CCNP ROUTE
Portable Command Guide

All the ROUTE 642-902 Commands in
One Compact, Portable Resource

Scott Empson
Hans Roth

ciscopress.com

CCNP ROUTE Portable Command Guide

Scott Empson, Hans Roth

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America

First Printing March 2010

Library of Congress Cataloging-in-Publication Data is on file.

ISBN-13: 978-1-58720-249-0

ISBN-10: 1-58720-249-2

Warning and Disclaimer

This book is designed to provide information about the Cisco ROUTE exam (642-902). Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Introduction

Welcome to *CCNP ROUTE Portable Command Guide*! When Cisco Press approached me about updating the four-volume *CCNP Portable Command Guides*, two thoughts immediately jumped into my head: "Is it time for revisions already?" and "Yikes! I am in the middle of pursuing my master's degree. Where will I find the time?" Because of those thoughts, two more soon followed: "I wonder what Hans is up to?" and "I hope Carol is in a good mood, as I am about to ask to take Hans away again…." The result is what you now have before you; a new *Portable Command Guide* for the latest version of the CCNP exam that focuses on routing: CCNP ROUTE.

For those of you who have worked with my books before, thank you for looking at this one. I hope that it will help you as you prepare for the vendor exam, or assist you in your daily activities as a Cisco network administrator/manager. For those of you who are new to my books, you are reading what is essentially a cleaned-up version of my own personal engineering journals—a small notebook that I carry around with me that contains little nuggets of information; commands that I use but then forget; IP address schemes for the parts of the network I work with only occasionally; and quick refreshers for those concepts that I work with only once or twice a year. Although I teach these topics to postsecondary students, the classes I teach sometimes occur only once a year; as you can attest to, it is extremely difficult to remember all those commands all the time. Having a journal of commands at your fingertips, without having to search the Cisco website, can be a real time-saver (or a job-saver if the network is down and you are responsible for getting it back online).

With the creation of the new CCNP exam objectives, there is always something new to read; or a new podcast to listen to; or another slideshow from Cisco Live that you missed or want to review. The engineering journal can be that central repository of information that won't weigh you down as you carry it from the office or cubicle to the server and infrastructure rooms in some remote part of the building or some branch office.

To make this guide a more realistic one for you to use, the folks at Cisco Press have decided to continue with an appendix of blank pages—pages on which you can write your own personal notes, such as your own configurations, commands that are not in this book but are needed in your world, and so on. That way, this book will look less like the authors' journals and more like your own.

Networking Devices Used in the Preparation of This Book

To verify the commands that are in this new series of *CCNP Portable Command Guides*, many different devices were used. The following is a list of the equipment used in the preparation of these books:

- C2620 router running Cisco IOS Release 12.3(7)T, with a fixed Fast Ethernet interface, a WIC 2A/S serial interface card, and an NM-1E Ethernet interface
- C2811 ISR bundle with PVDM2, CMME, a WIC-2T, FXS and FXO VICs, running Cisco IOS Release 12.4(3g)

- C2821 ISR bundle with HWICD 9ESW, a WIC 2A/S, running 12.4(16) Advanced Security IOS
- WS-C3560-24-EMI Catalyst Switch, running Cisco IOS Release 12.2(25)SE
- WS-C3550-24-EMI Catalyst Switch, running Cisco IOS Release 12.1(9)EA1c
- WS-2960-24TT-L Catalyst Switch, running Cisco IOS Release 12.2(25)SE
- WS-2950-12 Catalyst Switch, running version C2950-C3.0(5.3)WC(1) Enterprise Edition Software
- WS-C3750-24TS Catalyst Switches, running ipservicesk9 release 12.2(52)SE
- C1760-V Voice Router with PVDM-256K-20, WIC-4ESW, VIC-2FXO, VIC-2FXS running ENTSERVICESK9 release 12.4(11)T2

You might notice that some of the devices were not running the latest and greatest IOS. Some of them are running code that is quite old.

Those of you familiar with Cisco devices will recognize that a majority of these commands work across the entire range of the Cisco product line. These commands are not limited to the platforms and IOS versions listed. In fact, in most cases, these devices are adequate for someone to continue their studies beyond the CCNP level as well. We have endeavored to identify throughout the book commands that are specific to a platform and/or IOS version.

Who Should Read This Book?

This book is for those people preparing for the CCNP ROUTE exam, whether through self-study, on-the-job training and practice, study within the Cisco Academy Program, or study through the use of a Cisco Training Partner. This book includes some handy hints and tips along the way to make life a bit easier for you in this endeavor. It is small enough that you will find it easy to carry around with you. Big, heavy textbooks might look impressive on your bookshelf in your office, but can you really carry them all around with you when you are working in some server room or equipment closet somewhere?

Strategies for Exam Preparation

The strategy that you use for CCNP ROUTE might be slightly different from strategies that other readers use, mainly based on the skills, knowledge, and experience you already have obtained. For instance, if you have attended the ROUTE course, you might take a different approach than someone who learned routing via on-the-job training. Regardless of the strategy you use or the background you have, the book is designed to help you get to the point where you can pass the exam with the least amount of time required. For instance, there is no need for you to practice or read about EIGRP or OSPF if you fully understand it already. However, many people like to make sure that they truly know a topic and thus read over material that they already know. Several book features will help you not only to gain the confidence that you need to be convinced that you know some material already, but also to determine which topics you need to study more.

Organization of This Book

Although this book could be read cover-to-cover, we strongly advise against it. The book is designed to be a simple listing of those commands that you need to understand to pass the ROUTE exam. Very little theory is included in the *Portable Command Guides*; they are designed to list commands needed at this level of study.

This book follows the list of objectives for the CCNP ROUTE exam:

- **Chapter 1: "Network Design Requirements"**—This chapter shows the Cisco Hierarchical Model of Network Design; the Cisco Enterprise Composite Network Model; the Cisco Service-Oriented Network Architecture (SONA); a comparison of routing protocols; a chart outlining where protocols should be implemented; and the PPDIOO network lifecycle.
- **Chapter 2: "Implementing an EIGRP-based Solution"**—This chapter covers EIGRP, including the design, implementation, verification, and troubleshooting of this protocol.
- **Chapter 3: "Implementing a Scalable Multiarea Network OSPF-based Solution"**—This chapter deals with OSPF, including a review of configuring OSPF, both single area (as a review) and multiarea. Topics include the design, implementation, verification, and troubleshooting of the protocol.
- **Chapter 4: "Implementing an IPv4-based Redistribution Solution"**—This chapter shows how to manipulate routing information. Topics include prefix lists, distribution lists, route maps, route redistribution, administrative distances, and static routes.
- **Chapter 5: "Implementing Path Control"**—This chapter deals with those tools and commands that can be used to help evaluate network performance issues and control the path. Topics include offset lists, Cisco IOS IP Service Level Agreements (SLAs), and policy-based routing using route maps.
- **Chapter 6: "Enterprise to ISP Connectivity"**—This chapter deals with the use of BGP to connect an enterprise network to a service provider. Topics include the configuration, verification, and troubleshooting of a BGP-based solution; BGP attributes; regular expressions; and BGP route filtering using access lists.
- **Chapter 7: "Implementing IPv6"**—This chapter provides information and commands regarding the implementation of IPv6. Topics include assigning IPv6 addresses; CEF and dCEF for IPv6; RIPng; OSPFv3; IPv6 and EIGRP; route redistribution; IPv6 transition techniques; NAT-PC for IPv6; static routes; and verifying and troubleshooting IPv6.
- **Chapter 8: "Routing for Branch Offices and Mobile Workers"**—This chapter deals with the connection, verification, and troubleshooting of remote locations within your network. Topics include verifying existing services; configuring DSL; configuring PPPoA; configuring a cable modem connection; connecting a teleworker to a branch office VPN; configuring IPsec site-to-site VPNs; and configuring GRE tunnels over IPsec.

Did We Miss Anything?

As educators, we are always interested to hear how our students, and now readers of our books, do on both vendor exams and future studies. If you would like to contact either of us and let us know how this book helped you in your certification goals, please do so. Did we miss anything? Let us know. Contact us at ccnpguide@empson.ca or through the Cisco Press website, www.ciscopress.com.

CHAPTER 3

Implementing a Scalable Multiarea Network OSPF-based Solution

This chapter provides information and commands concerning the following Open Shortest Path First (OSPF) topics:

- Configuring OSPF
- Using wildcard masks with OSPF areas
- Configuring multiarea OSPF
- Loopback interfaces
- Router ID
- DR/BDR elections
- Passive interfaces
- Modifying cost metrics
- OSPF LSDB overload protection
- OSPF **auto-cost reference-bandwidth**
- Authentication: simple
- Authentication: using MD5 encryption
- Timers
- Propagating a default route
- OSPF special area types
  - Stub areas
  - Totally stubby areas
  - Not-so-stubby areas (NSSA) stub area
  - NSSA totally stubby areas
- Route summarization
  - Inter-area route summarization
  - External route summarization
- Configuration example: virtual links
- OSPF and NBMA networks
  - Full-mesh Frame Relay: NBMA on physical interfaces
  - Full-mesh Frame Relay: broadcast on physical interfaces
  - Full-mesh Frame Relay: point-to-multipoint networks
  - Full-mesh Frame Relay: point-to-point networks on subinterfaces
- OSPF over NBMA topology summary
- Verifying OSPF Configuration
- Troubleshooting OSPF

- Configuration example: single-area OSPF
- Configuration example: multiarea OSPF
- Configuration example: OSPF and NBMA networks
- Configuration example: OSPF and broadcast networks
- Configuration example: OSPF and point-to-multipoint networks
- Configuration example: OSPF and point-to-point networks using subinterfaces

Configuring OSPF

`Router(config)#router ospf 123`	Starts OSPF process 123. The process ID is any positive integer value between 1 and 65,535. The process ID is *not* related to the OSPF area. The process ID merely distinguishes one process from another within the device.
`Router(config-router)#network 172.16.10.0 0.0.0.255 area 0`	OSPF advertises interfaces, not networks. Uses the wildcard mask to determine which interfaces to advertise. Read this line to say, "Any interface with an address of 172.16.10.x is to be put into area 0."
	NOTE: The process ID number of one router does not have to match the process ID of any other router. Unlike Enhanced Interior Gateway Routing Protocol (EIGRP), matching this number across all routers does *not* ensure that network adjacencies will form.
`Router(config-router)#log-adjacency-changes detail`	Configures the router to send a syslog message when there is a change of state between OSPF neighbors.
	TIP: Although the **log-adjacency-changes** command is on by default, only up/down events are reported unless you use the **detail** keyword.

Using Wildcard Masks with OSPF Areas

When compared to an IP address, a wildcard mask will identify what addresses get matched for placement into an area:

- A 0 (zero) in a wildcard mask means to check the corresponding bit in the address for an exact match.
- A 1 (one) in a wildcard mask means to ignore the corresponding bit in the address—can be either 1 or 0.

Example 1: 172.16.0.0 0.0.255.255

172.16.0.0 =	10101100.00010000.00000000.00000000
0.0.255.255 =	00000000.00000000.11111111.11111111
result =	10101100.00010000.*xxxxxxxx*.*xxxxxxxx*

172.16.*x*.*x* (Anything between 172.16.0.0 and 172.16.255.255 will match the example statement.)

TIP: An octet of all zeros means that the octet has to match the address exactly. An octet of all ones means that the octet can be ignored.

Example 2: 172.16.8.0 0.0.7.255

172.168.8.0 =	10101100.00010000.00001000.00000000
0.0.0.7.255 =	00000000.00000000.00000111.11111111
result =	10101100.00010000.00001*xxx*.*xxxxxxxx*
00001*xxx* =	00001*000* to 00001*111* = 8 – 15
xxxxxxxx =	00000000 to 11111111 = 0 – 255

Anything between 172.16.8.0 and 172.16.15.255 will match the example statement.

`Router(config-router)#network 172.16.10.1 0.0.0.0 area 0`	Read this line to say, "Any interface with an exact address of 172.16.10.1 is to be put into area 0."
`Router(config-router)#network 172.16.10.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an address of 172.16.*x*.*x* is to be put into area 0."
`Router(config-router)#network 0.0.0.0 255.255.255.255 area 0`	Read this line to say, "Any interface with any address is to be put into area 0."

Configuring Multiarea OSPF

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#network 172.16.10.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.10.*x* is to be put into area 0."
`Router(config-router)#network 10.10.10.1 0.0.0.0 area 51`	Read this line to say, "Any interface with an exact address of 10.10.10.1 is to be put into area 51."

Loopback Interfaces

`Router(config)#interface loopback0`	Creates a virtual interface named Loopback 0 and then moves the router to interface configuration mode.
`Router(config-if)#ip address 192.168.100.1 255.255.255.255`	Assigns the IP address to the interface.
	NOTE: Loopback interfaces are always "up and up" and do not go down unless manually shut down. This makes loopback interfaces great for use as an OSPF router ID.

Router ID

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#router-id 10.1.1.1`	Sets the router ID to 10.1.1.1. If this command is used on an OSPF router process that is already active (has neighbors), the new router ID is used at the next reload or at a manual OSPF process restart.
`Router(config-router)#no router-id 10.1.1.1`	Removes the static router ID from the configuration. If this command is used on an OSPF router process that is already active (has neighbors), the old router ID behavior is used at the next reload or at a manual OSPF process restart.

DR/BDR Elections

`Router(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Router(config-if)#ip ospf priority 50`	Changes the OSPF interface priority to 50.
	NOTE: The assigned priority can be between 0 and 255. A priority of 0 makes the router ineligible to become a designated router (DR) or backup designated router (BDR). The highest priority wins the election. A priority of 255 guarantees a tie in the election. If all routers have the same priority, regardless of the priority number, they tie. Ties are broken by the highest router ID.

Passive Interfaces

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#network 172.16.10.0 0.0.0.255 area 0`	Read this line to say "Any interface with an address of 172.16.10.*x* is to be put into area 0."
`Router(config-router)#passive-interface fastethernet 0/0`	Disables the sending of routing updates on this interface.
`Router(config-router)#passive-interface default`	Disables the sending of routing updates out all interfaces.
`Router(config-router)#no passive-interface serial 0/0/1`	Enables routing updates to be sent out interface serial 0/0/1, thereby allowing neighbor adjacencies to form.

Modifying Cost Metrics

`Router(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Router(config-if)#bandwidth 128`	If you change the bandwidth, OSPF will recalculate the cost of the link.
Or	
`Router(config-if)#ip ospf cost 1564`	Changes the cost to a value of 1564.
	NOTE: The cost of a link is determined by dividing the reference bandwidth by the interface bandwidth. The bandwidth of the interface is a number between 1 and 10,000,000. The unit of measurement is kilobits. The cost is a number between 1 and 65,535. The cost has no unit of measurement; it is just a number.

OSPF LSDB Overload Protection

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-if)#max-lsa 12000`	Limits the number of nonself-generated link-state advertisements (LSA).

> **NOTE:** If other routers are configured incorrectly, causing, for example, a redistribution of a large number of prefixes, large numbers of LSAs can be generated. This can drain local CPU and memory resources. With the **max-lsa** *x* feature enabled, the router keeps count of the number of received (nonself-generated) LSAs that it keeps in its link-state database (LSDB). An error message is logged when this number reaches a configured threshold number, and a notification is sent when it exceeds the threshold number.

If the LSA count still exceeds the threshold after one minute, the OSPF process takes down all adjacencies and clears the OSPF database. This is called the *ignore state*. In the ignore state, no OSPF packets are sent or received by interfaces that belong to the OSPF process.

The OSPF process will remain in the ignore state for the time that is defined by the **ignore-time** parameter. If the OSPF process remains normal for the time that is defined by the **reset-time** parameter, the ignore state counter is reset to 0.

OSPF auto-cost reference-bandwidth

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#auto-cost reference-bandwidth 1000`	Changes the reference bandwidth that OSPF uses to calculate the cost of an interface.
	NOTE: The range of the reference bandwidth is 1 to 4,294,967. The default is 100. The unit of measurement is Mbps.
	NOTE: The value set by the **ip ospf cost** command overrides the cost resulting from the **auto-cost** command.
	TIP: If you use the command **auto-cost reference-bandwidth** *reference-bandwidth*, configure all the routers to use the same value. Failure to do so will result in routers using a different reference cost to calculate the shortest path, resulting in potential suboptimum routing paths.

Authentication: Simple

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#area 0 authentication`	Enables simple authentication; the password will be sent in clear text.
	NOTE: Another way to enable authentication is to move to the interface in which you want authentication to take place and enter the following command: `Router(config-if)#ip ospf authentication`
`Router(config-router)#exit`	Returns to global configuration mode.

`Router(config)#interface fastethernet 0/0`	Enters interface configuration mode.
`Router(config-if)#ip ospf authentication-key fred`	Sets key (password) to fred.
	NOTE: The password can be any continuous string of characters that can be entered from the keyboard, up to eight characters in length. To be able to exchange OSPF information, all neighboring routers on the same network must have the same password.
	NOTE: In Cisco IOS Release 12.4, the router will give a warning if you try to configure a password longer than eight characters; only the first eight characters will be used. Some earlier Cisco IOS releases did not provide this warning.

Authentication: Using MD5 Encryption

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#area 0 authentication message-digest`	Enables authentication with MD5 password encryption.
	NOTE: Another way to enable authentication is to move to the interface in which you want authentication to take place and enter the following command: `Router(config-if)#ip ospf authentication message-digest`
`Router(config-router)#exit`	Returns to global configuration mode.
`Router(config)#interface fastethernet 0/0`	Enters interface configuration mode.
`Router(config-if)#ip ospf message-digest-key 1 md5 fred`	1 is the *key-id*. This value must be the same as that of your neighboring router. **md5** indicates that the MD5 hash algorithm will be used. **fred** is the key (password) and must be the same as that of your neighboring router.

	NOTE: If the **service password-encryption** command is not used when implementing OSPF MD5 authentication, the MD5 secret will be stored as plain text in NVRAM.
	NOTE: In Cisco IOS Release 12.4, the router will give a warning if you try to configure a password longer than 16 characters; only the first 16 characters will be used. Some earlier Cisco IOS releases did not provide this warning.

TIP: It is recommended that you keep no more than one key per interface. Every time you add a new key, you should remove the old key to prevent the local system from continuing to communicate with a hostile system that knows the old key.

NOTE: If the **service password-encryption** command is not used when configuring OSPF authentication, the key will be stored as plain text in the router configuration. If you use the **service password-encryption** command, there will be an *encryption-type* of 7 specified before the encrypted key.

Timers

`Router(config-if)#ip ospf hello-interval timer 20`	Changes the Hello Interval timer to 20 seconds.
`Router(config-if)#ip ospf dead-interval 80`	Changes the Dead Interval timer to 80 seconds.
	NOTE: Hello and Dead Interval timers must match for routers to become neighbors.

NOTE: If you change the Hello Interval timer, the Dead Interval timer will automatically be adjusted to four times the new Hello Interval timer.

Propagating a Default Route

`Router(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0`	Creates a default route.
`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#default-information originate`	Sets the default route to be propagated to all OSPF routers.
`Router(config-router)#default-information originate always`	The **always** option will propagate a default "quad-zero" route even if one is not configured on this router.
	NOTE: The **default-information originate** command or the **default-information originate always** command is usually only to be configured on your "entrance" or "gateway" router, the router that connects your network to the outside world—the Autonomous System Boundary Router (ASBR).

OSPF Special Area Types

This section covers four different special areas with respect to OSPF:

- Stub areas
- Totally stubby areas
- Not-so-stubby areas (NSSA) stub area
- NSSA totally stubby areas

Stub Areas

`ABR(config)#router ospf 1`	Starts OSPF process 1.
`ABR(config-router)#network 172.16.10.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.10.*x* is to be put into area 0."
`ABR(config-router)#network 172.16.20.0 0.0.0.255 area 51`	Read this line to say, "Any interface with an address of 172.16.20.x is to be put into area 51."

`ABR(config-router)#area 51 stub`	Defines area 51 as a stub area.
`ABR(config-router)#area 51 default-cost 10`	Defines the cost of a default route sent into the stub area. Default is 1.
	NOTE: This is an optional command.
`Internal(config)#router ospf 1`	Starts OSPF process 1.
`Internal(config-router)#network 172.16.20.0 0.0.0.255 area 51`	Read this line to say, "Any interface with an address of 172.16.20.*x* is to be put into area 51."
`Internal(config-router)#area 51 stub`	Defines area 51 as a stub area.
	NOTE: All routers in the stub area must be configured with the **area** *x* **stub** command, including the Area Border Router (ABR).

Totally Stubby Areas

`ABR(config)#router ospf 1`	Starts OSPF process 1.
`ABR(config-router)#network 172.16.10.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.10.*x* is to be put into area 0."
`ABR(config-router)#network 172.16.20.0 0.0.0.255 area 51`	Read this line to say, "Any interface with an address of 172.16.20.*x* is to be put into area 51."
`ABR(config-router)#area 51 stub no-summary`	Defines area 51 as a totally stubby area.
`Internal(config)#router ospf 1`	Starts OSPF process 1.
`Internal(config-router)#network 172.16.20.0 0.0.0.255 area 51`	Read this line to say, "Any interface with an address of 172.16.20.*x* is to be put into area 51."
`Internal(config-router)#area 51 stub`	Defines area 51 as a stub area.
	NOTE: Whereas all internal routers in the area are configured with the **area** *x* **stub** command, the ABR is configured with the **area** *x* **stub no-summary** command.

Not-So-Stubby Areas (NSSA) Stub Area

`ABR(config)#router ospf 1`	Starts OSPF process 1.
`ABR(config-router)#network 172.16.10.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.10.*x* is to be put into area 0."
`ABR(config-router)#network 172.16.20.0 0.0.0.255 area 1`	Read this line to say, "Any interface with an address of 172.16.20.*x* is to be put into area 1."
`ABR(config-router)#area 1 nssa`	Defines area 1 as an NSSA stub area.
`Internal(config)#router ospf 1`	Starts OSPF process 1.
`Internal(config-router)#network 172.16.20.0 0.0.0.255 area 1`	Read this line to say, "Any interface with an address of 172.16.20.*x* is to be put into area 1."
`Internal(config-router)#area 1 nssa`	Defines area 1 as an NSSA stub area.
	NOTE: All routers in the NSSA stub area must be configured with the **area** *x* **nssa** command.

NSSA Totally Stubby Areas

`ABR(config)#router ospf 1`	Starts OSPF process 1.
`ABR(config-router)#network 172.16.10.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.10.*x* is to be put into area 0."
`ABR(config-router)#network 172.16.20.0 0.0.0.255 area 11`	Read this line to say, "Any interface with an address of 172.16.20.*x* is to be put into area 11."
`ABR(config-router)#area 11 nssa no-summary`	Defines area 11 as an NSSA totally stubby area.
`Internal(config)#router ospf 1`	Starts OSPF process 1.
`Internal(config-router)#network 172.16.20.0 0.0.0.255 area 11`	Read this line to say, "Any interface with an address of 172.16.20.*x* is to be put into area 11."

`Internal(config-router)#area 11 nssa`	Defines area 11 as an NSSA stub area.
	NOTE: Whereas all internal routers in the area are configured with the **area** *x* **nssa** command, the ABR is configured with the **area** *x* **nssa no-summary** command.

Route Summarization

In OSPF, there are two different types of summarization:

- Inter-area route summarization
- External route summarization

The sections that follow provide the commands necessary to configure both types of summarization.

Inter-Area Route Summarization

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#area 1 range 192.168.64.0 255.255.224.0`	Configures the ABR to consolidate routes to this summary address before injecting them into a different area.
	NOTE: This command is to be configured on an ABR only.
	NOTE: By default, ABRs do *not* summarize routes between areas.

External Route Summarization

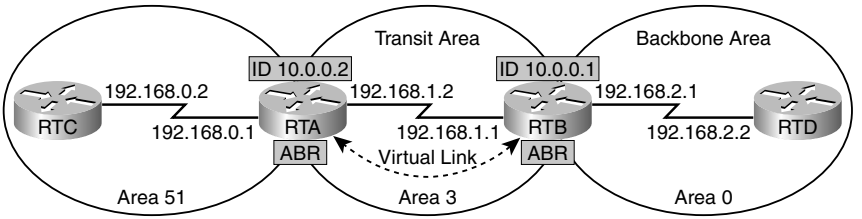
`Router(config)#router ospf 123`	Starts OSPF process 1.
`Router(config-router)#summary-address 192.168.64.0 255.255.224.0`	Advertises a single route for all the redistributed routes that are covered by a specified network address and netmask.

	NOTE: This command is to be configured on an ASBR only.
	NOTE: By default, ASBRs do *not* summarize routes.

Configuration Example: Virtual Links

Figure 3-1 shows the network topology for the configuration that follows, which demonstrates how to create a virtual link.

Figure 3-1 Virtual Areas: OSPF



`RTA(config)#router ospf 1`	Starts OSPF process 1.
`RTA(config-router)#router-id 10.0.0.2`	Sets the router ID to 10.0.0.2.
`RTA(config-router)#network 192.168.0.0 0.0.0.255 area 51`	Read this line to say, "Any interface with an address of 192.168.0.*x* is to be put into area 51."
`RTA(config-router)#network 192.168.1.0 0.0.0.255 area 3`	Read this line to say, "Any interface with an address of 192.168.1.*x* is to be put into area 3."
`RTA(config-router)#area 3 virtual-link 10.0.0.1`	Creates a virtual link with RTB.
`RTB(config)#router ospf 1`	Starts OSPF process 1.
`RTB(config-router)#router-id 10.0.0.1`	Sets the router ID to 10.0.0.1.
`RTB(config-router)#network 192.168.1.0 0.0.0.255 area 3`	Read this line to say, "Any interface with an address of 192.168.1.*x* is to be put into area 3."
`RTB(config-router)#network 192.168.2.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 192.168.2.*x* is to be put into area 0."

`RTB(config-router)#area 3 virtual-link 10.0.0.2`	Creates a virtual link with RTA.
	NOTE: A virtual link has the following two requirements: It must be established between two routers that share a common area and are both ABRs. One of these two routers must be connected to the backbone.
	NOTE: A virtual link is a temporary solution to a topology problem.
	NOTE: A virtual link cannot be configured through stub areas.

OSPF and NBMA Networks

OSPF is not well suited for nonbroadcast multiaccess (NBMA) networks such as Frame Relay or ATM. The term *multiaccess* means that an NBMA cloud is seen as a single network that has multiple devices attached to it, much like an Ethernet network. However, the *nonbroadcast* part of NBMA means that a packet sent into this network might not be seen by all other routers, which differs from broadcast technologies such as Ethernet. OSPF will want to elect a DR and BDR because an NBMA network is multiaccess; however, because the network is also nonbroadcast, there is no guarantee that all OSPF packets, such as Hello packets, would be received by other routers. This could affect the election of the DR because not all routers would know about all the other routers. The following sections list some possible solutions to dealing with OSPF in NBMA networks.

Full-Mesh Frame Relay: NBMA on Physical Interfaces

`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#network 10.1.1.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 10.1.1.*x* is to be put into area 0."
`Router(config-router)#neighbor 10.1.1.2`	Manually identifies this router's neighbor at IP address 10.1.1.2.

`Router(config-router)#neighbor 10.1.1.3 priority 15`	Manually identifies this router's neighbor at IP address 10.1.1.3 and assigns a priority value of 15 to determine the DR.
`Router(config-router)#exit`	Returns to global configuration mode.
`Router(config)#interface serial 0/0/0`	Moves to interface configuration mode.
`Router(config-if)#encapsulation frame-relay`	Enables Frame Relay on this interface.
`Router(config-if)#ip address 10.1.1.1 255.255.255.0`	Assigns an IP address and netmask to this interface.
`Router(config-if)#ip ospf network non-broadcast`	Defines OSPF nonbroadcast network type.
`Router(config-if)#frame-relay map ip 10.1.1.2 100`	Maps the remote IP address 10.1.1.2 to data-link connection identifier (DLCI) 100.
`Router(config-if)#frame-relay map ip 10.1.1.3 200`	Maps the remote IP address 10.1.1.3 to DLCI 200.
	NOTE: Using the **neighbor** command will allow for an OSPF router to exchange routing information without multicasts and instead use unicasts to the manually entered neighbor IP address.
	NOTE: Prior to Cisco IOS Release 12.0, the **neighbor** command applied to NBMA networks only. With Release 12.0, the **neighbor** command applies to NBMA networks and point-to-multipoint networks.

Full-Mesh Frame Relay: Broadcast on Physical Interfaces

`Router(config)#interface serial 0/0/0`	Moves to interface configuration mode.
`Router(config-if)#encapsulation frame-relay`	Enables Frame Relay on this interface.

`Router(config-if)#ip address 10.1.1.1 255.255.255.0`	Assigns an IP address and netmask to this interface.
`Router(config-if)#ip ospf network broadcast`	Changes the network type from the default *NBMA* to *broadcast*.
`Router(config-if)#ip ospf priority 15`	Changes the OSPF interface priority to 15.
`Router(config-if)#frame-relay map ip 10.1.1.2 100 broadcast`	Maps the remote IP address 10.1.1.2 to DLCI 100. Broadcast and multicast addresses will now be forwarded.
`Router(config-if)#frame-relay map ip 10.1.1.3 200 broadcast`	Maps the remote IP address 10.1.1.3 to DLCI 200. Broadcast and multicast addresses will now be forwarded.
`Router(config-if)#no shutdown`	Enables the interface.
`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#network 10.1.1.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 10.1.1.*x* is to be put into area 0."
	NOTE: If you want to influence the DR election, another option is to create a fully meshed topology where every router has a permanent virtual circuit (PVC) to every other router. This is efficient in terms of NBMA network designs. However, this is also a costly solution. For *n* routers, you need $n(n-1)/2$ PVCs. A 5-router full mesh would mean 10 PVCs are needed; 20 routers would mean 190 PVCs are needed.

Full-Mesh Frame Relay: Point-to-Multipoint Networks

`Router(config)#interface serial 0/0/0`	Moves to interface configuration mode.
`Router(config-if)#encapsulation frame-relay`	Enables Frame Relay on this interface.
`Router(config-if)#ip address 10.1.1.1 255.255.255.0`	Assigns an IP address and netmask to this interface.
`Router(config-if)#ip ospf network point-to-multipoint`	Changes the network to a point-to-multipoint network.
`Router(config-if)#exit`	Returns to global configuration mode.
`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#network 10.1.1.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 10.1.1.*x* is to be put into area 0."
	NOTE: In this example, Inverse ARP is used to dynamically map IP addresses to DLCIs. Static maps could have been used, if desired.
	NOTE: Point-to-multipoint networks treat PVCs as a collection of point-to-point links rather than a multiaccess network. No DR/BDR election will take place.
	NOTE: Point-to-multipoint networks might be your only alternative to broadcast networks in a multivendor environment.
`Router(config-router)#exit`	Returns to global configuration mode.
`Router(config)#interface serial 0/0/1`	Moves to interface configuration mode.
`Router(config)#ip ospf network point-to-multipoint non-broadcast`	Creates a point-to-multipoint nonbroadcast mode.

	NOTE: Point-to-multipoint nonbroadcast mode is a Cisco extension to the RFC-compliant point-to-multipoint mode. **NOTE:** Neighbors must be manually defined in this mode. **NOTE:** DR/BDRs are not used in this mode. **NOTE:** Point-to-multipoint nonbroadcast mode is used in special cases where neighbors cannot be automatically discovered.

NOTE: Point-to-multipoint networks can also be created on a subinterface with the following command:

```
Router(config)#interface serial 0/0/0.1 multipoint
```

NOTE: OSPF defaults to point-to-point mode on the point-to-point subinterface. OSPF defaults to nonbroadcast mode on the point-to-multipoint interface.

Full-Mesh Frame Relay: Point-to-Point Networks with Subinterfaces

`Router(config)#interface serial 0/0/0`	Moves to interface configuration mode.
`Router(config-if)#encapsulation frame-relay`	Enables Frame Relay on this interface.
`Router(config-if)#no shutdown`	Enables the interface.
`Router(config-if)#interface serial 0/0/0.300 point-to-point`	Creates subinterface 300 and makes it a point-to-point network.
`Router(config-subif)#ip address 192.168.1.1 255.255.255.252`	Assigns an IP address and netmask.
`Router(config-subif)#frame-relay interface-dlci 300`	Assigns DLCI 300 to the subinterface.
`Router(config-subif)#interface serial 0/0/0.400 point-to-point`	Creates subinterface 400 and makes it a point-to-point network.
`Router(config-subif)#ip address 192.168.1.5 255.255.255.252`	Assigns an IP address and netmask.

`Router(config-subif)#frame-relay interface-dlci 400`	Assigns DLCI 400 to the subinterface.
`Router(config-subif)#exit`	Returns to interface configuration mode.
`Router(config-if)#exit`	Returns to global configuration mode.
`Router(config)#router ospf 1`	Starts OSPF process 1.
`Router(config-router)#network 192.168.1.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 192.168.1.*x* is to be put into area 0."
	NOTE: Point-to-point subinterfaces allow each PVC to be configured as a separate subnet. No DR/BDR election will take place.
	NOTE: The use of subinterfaces increases the amount of memory used on the router.

OSPF over NBMA Topology Summary

OSPF Mode	NBMA Preferred Topology	Subnet Address	Hello Timer	Adjacency	RFC or Cisco
Broadcast	Full or partial mesh	Same	10 seconds	Automatic, DR/BDR elected	Cisco
Nonbroadcast (NBMA)	Full or partial mesh	Same	30 seconds	Manual configuration, DR/BDR elected	RFC
Point-to-multipoint	Partial mesh or star	Same	30 seconds	Automatic, no DR/BDR	RFC
Point-to-multipoint nonbroadcast	Partial mesh or star	Same	30 seconds	Manual Configuration, no DR/BDR	Cisco
Point-to-point	Partial mesh or star, using subinterface	Different for each Subinterface	10 seconds	Automatic, no DR/BDR	Cisco

Verifying OSPF Configuration

`Router#show ip protocol`	Displays parameters for all protocols running on the router.
`Router#show ip route`	Displays a complete IP routing table.
`Router#show ip ospf`	Displays basic information about OSPF routing processes.
`Router#show ip ospf border-routers`	Displays border and boundary router information.
`Router#show ip ospf database`	Displays the contents of the OSPF database.
`Router#show ip ospf database nssa-external`	Displays NSSA external link states.
`Router#show ip ospf database summary`	Displays a summary of the OSPF database.
`Router#show ip ospf interface`	Displays OSPF info as it relates to all interfaces.
`Router#show ip ospf interface fastethernet 0/0`	Displays OSPF information for interface fastethernet 0/0.
`Router#show ip ospf neighbor`	Lists all OSPF neighbors and their states.
`Router#show ip ospf neighbor detail`	Displays a detailed list of neighbors.
`Router#show ip ospf virtual-links`	Displays information about virtual links.

Troubleshooting OSPF

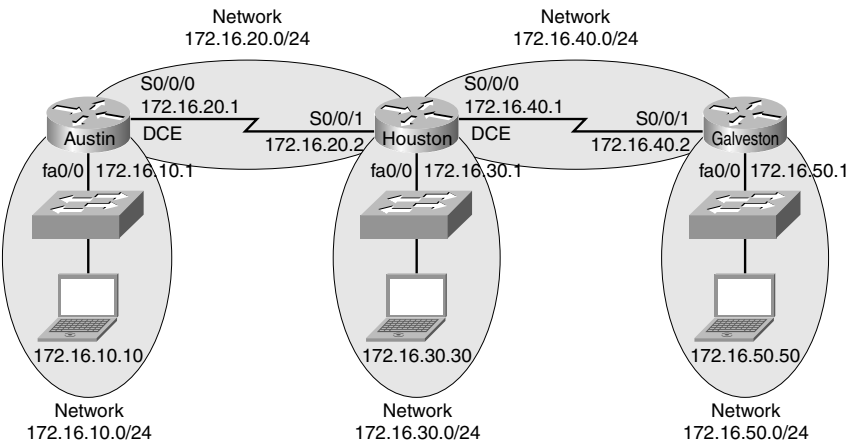
`Router#clear ip route *`	Clears the entire routing table, forcing it to rebuild.
`Router#clear ip route a.b.c.d`	Clears a specific route to network a.b.c.d.
`Router#clear ip ospf counters`	Resets OSPF counters.
`Router#clear ip ospf process`	Resets the *entire* OSPF process, forcing OSPF to re-create neighbors, database, and routing table.

`Router#debug ip ospf events`	Displays *all* OSPF events.
`Router#debug ip ospf adjacency`	Displays various OSPF states and DR/BDR election between adjacent routers.
`Router#debug ip ospf packets`	Displays OSPF packets.

Configuration Example: Single-Area OSPF

Figure 3-2 shows the network topology for the configuration that follows, which demonstrates how to configure single-area OSPF using the commands covered in this chapter.

Figure 3-2 Network Topology for Single-Area OSPF Configuration



Austin Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Austin`	Sets the hostname.
`Austin(config)#interface fastethernet 0/0`	Moves to interface configuration mode.
`Austin(config-if)#ip address 172.16.10.1 255.255.255.0`	Assigns an IP address and netmask.

`Austin(config-if)#no shutdown`	Enables the interface.
`Austin(config-if)#interface serial 0/0/0`	Moves to interface configuration mode.
`Austin(config-if)#ip address 172.16.20.1 255.255.255.252`	Assigns an IP address and netmask.
`Austin(config-if)#clock rate 56000`	Sets a clock rate on this interface. A DCE cable must be plugged in to this side.
`Austin(config-if)#no shutdown`	Enables the interface.
`Austin(config-if)#exit`	Returns to global configuration mode.
`Austin(config)#router ospf 1`	Starts OSPF process 1.
`Austin(config-router)#network 172.16.10.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.10.*x* is to be put into area 0."
`Austin(config-router)#network 172.16.20.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.20.*x* is to be put into area 0.
`Austin(config-router)#<ctrl> z`	Returns to privileged mode.
`Austin#copy running-config startup-config`	Saves the configuration to NVRAM.

Houston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Houston`	Sets the hostname.
`Houston(config)#interface fastethernet 0/0`	Moves to interface configuration mode.
`Houston(config-if)#ip address 172.16.30.1 255.255.255.0`	Assigns an IP address and netmask.
`Houston(config-if)#no shutdown`	Enables the interface.

`Houston(config-if)#interface serial 0/0/0`	Moves to interface configuration mode.
`Houston(config-if)#ip address 172.16.40.1 255.255.255.252`	Assigns an IP address and netmask.
`Houston(config-if)#clock rate 56000`	Sets a clock rate on this interface. A DCE cable must be plugged in to this side.
`Houston(config-if)#no shutdown`	Enables the interface.
`Houston(config)#interface serial 0/0/1`	Moves to interface configuration mode.
`Houston(config-if)#ip address 172.16.20.2 255.255.255.252`	Assigns an IP address and netmask.
`Houston(config-if)#no shutdown`	Enables the interface.
`Houston(config-if)#exit`	Returns to global configuration mode.
`Houston(config)#router ospf 1`	Starts OSPF process 1.
`Houston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an address of 172.16.*x.x* is to be put into area 0." One statement will now advertise all three interfaces.
`Houston(config-router)#<ctrl> z`	Returns to privileged mode.
`Houston#copy running-config startup-config`	Saves the configuration to NVRAM.

Galveston Router

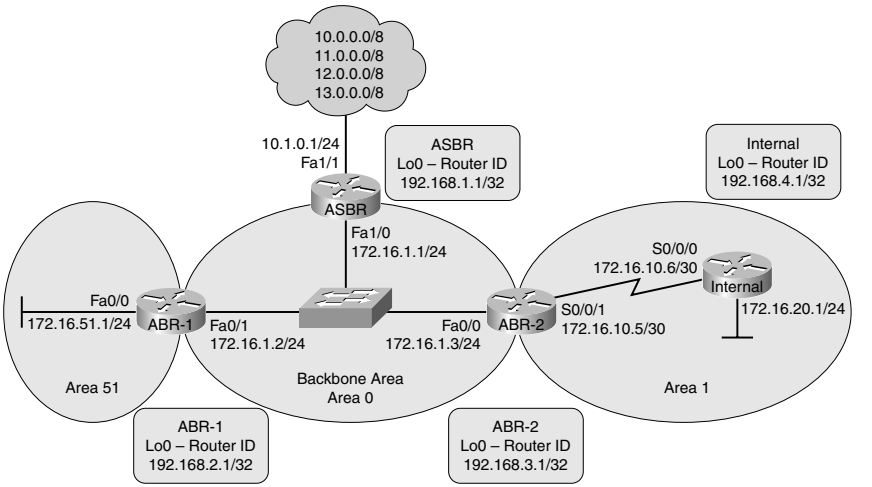
`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Galveston`	Sets the hostname.
`Galveston(config)#interface fastethernet 0/0`	Moves to interface configuration mode.
`Galveston(config-if)#ip address 172.16.50.1 255.255.255.0`	Assigns an IP address and netmask.

`Galveston(config-if)#no shutdown`	Enables the interface.
`Galveston(config-if)#interface serial 0/0/1`	Moves to interface configuration mode.
`Galveston(config-if)#ip address 172.16.40.2 255.255.255.252`	Assigns an IP address and netmask.
`Galveston(config-if)#no shutdown`	Enables the interface.
`Galveston(config-if)#exit`	Returns to global configuration mode.
`Galveston(config)#router ospf 1`	Starts OSPF process 1.
`Galveston(config-router)#network 172.16.40.2 0.0.0.0 area 0`	Any interface with an exact address of 172.16.40.2 is to be put into area 0. This is the most precise way to place an exact address into the OSPF routing process.
`Galveston(config-router)#network 172.16.50.1 0.0.0.0 area 0`	Read this line to say, "Any interface with an exact address of 172.16.50.1 is to be put into area 0."
`Galveston(config-router)#<ctrl> z`	Returns to privileged mode.
`Galveston#copy running-config startup-config`	Saves the configuration to NVRAM.

Configuration Example: Multiarea OSPF

Figure 3-3 shows the network topology for the configuration that follows, which demonstrates how to configure multiarea OSPF using the commands covered in this chapter.

Figure 3-3 Network Topology for Multiarea OSPF Configuration



ASBR Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname ASBR`	Sets the router hostname.
`ASBR(config)#interface loopback 0`	Enters loopback interface mode.
`ASBR(config-if)#ip address 192.168.1.1 255.255.255.255`	Assigns an IP address and netmask.
`ASBR(config-if)#description Router ID`	Sets a locally significant description.
`ASBR(config-if)#exit`	Returns to global configuration mode.
`ASBR(config)#interface fastethernet 0/0`	Enters interface configuration mode.
`ASBR(config-if)#ip address 172.16.1.1 255.255.255.0`	Assigns an IP address and netmask.
`ASBR(config-if)#no shutdown`	Enables the interface.
`ASBR(config-if)#exit`	Returns to global configuration mode.
`ASBR(config)#ip route 0.0.0.0 0.0.0.0 10.1.0.2 fastethernet 0/1`	Creates default route. Using both an exit interface and next-hop address on a Fast Ethernet interface prevents recursive lookups in the routing table.

`ASBR(config)#ip route 11.0.0.0 0.0.0.0 null0`	Creates a static route to a null interface. In this example, these routes represent a simulated remote destination.
`ASBR(config)#ip route 12.0.0.0 0.0.0.0 null0`	Creates a static route to a null interface. In this example, these routes represent a simulated remote destination.
`ASBR(config)#ip route 13.0.0.0 0.0.0.0 null0`	Creates a static route to a null interface. In this example, these routes represent a simulated remote destination.
`ASBR(config)#router ospf 1`	Starts OSPF process 1.
`ASBR(config-router)#network 172.16.1.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.1.*x* is to be put into area 0."
`ASBR(config-router)#default-information originate`	Sets the default route to be propagated to all OSPF routers.
`ASBR(config-router)#redistribute static`	Redistributes static routes into the OSPF process. This turns the router into an ASBR because static routes are not part of OSPF, and the definition of an ASBR is a router that sits between OSPF and another routing process—in this case, static routing.
`ASBR(config-router)#exit`	Returns to global configuration mode.
`ASBR(config)#exit`	Returns to privileged mode.
`ASBR#copy running-config startup-config`	Saves the configuration to NVRAM.

ABR-1 Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname ABR-1`	Sets the router hostname.
`ABR-1(config)#interface loopback 0`	Enters loopback interface mode.
`ABR-1(config-if)#ip address 192.168.2.1 255.255.255.255`	Assigns an IP address and netmask.

`ABR-1(config-if)#description Router ID`	Sets a locally significant description.
`ABR-1(config-if)#exit`	Returns to global configuration mode.
`ABR-1(config)#interface fastethernet 0/1`	Enters interface configuration mode.
`ABR-1(config-if)#ip address 172.16.1.2 255.255.255.0`	Assigns an IP address and netmask.
`ABR-1(config-if)#ip ospf priority 200`	Sets the priority for the DR/BDR election process. This router will win and become the DR.
`ABR-1(config-if)#no shutdown`	Enables the interface.
`ABR-1(config-if)#exit`	Returns to global configuration mode.
`ABR-1(config)#interface fastethernet 0/0`	Enters interface configuration mode.
`ABR-1(config-if)#ip address 172.16.51.1 255.255.255.0`	Assigns an IP address and netmask.
`ABR-1(config-if)#no shutdown`	Enables the interface.
`ABR-1(config-if)#exit`	Returns to global configuration mode.
`ABR-1(config)#router ospf 1`	Starts OSPF process 1.
`ABR-1(config-router)#network 172.16.1.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.1.*x* is to be put into area 0."
`ABR-1(config-router)#network 172.16.51.1 0.0.0.0 area 51`	Read this line to say, "Any interface with an exact address of 172.16.51.1 is to be put into area 51."
`ABR-1(config-router)#exit`	Returns to global configuration mode.
`ABR-1(config)#exit`	Returns to privileged mode.
`ABR-1(config)#copy running-config startup-config`	Saves the configuration to NVRAM.

ABR-2 Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname ABR-2`	Sets the router hostname.
`ABR-2(config)#interface loopback 0`	Enters loopback interface mode.
`ABR-2(config-if)#ip address 192.168.3.1 255.255.255.255`	Assigns an IP address and netmask.
`ABR-2(config-if)#description Router ID`	Sets a locally significant description.
`ABR-2(config-if)#exit`	Returns to global configuration mode.
`ABR-2(config)#interface fastethernet 0/0`	Enters interface configuration mode.
`ABR-2(config-if)#ip address 172.16.1.3 255.255.255.0`	Assigns an IP address and netmask.
`ABR-2(config-if)#ip ospf priority 100`	Sets the priority for the DR/BDR election process. This router will become the BDR to ABR-1's DR.
`ABR-2(config-if)#no shutdown`	Enables the interface.
`ABR-2(config-if)#exit`	Returns to global configuration mode.
`ABR-2(config)#interface serial 0/0/1`	Enters interface configuration mode.
`ABR-2(config-if)#ip address 172.16.10.5 255.255.255.252`	Assigns an IP address and netmask.
`ABR-2(config-if)#clock rate 56000`	Assigns a clock rate to the interface.
`ABR-2(config-if)#no shutdown`	Enables the interface.
`ABR-2(config-if)#exit`	Returns to global configuration mode.
`ABR-2(config)#router ospf 1`	Starts OSPF process 1.
`ABR-2(config-router)#network 172.16.1.0 0.0.0.255 area 0`	Read this line to say, "Any interface with an address of 172.16.1.*x* is to be put into area 0."

`ABR-2(config-router)#network 172.16.10.4 0.0.0.3 area 1`	Read this line to say, "Any interface with an address of 172.16.10.4–7 is to be put into area 1."
`ABR-2(config-router)#area 1 stub`	Makes area 1 a stub area. LSA type 4 and type 5s are blocked and not sent into area 1. A default route is injected into the stub area, pointing to the ABR.
`ABR-2(config-router)#exit`	Returns to global configuration mode.
`ABR-2(config)#exit`	Returns to privileged mode.
`ABR-2(config)#copy running-config startup-config`	Saves the configuration to NVRAM.

Internal Router

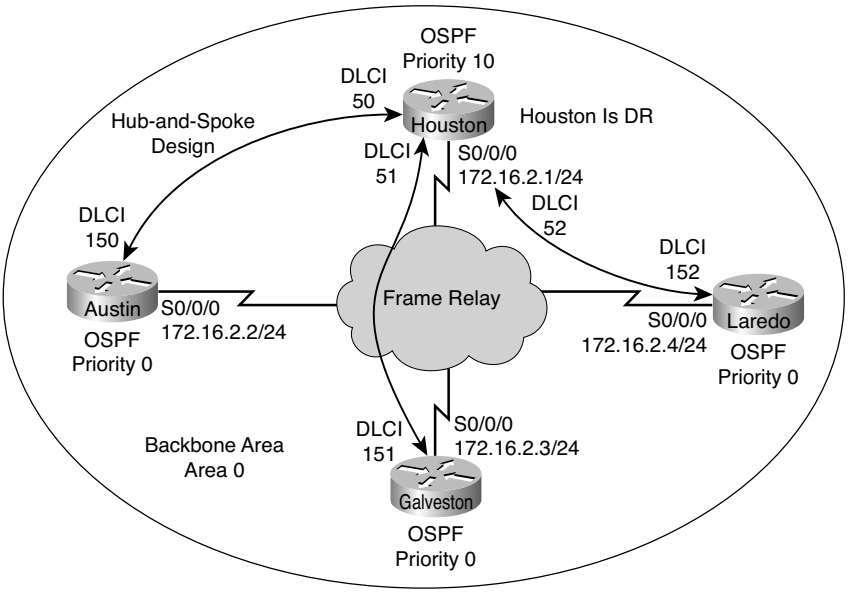
`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Internal`	Sets the router hostname.
`Internal(config)#interface loopback 0`	Enters loopback interface mode.
`Internal(config-if)#ip address 192.168.4.1 255.255.255.255`	Assigns an IP address and netmask.
`Internal(config-if)#description Router ID`	Sets a locally significant description.
`Internal(config-if)#exit`	Returns to global configuration mode.
`Internal(config)#interface fastethernet 0/0`	Enters interface configuration mode.
`Internal(config-if)#ip address 172.16.20.1 255.255.255.0`	Assigns an IP address and netmask.
`Internal(config-if)#no shutdown`	Enables the interface.
`Internal(config-if)#exit`	Returns to global configuration mode.
`Internal(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Internal(config-if)#ip address 172.16.10.6 255.255.255.252`	Assigns an IP address and netmask.
`Internal(config-if)#no shutdown`	Enables the interface.
`Internal(config-if)#exit`	Returns to global configuration mode.

`Internal(config)#router ospf 1`	Starts OSPF process 1.
`Internal(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an address of 172.16.*x*.*x* is to be put into area 0."
`Internal(config-router)#area 1 stub`	Makes area 1 a stub area.
`Internal(config-router)#exit`	Returns to global configuration mode.
`Internal(config)#exit`	Returns to privileged mode.
`Internal(config)#copy running-config startup-config`	Saves the configuration to NVRAM.

Configuration Example: OSPF and NBMA Networks

Figure 3-4 shows the network topology for the configuration that follows, which demonstrates how to configure OSPF on an NBMA network using the commands covered in this chapter.

Figure 3-4 Network Topology for OSPF Configuration on an NBMA Network



Houston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Houston`	Sets the router hostname.
`Houston(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Houston(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Houston(config-if)#ip address 172.16.2.1 255.255.255.0`	Assigns an IP address and netmask.
`Houston(config-if)#frame-relay map ip 172.16.2.2 50`	Maps the remote IP address to local DLCI 50.
`Houston(config-if)#frame-relay map ip 172.16.2.3 51`	Maps the remote IP address to local DLCI 51.
`Houston(config-if)#frame-relay map ip 172.16.2.4 52`	Maps the remote IP address to local DLCI 52.
`Houston(config-if)#no shutdown`	Enables the interface.
`Houston(config-if)#exit`	Returns to global configuration mode.
`Houston(config)#router ospf 1`	Starts OSPF process 1.
`Houston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Houston(config-router)#neighbor 172.16.2.2`	Identifies neighbor (Austin) to Houston.
`Houston(config-router)#neighbor 172.16.2.3`	Identifies neighbor (Galveston) to Houston.
`Houston(config-router)#neighbor 172.16.2.4`	Identifies neighbor (Laredo) to Houston.
`Houston(config-router)#exit`	Returns to global configuration mode
`Houston(config)#exit`	Returns to privileged mode.
`Houston#copy running-config startup-config`	Saves the configuration to NVRAM.

Austin Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Austin`	Sets the router hostname.
`Austin(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Austin(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Austin(config-if)#ip address 172.16.2.2 255.255.255.0`	Assigns an IP address and netmask.
`Austin(config-if)#frame-relay map ip 172.16.2.1 150`	Maps the remote IP address to local DLCI 150.
`Austin(config-if)#frame-relay map ip 172.16.2.3 150`	Maps the remote IP address to local DLCI 150.
`Austin(config-if)#frame-relay map ip 172.16.2.4 150`	Maps the remote IP address to local DLCI 150.
`Austin(config-if)#no shutdown`	Enables the interface.
`Austin(config-if)#exit`	Returns to global configuration mode.
`Austin(config)#router ospf 1`	Starts OSPF process 1.
`Austin(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Austin(config-router)#neighbor 172.16.2.1 priority 10`	Identifies neighbor (Houston) to Austin and assigns a priority of 10 to Houston for DR/BDR election.
`Austin(config-router)#exit`	Returns to global configuration mode.
`Austin(config)#exit`	Returns to privileged mode.
`Austin#copy running-config startup-config`	Saves the configuration to NVRAM.

Galveston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Galveston`	Sets the router hostname.
`Galveston(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Galveston(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Galveston(config-if)#ip address 172.16.2.3 255.255.255.0`	Assigns an IP address and netmask.
`Galveston(config-if)#frame-relay map ip 172.16.2.1 151`	Maps the remote IP address to local DLCI 151.
`Galveston(config-if)#frame-relay map ip 172.16.2.2 151`	Maps the remote IP address to local DLCI 151.
`Galveston(config-if)#frame-relay map ip 172.16.2.4 151`	Maps the remote IP address to local DLCI 151.
`Galveston(config-if)#no shutdown`	Enables the interface.
`Galveston(config-if)#exit`	Returns to global configuration mode.
`Galveston(config)#router ospf 1`	Starts OSPF process 1.
`Galveston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x.x* will be placed into area 0."
`Galveston(config-router)#neighbor 172.16.2.1 priority 10`	Identifies neighbor (Houston) to Galveston and assigns a priority of 10 to Houston for DR/BDR election.
`Galveston(config-router)#exit`	Returns to global configuration mode.
`Galveston(config)#exit`	Returns to privileged mode.
`Galveston#copy running-config startup-config`	Saves the configuration to NVRAM.

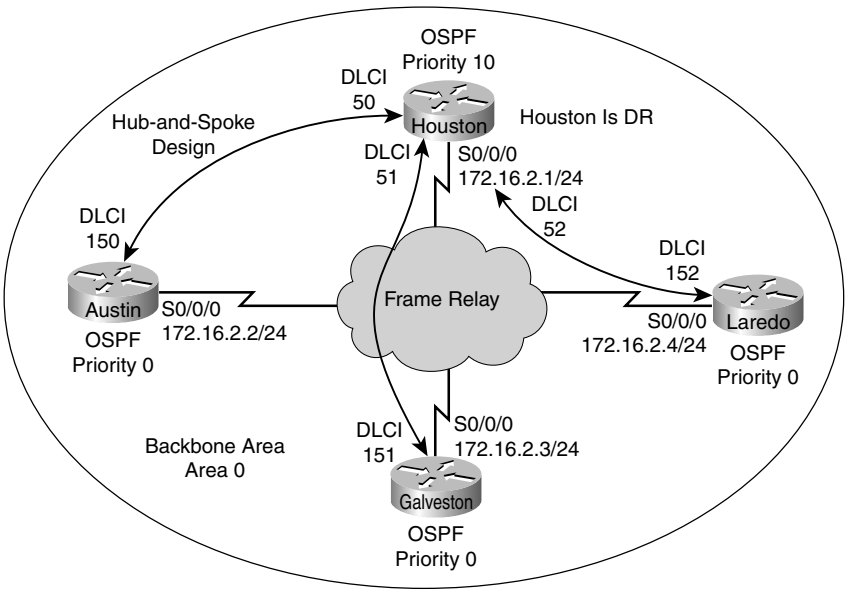
Laredo Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Laredo`	Sets the router hostname.
`Laredo(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Laredo(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Laredo(config-if)#ip address 172.16.2.4 255.255.255.0`	Assigns an IP address and netmask.
`Laredo(config-if)#frame-relay map ip 172.16.2.1 152`	Maps the remote IP address to local DLCI 152.
`Laredo(config-if)#frame-relay map ip 172.16.2.2 152`	Maps the remote IP address to local DLCI 152.
`Laredo(config-if)#frame-relay map ip 172.16.2.3 152`	Maps the remote IP address to local DLCI 152.
`Laredo(config-if)#no shutdown`	Enables the interface.
`Laredo(config-if)#exit`	Returns to global configuration mode.
`Laredo(config)#router ospf 1`	Starts OSPF process 1.
`Laredo(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Laredo(config-router)#neighbor 172.16.2.1 priority 10`	Identifies neighbor (Houston) to Laredo and assigns a priority of 10 to Houston for DR/BDR election.
`Laredo(config-router)#exit`	Returns to global configuration mode.
`Laredo(config)#exit`	Returns to privileged mode.
`Laredo#copy running-config startup-config`	Saves the configuration to NVRAM.

Configuration Example: OSPF and Broadcast Networks

Figure 3-5 shows the network topology for the configuration that follows, which demonstrates how to configure OSPF on a broadcast network using the commands covered in this chapter.

Figure 3-5 Network Topology for OSPF Configuration on a Broadcast Network



Houston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Houston`	Sets the router hostname.
`Houston(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Houston(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Houston(config-if)#ip address 172.16.2.1 255.255.255.0`	Assigns an IP address and netmask.
`Houston(config-if)#ip ospf network broadcast`	Changes the network type from the default NBMA to broadcast.

`Houston(config-if)#ip ospf priority 10`	Sets the priority to 10 for the DR/BDR election process.
`Houston(config-if)#frame-relay map ip 172.16.2.2 50 broadcast`	Maps the remote IP address to local DLCI 50. Broadcast and multicasts will now be forwarded.
`Houston(config-if)#frame-relay map ip 172.16.2.3 51 broadcast`	Maps the remote IP address to local DLCI 51. Broadcast and multicasts will now be forwarded.
`Houston(config-if)#frame-relay map ip 172.16.2.4 52 broadcast`	Maps the remote IP address to local DLCI 52. Broadcast and multicasts will now be forwarded.
`Houston(config-if)#no shut`	Enables the interface.
`Houston(config-if)#exit`	Returns to global configuration mode.
`Houston(config)#router ospf 1`	Starts OSPF process 1.
`Houston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Houston(config-router)#exit`	Returns to global configuration mode.
`Houston(config)#exit`	Returns to privileged mode.
`Houston#copy running-config startup-config`	Saves the configuration to NVRAM.

Austin Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Austin`	Sets the router hostname.
`Austin(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Austin(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Austin(config-if)#ip address 172.16.2.2 255.255.255.0`	Assigns an IP address and netmask.

`Austin(config-if)#ip ospf network broadcast`	Changes the network type from the default NBMA to broadcast.
`Austin(config-if)#ip ospf priority 0`	Sets the priority to 0 for the DR/BDR election process. Austin will not participate in the election process.
`Austin(config-if)#frame-relay map ip 172.16.2.1 150 broadcast`	Maps the remote IP address to local DLCI 150. Broadcast and multicasts will now be forwarded.
`Austin(config-if)#frame-relay map ip 172.16.2.3 150 broadcast`	Maps the remote IP address to local DLCI 150. Broadcast and multicasts will now be forwarded.
`Austin(config-if)#frame-relay map ip 172.16.2.4 150 broadcast`	Maps the remote IP address to local DLCI 150. Broadcast and multicasts will now be forwarded.
`Austin(config-if)#no shutdown`	Enables the interface.
`Austin(config-if)#exit`	Returns to global configuration mode.
`Austin(config)#router ospf 1`	Starts OSPF process 1.
`Austin(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Austin(config-router)#exit`	Returns to global configuration mode.
`Austin(config)#exit`	Returns to privileged mode.
`Austin#copy running-config startup-config`	Saves the configuration to NVRAM.

Galveston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Galveston`	Sets the router hostname.
`Galveston(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Galveston(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.

`Galveston(config-if)#ip address 172.16.2.3 255.255.255.0`	Assigns an IP address and netmask.
`Galveston(config-if)#ip ospf network broadcast`	Changes the network type from the default NBMA to broadcast.
`Galveston(config-if)#ip ospf priority 0`	Sets the priority to 0 for the DR/BDR election process. Galveston will not participate in the election process.
`Galveston(config-if)#frame-relay map ip 172.16.2.1 151 broadcast`	Maps the remote IP address to local DLCI 151. Broadcast and multicasts will now be forwarded.
`Galveston(config-if)#frame-relay map ip 172.16.2.2 151 broadcast`	Maps the remote IP address to local DLCI 151. Broadcast and multicasts will now be forwarded.
`Galveston(config-if)#frame-relay map ip 172.16.2.4 151 broadcast`	Maps the remote IP address to local DLCI 151. Broadcast and multicasts will now be forwarded.
`Galveston(config-if)#no shutdown`	Enables the interface.
`Galveston(config-if)#exit`	Returns to global configuration mode.
`Galveston(config)#router ospf 1`	Starts OSPF process 1.
`Galveston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Galveston(config-router)#exit`	Returns to global configuration mode.
`Galveston(config)#exit`	Returns to privileged mode.
`Galveston#copy running-config startup-config`	Saves the configuration to NVRAM.

Laredo Router

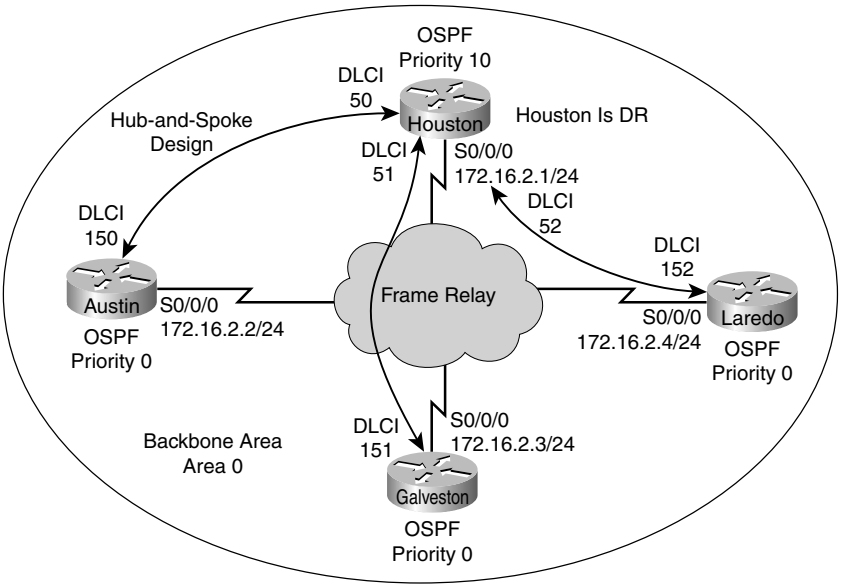
`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Laredo`	Sets the router hostname.
`Laredo(config)#interface serial 0/0/0`	Enters interface configuration mode.

`Laredo(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Laredo(config-if)#ip address 172.16.2.4 255.255.255.0`	Assigns an IP address and netmask.
`Laredo(config-if)#ip ospf network broadcast`	Changes the network type from the default NBMA to broadcast.
`Laredo(config-if)#ip ospf priority 0`	Sets the priority to 0 for the DR/BDR election process. Laredo will not participate in the election process.
`Laredo(config-if)#frame-relay map ip 172.16.2.1 152 broadcast`	Maps the remote IP address to local DLCI 152. Broadcast and multicasts will now be forwarded.
`Laredo(config-if)#frame-relay map ip 172.16.2.2 152 broadcast`	Maps the remote IP address to local DLCI 152. Broadcast and multicasts will now be forwarded.
`Laredo(config-if)#frame-relay map ip 172.16.2.3 152 broadcast`	Maps the remote IP address to local DLCI 152. Broadcast and multicasts will now be forwarded.
`Laredo(config-if)#no shutdown`	Enables the interface.
`Laredo(config-if)#exit`	Returns to global configuration mode.
`Laredo(config)#router ospf 1`	Starts OSPF process 1.
`Laredo(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Laredo(config-router)#exit`	Returns to global configuration mode.
`Laredo(config)#exit`	Returns to privileged mode.
`Laredo#copy running-config startup-config`	Saves the configuration to NVRAM.

Configuration Example: OSPF and Point-to-Multipoint Networks

Figure 3-6 shows the network topology for the configuration that follows, which demonstrates how to configure OSPF on a point-to-multipoint network using the commands covered in this chapter.

Figure 3-6 Network Topology for OSPF Configuration on a Point-to-Multipoint Network



Houston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to privileged mode.
`Router(config)#hostname Houston`	Sets the router hostname.
`Houston(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Houston(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Houston(config-if)#ip address 172.16.2.1 255.255.255.0`	Assigns an IP address and netmask.
`Houston(config-if)#ip ospf network point-to-multipoint`	Changes the network type from the default NBMA to point-to-multipoint.
`Houston(config-if)#frame-relay map ip 172.16.2.2 50`	Maps the remote IP address to local DLCI 50.
`Houston(config-if)#frame-relay map ip 172.16.2.3 51`	Maps the remote IP address to local DLCI 51.

`Houston(config-if)#frame-relay map ip 172.16.2.4 52`	Maps the remote IP address to local DLCI 52.
`Houston(config-if)#no shutdown`	Enables the interface.
`Houston(config-if)#exit`	Returns to global configuration mode.
`Houston(config)#router ospf 1`	Enables OSPF process 1.
`Houston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Houston(config-router)#exit`	Returns to global configuration mode.
`Houston(config)#exit`	Returns to privileged mode.
`Houston#copy running-config startup-config`	Saves the configuration to NVRAM.

Austin Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Austin`	Sets the router hostname.
`Austin(config)#interface serial 0/0/0`	Enters serial interface mode.
`Austin(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Austin(config-if)#ip address 172.16.2.2 255.255.255.0`	Assigns an IP address and netmask.
`Austin(config-if)#ip ospf network point-to-multipoint`	Changes the network type from the default NBMA to point-to-multipoint.
`Austin(config-if)#frame-relay map ip 172.16.2.1 150`	Maps the remote IP address to local DLCI 150.
`Austin(config-if)#frame-relay map ip 172.16.2.3 150`	Maps the remote IP address to local DLCI 150.
`Austin(config-if)#frame-relay map ip 172.16.2.4 150`	Maps the remote IP address to local DLCI 150.

`Austin(config-if)#no shutdown`	Enables the interface.
`Austin(config-if)#exit`	Returns to global configuration mode.
`Austin(config)#router ospf 1`	Starts OSPF process 1.
`Austin(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Austin(config-router)#exit`	Returns to global configuration mode.
`Austin(config)#exit`	Returns to privileged mode.
`Austin#copy running-config startup-config`	Saves the configuration to NVRAM.

Galveston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Galveston`	Sets the router hostname.
`Galveston(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Galveston(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Galveston(config-if)#ip address 172.16.2.3 255.255.255.0`	Assigns an IP address and netmask.
`Galveston(config-if)#ip ospf network point-to-multipoint`	Changes the network type from the default NBMA to point-to-multipoint.
`Galveston(config-if)#frame-relay map ip 172.16.2.1 151`	Maps the remote IP address to local DLCI 151.
`Galveston(config-if)#frame-relay map ip 172.16.2.2 151`	Maps the remote IP address to local DLCI 151.
`Galveston(config-if)#frame-relay map ip 172.16.2.4 151`	Maps the remote IP address to local DLCI 151.
`Galveston(config-if)#no shutdown`	Enables the interface.

`Galveston(config-if)#exit`	Returns to global configuration mode.
`Galveston(config)#router ospf 1`	Starts OSPF process 1.
`Galveston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x.x* will be placed into area 0."
`Galveston(config-router)#exit`	Returns to global configuration mode.
`Galveston(config)#exit`	Returns to privileged mode.
`Galveston#copy running-config startup-config`	Saves the configuration to NVRAM.

Laredo Router

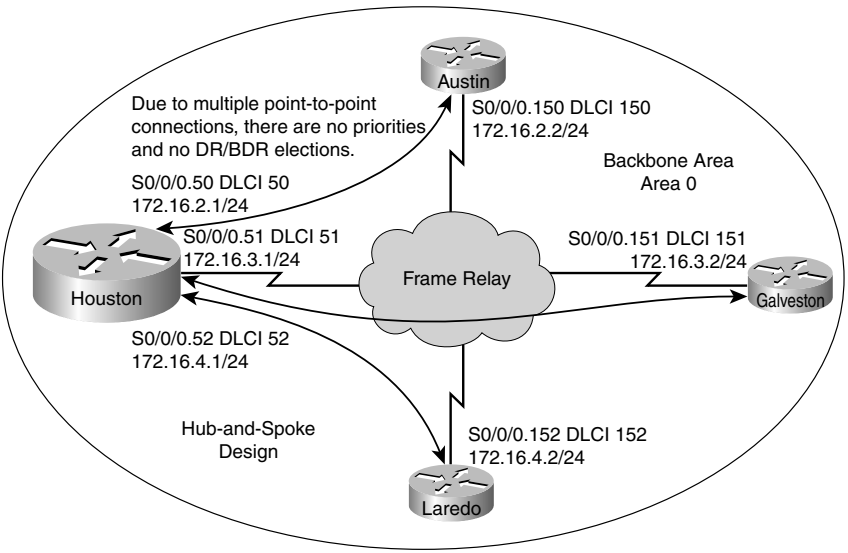
`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Laredo`	Sets the router hostname.
`Laredo(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Laredo(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Laredo(config-if)#ip address 172.16.2.4 255.255.255.0`	Assigns an IP address and netmask.
`Laredo(config-if)#ip ospf network point-to-multipoint`	Changes the network type from the default NBMA to point-to-multipoint.
`Laredo(config-if)#frame-relay map ip 172.16.2.1 152`	Maps the remote IP address to local DLCI 152.
`Laredo(config-if)#frame-relay map ip 172.16.2.2 152`	Maps the remote IP address to local DLCI 152.
`Laredo(config-if)#frame-relay map ip 172.16.2.3 152`	Maps the remote IP address to local DLCI 152.
`Laredo(config-if)#no shutdown`	Enables the interface.
`Laredo(config-if)#exit`	Returns to global configuration mode.

`Laredo(config)#router ospf 1`	Starts OSPF process 1.
`Laredo(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Laredo(config-router)#exit`	Returns to global configuration mode.
`Laredo(config)#exit`	Returns to privileged mode.
`Laredo#copy running-config startup-config`	Saves the configuration to NVRAM.

Configuration Example: OSPF and Point-to-Point Networks Using Subinterfaces

Figure 3-7 shows the network topology for the configuration that follows, which demonstrates how to configure OSPF on a point-to-point network using subinterfaces, using the commands covered in this chapter.

Figure 3-7 Network Topology for OSPF Configuration on a Point-to-Point Network Using Subinterfaces



Houston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Houston`	Sets the router hostname.
`Houston(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Houston(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Houston(config-if)#no shutdown`	Enables the interface.
`Houston(config-if)#interface serial 0/0/0.50 point-to-point`	Creates a subinterface.
`Houston(config-subif)#description Link to Austin`	Creates a locally significant description of the interface.
`Houston(config-subif)#ip address 172.16.2.1 255.255.255.252`	Assigns an IP address and netmask.
`Houston(config-subif)#frame-relay interface-dlci 50`	Assigns a DLCI to the subinterface.
`Houston(config-subif)#exit`	Returns to interface configuration mode.
`Houston(config-if)#interface serial 0/0/0.51 point-to-point`	Creates a subinterface.
`Houston(config-subif)#description Link to Galveston`	Creates a locally significant description of the interface.
`Houston(config-subif)#ip address 172.16.3.1 255.255.255.252`	Assigns an IP address and netmask.
`Houston(config-subif)#frame-relay interface-dlci 51`	Assigns a DLCI to the subinterface.
`Houston(config-subif)#exit`	Returns to interface configuration mode.
`Houston(config-if)#interface serial 0/0/0.52 point-to-point`	Creates a subinterface.
`Houston(config-subif)#description Link to Laredo`	Creates a locally significant description of the interface.

`Houston(config-subif)#ip address 172.16.4.1 255.255.255.252`	Assigns an IP address and netmask.
`Houston(config-subif)#frame-relay interface-dlci 52`	Assigns a DLCI to the subinterface.
`Houston(config-subif)#exit`	Returns to interface configuration mode.
`Houston(config-if)#exit`	Returns to global configuration mode.
`Houston(config)#router ospf 1`	Starts OSPF process 1.
`Houston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Houston(config-router)#exit`	Returns to global configuration mode.
`Houston(config)#exit`	Returns to privileged mode.
`Houston#copy running-config startup-config`	Saves the configuration to NVRAM.

Austin Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Austin`	Sets the router hostname.
`Austin(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Austin(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Austin(config-if)#no shutdown`	Enables the interface.
`Austin(config-if)#interface serial 0/0/0.150 point-to-point`	Creates a subinterface.
`Austin(config-subif)#description Link to Houston`	Creates a locally significant description of the interface.
`Austin(config-subif)#ip address 172.16.2.2 255.255.255.252`	Assigns an IP address and netmask.
`Austin(config-subif)#frame-relay interface-dlci 150`	Assigns a DLCI to the subinterface.

`Austin(config-subif)#exit`	Returns to interface configuration mode.
`Austin(config-if)#exit`	Returns to global configuration mode.
`Austin(config)#router ospf 1`	Starts OSPF process 1.
`Austin(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Austin(config-router)#exit`	Returns to global configuration mode.
`Austin(config)#exit`	Returns to privileged mode.
`Austin#copy running-config startup-config`	Saves the configuration to NVRAM.

Galveston Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Galveston`	Sets the router hostname.
`Galveston(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Galveston(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Galveston(config-if)#no shutdown`	Enables the interface.
`Galveston(config-if)#interface serial 0/0/0.151 point-to-point`	Creates a subinterface.
`Galveston(config-subif)#description Link to Houston`	Creates a locally significant description of the interface.
`Galveston(config-subif)#ip address 172.16.3.2 255.255.255.252`	Assigns an IP address and netmask.
`Galveston(config-subif)#frame-relay interface-dlci 151`	Assigns a DLCI to the subinterface.
`Galveston(config-subif)#exit`	Returns to interface configuration mode.
`Galveston(config-if)#exit`	Returns to global configuration mode.

`Galveston(config)#router ospf 1`	Starts OSPF process 1.
`Galveston(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Galveston(config-router)#exit`	Returns to global configuration mode.
`Galveston(config)#exit`	Returns to privileged mode.
`Galveston#copy running-config startup-config`	Saves the configuration to NVRAM.

Laredo Router

`Router>enable`	Moves to privileged mode.
`Router#configure terminal`	Moves to global configuration mode.
`Router(config)#hostname Laredo`	Sets the router hostname.
`Laredo(config)#interface serial 0/0/0`	Enters interface configuration mode.
`Laredo(config-if)#encapsulation frame-relay`	Enables Frame Relay encapsulation.
`Laredo(config-if)#no shutdown`	Enables the interface.
`Laredo(config-if)#interface serial 0/0/0.152 point-to-point`	Creates a subinterface.
`Laredo(config-subif)#description Link to Houston`	Creates a locally significant description of the interface.
`Laredo(config-subif)#ip address 172.16.4.2 255.255.255.252`	Assigns an IP address and netmask.
`Laredo(config-subif)#frame-relay interface-dlci 152`	Assigns a DLCI to the subinterface.
`Laredo(config-subif)#exit`	Returns to interface configuration mode.
`Laredo(config-if)#exit`	Returns to global configuration mode.
`Laredo(config)#router ospf 1`	Starts OSPF process 1.

`Laredo(config-router)#network 172.16.0.0 0.0.255.255 area 0`	Read this line to say, "Any interface with an IP address of 172.16.*x*.*x* will be placed into area 0."
`Laredo(config-router)#exit`	Returns to global configuration mode.
`Laredo(config)#exit`	Returns to privileged mode.
`Laredo#copy running-config startup-config`	Saves the configuration to NVRAM.