying Area 2. This area is not configured as a stub area and should see the default type-5 LSA in its OSPF database:

```
root@R4> show ospf database external
    OSPF AS SCOPE link-state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   0.0.0.0          10.1.1.1         0x80000001  1006  0x22 0x9114  36
Extern   0.0.0.0          10.1.1.2         0x80000001  1037  0x22 0x8b19  36
Extern  *192.168.29.0     10.1.1.4         0x80000003   426  0x22 0x8990  36
Extern  *192.168.30.0     10.1.1.4         0x80000002  2033  0x22 0x8099  36
Extern   192.168.71.0     10.1.1.1         0x80000003   162  0x22 0x9150  36
Extern   192.168.72.0     10.1.1.1         0x80000002  2021  0x22 0x8859  36
```

Issuing the `show ospf database external` command, the two 0.0.0.0/0 routes are shown as external type-5 LSA routes in the LSBD. From the perspective of outlying Area 1 there is already a default route being advertised, since it is a stub network:

```
root@R3> show ospf database nssa
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
NSSA     0.0.0.0         10.1.1.1         0x8000007f   658  0x20 0x1d89  36
NSSA    *192.168.71.0     10.1.1.3         0x80000001   638  0x28 0x13c6  36
NSSA    *192.168.72.0     10.1.1.3         0x80000001   638  0x28 0x8d0   36
```

Every router in the routing domain should now have access to the internet via the default route. Now let's cover redistribution of another routing protocol into the OSPF process.

Redistribution of Another IGP

One of this book's OSFP examples from Chapter 3 used a redistribution to place routes learned via RIP into the OSPF database. This section provides more details of the procedures involved in that process.

To redistribute another IGP, the RIP routing process from R5 will be redistributed into the OSPF LSDB of Area 1.

Step 1

Again, this is done through a policy statement that is applied to the OSPF process. Let's use the following policy:

```
set policy-options policy-statement redistribute-rip term 1 from protocol rip
set policy-options policy-statement redistribute-rip term 1 then accept
```

The only difference between this example of redistribution and the one shown in the previous section is the `from protocol rip option`, which states that the source protocol is RIP instead of static.

Step 2

Now use the `set protocols ospf export redistribute RIP` command to take the policy and put all routes learned through the RIP routing process into the OSPF routing process of the router.

Once complete, here is the LSDB after application of the RIP redistribution policy:

```
root@R3# run show route receive-protocol rip 192.168.70.2
inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.71.0/24    *[RIP/100] 00:56:33, metric 2, tag 0
                    > to 192.168.70.2 via ge-0/0/1.0
```

```
192.168.72.0/24    *[RIP/100] 00:56:33, metric 2, tag 0
                    > to 192.168.70.2 via ge-0/0/1.0
```

And these are the LSA's that will be advertised into the OSPF process:

```
root@R3# run show ospf database
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.1         10.1.1.1         0x80000003  1643  0x20 0x9f82  48
Router  *10.1.1.3         10.1.1.3         0x80000005  2013  0x20 0xe41a  60
Summary  10.1.1.1         10.1.1.1         0x8000007a  2205  0x20 0xa10a  28
Summary  10.1.1.2         10.1.1.1         0x80000079  2393  0x20 0xa307  28
Summary  10.1.1.4         10.1.1.1         0x80000002   707  0x20 0x8896  28
Summary  192.168.1.0      10.1.1.1         0x80000091   332  0x20 0x52e6  28
Summary  192.168.3.0      10.1.1.1         0x80000003   894  0x20 0x6361  28
Summary  192.168.16.0     10.1.1.1         0x80000002   519  0x20 0xd8e1  28
NSSA     0.0.0.0          10.1.1.1         0x80000080  2018  0x20 0x1b8a  36
NSSA    *192.168.71.0     10.1.1.3         0x80000002  1397  0x28 0x11c7  36
NSSA    *192.168.72.0     10.1.1.3         0x80000002   592  0x28 0x6d1   36
```

As shown here, the RIP routes are now in the OSPF LSDB. They are NSSA routes since the router is configured as an NSSA stub router.

Routers R1 and R2 will see the advertisements from R3 as follows:

```
lab@R2# run show ospf database external
    OSPF AS SCOPE link-state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   0.0.0.0          10.1.1.1         0x80000006   444  0x22 0x8719  36
Extern  *0.0.0.0          10.1.1.2         0x80000006  2925  0x22 0x811e  36
Extern   192.168.29.0     10.1.1.4         0x80000004   975  0x22 0x8791  36
Extern   192.168.30.0     10.1.1.4         0x80000004   197  0x22 0x7c9b  36
Extern   192.168.71.0     10.1.1.1         0x80000003  1944  0x22 0x9150  36
Extern   192.168.72.0     10.1.1.1         0x80000003  1757  0x22 0x865a  36
```

The routes from RIP advertised by OSPF Area 1 appear in R2's link-state database as external. They will be placed into the routing table accordingly, as shown here:

```
lab@R2# run show ospf route
Topology default Route Table:
Prefix              Path  Route      NH       Metric NextHop       Nexthop
                    Type  Type       Type            Interface     Address/LSP
10.1.1.1            Intra Area/AS BR IP            1 ge-0/0/0.0    192.168.1.1
10.1.1.3            Inter AS BR      IP            2 ge-0/0/0.0    192.168.1.1
10.1.1.4            Intra AS BR      IP            1 ge-0/0/1.0    192.168.3.2
192.168.16.2        Intra Router     IP            2 ge-0/0/1.0    192.168.3.2
0.0.0.0/0           Ext2  Network    IP            0 ge-0/0/0.0    192.168.1.1
10.1.1.1/32         Intra Network    IP            1 ge-0/0/0.0    192.168.1.1
10.1.1.2/32         Intra Network    IP            0 lo0.0
10.1.1.3/32         Inter Network    IP            2 ge-0/0/0.0    192.168.1.1
10.1.1.4/32         Intra Network    IP            1 ge-0/0/1.0    192.168.3.2
192.168.1.0/30      Intra Network    IP            1 ge-0/0/0.0
192.168.2.0/30      Inter Network    IP            2 ge-0/0/0.0    192.168.1.1
```

```
192.168.3.0/30     Intra Network    IP            1 ge-0/0/1.0
192.168.16.0/21    Inter Discard    IP       16777215
192.168.16.0/30    Intra Network    IP            2 ge-0/0/1.0    192.168.3.2
192.168.17.0/24    Intra Network    IP            3 ge-0/0/1.0    192.168.3.2
192.168.18.0/23    Intra Network    IP            2 ge-0/0/1.0    192.168.3.2
192.168.29.0/24    Ext2  Network    IP            0 ge-0/0/1.0    192.168.3.2
192.168.30.0/24    Ext2  Network    IP            0 ge-0/0/1.0    192.168.3.2
192.168.71.0/24    Ext2  Network    IP            2 ge-0/0/0.0    192.168.1.1
192.168.72.0/24    Ext2  Network    IP            2 ge-0/0/0.0    192.168.1.1
```

Summary

Hopefully, this chapter will reaffirm to the reader that redistribution of routes is necessary to best practice networking. There will always be a need for it, and making sure that the routes are inserted into the OSPF process correctly just helps keep the network running optimally.

Now let's get a little more complex in Chapter 5 with multiple domains.

Chapter 5

Multiple Homogeneous Domains

Multiple homogeneous domains within OSPF are sometimes referred to as *different OSPF processes*. They are useful because different routing domains can be assigned to different regions while still keeping the same routing protocol, which is often required when integrating the network of one company into that of another. The processor utilization of OSPF is used for all separate instances, which also provides the ability to keep several backbone areas within the network to regionalize all of the OSPF network.

NOTE In reality, this type of deployment should be avoided whenever possible and should be used only by very large entities, or as a temporary solution for an integration.

This chapter shows you how to deploy multiple homogeneous OSPF domains using Junos routers and switches. And, as with all the chapters in this book, it shows you how to verify that the routing tables and OSPF database are receiving the correct entries.

Subject Network

This chapter uses the network topology shown in Figure 5.1. Routers R6 and R7 are additions to the network created in Chapter 4. The additions represent a different OSPF domain that consists of a separate OSPF Area 0 and Area 1. The connection between these two domains will be on R6 and R1. The IP address list is captured in Table 5.1.

Table 5.1 The IP Address List for the Network Example in Chapter 5

Link	Interface	IP	Interface	IP
R1 - R2	ge-0/0/0.0	192.168.1.1/30	ge-0/0/0.0	192.168.1.2/30
R1 - R3	ge-0/0/1.0	192.168.2.1/30	ge-0/0/0.0	192.168.2.2/30
R2 - R4	ge-0/0/1.0	192.168.3.1/30	ge-0/0/0.0	192.168.3.2/30
R1 loopback	lo0.0	10.1.1.1/32		
R2 loopback	lo0.0	10.1.1.2/32		
R3 loopback	lo0.0	10.1.1.3/32		
R4 loopback	lo0.0	10.1.1.4/32		
R5 – R3	ge-0/0/0.0	192.168.70.2/30	ge-0/0/1.0	192.168.70.1/30
R1 - internet	fe-0/0/2.0	16.23.15.1/30		

R2 - internet	fe-0/0/2.0	16.23.15.5/30		
R6 – R1	ge-0/0/0.0	192.168.80.2/30	fe-0/0/3.0	192.168.80.1/30
R6 – R7	ge-0/0/1.0	192.168.80.5/30	ge-0/0/0.0	192.168.80.6/30
R6 loopback	lo0.0	10.1.1.6/32		
R7 loopback	lo0.0	10.1.1.7/32		

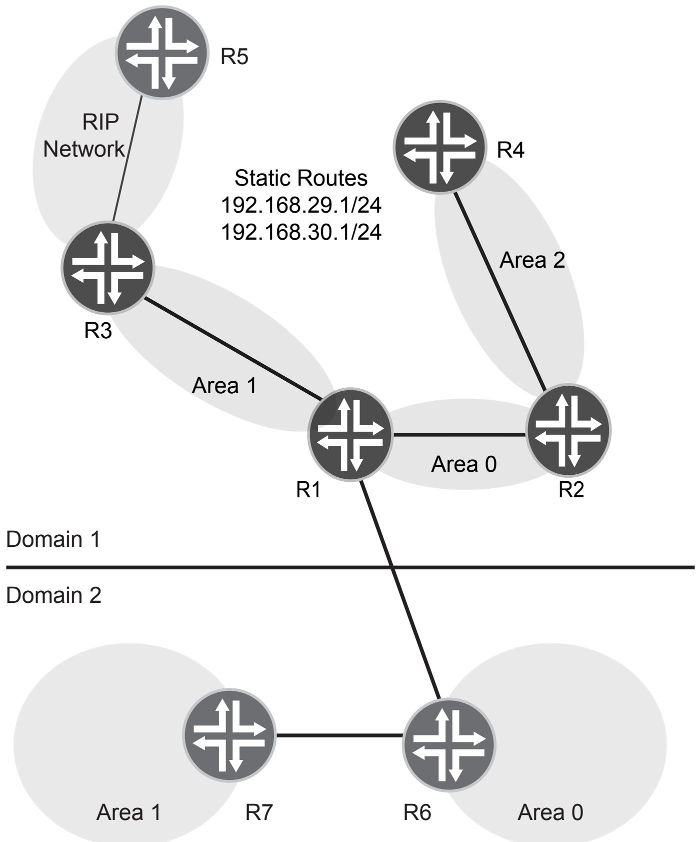

Figure 5.1 Network Example with Multiple Domains

The OSPF configurations for R6 and R7 are listed below:

```
[edit protocols]
root@R6# show
ospf {
    Area 0.0.0.0 {
        interface ge-0/0/0.0;
        interface lo0.0;
           passive;
         }
    }
    Area 0.0.0.1 {
        interface ge-0/0/1.0;
    }
}
```

And for R7:

```
[edit protocols]
root@R6# show
ospf {
    Area 0.0.0.1 {
        interface fe-0/0/0.0;
        interface lo0.0  {
           passive;
         }
    }
}
```

As you can see, the configuration for OSPF in this domain is very familiar and basic. There are two areas with an Area 0 backbone. For this demonstration, two host routes to the R7 have been advertised to R6 to show network connectivity for the OSPF network being added. R6 has the following routing table:

```
root@R6# run show route
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.1.1.6/32        *[Direct/0] 00:45:27
                    > via lo0.0
172.19.21.1/32     *[OSPF/10] 00:40:09, metric 2
                    > to 192.168.80.6 via ge-0/0/1.0
172.19.22.1/32     *[OSPF/10] 00:40:09, metric 2
                    > to 192.168.80.6 via ge-0/0/1.0
192.168.80.0/30    *[Direct/0] 01:25:05
                    > via ge-0/0/0.0
192.168.80.2/32    *[Local/0] 01:25:08
                      Local via ge-0/0/0.0
192.168.80.4/30    *[Direct/0] 00:59:46
                    > via ge-0/0/1.0
192.168.80.5/32    *[Local/0] 01:25:08
                      Local via ge-0/0/1.0
224.0.0.5/32       *[OSPF/10] 01:25:30, metric 1
```

```
                      MultiRecv
__juniper_private1__.
inet.0: 4 destinations, 4 routes (2 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both
10.0.0.1/32        *[Direct/0] 01:25:30
                    > via lo0.16385
10.0.0.16/32       *[Direct/0] 01:25:30
                    > via lo0.16385
```

The routes 172.19.21.1/32 and 172.19.22.1/32 are both shown in the routing table as learned from OSPF. These are the routes that were added from R7 that have been sent via LSA's to R6.

Configuration of Router R1

With multiple OSPF domains, one device is needed to connect the domains, and in this example this device will be R1. Most of the significant configuration will reside on this device and the Junos OS will create a new routing instance that houses the new domains in OSPF backbone Area 0.

MORE? A good book on routing instances and how they function can be found in *Juniper Networks Reference Guide*, by Thomas M Thomas, et. al., (Addison-Wesley Publishing Company, 2002).

Let's follow the configuration steps of this routing instance.

Step 1

First create the routing instance:

```
set routing-instances domain2 interface fe-0/0/3.0
set routing-instances domain2 protocols ospf area 0.0.0.0 interface fe-0/0/3.0
```

Items to note in this configuration are a new configuration stanza, which is under the `[routing-instances]` hierarchy. This is where you can create different virtual routing tables of many types, including MPLS, VPLS, virtual bridges, and virtual routers. In this case, it is a virtual router (the default configuration) that will house the connection to the other OSPF domain. The interface fe-0/0/3.0 command piece specifies an interface in the virtual-router.

Step 1

The rest of the configuration looks the same as any of the other OSPF configurations seen in previous chapters:

```
[edit routing-instances]
lab@R1# show
domain2 {
    interface fe-0/0/3.0;
    protocols {
        ospf {
            Area 0.0.0.0 {
                interface fe-0/0/3.0;
            }
        }
    }
}
```

After this configuration, it's necessary to see if the OSPF adjacency is up.

Step 2

This step will be a bit different than before in R1. Since a separate routing instance is being used for this domain, the command to see if there is an OSPF neighbor is:

```
lab@R1# run show ospf neighbor instance domain2
Address          Interface              State     ID               Pri  Dead
192.168.80.2     fe-0/0/3.0             Full      10.1.1.6         128    37
```

You can see from the above output that there is an OSPF adjacency to R6 and the state is full.

Step 3

The routing table will look very different than it has before, since there will be a new routing table representing our new routing instances:

```
lab@R1# run show route
inet.0: 15 destinations, 16 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 2w5d 07:08:02
                    > to 16.23.15.2 via fe-0/0/2.0
                    [OSPF/150] 2w5d 06:14:54, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
10.1.1.1/32        *[Direct/0] 3w2d 07:01:07
                    > via lo0.0
10.1.1.2/32        *[OSPF/10] 3w2d 06:58:59, metric 1
                    > to 192.168.1.2 via ge-0/0/0.0
10.1.1.4/32        *[OSPF/10] 00:10:52, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
10.10.2.99/32      *[Local/0] 3w2d 07:00:28
                      Reject
16.23.15.0/30      *[Direct/0] 2w5d 07:08:02
                    > via fe-0/0/2.0
```

```
16.23.15.1/32      *[Local/0] 2w5d 07:08:02
                      Local via fe-0/0/2.0
192.168.1.0/30     *[Direct/0] 3w2d 06:59:49
                    > via ge-0/0/0.0
192.168.1.1/32     *[Local/0] 3w2d 07:00:29
                      Local via ge-0/0/0.0
192.168.2.1/32     *[Local/0] 3w2d 07:00:29
                      Reject
192.168.3.0/30     *[OSPF/10] 00:11:02, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/10] 00:10:52, metric 4
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.29.0/24    *[OSPF/150] 00:10:52, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.30.0/24    *[OSPF/150] 00:10:52, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
224.0.0.5/32       *[OSPF/10] 3w2d 07:01:10, metric 1
                      MultiRecv
domain2.inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.1.1.6/32        *[OSPF/10] 00:31:42, metric 1
                    > to 192.168.80.2 via fe-0/0/3.0
172.19.21.1/32     *[OSPF/10] 00:31:42, metric 3
                    > to 192.168.80.2 via fe-0/0/3.0
172.19.22.1/32     *[OSPF/10] 00:31:42, metric 3
                    > to 192.168.80.2 via fe-0/0/3.0
192.168.80.0/30    *[Direct/0] 00:31:52
                    > via fe-0/0/3.0
192.168.80.1/32    *[Local/0] 00:31:52
                      Local via fe-0/0/3.0
192.168.80.4/30    *[OSPF/10] 00:31:42, metric 2
                    > to 192.168.80.2 via fe-0/0/3.0
224.0.0.5/32       *[OSPF/10] 00:31:52, metric 1
                      MultiRecv
```

As shown here, there is a new routing table called *domain2.inet.0* that contains the routes of the new OSPF domain. You can see the two host routes, but can you communicate to them from other routers in the network?

Step 4

Let's see if the routes have been put into the routing table:

```
lab@R2# run show route
inet.0: 17 destinations, 18 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 2w5d 07:13:00
                    > to 16.23.15.6 via fe-0/0/2.0
                    [OSPF/150] 2w5d 06:19:49, metric 0, tag 0
```

```
                    > to 192.168.1.1 via ge-0/0/0.0
10.1.1.1/32        *[OSPF/10] 3w2d 07:04:26, metric 1
                    > to 192.168.1.1 via ge-0/0/0.0
10.1.1.2/32        *[Direct/0] 3w2d 07:06:13
                    > via lo0.0
10.1.1.4/32        *[OSPF/10] 00:16:26, metric 1
                    > to 192.168.3.2 via ge-0/0/1.0
16.23.15.4/30      *[Direct/0] 2w5d 07:13:00
                    > via fe-0/0/2.0
16.23.15.5/32      *[Local/0] 2w5d 07:13:00
                      Local via fe-0/0/2.0
192.168.1.0/30     *[Direct/0] 3w2d 07:05:16
                    > via ge-0/0/0.0
192.168.1.2/32     *[Local/0] 3w2d 07:05:21
                      Local via ge-0/0/0.0
192.168.3.0/30     *[Direct/0] 00:16:37
                    > via ge-0/0/1.0
192.168.3.1/32     *[Local/0] 3w2d 07:05:21
                      Local via ge-0/0/1.0
192.168.16.0/21    *[OSPF/10] 00:16:26, metric 16777215
                      Discard
192.168.16.0/30    *[OSPF/10] 00:16:26, metric 2
                    > to 192.168.3.2 via ge-0/0/1.0
192.168.17.0/24    *[OSPF/10] 00:16:26, metric 3
                    > to 192.168.3.2 via ge-0/0/1.0
192.168.18.0/23    *[OSPF/10] 00:16:26, metric 2
                    > to 192.168.3.2 via ge-0/0/1.0
192.168.29.0/24    *[OSPF/150] 00:16:26, metric 0, tag 0
                    > to 192.168.3.2 via ge-0/0/1.0
192.168.30.0/24    *[OSPF/150] 00:16:26, metric 0, tag 0
                    > to 192.168.3.2 via ge-0/0/1.0
224.0.0.5/32       *[OSPF/10] 3w2d 07:06:14, metric 1
                      MultiRecv
```

And the routes to 172.19.21.1 and 172.19.22.1 are not present, so any communication to those network routes will end up using the 0/0 route, which is not what you want.

The goal is to get both domains to communicate with each other, of course, so let's move to the next section to find out how.

Communication from One OSPF Domain to the Other

Having two separate routing tables and two separate OSPF domains is nice if you are a service provider and want this type of separation, but within an enterprise network communication between the two domains is usually necessary. To accomplish this it's necessary to utilize a *rib-group,* which gives the user the ability to have a routing protocol place information in multiple routing tables.

In our case, inet.0 places information into domain2.inet.0 and vice versa. The configuration for this is not very complex, but it has been known to confuse network engineers.

Step 1

First let's configure R1:

```
set routing-options rib-groups sharing import-rib inet.0
set routing-options rib-groups sharing import-rib domain2.inet.0
set routing-options rib-groups sharing2 import-rib domain2.inet.0
set routing-options rib-groups sharing2 import-rib inet.0
```

It's important to note that a grouping is being created between two separate routing tables in this configuration. In this case it's the inet.0, which is the existing OSPF domain from our previous chapters, and domain2.inet.0, which is the new domain recently created. There are two separate groups created, both of which are important, one for each domain of OSPF. In the simplest terms, rib-groups share the routing tables between two domains.

MORE? There are options to filter out certain types of routes using routing policies associated with rib-groups, but they are beyond the scope of this book. For more information on rib-groups see: http://www.juniper.net/techpubs/en_US/junos11.3/information-products/topic-collections/config-guide-routing/index.html?topic-32752.html.

Step 2

Router R1's routing options configuration now looks like this:

```
lab@R1# show
static {
    route 0.0.0.0/0 next-hop 16.23.15.2;
}
rib-groups {
    sharing {
        import-rib [ inet.0 domain2.inet.0 ];
    }
    sharing2 {
        import-rib [ domain2.inet.0 inet.0 ];
    }
}
router-id 10.1.1.1;
```

Now that the rib-groups are created the next step is to apply them to the OSPF processes.

Step 3

Since R1 has two separate OSPF domains, the rib-groups will need to be applied to each one, which is done like this:

```
set protocols ospf rib-group sharing
set routing-instances domain2 protocols ospf rib-group sharing2
```

This configuration allows sharing of the routing tables between both OSPF domains. And the total OSPF configuration for both routers now looks like this:

```
[edit protocols ospf]
lab@R1# show
rib-group sharing;
export redistribution;
Area 0.0.0.0 {
    interface ge-0/0/0.0 {
        priority 255;
    }
    interface lo0.0 {
        passive;
    }
}
Area 0.0.0.1 {
    nssa {
        default-lsa {
            default-metric 1;
            type-7;
        }
        summaries;
    }
    interface ge-0/0/1.0 {
        interface-type p2p;
    }
}
[edit routing-instances]
lab@R1# show
domain2 {
    interface fe-0/0/3.0;
    protocols {
        ospf {
            rib-group sharing2;
            Area 0.0.0.0 {
                interface fe-0/0/3.0;
            }
        }
    }
}
```

This's all that is required to create two separate OSPF domains and allow them to communicate. As stated before, some enterprise networks may want to filter routes between domains, but that's not required to accomplish a separate domain OSPF network.

Verification of Functionality

With the configuration completed, the next task is to verify that both routing tables are complete with all routes and that the OSPF database is seeing the correct information.

Step 1

First you need to verify that both tables are seeing the same routes on R1:

```
lab@R1# run show route protocol ospf
inet.0: 19 destinations, 20 routes (19 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0           [OSPF/150] 00:41:04, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
10.1.1.2/32        *[OSPF/10] 00:41:04, metric 1
                    > to 192.168.1.2 via ge-0/0/0.0
10.1.1.4/32        *[OSPF/10] 00:41:04, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
10.1.1.6/32        *[OSPF/10] 00:36:24, metric 1
                    > to 192.168.80.2 via fe-0/0/3.0
172.19.21.1/32     *[OSPF/10] 00:36:24, metric 3
                    > to 192.168.80.2 via fe-0/0/3.0
172.19.22.1/32     *[OSPF/10] 00:36:24, metric 3
                    > to 192.168.80.2 via fe-0/0/3.0
192.168.3.0/30     *[OSPF/10] 00:41:04, metric 2
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/10] 00:41:04, metric 4
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.29.0/24    *[OSPF/150] 00:41:04, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.30.0/24    *[OSPF/150] 00:41:04, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
192.168.80.4/30    *[OSPF/10] 00:36:24, metric 2
                    > to 192.168.80.2 via fe-0/0/3.0
224.0.0.5/32       *[OSPF/10] 3w2d 08:09:53, metric 1
                      MultiRecv
domain2.inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[OSPF/150] 00:41:04, metric 0, tag 0
                    > to 192.168.1.2 via ge-0/0/0.0
10.1.1.2/32        *[OSPF/10] 00:41:04, metric 1
```

```
                      > to 192.168.1.2 via ge-0/0/0.0
10.1.1.4/32        *[OSPF/10] 00:41:04, metric 2
                      > to 192.168.1.2 via ge-0/0/0.0
10.1.1.6/32        *[OSPF/10] 00:36:24, metric 1
                      > to 192.168.80.2 via fe-0/0/3.0
172.19.21.1/32     *[OSPF/10] 00:36:24, metric 3
                      > to 192.168.80.2 via fe-0/0/3.0
172.19.22.1/32     *[OSPF/10] 00:36:24, metric 3
                      > to 192.168.80.2 via fe-0/0/3.0
192.168.3.0/30     *[OSPF/10] 00:41:04, metric 2
                      > to 192.168.1.2 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/10] 00:41:04, metric 4
                      > to 192.168.1.2 via ge-0/0/0.0
192.168.29.0/24    *[OSPF/150] 00:41:04, metric 0, tag 0
                      > to 192.168.1.2 via ge-0/0/0.0
192.168.30.0/24    *[OSPF/150] 00:41:04, metric 0, tag 0
                      > to 192.168.1.2 via ge-0/0/0.0
192.168.80.4/30    *[OSPF/10] 00:36:24, metric 2
                      > to 192.168.80.2 via fe-0/0/3.0
224.0.0.5/32       *[OSPF/10] 01:40:35, metric 1
                        MultiRecv
```

You can see that both tables have the same destinations. Now let's take a look at R2, and the routing table there:

```
lab@R2# run show route protocol ospf
inet.0: 21 destinations, 22 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0           [OSPF/150] 00:20:56, metric 0, tag 0
                      > to 192.168.1.1 via ge-0/0/0.0
10.1.1.1/32        *[OSPF/10] 00:20:56, metric 1
                      > to 192.168.1.1 via ge-0/0/0.0
10.1.1.3/32        *[OSPF/10] 00:20:56, metric 2
                      > to 192.168.1.1 via ge-0/0/0.0
10.1.1.4/32        *[OSPF/10] 00:30:11, metric 1
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.2.0/30     *[OSPF/10] 00:20:56, metric 2
                      > to 192.168.1.1 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/10] 00:30:11, metric 16777215
                        Discard
192.168.16.0/30    *[OSPF/10] 00:30:11, metric 2
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.17.0/24    *[OSPF/10] 00:30:11, metric 3
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.18.0/23    *[OSPF/10] 00:30:11, metric 2
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.29.0/24    *[OSPF/150] 00:30:11, metric 0, tag 0
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.30.0/24    *[OSPF/150] 00:30:11, metric 0, tag 0
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.71.0/24    *[OSPF/150] 00:20:56, metric 2, tag 0
```

```
                    > to 192.168.1.1 via ge-0/0/0.0
192.168.72.0/24    *[OSPF/150] 00:20:56, metric 2, tag 0
                    > to 192.168.1.1 via ge-0/0/0.0
224.0.0.5/32       *[OSPF/10] 3w2d 11:40:48, metric 1
                      MultiRecv
```

What is missing are the host routes learned from R7. The reason for this is seen in the OSPF database, for the inet.0 route table of R1:

```
root@R1# run show ospf database
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.1         10.1.1.1         0x800002bd   229  0x22 0xb8aa  48
Router   10.1.1.2         10.1.1.2         0x800002bd    84  0x22 0xcc92  48
Network *192.168.1.1      10.1.1.1         0x800002aa     3  0x22 0x3fa   32
Summary *10.1.1.3         10.1.1.1         0x80000002   481  0x22 0x6ab4  28
Summary  10.1.1.4         10.1.1.2         0x80000002   727  0x22 0x5ac2  28
Summary *192.168.2.0      10.1.1.1         0x80000003   833  0x22 0x467e  28
Summary  192.168.3.0      10.1.1.2         0x80000005  1156  0x22 0x318f  28
Summary  192.168.16.0     10.1.1.2         0x80000002   513  0x22 0xaa0e  28
ASBRSum *10.1.1.3         10.1.1.1         0x80000006   862  0x22 0x54c5  28
ASBRSum  10.1.1.4         10.1.1.2         0x8000000c  1376  0x22 0x38d9  28
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.1         10.1.1.1         0x80000005   229  0x20 0x9b84  48
Router   10.1.1.3         10.1.1.3         0x8000000a  1427  0x20 0xda1f  60
Summary *10.1.1.1         10.1.1.1         0x80000002  1009  0x20 0x9291  28
Summary *10.1.1.2         10.1.1.1         0x80000001  1373  0x20 0x948e  28
Summary *10.1.1.4         10.1.1.1         0x80000001  1373  0x20 0x8a95  28
Summary *192.168.1.0      10.1.1.1         0x80000008   229  0x20 0x655d  28
Summary *192.168.3.0      10.1.1.1         0x80000001  1373  0x20 0x675f  28
Summary *192.168.16.0     10.1.1.1         0x80000001  1373  0x20 0xdae0  28
NSSA    *0.0.0.0          10.1.1.1         0x80000003  1383  0x20 0x160d  36
NSSA     192.168.71.0     10.1.1.3         0x80000002   522  0x28 0x11c7  36
NSSA     192.168.72.0     10.1.1.3         0x80000001  2004  0x28 0x8d0   36
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *0.0.0.0          10.1.1.1         0x80000234   123  0x22 0x254b  36
Extern   0.0.0.0          10.1.1.2         0x80000003   298  0x22 0x871b  36
Extern   192.168.29.0     10.1.1.4         0x80000002   749  0x22 0x8b8f  36
Extern   192.168.30.0     10.1.1.4         0x80000002   142  0x22 0x8099  36
Extern  *192.168.71.0     10.1.1.1         0x80000002   362  0x22 0x934f  36
Extern  *192.168.72.0     10.1.1.1         0x80000002   242  0x22 0x8859  36
```

The routes are not present in the OSPF database for R1, therefore they will not be advertised to other routers. The reason for this is simply because they are considered completely external to the OSPF process of the original OSPF domain. With that said, a policy needs to be created to advertise these routes to the other routers within the OSPF domain. The same can be said for the OSPF domain 2.

Such a policy for exporting the routes learned from OSPF domain 2 is as follows:

```
set policy-options policy-statement from_domain2 term 1 from instance domain2
set policy-options policy-statement from_domain2 term 1 then tag 10
set policy-options policy-statement from_domain2 term 1 then accept
```

This configuration takes the routes from the routing table domain2 and sets the external tag to 10 (this can be whatever you want it to be), then accepts it. The next step is to apply configuration to the OSPF process.

Step 4

To apply the configuration to the OSPF process, use:

```
set protocols ospf export from_domain2
```

In total the configuration for OSPF and the associated policies is as follows:

```
root@R1# show
rib-group sharing;
export [ redistribution from_domain2 ];
area 0.0.0.0 {
    interface ge-0/0/0.0 {
        priority 255;
    }
    interface lo0.0 {
        passive;
    }
    interface fe-0/0/7.0 {
        disable;
    }
}
area 0.0.0.1 {
    nssa {
        default-lsa {
            default-metric 1;
            type-7;
        }
        summaries;
    }
    interface ge-0/0/1.0 {
        interface-type p2p;
    }
}
[edit routing-instances]
root@R1# show
domain2 {
```

```
    interface fe-0/0/3.0;
    protocols {
        ospf {
            rib-group sharing2;
            export from_domain1;
            area 0.0.0.0 {
                interface fe-0/0/3.0;
            }
        }
    }
}
[edit policy-options]
root@R1# show
policy-statement from_domain1 {
    term 1 {
        from protocol ospf;
        then {
            tag 11;
            accept;
        }
    }
}
policy-statement from_domain2 {
    term 1 {
        from instance domain2;
        then {
            tag 10;
            accept;
        }
    }
}
```

Now that the policy has been applied to the OSPF processes, Router R2 *should have* the complete route table. Let's see:

```
lab@R2# run show route protocol ospf
inet.0: 25 destinations, 26 routes (25 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0           [OSPF/150] 00:43:57, metric 0, tag 0
                    > to 192.168.1.1 via ge-0/0/0.0
10.1.1.1/32        *[OSPF/10] 00:43:57, metric 1
                    > to 192.168.1.1 via ge-0/0/0.0
10.1.1.3/32        *[OSPF/10] 00:43:57, metric 2
                    > to 192.168.1.1 via ge-0/0/0.0
10.1.1.4/32        *[OSPF/10] 00:53:12, metric 1
                    > to 192.168.3.2 via ge-0/0/1.0
10.1.1.6/32        *[OSPF/150] 00:09:49, metric 1, tag 10
                    > to 192.168.1.1 via ge-0/0/0.0
172.19.21.1/32     *[OSPF/150] 00:09:49, metric 3, tag 10
                    > to 192.168.1.1 via ge-0/0/0.0
172.19.22.1/32     *[OSPF/150] 00:09:49, metric 3, tag 10
```

```
                      > to 192.168.1.1 via ge-0/0/0.0
192.168.2.0/30     *[OSPF/10] 00:43:57, metric 2
                      > to 192.168.1.1 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/10] 00:53:12, metric 16777215
                        Discard
192.168.16.0/30    *[OSPF/10] 00:53:12, metric 2
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.17.0/24    *[OSPF/10] 00:53:12, metric 3
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.18.0/23    *[OSPF/10] 00:53:12, metric 2
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.29.0/24    *[OSPF/150] 00:53:12, metric 0, tag 0
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.30.0/24    *[OSPF/150] 00:53:12, metric 0, tag 0
                      > to 192.168.3.2 via ge-0/0/1.0
192.168.71.0/24    *[OSPF/150] 00:43:57, metric 2, tag 0
                      > to 192.168.1.1 via ge-0/0/0.0
192.168.72.0/24    *[OSPF/150] 00:43:57, metric 2, tag 0
                      > to 192.168.1.1 via ge-0/0/0.0
192.168.80.4/30    *[OSPF/150] 00:09:49, metric 2, tag 10
                      > to 192.168.1.1 via ge-0/0/0.0
224.0.0.5/32       *[OSPF/10] 3w2d 12:03:49, metric 1
                        MultiRecv
```

The routes from R7 are indeed in place in the table, and the OSPF database is showing them correctly as external routes.

```
lab@R2# run show ospf database area 0
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.1         10.1.1.1         0x800002be   632  0x22 0xb6ab  48
Router  *10.1.1.2         10.1.1.2         0x800002bd  1384  0x22 0xcc92  48
Network  192.168.1.1      10.1.1.1         0x800002aa  1305  0x22 0x3fa   32
Summary  10.1.1.3         10.1.1.1         0x80000003   137  0x22 0x68b5  28
Summary *10.1.1.4         10.1.1.2         0x80000002  2027  0x22 0x5ac2  28
Summary  192.168.2.0      10.1.1.1         0x80000004   262  0x22 0x447f  28
Summary *192.168.3.0      10.1.1.2         0x80000005  2456  0x22 0x318f  28
Summary *192.168.16.0     10.1.1.2         0x80000002  1813  0x22 0xaa0e  28
ASBRSum  10.1.1.3         10.1.1.1         0x80000007   395  0x22 0x52c6  28
ASBRSum *10.1.1.4         10.1.1.2         0x8000000c  2677  0x22 0x38d9  28
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   0.0.0.0          10.1.1.1         0x80000234  1425  0x22 0x254b  36
Extern  *0.0.0.0          10.1.1.2         0x80000003  1600  0x22 0x871b  36
Extern   10.1.1.6         10.1.1.1         0x80000001   633  0x22 0x7a0e  36
Extern   172.19.21.1      10.1.1.1         0x80000001   633  0x22 0xc8f9  36
Extern   172.19.22.1      10.1.1.1         0x80000001   633  0x22 0xbd04  36
Extern   192.168.29.0     10.1.1.4         0x80000002  2050  0x22 0x8b8f  36
Extern   192.168.30.0     10.1.1.4         0x80000002  1443  0x22 0x8099  36
Extern   192.168.71.0     10.1.1.1         0x80000003    13  0x22 0x9150  36
Extern   192.168.72.0     10.1.1.1         0x80000002  1545  0x22 0x8859  36
Extern   192.168.80.4     10.1.1.1         0x80000001   633  0x22 0xfae3  36
```

You can tell from this output that the routes are in the database correctly.

The next verification confirms that all of the routes from the original OSPF domain are showing up within OSPF domain 2:

```
root@R6# run show route protocol ospf
inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[OSPF/150] 00:34:54, metric 0, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
10.1.1.2/32        *[OSPF/150] 00:34:54, metric 1, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
10.1.1.3/32        *[OSPF/150] 00:34:54, metric 1, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
10.1.1.4/32        *[OSPF/150] 00:34:54, metric 2, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
172.19.21.1/32     *[OSPF/10] 00:56:18, metric 2
                    > to 192.168.80.6 via ge-0/0/1.0
172.19.22.1/32     *[OSPF/10] 00:56:18, metric 2
                    > to 192.168.80.6 via ge-0/0/1.0
192.168.3.0/30     *[OSPF/150] 00:34:54, metric 2, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
192.168.16.0/21    *[OSPF/150] 00:34:54, metric 4, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
192.168.29.0/24    *[OSPF/150] 00:34:54, metric 0, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
192.168.30.0/24    *[OSPF/150] 00:34:54, metric 0, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
192.168.71.0/24    *[OSPF/150] 00:34:54, metric 2, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
192.168.72.0/24    *[OSPF/150] 00:34:54, metric 2, tag 11
                    > to 192.168.80.1 via ge-0/0/0.0
224.0.0.5/32       *[OSPF/10] 00:57:34, metric 1
                      MultiRecv
```

You can see that all of the OSPF routes from the previous OSPF domain are inserted into the routing table of R6.

Of course, this means that the routes will appear in the OSPF database as external, as expected. Let’s check:

```
root@R6# run show ospf database area 0
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.6         10.1.1.6         0x8000000c   501  0x22 0x1951  48
Router   192.168.80.1     192.168.80.1     0x8000000a   726  0x22 0x2617  36
Network  192.168.80.1     192.168.80.1     0x80000004  1743  0x22 0x57a0  32
Summary *172.19.21.1      10.1.1.6         0x80000002  1735  0x22 0x72df  28
Summary *172.19.22.1      10.1.1.6         0x80000002  1413  0x22 0x67e9  28
Summary *192.168.80.4     10.1.1.6         0x80000004  1090  0x22 0xa0cb  28
```

```
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   0.0.0.0          192.168.80.1     0x80000002  1267  0x22 0x816a  36
Extern   10.1.1.2         192.168.80.1     0x80000002  1029  0x22 0xddfe  36
Extern   10.1.1.3         192.168.80.1     0x80000002   791  0x22 0xd308  36
Extern   10.1.1.4         192.168.80.1     0x80000002   554  0x22 0xd306  36
Extern   192.168.3.0      192.168.80.1     0x80000002   316  0x22 0xb0cf  36
Extern   192.168.16.0     192.168.80.1     0x80000002    78  0x22 0x2451  36
Extern   192.168.29.0     192.168.80.1     0x80000001  2126  0x22 0x91d4  36
Extern   192.168.30.0     192.168.80.1     0x80000001  2126  0x22 0x86de  36
Extern   192.168.71.0     192.168.80.1     0x80000001  2126  0x22 0xd564  36
Extern   192.168.72.0     192.168.80.1     0x80000001  2126  0x22 0xca6e  36
```

It's clear that now R1 and R6 are both ABR's and ASBR's. They are both receiving and sending type-4 and type-5 LSA's to each other.

Summary

This chapter has shown that using multiple OSPF domains can be very useful to large organizations that have different regions or during the integration of two separate enterprises. You are able to maintain a single routing process for ease of troubleshooting and keep the knowledge of complex routing protocols within the operations organization.

Multiple OSPF domains can also be used as a mechanism to integrate two large enterprises in an acquisition or merger situation. Network engineers must be careful in such situations, however, because in the case of dual connections, one could easily create routing loops since the routes between OSPF domains will be flagged as external. Careful use of policies and how they are tagged during export, combined with metrics, can alleviate most of the problems.

Let's continue our OSPF exploration with multiple autonomous systems (AS) in Chapter 6.

Chapter 6

Multiple Autonomous Systems

In the previous chapter, multiple homogeneous domains were used to show two different OSPF routing domains for large enterprises that might have offices in multiple regions, or might need them for the purpose of integration. Another routing domain option for these purposes is using a separate routing protocol and/or static routes between the two domains. This technique is becoming more and more prevalent with the advent of carrier-provided MPLS.

Provider MPLS L3VPNs usually require the enterprise network to use static routes or BGP to connect to them. This requires either converting to all one protocol or to *multiple autonomous systems* routing management. The latter kind of management may be a bit more complex because of redistribution and the support of another routing protocol, but the advantages are the clean separation of the two autonomous systems within the network. The redistribution process adds more control over what routes are seen between the two systems as well.

In this chapter a new routing protocol, BGP, is used for connectivity between the two autonomous systems. The chapter does not cover how BGP works in any depth, it simply shows how the routes learned from BGP are interpreted by OSPF.

MORE? For more information on BGP as it pertains to the enterprise see *Juniper Enterprise Routing, 2nd Edition*, by Peter Southwick, Doug Marshcke, and Harry Reynolds, at http://www.juniper.net/books.

At the end of this chapter you should be able to connect the two autonomous systems using a redistribution process, and you should see the routes in the OSPF database and the OSPF routing table. You should also be able to verify the functionality of the process and understand how to interpret the OSPF database.

Configuration of the Separate Autonomous System

This exercise adds routers R8 and R9. These routers are a separate autonomous system having their own OSPF backbone Area 0, represented by R8, and another OSPF Area 1, represented by R9, as shown in Figure 6.1.

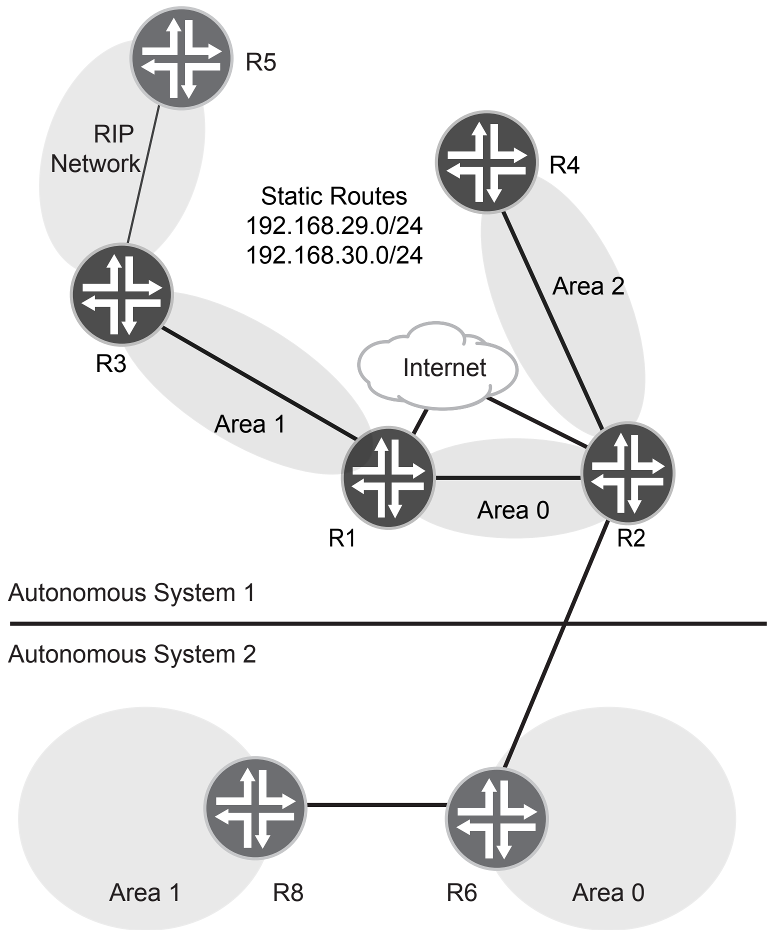

Figure 6.1 OSPF Network with Multiple Autonomous Systems

The routing protocol BGP will be used to connect the two separate autonomous systems. The type of BGP used will be EBGP since two separate autonomous system IDs are being used. Also, with the addition of two more routers in the network, it's necessary to update the IP address and interface list. And these are listed in Table 6.1.

Table 6.1 Updated IP Address and Interface List for Chapter 6

Link	Interface	IP	Interface	IP
R1 - R2	ge-0/0/0.0	192.168.1.1/30	ge-0/0/0.0	192.168.1.2/30
R1 - R3	ge-0/0/1.0	192.168.2.1/30	ge-0/0/0.0	192.168.2.2/30
R2 - R4	ge-0/0/1.0	192.168.3.1/30	ge-0/0/0.0	192.168.3.2/30
R1 loopback	lo0.0	10.1.1.1/32		
R2 loopback	lo0.0	10.1.1.2/32		
R3 loopback	lo0.0	10.1.1.3/32		
R4 loopback	lo0.0	10.1.1.4/32		
R5 – R3	ge-0/0/0.0	192.168.70.2/30	ge-0/0/1.0	192.168.70.1/30
R1 - internet	fe-0/0/2.0	16.23.15.1/30		
R2 - internet	fe-0/0/2.0	16.23.15.5/30		
R6 – R1	ge-0/0/0.0	192.168.80.2/30	fe-0/0/3.0	192.168.80.1/30
R6 – R7	ge-0/0/1.0	192.168.80.5/30	ge-0/0/0.0	192.168.80.6/30
R6 loopback	lo0.0	10.1.1.6/32		
R7 loopback	lo0.0	10.1.1.7/32		
R8 – R9	Ge-0/0/1.0	192.168.90.5/30	Ge-0/0/0.0	192.168.90.6/30
R8 – R2	Ge-0/0/0.0	192.168.90.2/30	Fe-0/0/4.0	192.168.90.1/30
R8 loopback	lo0.0	10.1.1.8/32		
R9 loopback	lo0.0	10.1.1.9/32		

With the design set, let's start configuring the new additions.

Step 1

The OSPF configurations for R8 and R9 are listed below:

```
[edit protocols]
root@R8# show
ospf {
    Area 0.0.0.0 {
        interface lo0.0;   {
               passive;
```

```
        }
    }
    Area 0.0.0.1 {
        interface ge-0/0/1.0;
    }
}
[edit protocols]
root@R9# show
ospf {
    Area 0.0.0.1 {
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
        interface fe-0/0/2.0;
        interface lo0.0 {
            passive;
        }
    }
}
```

You should notice that the configuration is a standard OSPF approach. All of the basic elements are there, a backbone Area 0, and a remote Area 1 with interfaces to external sources that the core needs to reach.

Let's check R8's OSPF routing table, and database:

```
root@R8# run show ospf route
Topology default Route Table:
Prefix              Path   Route       NH   Metric  NextHop        Nexthop
                    Type   Type        Type         Interface      addr/label
10.1.1.9            Intra  Router      IP        1  ge-0/0/1.0     192.168.90.6
10.1.1.8/32         Intra  Network     IP        0  lo0.0
10.1.1.9/32         Intra  Network     IP        1  ge-0/0/1.0     192.168.90.6
192.168.90.4/30     Intra  Network     IP        1  ge-0/0/1.0
192.168.91.0/24     Intra  Network     IP        2  ge-0/0/1.0     192.168.90.6
192.168.92.0/24     Intra  Network     IP        2  ge-0/0/1.0     192.168.90.6
root@R8# run show ospf database
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.8         10.1.1.8         0x8000000b     3  0x22 0x2e7    36
Summary *10.1.1.9         10.1.1.8         0x80000003     3  0x22 0x20f    28
Summary *192.168.90.4     10.1.1.8         0x8000000c     3  0x22 0x1642   28
Summary *192.168.91.0     10.1.1.8         0x80000006     3  0x22 0x5b02   28
Summary *192.168.92.0     10.1.1.8         0x80000006     3  0x22 0x500c   28
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.8         10.1.1.8         0x80000017     3  0x22 0x4619   36
Router   10.1.1.9         10.1.1.9         0x80000014     4  0x22 0xc8c7   72
Network  192.168.90.6     10.1.1.9         0x8000000b     4  0x22 0xb377   32
Summary *10.1.1.8         10.1.1.8         0x80000008     3  0x22 0xf716   28
```

All of the routes are present and accounted for in the OSPF database. The next step is getting this network integrated into the existing OSPF network that was created in the earlier chapters of this book.

Configuration of the BGP Network

As stated previously, diving deeply into BGP is not the goal of this chapter, but it is necessary to show the BGP configuration and neighboring to complete the exercise. So, in this section, the goal is to configure a basic EBGP neighboring between R8 and R2, and to verify that they are in a neighboring state.

Step 1

First the configuration of the autonomous system ID:

```
set routing-options autonomous-system 65001
```

This is required when using BGP since BGP uses AS numbers to create route paths.

Step 2

Now configure the BGP routing protocol:

```
set protocols bgp group autonomous type external
set protocols bgp group autonomous neighbor 192.168.90.2 peer-as 65002
```

The purpose of this configuration is for you to get the very basics of BGP routing. The required configuration tells the router what type of BGP to use. In this example, `external` is displayed in the configuration since it's peering to a different autonomous system number (EBGP). Next the neighbor IP address and the neighbor's `peer-as` number to peer with are needed. Again, these are the very basics of BGP configuration and all that is needed to display this example.

The complete BGP stanza configuration shows as:

```
[edit protocols]
lab@R2# show
bgp {
    group autonomous {
        type external;
        neighbor 192.168.90.2 {
            peer-as 65002;
        }
    }
}
```

Step 3

Now let's configure R8, which is almost exactly the same as R2's:

```
set routing-options autonomous-system 65002
set protocols bgp group autonomous peer-as 65001
set protocols bgp group autonomous neighbor 192.168.90.1
```

The configuration difference between Router R2 and R8 is basically the peer-as and the neighbor IP address. The configuration stanza looks like this for R8:

```
[edit protocols]
root@R8# show
bgp {
    group autonomous {
        peer-as 65001;
        neighbor 192.168.90.1;
    }
}
```

Now that both routers are configured for BGP, they should be in a peering state. Let's see.

Step 4

The command issued on both routers is below:

```
lab@R2# run show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/
Dwn State|#Active/Received/Accepted/Damped...
192.168.90.2             65002         38         35       0       1       14:50
0/0/0/0       0/0/0/0

root@R8# run show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/
Dwn State|#Active/Received/Accepted/Damped...
192.168.90.1             65001        444        450       0       3       13:55
0/0/0/0        0/0/0/0
```

There is a peering relationship between the two routers and BGP is up, but there are no routes being shared between the routers as shown in the bolded output. The reason for this is clear, our learned OSPF routes between the routers are not being shared and a redistribution between the two devices is needed.

Connecting the Two Autonomous Systems

You can use what you learned about redistribution in Chapter 4 to put the OSPF routes into BGP and send them to another autonomous network.

Step 1

The configuration is as follows:

```
set policy-options policy-statement ospf-to-bgp term 1 from protocol ospf
set policy-options policy-statement ospf-to-bgp term 1 then accept
```

This policy puts all of the routes learned from OSPF into BGP as an export within the BGP configuration once applied to the routing protocol.

Step 2

Let's apply it:

```
set protocols bgp group autonomous export ospf-to-bgp
```

Step 3

Now let's show a BGP summary:

```
root@R8# run show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                19         19          0          0          0          0
Peer                  AS      InPkt     OutPkt    OutQ   Flaps Last Up/
Dwn State|#Active/Received/Accepted/Damped...
192.168.90.1     65001        521        523       0       3       46:23 19/19/19/0
0/0/0/0
```

You can see that routes are being learned from R2 via the BGP protocol. It shows that there are 19 routes learned via BGP, but are these routes in the OSPF database? Let's take a look:

```
root@R8# run show ospf database
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.8         10.1.1.8         0x80000012   108  0x22 0xf3ee  36
Summary *10.1.1.9         10.1.1.8         0x80000008   550  0x22 0xf714  28
Summary *192.168.90.4     10.1.1.8         0x80000012   108  0x22 0xa48   28
Summary *192.168.91.0     10.1.1.8         0x8000000a   180  0x22 0x5306  28
```

```
Summary *192.168.92.0     10.1.1.8          0x80000009  1442  0x22 0x4a0f  28
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   *10.1.1.8          10.1.1.8          0x8000001d   108  0x22 0x3a1f  36
Router    10.1.1.9          10.1.1.9          0x80000017  1443  0x22 0xc2ca  72
Network   192.168.90.6      10.1.1.9          0x8000000e  1443  0x22 0xad7a  32
Summary *10.1.1.8          10.1.1.8          0x8000000b  1442  0x22 0xf119  28
```

There are no routes from the original OSPF Autonomous system in the OSPF database of R8. This is because it's necessary to redistribute the BGP routes *into* the OSPF process.

Step 4

The configuration to redistribute the BGP routes in the OSPF process is as follows:

```
set policy-options policy-statement bgp-to-ospf term 1 from protocol bgp
set policy-options policy-statement bgp-to-ospf term 1 then accept
set protocols ospf export bgp-to-ospf
```

With this addition, the OSPF database for R8 now looks like this:

```
root@R8# run show ospf database
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   *10.1.1.8          10.1.1.8          0x80000012   485  0x22 0xf3ee  36
Summary *10.1.1.9          10.1.1.8          0x80000008   927  0x22 0xf714  28
Summary *192.168.90.4      10.1.1.8          0x80000012   485  0x22 0xa48   28
Summary *192.168.91.0      10.1.1.8          0x8000000a   557  0x22 0x5306  28
Summary *192.168.92.0      10.1.1.8          0x8000000a   187  0x22 0x4810  28
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   *10.1.1.8          10.1.1.8          0x8000001d   485  0x22 0x3a1f  36
Router    10.1.1.9          10.1.1.9          0x80000017  1820  0x22 0xc2ca  72
Network   192.168.90.6      10.1.1.9          0x8000000e  1820  0x22 0xad7a  32
Summary *10.1.1.8          10.1.1.8          0x8000000c    82  0x22 0xef1a  28
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   *10.1.1.1          10.1.1.8          0x80000001   485  0x22 0xcdc2  36
Extern   *10.1.1.3          10.1.1.8          0x80000001   485  0x22 0xc3c9  36
Extern   *10.1.1.4          10.1.1.8          0x80000001   485  0x22 0xafdd  36
Extern   *10.1.1.6          10.1.1.8          0x80000001   485  0x22 0x9bef  36
Extern   *172.19.21.1       10.1.1.8          0x80000001   485  0x22 0xe9db  36
Extern   *172.19.22.1       10.1.1.8          0x80000001   485  0x22 0xdee5  36
Extern   *192.168.2.0       10.1.1.8          0x80000001   485  0x22 0xa192  36
Extern   *192.168.16.0      10.1.1.8          0x80000001   485  0x22 0xd755  36
Extern   *192.168.16.3      10.1.1.8          0x80000001   485  0x22 0xe83a  36
Extern   *192.168.17.0      10.1.1.8          0x80000001   485  0x22 0x1809  36
Extern   *192.168.18.0      10.1.1.8          0x80000001   485  0x22 0xfd24  36
Extern   *192.168.29.0      10.1.1.8          0x80000001   485  0x22 0x75a2  36
```

```
Extern  *192.168.30.0     10.1.1.8         0x80000001   485  0x22 0x6aac  36
Extern  *192.168.71.0     10.1.1.8         0x80000001   485  0x22 0xb932  36
Extern  *192.168.72.0     10.1.1.8         0x80000001   485  0x22 0xae3c  36
Extern  *192.168.75.0     10.1.1.8         0x80000001   485  0x22 0x7b6f  36
Extern  *192.168.76.0     10.1.1.8         0x80000001   485  0x22 0x8c59  36
Extern  *192.168.77.0     10.1.1.8         0x80000001   485  0x22 0x8163  36
Extern  *192.168.80.4     10.1.1.8         0x80000001   485  0x22 0x1cc5  36
```

Now, all of the routes appear in the OSPF database and will be advertised to R9 via OSPF. Let's double-check:

```
root@R9# run show ospf route
Topology default Route Table:
Prefix              Path   Route       NH   Metric  NextHop       Nexthop
                    Type   Type        Type         Interface     addr/label
10.1.1.8            Intra  Area/AS BR  IP        1  ge-0/0/0.0    192.168.90.5
10.1.1.1/32         Ext2   Network     IP        1  ge-0/0/0.0    192.168.90.5
10.1.1.3/32         Ext2   Network     IP        2  ge-0/0/0.0    192.168.90.5
10.1.1.4/32         Ext2   Network     IP        1  ge-0/0/0.0    192.168.90.5
10.1.1.6/32         Ext2   Network     IP        1  ge-0/0/0.0    192.168.90.5
10.1.1.8/32         Inter  Network     IP        1  ge-0/0/0.0    192.168.90.5
10.1.1.9/32         Intra  Network     IP        0  lo0.0
172.19.21.1/32      Ext2   Network     IP        3  ge-0/0/0.0    192.168.90.5
172.19.22.1/32      Ext2   Network     IP        3  ge-0/0/0.0    192.168.90.5
192.168.2.0/30      Ext2   Network     IP        2  ge-0/0/0.0    192.168.90.5
192.168.16.0/21     Ext2   Network     IP   16777214 ge-0/0/0.0   192.168.90.5
192.168.16.0/30     Ext2   Network     IP        2  ge-0/0/0.0    192.168.90.5
192.168.17.0/24     Ext2   Network     IP        3  ge-0/0/0.0    192.168.90.5
192.168.18.0/23     Ext2   Network     IP        2  ge-0/0/0.0    192.168.90.5
192.168.29.0/24     Ext2   Network     IP        0  ge-0/0/0.0    192.168.90.5
192.168.30.0/24     Ext2   Network     IP        0  ge-0/0/0.0    192.168.90.5
192.168.71.0/24     Ext2   Network     IP        2  ge-0/0/0.0    192.168.90.5
192.168.72.0/24     Ext2   Network     IP        2  ge-0/0/0.0    192.168.90.5
192.168.75.0/30     Ext2   Network     IP        2  ge-0/0/0.0    192.168.90.5
192.168.76.0/24     Ext2   Network     IP        3  ge-0/0/0.0    192.168.90.5
192.168.77.0/24     Ext2   Network     IP        3  ge-0/0/0.0    192.168.90.5
192.168.80.4/30     Ext2   Network     IP        2  ge-0/0/0.0    192.168.90.5
192.168.90.4/30     Intra  Network     IP        1  ge-0/0/0.0
192.168.91.0/24     Intra  Network     IP        1  ge-0/0/1.0
192.168.92.0/24     Intra  Network     IP        1  fe-0/0/2.0
```

All of the OSPF routes from the original autonomous system are now in the new autonomous system. The last step is reversing the configuration and adding the new OSPF AS into the original OSPF AS.

The total protocol configuration for R8 and R2 is:

```
[edit protocols]
lab@R2# show
bgp {
    group autonomous {
        type external;
        export ospf-to-bgp;
        neighbor 192.168.90.2 {
            peer-as 65002;
        }
    }
}
ospf {
    export [ redistribution bgp-to-ospf ];
    area 0.0.0.0 {
        interface ge-0/0/0.0 {
            priority 100;
        }
        interface lo0.0 {
            passive;
        }
    }
    area 0.0.0.2 {
        area-range 192.168.16.0/21;
        interface ge-0/0/1.0 {
            interface-type p2p;
        }
    }
}
[edit protocols]
root@R8# show
bgp {
    group autonomous {
        peer-as 65001;
        neighbor 192.168.90.1;
    }
}
ospf {
    export bgp-to-ospf;
    area 0.0.0.0 {
        interface lo0.0 {
            passive;
        }
    }
    area 0.0.0.1 {
        interface ge-0/0/1.0;
    }
}
```

Lastly, from the above output there will be type-4 and type-5 LSA's shared between R8 and R2. Both of these routers will be classified as ABR's, ASBR's, and BR's.

Summary

Using multiple autonomous systems to connect networks together is a common practice. It provides a very clean separation between networks and allows for separate management of those networks as well.

Compared to separate OSPF domains from the previous chapter, using multiple autonomous systems can keep routing processes separate due to the use of a second routing protocol. The drawback, of course, is understanding and managing this other protocol. There are also redistribution considerations that need to be made when using multiple autonomous systems, and if multiple routers are used, care must be taken to prevent loops through the tagging, and careful manipulation of the routing tables.

Our final chapter, *Chapter 7: Virutal Links*, is next.

Chapter 7

Virtual Links

Virtual links within OSPF have two basic functions. The first is to connect a physically discontiguous backbone Area 0 and the other is to connect a physically detached area to the backbone Area 0 through another area.

One may ask, when would this be useful? Well in the event of company mergers, as mentioned in previous examples, there will be multiple backbone Area 0s that are physically separated. In most cases the long term plan would be to physically connect the backbone areas, but as an interim solution one could extend an existing outlying area and connect the newly obtained backbone Area 0 to the existing one. The outlying area would be a transit area for the backbone Area 0, and allow connectivity of the two area border routers of the newly formed Area 0 connection through the outlying area's network. This is called *OSPF virtual links*, and it is described in RFC2328.

In this chapter you learn how to configure the virtual OSPF link and verify its functionality across the network. The chapter also updates our existing network topology as shown in Figure 7.1.

Configuration of the Virtual Link

You can see that with the addition of R10, and with Area 3 connected to it, there is no direct path to the backbone Area 0, and so it requires a virtual link. The outlying Area 2 will be a transit area logically connecting R10 to R2 as a backbone ABR router.

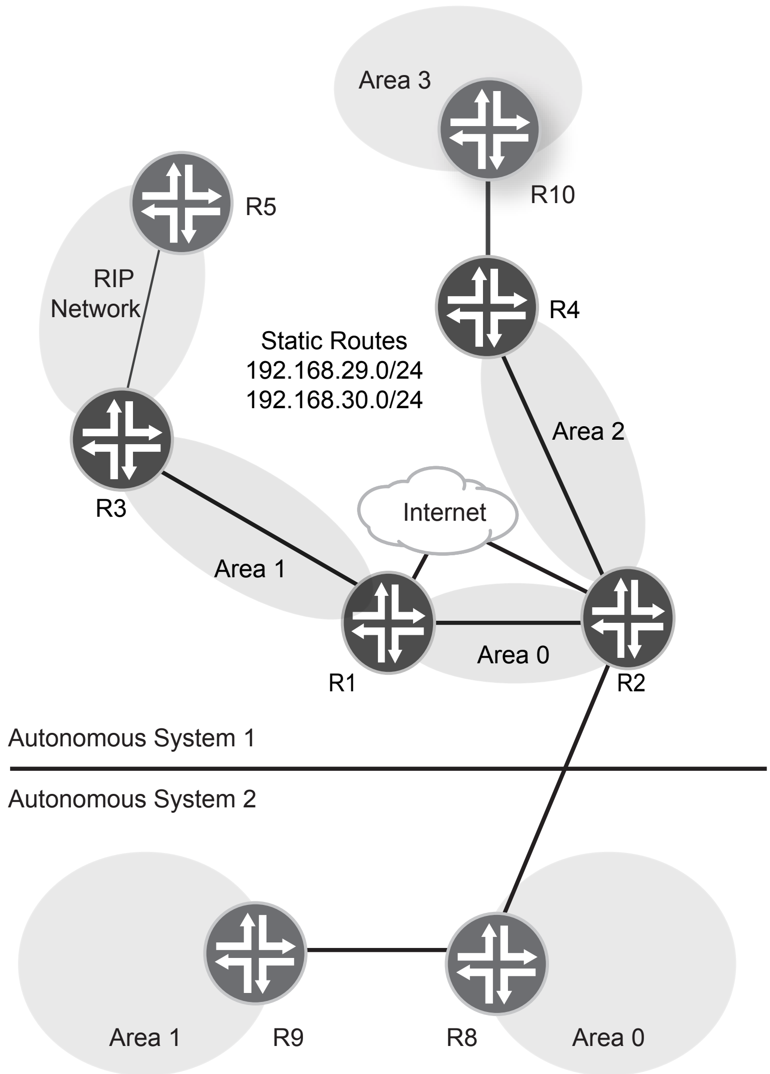

Figure 7.1 Example Network with Virtual Link

The interface and IP addresses for our virtual link assignment are found in Table 7.1.

Table 7.1 Interface and IP Addresses for Virtual Links

Link	Interface	IP	Interface	IP
R1 - R2	ge-0/0/0.0	192.168.1.1/30	ge-0/0/0.0	192.168.1.2/30
R1 - R3	ge-0/0/1.0	192.168.2.1/30	ge-0/0/0.0	192.168.2.2/30
R2 - R4	ge-0/0/1.0	192.168.3.1/30	ge-0/0/0.0	192.168.3.2/30
R1 loopback	lo0.0	10.1.1.1/32		
R2 loopback	lo0.0	10.1.1.2/32		
R3 loopback	lo0.0	10.1.1.3/32		
R4 loopback	lo0.0	10.1.1.4/32		
R5 – R3	ge-0/0/0.0	192.168.70.2/30	ge-0/0/1.0	192.168.70.1/30
R1 - internet	fe-0/0/2.0	16.23.15.1/30		
R2 - internet	fe-0/0/2.0	16.23.15.5/30		
R6 – R1	ge-0/0/0.0	192.168.80.2/30	fe-0/0/3.0	192.168.80.1/30
R6 – R7	ge-0/0/1.0	192.168.80.5/30	ge-0/0/0.0	192.168.80.6/30
R6 loopback	lo0.0	10.1.1.6/32		
R7 loopback	lo0.0	10.1.1.7/32		
R8 – R9	Ge-0/0/1.0	192.168.90.5/30	Ge-0/0/0.0	192.168.90.6/30
R8 – R2	Ge-0/0/0.0	192.168.90.2/30	Fe-0/0/4.0	192.168.90.1/30
R8 loopback	lo0.0	10.1.1.8/32		
R9 loopback	lo0.0	10.1.1.9/32		
R10 – R4	Ge-0/0/0.0	192.168.75.2/30	Ge-0/0/3.0	192.168.75.1/30
R10 loopback	lo0.0	10.1.1.10/32		

For the virtual link to work, each router in the virtual path is an ABR, so in this case routers R10 and R2 will be the ABRs, with R4 as the transit for the virtual link.

Step 1

The initial configuration of R10 is:

```
[edit protocols ospf]
root@R10# show
Area 0.0.0.2 {
    interface ge-0/0/0.0;
}
Area 0.0.0.3 {
    interface ge-0/0/1.0;
    interface fe-0/0/2.0;
}
```

With this configuration you can see that R2 will not receive any routes from R10's Area 3. Let's confirm:

```
lab@R2# run show route 192.168.76.0
inet.0: 25 destinations, 26 routes (25 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 00:27:49
                    > to 16.23.15.6 via fe-0/0/2.0
                    [OSPF/150] 00:26:57, metric 0, tag 0
                    > to 192.168.1.1 via ge-0/0/0.0
```

Instead it's pointing to the default route that was added in Chapter 4.

From R10's perspective, there is a neighboring relationship to R4, and LSA's are being sent and received, so the LSA database looks like the following:

```
root@R10# run show ospf database
    OSPF database, Area 0.0.0.2
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.2         10.1.1.2         0x80000004  1154  0x22 0x9b7e  48
Router   10.1.1.4         10.1.1.4         0x8000000a  1796  0x22 0xc993 108
Router  *10.1.1.10        10.1.1.10        0x80000006   703  0x22 0xa4ee  36
Router   192.168.16.2     192.168.16.2     0x80000003  2430  0x22 0xdfbc  60
Network *192.168.75.2     10.1.1.10        0x80000003   919  0x22 0x5dea  32
Summary  10.1.1.1         10.1.1.2         0x80000002   948  0x22 0x78a7  28
Summary  10.1.1.2         10.1.1.2         0x80000003   148  0x22 0x62bc  28
Summary  10.1.1.3         10.1.1.2         0x80000002   748  0x22 0x6eae  28
Summary  192.168.1.0      10.1.1.2         0x8000000d  1154  0x22 0x3783  28
Summary  192.168.2.0      10.1.1.2         0x80000002   548  0x22 0x4c77  28
Summary *192.168.76.0     10.1.1.10        0x80000004  1188  0x22 0xee7e  28
Summary *192.168.77.0     10.1.1.10        0x80000003  1188  0x22 0xe587  28
ASBRSum  10.1.1.1         10.1.1.2         0x8000000a  1154  0x22 0x5abc  28
ASBRSum  10.1.1.3         10.1.1.2         0x80000002   348  0x22 0x60bb  28
    OSPF database, Area 0.0.0.3
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.10        10.1.1.10        0x80000002  1148  0x22 0xe5a0  48
```

```
Summary *10.1.1.4        10.1.1.10        0x80000002   487  0x22 0x2aea  28
Summary *192.168.3.0     10.1.1.10        0x80000002   271  0x22 0x11a9  28
Summary *192.168.16.0    10.1.1.10        0x80000002    57  0x22 0x812c  28
Summary *192.168.17.0    10.1.1.10        0x80000001  1188  0x22 0x9415  28
Summary *192.168.18.0    10.1.1.10        0x80000001  1188  0x22 0x7a30  28
Summary *192.168.75.0    10.1.1.10        0x80000004  1151  0x22 0xe789  28
ASBRSum *10.1.1.2        10.1.1.10        0x80000004  1151  0x22 0x36dc  28
ASBRSum *10.1.1.4        10.1.1.10        0x80000004  1151  0x22 0x18f9  28
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   0.0.0.0          10.1.1.1         0x80000003  1169  0x22 0x8d16  36
Extern   0.0.0.0          10.1.1.2         0x80000002  1748  0x22 0x891a  36
Extern   10.1.1.6         10.1.1.1         0x80000002  1046  0x22 0x780f  36
Extern   192.168.29.0     10.1.1.4         0x80000002  1250  0x22 0x8b8f  36
Extern   192.168.30.0     10.1.1.4         0x80000002   422  0x22 0x8099  36
Extern   192.168.71.0     10.1.1.1         0x80000002  1417  0x22 0x934f  36
Extern   192.168.72.0     10.1.1.1         0x80000002  1293  0x22 0x8859  36
```

From R10's perspective everything looks normal and it is sending out LSA's for Area 3.

Now let's take a look at R2's LSA database to determine why these routes are not being accepted:

```
lab@R2# run show ospf database
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.1         10.1.1.1         0x80000008   906  0x22 0x2af0  48
Router  *10.1.1.2         10.1.1.2         0x80000008  1385  0x22 0x3ed8  48
Router   10.1.1.10        10.1.1.10        0x80000005  2025  0x22 0xebbf  60
Network  192.168.1.1      10.1.1.1         0x80000003   782  0x22 0x584f  32
Summary  10.1.1.3         10.1.1.1         0x80000002  1771  0x22 0x6ab4  28
Summary *10.1.1.4         10.1.1.2         0x80000002  1578  0x22 0x5ac2  28
Summary  10.1.1.4         10.1.1.10        0x80000001  2029  0x22 0x2ce9  28
Summary  192.168.2.0      10.1.1.1         0x80000003  2085  0x22 0x467e  28
Summary *192.168.3.0      10.1.1.2         0x80000003  1778  0x22 0x358d  28
Summary  192.168.3.0      10.1.1.10        0x80000001  2029  0x22 0x13a8  28
Summary *192.168.16.0     10.1.1.2         0x80000002  1379  0x22 0xaa0e  28
Summary  192.168.16.0     10.1.1.10        0x80000001  2029  0x22 0x832b  28
Summary  192.168.17.0     10.1.1.10        0x80000001  2029  0x22 0x9415  28
Summary  192.168.18.0     10.1.1.10        0x80000001  2029  0x22 0x7a30  28
Summary *192.168.75.0     10.1.1.2         0x80000004  1385  0x22 0x2256  28
Summary  192.168.75.0     10.1.1.10        0x80000005  2025  0x22 0xe58a  28
ASBRSum  10.1.1.2         10.1.1.10        0x80000004  2025  0x22 0x36dc  28
ASBRSum  10.1.1.3         10.1.1.1         0x80000005   164  0x22 0x56c4  28
ASBRSum *10.1.1.4         10.1.1.2         0x80000009  1385  0x22 0x3ed6  28
ASBRSum  10.1.1.4         10.1.1.10        0x80000004  2025  0x22 0x18f9  28

    OSPF database, Area 0.0.0.2
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.1.1.2         10.1.1.2         0x80000004  1385  0x22 0x9b7e  48
```

```
Router   10.1.1.4         10.1.1.4         0x8000000a  2029  0x22 0xc993 108
Router   10.1.1.10        10.1.1.10        0x80000006   938  0x22 0xa4ee  36
Router   192.168.16.2     192.168.16.2     0x80000003  2663  0x22 0xdfbc  60
Network  192.168.75.2     10.1.1.10        0x80000003  1154  0x22 0x5dea  32
Summary *10.1.1.1         10.1.1.2         0x80000002  1179  0x22 0x78a7  28
Summary *10.1.1.2         10.1.1.2         0x80000003   379  0x22 0x62bc  28
Summary *10.1.1.3         10.1.1.2         0x80000002   979  0x22 0x6eae  28
Summary *192.168.1.0      10.1.1.2         0x8000000d  1385  0x22 0x3783  28
Summary *192.168.2.0      10.1.1.2         0x80000002   779  0x22 0x4c77  28
Summary  192.168.76.0     10.1.1.10        0x80000004  1423  0x22 0xee7e  28
Summary  192.168.77.0     10.1.1.10        0x80000003  1423  0x22 0xe587  28
ASBRSum *10.1.1.1         10.1.1.2         0x8000000a  1385  0x22 0x5abc  28
ASBRSum *10.1.1.3         10.1.1.2         0x80000002   579  0x22 0x60bb  28
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   0.0.0.0          10.1.1.1         0x80000003  1400  0x22 0x8d16  36
Extern  *0.0.0.0          10.1.1.2         0x80000003   179  0x22 0x871b  36
Extern   10.1.1.6         10.1.1.1         0x80000002  1277  0x22 0x780f  36
Extern   192.168.29.0     10.1.1.4         0x80000002  1483  0x22 0x8b8f  36
Extern   192.168.30.0     10.1.1.4         0x80000002   655  0x22 0x8099  36
Extern   192.168.71.0     10.1.1.1         0x80000002  1648  0x22 0x934f  36
Extern   192.168.72.0     10.1.1.1         0x80000002  1524  0x22 0x8859  36
```

The routes are not showing up in the LSA Database, let's do a trace options on the OSPF process within R2 to see why.

Step 1

The configuration for traceoptions is as follows:

```
set protocols ospf traceoptions file ospfdebug
set protocols ospf traceoptions flag all
```

This configuration flags all OSPF traces to a file called *ospfdebug*. Now you can take a look at this file for the purpose of understanding why it should not be added into the database:

```
[edit protocols ospf]
lab@R2# run show log ospfdebug | match 192.168.76.
Nov 14 00:05:02.570189 Deleting LSA Summary 192.168.76.0 10.1.1.10 (flood state Idle)
Nov 14 00:05:02.879940   id 192.168.76.0, type Summary (0x3), age 0xe7
Nov 14 00:05:02.924009   type Summary (3), id 192.168.76.0, adv rtr 10.1.1.10
Nov 14 00:05:02.985456   id 192.168.76.0, type Summary (0x3), age 0xe8
Nov 14 00:05:03.056223 OSPF LSA Summary 192.168.76.0 10.1.1.10 from 192.168.3.2 newer
than db
Nov 14 00:05:03.056527 LSA Summary 192.168.76.0 10.1.1.10 flood state Idle -> Idle, new
LSA
Nov 14 00:05:03.056716 ospf_set_lsdb_state: Summary LSA 192.168.76.0 adv-rtr 10.1.1.10
state QUIET->QUIET
Nov 14 00:05:03.057400 OSPF LSA Summary 192.168.76.0 10.1.1.10 from 192.168.3.2, LSA
```

```
changed from its last instance
Nov 14 00:05:03.057598 OSPF LSREQ for LSA Summary 192.168.76.0 10.1.1.10 satisfied for
nbr 192.168.3.2 on ge-0/0/1.0 area 0.0.0.2
Nov 14 00:05:03.057800 LSA Summary 192.168.76.0 10.1.1.10 flood state Idle -> Standby
send, flooding
Nov 14 00:05:03.057972 Updating LSA Summary 192.168.76.0 10.1.1.10 (flood state Standby
send)
Nov 14 00:05:03.058511 LSA Summary 192.168.76.0 10.1.1.10 flood state Standby send ->
Wait nbr ack, not queued
Nov 14 00:05:03.058716 OSPF LSA Summary 192.168.76.0 10.1.1.10 newer, delayed ack
Nov 14 00:05:03.132603 OSPF LSA Summary 192.168.76.0 10.1.1.10 same as ge-0/0/1.0 area
0.0.0.2 192.168.3.2 LSREQ
Nov 14 00:05:03.132887 OSPF LSA Summary 192.168.76.0 10.1.1.10 on no ge-0/0/1.0 area
0.0.0.2 rexmit lists, no flood
Nov 14 00:05:03.133080 LSA Summary 192.168.76.0 10.1.1.10 flood state Wait nbr ack ->
Idle, not queued
Nov 14 00:05:03.348101   Adding Network summary route 192.168.76.0/24: origin 10.1.1.10
Nov 14 00:05:03.391055 CHANGE   192.168.76.0/24    nhid 0 gw 192.168.3.2    OSPF    pref
10/0 metric 3/0 ge-0/0/1.0 <Active Int>
Nov 14 00:05:03.391489 ADD     192.168.76.0/24    nhid 0 gw 192.168.3.2    OSPF    pref
10/0 metric 3/0 ge-0/0/1.0 <Active Int>
Nov 14 00:05:03.391804   Route 192.168.76.0/24 has changed (other)
Nov 14 00:05:03.392071   Considering autosummary for 192.168.76.0/24, summary possible=1
Nov 14 00:05:03.392714   Considering NSSA autosummary for 192.168.76.0/24, summary not
possible
Nov 14 00:05:03.590700   Adding Network summary route 192.168.76.0/24: origin 10.1.1.10
Nov 14 00:05:03.598065   Route 192.168.76.0/24 is unchanged
Nov 14 00:05:04.027834   id 192.168.76.0, type Summary (0x3), age 0xe8
Nov 14 00:05:11.896209    id 192.168.76.0, data 255.255.255.0, type Stub (3)
Nov 14 00:05:12.367939 CHANGE   192.168.76.0/24    nhid 565 gw 192.168.3.2    OSPF
pref 10/0 metric 3/0 ge-0/0/1.0 <Delete Int>
Nov 14 00:05:12.368227   Route 192.168.76.0/24 has been deleted
Nov 14 00:05:12.368486   Considering autosummary for 192.168.76.0/24, summary possible=0
Nov 14 00:05:12.368978   Considering NSSA autosummary for 192.168.76.0/24, summary not
possible
```

From this output you can see that the route 192.168.76.0/24 was removed from the LSA database and was not put into the routing table. The reason is that within OSPF every area must be connected to the backbone Area 0. In this case, R10 is not. For the configurations to function properly a virtual link must be added to connect R10 to the backbone Area 0 through R2.

Configuration of the Virtual Link

In order to see the Area 3 routes within the backbone Area 0, a virtual link configuration must be done on both R10 and R2.

Step 1

The configuration for R2:

```
set protocols ospf area 0.0.0.0 virtual-link neighbor-id 10.1.1.10 transit-
area 0.0.0.2
```

Step 2

The configuration for R10:

```
set protocols ospf area 0.0.0.0 virtual-link neighbor-id 10.1.1.2 transit-
area 0.0.0.2
```

In this configuration each virtual link has a transit area of 0.0.0.2 because Area 2 is the “connecting” area for the two routers. Another piece of the configuration shows that the neighbor-ids listed are the loopback interfaces of the R10 and R4. In fact, the OSPF router-id must match the router-id and must either be configured manually (as configured in Chapter 2) or automatically. If you do not set the neighbor-id to the remote router’s router-id, the link will not come up.

Once the configuration is committed to the routers the following command shows there is a neighboring relationship between the two routers:

```
root@R10# run show ospf neighbor
Address          Interface              State     ID               Pri  Dead
192.168.75.1     ge-0/0/0.0             Full      10.1.1.4         128    32
192.168.3.1      vl-10.1.1.2            Full      10.1.1.2           0    32
```

The virtual interface vl-10.1.1.2 is the created virtual link to R2. This shows that there is a peer and the state is in full. A closer look reveals:

```
[edit protocols ospf]
root@R10# run show ospf interface vl-10.1.1.2 detail
Interface           State   Area            DR ID           BDR ID          Nbrs
vl-10.1.1.2         PtToPt  0.0.0.0         0.0.0.0         0.0.0.0            1
  Type: Virtual, Address: 192.168.75.2, Mask: 0.0.0.0, MTU: 0, Cost: 2
  Transit Area: 0.0.0.2, Destination: 192.168.3.1
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Topology default (ID 0) -> Cost: 0
```

This detail output shows more of the peering and also shows the transit Area 2, which is important to note for any operations personnel who would be troubleshooting a network issue.

From R2's perspective, the LSA database can see the peer of R10 as an ABR with a type-4 link, which represents a virtual link. This is confirmed by issuing the following command:

```
lab@R2# run show ospf database lsa-id 10.1.1.10 detail
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.1.1.10        10.1.1.10        0x80000008   419  0x22 0xf73f  36
  bits 0x1, link count 1
  id 10.1.1.2, data 192.168.75.2, Type Virtual (4)
    Topology count: 0, Default metric: 2
  Topology default (ID 0)
    Type: Virtual, Node ID: 10.1.1.2
      Metric: 2, Bidirectional
```

Now that there is a peering relationship between R2 and R10, the routes for 192.168.76.0/24 and 192.168.77.0/24 from Area 3 are in the routing table for R2, as shown here:

```
[edit protocols ospf]
lab@R2# run show route 192.168.76.0
inet.0: 27 destinations, 28 routes (27 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.76.0/24    *[OSPF/10] 00:35:21, metric 3
                    > to 192.168.3.2 via ge-0/0/1.0
[edit protocols ospf]
lab@R2# run show route 192.168.77.0
inet.0: 27 destinations, 28 routes (27 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.77.0/24    *[OSPF/10] 00:35:26, metric 3
                    > to 192.168.3.2 via ge-0/0/1.0
```

And this verifies that the virtual link is functioning and that the routes from Area 3 will be seen throughout the network.

Summary

Using Virtual Links can help network administrators integrate backbone areas or extend them for redundancy purposes. Note that this type of connectivity should really be considered a short term solution to a problem. At some point it should be replaced with a more permanent link to the existing backbone Area 0. The main reasons for making this a temporary configuration are for troubleshooting purposes – virtual links add complexity to networks that should be avoided if at all possible.

What to Do Next & Where to Go

http://www.juniper.net/books

The following books may assist your further exploration of OSPF:
- *OSPF and IS-IS*, by Jeff Doyle
- *Junos Enterprise Routing, 2nd Edition*, by Peter Southwick, et. al.
- *Junos Cookbook*, by Aviva Garrett
- *Day One: Migrating EIGRP to OSPF*, by Jack Parks

http://www.juniper.net/dayone

The *Day One* book series is available for free download in PDF format. Select titles also feature a *Copy and Paste* edition for direct placement of Junos configurations. (The library is available in eBook format for iPads and iPhones from the Apple iBookstore, or download to Kindles, Androids, Blackberrys, Macs and PCs by visiting the Kindle Store. In addition, print copies are available for sale at Amazon or www.vervante.com.)

http://forums.juniper.net/jnet

The Juniper-sponsored J-Net Communities forum is dedicated to sharing information, best practices, and questions about Juniper products, technologies, and solutions. Register to participate in this free forum.

www.juniper.net/techpubs/

Juniper Networks technical documentation includes everything you need to understand and configure all aspects of Junos, including MPLS. The documentation set is both comprehensive and thoroughly reviewed by Juniper engineering.

www.juniper.net/training/fasttrack

Take courses online, on location, or at one of the partner training centers around the world. The Juniper Network Technical Certification Program (JNTCP) allows you to earn certifications by demonstrating competence in configuration and troubleshooting of Juniper products. If you want the fast track to earning your certifications in enterprise routing, switching, or security use the available online courses, student guides, and lab guides.