

CISCO ROUTERS FOR THE DESPERATE™ 2ND EDITION

ROUTER AND SWITCH MANAGEMENT,
THE EASY WAY

MICHAEL W. LUCAS



Cisco Routers for the Desperate, 2nd Edition

Michael W. Lucas

Michael W. Lucas

Copyright © 2009

For information on book distributors or translations, please contact No Starch Press, Inc. directly:
No Starch Press, Inc.

555 De Haro Street, Suite 250, San Francisco, CA 94107

phone: 415.863.9900; fax: 415.863.9950; info@nostarch.com; www.nostarch.com

Library of Congress Cataloging-in-Publication Data:

Lucas, Michael, 1967-

Cisco routers for the desperate : router and switch management, the easy way / Michael W. Lucas. -- 2nd ed.

p. cm.

Includes index.

ISBN-13: 978-1-59327-193-0

ISBN-10: 1-59327-193-X

1. Routers (Computer networks) 2. Routing (Computer network management) 3. Cisco IOS.
4. Cisco Systems, Inc.

I. Title.

TK5105.543.L83 2009

004.6--dc22

2008050188

No Starch Press and the No Starch Press logo are registered trademarks of No Starch Press, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an "As Is" basis, without warranty. While every precaution has been taken in the preparation of this work, neither the author nor No Starch Press, Inc. shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in it.

No Starch Press

Dedication

To all those poor bastards who are awake at oh-dark-thirty trying to get their router working.

Thanks to Liz, for not smothering me in my sleep as I wrote this book.

Introduction



Welcome to *Cisco Routers for the Desperate!* While network and system administrators know that they should intimately understand every piece of equipment in their control, in the real world, we all become most familiar with the things that require the most care and feeding. We all spend more time than we'd like arguing with buggy operating systems and vexing applications, and as a result, we are painfully familiar with their foibles. Cisco routers "just work" almost all the time, so we don't get much chance to become familiar with them. This gives the router a certain air of the "mysterious black box."

When the router actually breaks, the mysterious black box immediately becomes an object of fear and dread. Loss of Internet access triggers the fight-or-flight instinct. People panic. Pretty soon, everyone's running around as if they have a drunken badger loose in their undies. While that can be amusing to watch, it doesn't get the Internet fixed. Perhaps you call your Internet service provider and beg for help—always a legitimate tactic, but perhaps not the quickest and certainly not the best for your self-respect. You could go buy a book on Cisco routers, but most of those that actually contain useful information weigh roughly the same as an automobile transmission and are almost as digestible. And we all know you won't have time to sit down and read it during an actual emergency, will you?

This book is aimed at the administrator of a small network who is responsible for anywhere from one to three Cisco routers in addition to his more time-consuming duties. Our goal is to make you comfortable with the Cisco environment and provide exactly the information you need to troubleshoot and resolve the most common problems that either happen on Cisco routers or intimately involve Cisco routers. Once you have that under your belt, we go a little further and provide the basic information you need to address some of the questions that frequently arise when working with Cisco routers, network connections, and switches. This book is also short enough to actually be read by someone who doesn't have to manage routers every single day.

This is the book I wish I had read before I first touched a Cisco device.

This book is not a complete tutorial on all of the things that can be done with Cisco equipment. Cisco almost invented the networking business, and its hardware has been used to interconnect almost every sort of network hardware ever created. We focus squarely on TCP/IP networking in an Internet environment and do not detour into channelizing circuits, VoIP, IPX, AppleTalk, MPLS, prioritization, or any of the other features available in Cisco-land. Instead, you'll discover how to learn these things yourself. Once you're comfortable with Cisco routers, everything else follows.

What This Book Will Teach You

The main goal of this book is to make you comfortable with the environment and technology found inside every Cisco router. First, you have to connect to the router and log in. You also need to examine interfaces, check the configuration, and change the system's current and startup configurations as well as run some basic troubleshooting commands.

To make the router work properly, you need a basic comprehension of the network connections most commonly found on modern TCP/IP networks. We'll briefly cover Ethernet and serial lines and give some pointers to further information. This isn't a primer on networking, but you need to understand enough theory to understand why things aren't working.

We'll then discuss the most common router usage, connecting a small network to the Internet. A step-by-step discussion of the typical configuration choices will help you preconfigure a router so that you can quickly and easily establish a new connection with a minimum of fuss. This will allow you to ask your ISP for the specific information you need, rather than their install tech telling you what you need to obediently type. To make your Internet connection work, you'll also have to configure the router's Ethernet interface, so we'll discuss that as well.

Perhaps the second most common use of a router is to connect two offices with a private leased line. This is a step up from connecting a network to an ISP, and we'll cover how to make this happen, beginning with ordering the circuit from the telephone company. Because you'll be responsible for the circuit, we'll cover troubleshooting as well.

As good as Cisco equipment is, its products require occasional software updates to solve stability and security problems. We'll take you through doing this safely and reliably, and we'll discuss recovering from the most common problems.

Whenever the office Internet connection goes down, managers all have the same question: "What would it take to get redundancy?" We'll unveil the mysteries of BGP, the Border Gateway Protocol, and show how you, as a small network provider, can use BGP to provide some measure of network redundancy. We'll also discuss the problems associated with BGP and some workarounds that can solve those problems.

Once you've mastered all of this, adding Ethernet switch expertise is easy. We'll detour into the Ethernet switching realm so that you can manage the hardware supporting your LAN as well as your WAN.

Finally, Cisco routers can take advantage of a variety of network services that you might already offer and include other interfaces to more easily fit with your network. These include such basic protocols as SSH (Secure Shell), NTP (Network Time Protocol), and SNMP (Simple Network Management Protocol). We'll briefly discuss how to configure these and more.

Where You Can Learn the Rest

If you need more information, it's available elsewhere for either a small fee or at no cost.

By far, my favorite source of information on a Cisco problem (or any problem) is Google. Usually, a quick web query with well-chosen search terms will return a helpful answer on the first page. The problem, of course, is finding those "well-chosen search terms." Usually, the word "Cisco" and a snippet of the technology or error message will suffice to provide an answer. Be sure to check both the "Groups" and "Web" search functions, because Google indexes the entire history of Usenet (including even the early days of Cisco). Some error messages have not changed for 20 years, and neither have their fixes. (Feel free to use your preferred search engine, of course.)

If you're a Usenet (aka "Internet News") user, check the newsgroup `comp.dcom.sys.cisco`. While this group hasn't had an FAQ update in several years, it's still actively used and is an excellent place to get pointers to further information on almost any Cisco-related topic.

The single most authoritative information on your router is available on the Cisco website, <http://www.cisco.com>. Documentation for all router models, modules, and other hardware appears there along with tutorials for configuring features specific to each model or module.

Although all of the above-mentioned items are free, I cannot recommend a Cisco SmartNet contract strongly enough. The cost is small compared to the cost of the router, and having a contract ensures that a Cisco engineer will immediately be available to you whenever a critical problem occurs. If you have to resolve a router problem on your own, you may find yourself struggling for hours or days. But if you have a contract, your service might well be restored within minutes of a simple phone call. The cost of downtime when you're struggling with a problem on your own almost certainly far exceeds the cost of the SmartNet contract.

Additionally, Cisco tends to very quickly respond to low-priority Technical Assistance Requests that SmartNet customers can enter on its website. I've come to expect answers within an hour on even trivial questions and have never waited more than a day. While Cisco does not guarantee this level of service, rapid response is very routine for them. You can spend far longer than that sorting through almost-but-not-quite-right answers on the Internet.

Unpacking the Router

If you're lucky, your first introduction to Cisco routers will be a brown box arriving on your loading dock. This means that you'll learn how to configure the router by actually installing it, which is much easier than having to thoroughly understand a working setup. If you aren't this lucky, just pretend that your server room is a really *big* brown box and follow along. You may have to search the office for the place where your predecessor stashed some of the router accessories, but at least you'll know what you're looking for!

While the most interesting part is probably the router itself, be sure to find the console cable and adapter as well as the manuals and CD-ROMs. In a modern Cisco router, the console cable is usually a flat blue cable with an RJ-45 connector on each end and an RJ-45-to-DB-9 adapter to let you attach the serial port on your computer. Hang on to this cable because you will need it later. (Because Cisco console cables tend to wander off if left unguarded, I keep a spare locked in my workshop in a fireproof safe labeled "Caution: Live Plague Virus.")

The manuals are generally available on Cisco's website if you have a SmartNet contract, but a paper copy is nicer to read while sprawled on the couch. Many modern Cisco routers are *modular*; the router chassis has several slots where cards containing connectors for different types of networks can be plugged in. Each card or component has its own manual. While we will cover basic configuration of the most common network interfaces, you should definitely check the manual for any special requirements for your modules.

Lastly, the CD-ROMs contain router software, additional documentation, and add-on tools that can help you manage your router. Keep this software; it can help greatly in disaster-recovery situations.

Break out the router chassis manual and look at the router itself. Things you should identify include the console port, the aux port, and the network interfaces. Let's start with the console port.

Warning

Do not power your new router on yet!

Console Setup

Your Cisco router has a port resembling a standard Ethernet RJ-45 port, labeled "console." You received a console cable with your router; plug one end into your router and attach the serial port to a 9-pin serial port on a computer.

Now you need serial port client software on your computer. We'll assume that you're using HyperTerminal for Windows to connect to the router. HyperTerminal is available on all versions of Windows. If you're using another operating system, it must have a serial port program available. Free Unix-like operating systems usually include "tip" or "cu," and "kermit" and "mini-com" are also available for many different operating systems. Specifics of configuring these programs vary, but the settings we use here will work for any of them.

In modern Windows versions, HyperTerminal may not be installed by default. If you look under

Start ► Programs ► Accessories ► Communications and don't see HyperTerminal, it's not yet installed on your system. Go to the Add/Remove Programs control panel and select "Windows Components." HyperTerminal is part of the "Communications" feature set.

Once you have your serial client software installed, set up a new connection to your serial port. Use the following settings:

- 9600 baud
- No parity
- 8 data bits
- 1 stop bit
- No flow control

Click the "connect" icon. Because your router has no power yet, you shouldn't see anything happen.

Other Router Ports

The aux port looks just like the console port, and you can even plug in your console cable and get a command line on your serial client. A router doesn't display its bootup messages on the aux port, however. For the most part, the aux port is there only in case something goes wrong. If you misconfigure your console port and find yourself locked out of the system, you can log in via the aux port and fix the router's configuration. In the event that you really botch an upgrade, you can set your aux port to use a high-speed connection and upload a new software image to the router over it. You might connect a modem to your console port for working remotely and use the aux port for local work.

Network interfaces look different from router to router, but for the most part, they have a connector appropriate to the type of network they connect to. Ethernet interfaces usually have either a standard Cat 5 or fiber connection, while T1 interfaces look like Cat 5 connectors. Larger circuits, such as DS3s and OC3s, have coaxial connections. Look at your router and identify the type of network interfaces it supports; you'll need to know that later.

Power-On

With your serial connection in the console port, plug in the router, and in a few seconds you'll see some activity.

```
System Bootstrap, (b)Version 12.2(1r) [dchih 1r], RELEASE SOFTWARE (fc1)
Copyright (c) 2001 by cisco Systems, Inc.
(c) C7200 platform with 131072 Kbytes of main memory

(d) Self decompressing the image : #####
...

```

You'll see a variety of facts about the equipment, such as the boot loader version (b) and the hardware platform (c). The system then starts loading the operating system (d) and heads for the main boot sequence. We'll interpret most of this output later. Let the router keep booting until it wants to talk to you. If this is the first boot, the router will offer to let you configure it:

```
Would you like to enter the initial configuration dialog? [yes/no]:
```

The configuration dialog is designed to get a bare minimum system up on the network, allowing you to use a configuration tool such as CiscoWorks to complete the setup. For reasons we'll discuss in the

next chapter, I encourage you to perform all the configuration at the command line instead of via one of these tools. ~~(Once you've read this book and understand the implications of your choices, you might want to use the configuration dialog to perform basic setup.)~~ For that reason, I recommend that you skip the configuration dialog. This will drop you to a router prompt.

```
router>
```

Congratulations! You're now actually logged in to the router. Now, let's see what you can do with it.

Chapter 1. Befriending the Command Line



This book focuses entirely on the command line. Why? Some Cisco routers include a web interface; others don't. Cisco sells advanced router management packages, but they require a complicated client environment that you might not have available when you really need it. You might only need to log in to the router once a year or less; what are the chances that your desktop system with all the fancy management tools has undergone some drastic surgery in the meantime, and the router tools no longer work the way you expect? Personally, I detest trying to solve Windows problems when the router is down. The one environment available on every Cisco router is the command line.

The command line gives you absolute control over every aspect of your router's behavior. Some configuration changes can only be realistically made at the command line. If your router loses its configuration, you'll need to use the command line to restore enough of its mind that your fancy tool can talk to it. So, if you must use the command line in dire circumstances, it's best that you learn enough to be comfortable there.

EXEC versus Privileged EXEC Mode

By default, when you first log in to a Cisco router you are in *EXEC mode*. You can execute basic diagnostic commands and look at things, but you cannot actually change configuration settings or view sensitive information. For example, in EXEC mode, you can see if an interface is receiving errors, and you can ping across a circuit to see if it is working, but you cannot reset the interface. In EXEC mode, the command prompt ends in a "greater than" sign.

```
router>
```

To change anything, or to run some more intrusive commands, you must use *privileged EXEC mode*, which is a superuser or administrator security level protected by a unique password. Privileged EXEC mode is usually called "enable mode." In enable mode, you can configure the router in any way desired, reboot the router, or take any other action possible in the software.

To enter privileged EXEC mode, use the `enable` command. When you actually have a password set to access enable mode, the router will prompt for it when you try to enable. See "Passwords" on [Passwords](#).

```
router>enable
router#
```

In enable mode, the command prompt ends in a pound sign.

The old method of Cisco access control was to give the standard EXEC password to all the technicians but restrict the enable password to the senior techs. Setting up discrete usernames is a better way to achieve a similar effect, but many older routers still only use unprivileged and enable passwords—either because they haven't been updated or because that's all that they can support. There's nothing

wrong with restricting the use of privileged mode in this way, especially in a small shop where only one or two people connect to the equipment, but if you have more people, you'll probably want to set up separate usernames and allow individuals to be privileged or not. We'll see how to do this in [Chapter 9](#).

Integrated Help

"Help" might be too strong a word, but routers offer hints on commands, command syntax, and the features available in the router at that privilege level. Different versions of the Cisco IOS have different features, and it's not easy to identify them all. The simplest way to see what commands are available is to ask the router. Once you've logged in, enter a single question mark at the command prompt.

```
router# ?
Exec commands:
 <1-99>          Session number to resume
 clear          Reset functions
 clock         Manage the system clock
 ...
```

My test router has four full screens of commands available from a single question mark. Many of these commands are completely irrelevant for a router that is providing basic Internet capability; such activities require only a very small subset of Cisco's features. Still, this can be helpful when you can't quite remember the name of a particular command.

You can request hints about individual commands. Many actions require one or more words to complete, and Cisco's integrated help system will provide helpful reminders about syntax or required information upon request. For example, the `show` command is used to view system information. If you want to know what arguments a `show` command can take, ask the router by entering `show ?`.

```
router# show ?
 access-expression      List access expression
 access-lists          List access lists
 accounting             Accounting data for active sessions
 adjacency             Adjacent nodes
 ...
```

On some versions of the IOS, this output can go on for screens and screens. If you know the first part of a command's name, even just the first letter, you can narrow it down by giving what you remember of the command and a question mark.

```
router# show a?
access-expression  access-lists  accounting  adjacency  aliases
alps              arp           async
router# ? show a
```

Note that on your next command-line prompt, the router automatically prints the chunk of command that you gave before the question mark **?**. The system knows that you want to type something that begins with `show a` and is trying to make it easier for you.

One thing to remember is that hints on a particular command are only available in the mode in which the command is available. For example, the commands beginning with `ip`, such as `ip route`, are only available in enable mode. Entering `ip ?` in standard EXEC mode will generate an "unrecognized command" error.

Command Abbreviation

One interesting feature of the Cisco IOS is that it allows the user to abbreviate command names to the shortest unique abbreviation for a word. For example, one common command we'll use again and again to view settings is `show`. If you ask your router for all the commands beginning with the letter `S`, you'll get a whole list.

```
router# s?  
❶ *s=show  sdlc  send      set      setup  
❷ show     slip  start-chat  systat  
router#
```

Note that only the command `show` begins with the letters `sh` ❷. You can use the letters `sh` as an abbreviation for the full word `show`; the router is smart enough to know that you couldn't possibly be typing any other legitimate command that begins with those letters. In this particular case, the letter `s` is also specially marked to indicate that it is an abbreviation for `show` ❶; apparently, Cisco thought that `show` was so commonly used that it made sense to abbreviate it further.

In this book, we will give commands in the text by their full forms. Example commands might be in abbreviated form if they are commonly used that way.

Chapter 2. Router Configuration



Now that you're logged in and have a basic understanding of how to enter commands, let's set up the router. The `show version` command explains some basic facts about your router, such as the software version, hardware type, and supported interfaces. Because the output is quite long, I won't include it all here, but we'll look at a few important snippets.

```
router# show version
Cisco Internetwork Operating System Software
IOS (tm) ❶ 3600 Software (C3640-IS56I-M), ❷ Version 12.0(7)XK1, EARLY
DEPLOYMENT RELEASE SOFTWARE (fc1)
...
```

The second line gives you the hardware platform this IOS is meant for ❶ and its version number ❷. If you contact Cisco for support, you must provide the version number. Next are details about who built this software and when, the ROM version, and so on. While Cisco would certainly want to see all of this information if you called for support, you can't do much with it by yourself.

```
...
router

❸ uptime is 4 weeks, 4 days, 9 hours, 53 minutes
System returned to ROM by ❹ reload
System image file is ❺ "flash:c3640-is56i-mz-120-7-XK1"
cisco ❻ 3640 (R4700) processor (revision 0x00) with 123904K/7168K bytes of
memory.
...
❼ 4 FastEthernet/IEEE 802.3 interface(s)
6 Serial network interface(s)
...
❽ Configuration register is 0x2102
```

The uptime ❸ tells how long the router has been running (4 weeks, 4 days, 9 hours, and 53 minutes), and the next line explains why it last went down ❹. (A *reload* is a software-driven reboot, as we'll discuss in [Chapter 5](#).) When a coworker asks, "Are we having Internet trouble?" it's nice to be able to say that the router has been working all along.

The system image file ❺ (in this listing, "flash:c3640-is56i-mz-120-7-XK1") is the location and name of the file that the router loaded as its operating system; this information will be important when the time comes to upgrade your IOS. The model information and physical characteristics of the router (shown below the system image file) ❻ can be important. If your router is in a remote office, looking at this information may be the easiest way to learn what sort of equipment it is. (It would have been even easier for you to document the router before sending it to the remote location, but—despite our best efforts—generally, it's standard practice to just ship it and forget it.)

After a few more lines that detail software features, we'll see the interface types installed in this router.

6. The router will then list some model descriptions for components and end with the configuration register 0 (which probably means nothing to you right now but will be important during IOS upgrades).

Running versus Startup

A router has two configurations at any given time: the *startup configuration* and the *running configuration*. The startup configuration is stored in the router's nonvolatile memory. When the router boots, it loads the startup configuration as the current configuration. At that point, a copy of the startup configuration becomes the running configuration. If you change the router's configuration while it's running, you're changing the running configuration. When you save the running configuration, it overwrites the previous startup configuration and becomes the new startup configuration. If you do not save your changes before you reboot the router, the changes are lost.

Note

This is a long-winded way of saying: save your changes or you'll lose them at the next reboot! Changing a router configuration is no different than editing a document in a word processor; if you exit without saving, you lose your work. This can be good or bad; don't be afraid to use this to your advantage if you really screw up your router's configuration!

Technically, you could say that the startup configuration is a configuration file; it's stored in nonvolatile RAM, just like a file on disk. Cisco experts don't generally refer to the startup configuration as a configuration file, however; it's just a "configuration." The running configuration is certainly not a file; it's a (possibly modified) copy of the startup configuration held in memory.

Startup Configuration

To view the configuration the router will boot with, enter the `show startup-config` command.

```
router# show startup-config
  1 Using 5620 out of 129016 bytes
  !
  2 version 12.0
  ...
```

The router reads the configuration from permanent storage and tells you how much space it takes up before displaying the configuration properly.

Running Configuration

To view the current (running) configuration, enter the command `show running-config`. Old Cisco geezers might remember this as `write terminal` or `wr t`. There's nothing inherently evil about `wr t`, but it's considered obsolete.

```
router# show running-config
  1 Building configuration...
  2 Current configuration:
  !
```

version 12.0

...

Note the first line of the output, where the router tells you that it is literally assembling the current configuration from the router's memory ❶. It then reminds you that this is the configuration in memory at the moment ❷, which may or may not match what is in NVRAM. It then displays the configuration one screen at a time.

Reading a Configuration

We won't dissect a single complete router configuration in this book because a working configuration for even a small router can easily reach up to 200 lines. Here's how to read this configuration, though.

A Cisco IOS configuration consists of a series of statements. Each statement either activates or deactivates a feature for an interface, a protocol, or the router as a whole, or defines some global characteristic of the router. You'll also see exclamation points, which IOS uses to separate sections of a configuration. (If you store copies of your router configurations in a place other than on the router, you can use exclamation points to indicate comments, much as many program configuration files use the pound sign. The router will strip out these comments when loading the configuration, so very few people bother with them.) For example, here are some snippets of a small router configuration file.

```
...
❶ service timestamps debug uptime
❷ !
❸ hostname router.blackhelicopters.org
...
```

This router supports the network services `timestamps`, `debug`, and `uptime`. ❶ Their presence in the configuration file is enough to enable them.

We also have a global configuration variable, the router's hostname. ❷ If the router thinks that configuration sections are sufficiently different, it puts a blank line between them. ❸

```
...
❹ ip subnet-zero
❺ ip domain-name blackhelicopters.org
❻ no ip finger
...
```

This router has two settings for Internet Protocol (IP): the `subnet-zero` feature is enabled ❹, and the IP domain name is set. ❺ Conversely, the "no" setting disables a particular service; in this case, the `finger` service is turned off. ❻

```
...
❶ interface Serial1/1
❷ description main office T1
❸ no ip redirects
...
```

Here we have a configuration for a particular router interface. ❶ Configuration information for this interface appears directly beneath it and has a single space in front of it. ❷ Note that we specifically turned off a particular feature here, `ip redirects`. ❸ We could enable this feature on the router as a whole and then disable it on an interface-by-interface basis.

```
...
❹ router bgp 8292
❺ bgp dampening
...
```

Similarly, major routing protocols configured on the router each have their own section ❶. Configuration for a protocol appears directly beneath it, set off by a leading space ❷.

Configuring the Router

When you're in privileged EXEC mode, you can not only issue more powerful commands but also change the router's configuration.

To change the router's configuration, you must enter *configure mode*. The most common way to configure the router is at the command line you're logged in at, also known as the *terminal*. Enter `configure terminal` at the privileged mode prompt.

```
router# conf t
Enter configuration commands, one per line. End with CNTL/Z.
❶ router(config)#
```

Note that the command prompt has changed from simply `router` to `router(config) ❶`. The router is making it explicit that you are configuring the system. At the configure prompt, you can enter configuration statements, one statement per line.

Configuration commands look exactly like those in the system configuration and are added directly to the router's configuration. For example, to add the line `service password-encryption` to your configuration, just enter it at the configure prompt.

```
router(config)# service password-encryption
```

The router will place this in an appropriate spot in the global configuration. (No, you can't put the statement in a place of your own choosing in the configuration; the router knows far better than you do where each line belongs and will blatantly ignore any attempts to reorder the configuration.) Entering a command like `reload` or `ping` at the configure prompt will only generate an error because these are not legitimate configuration statements.

When you have completed your configuration, leave configuration mode with `ctrl-Z`.

```
router(config)# ^z
router#
```

The prompt changes back to simply `router`.

Configuring a Particular Interface

When you need to configure a particular interface, just enter the interface name at the configure prompt. The router will place any further statements under the interface configuration.

```
router(config)# int s1/0
❶ router(config-if)#
```

Note that the prompt changes to `router(config-if) ❶` to remind you that you're configuring an interface, not the entire router.

Why is this important? Do you remember our example configuration that had `no ip redirects` on a single interface? Presumably, that router needed the feature enabled on some other interface or on a global level. Enabling and disabling features and protocols on a per-interface basis gives you fine-grained control of your router.

Routing protocols have a similar configuration subprompt.

```
router(config)# router bgp 8292
router(config-route)#
```

Any further statements will appear in the `router bgp 8292` section, as shown in our example earlier.

If you want to stop configuring a particular interface or protocol but remain in configure mode, use `exit`.

Saving Changes

Entering all these configuration statements will alter the running configuration, but that configuration won't persist through a reboot unless you save the configuration.

```
router# copy running-config startup-config
```

Again, old Cisco hands might remember this as `write memory` or `wr mem`. Cisco hasn't taken that away from you either.

Backing Up Router Configurations

Router configurations are just plain text. The simplest way to back up your router's configuration is to copy the configuration statements to another system, perhaps a plain text file on a server or even to a piece of paper in a logbook. Should your router suffer a critical failure and lose its mind, you can restore service by just going into configure mode and pasting in the entire router configuration. If you have Unix-like systems available, programs such as RANCID automate configuration backup.

It is also possible to copy your router's configuration to an FTP server with the `copy ftp` command. The process is quite similar to the FTP process used for IOS upgrades, and it is an *excellent* precaution to take before an upgrade, so we'll discuss it in [Chapter 6](#).

Chapter 3. Router Interfaces



One of the main functions of a Cisco router is to connect different types of networks. Connecting Ethernet systems is quite straightforward, and connecting WAN links is just as easy, but these two very common network types simply refuse to talk to each other without an intermediary. The router allows you to treat very different physical networks as a single, continuous entity.

Cisco routers support almost any type of network interface: Ethernet, serial, token ring, DS3, OC3, asynchronous modem, and so on. These interfaces might be on add-on cards that slide into a system, much like those used in a laptop, or they might be integrated with the system. And when some bright scientist develops direct neural links into the human mind, Cisco will have an interface for that, too. But until then, the most common interface types are Ethernet and serial, so we'll focus on them.

Got Interface?

By looking at the system interfaces, you can see not only what sorts of interfaces the router has, but also how much traffic each interface is handling, what sort of network errors the router senses, and a whole slew of further detail about the networks the router is attached to. To see every interface on your router, type `show interfaces`. Every interface on the router will show up with an entry starting like this.

```
router# sho int
FastEthernet1/0 is administratively down, line protocol is down
  Hardware is AmdFE, address is 0003.e35e.d191 (bia 0003.e35e.d191)
...
```

Routers list their interfaces by their internal order in the system. Some routers clearly print this order on the chassis, while you might wonder where other routers learned to count.

Each interface is uniquely named by a type (`FastEthernet` in this example) and a unique number for that type (`1/0`).

The first interface of any type is numbered `0`. A split number like the one in this example tells us that multiple interfaces are loaded into a single module, or *card*. For example, the interface `FastEthernet 1/0` is the first Ethernet interface on card 1. Numbering depends on how the router thinks the interfaces are attached.

On a router with many interfaces, you might only want to see a particular interface. For example, if you want to see if your Internet circuit is working, you only need to look at the interface that's connected to that circuit, not every interface on the router. To display a particular interface, add its name to the `show int` command. For example, to see only the interface `serial 0`, enter `show int serial0`.

As with commands, you can abbreviate interface names to the shortest unique identifier: these

abbreviations usually contain just enough letters to uniquely identify the interface type and number. For example, ~~serial 1~~ can be ~~s1~~, ~~ethernet 0~~ can be ~~e0~~, and ~~fast ethernet 2/1~~ can be ~~faste2/1~~. Have a look at your router's interface names to see how they can be abbreviated.

Common Interface Characteristics

When you run `sho int` on most types of interfaces, including serial and Ethernet, you will see a great deal of similar information in the resulting output for each. The example below shows the first part of `sho int` output for an Ethernet interface, but everything we'll discuss relates to serial interfaces as well.

```
router# sho int fastethernet2/0
1 FastEthernet2/0 is up, 2 line protocol is up
3   Hardware is AmdFE, address is 0003.e35e.d1a1 (bia 0003.e35e.d1a1)
4   Description: Main office Ethernet hub
5   Internet address is 198.88.118.129/25
6   MTU 1500 bytes, 7 BW 10000 Kbit, DLY 1000 usec,
   reliability 255/255, txload 1/255, rxload 13/255
8   Encapsulation ARPA, loopback not set
...

```

The output for each interface continues, but it's mostly debugging information and not useful at the moment. We'll look at that additional output in [Chapter 5](#).

Looking at the output above, we first see that this interface is **up** 1. This means that a physical cable plugged into the interface and that the router sees a comprehensible signal from it. If the interface is unused or empty, or if there is some physical problem with the circuit, the status would be **down**. If your interface is up, chances are good that there is no physical problem with this network. If it is down, the problem can quite possibly be identified by walking along the wire looking for problems. In the case of serial lines, however, that wire might be very long ...

Next is the line protocol status 2. The line protocol tells us if the router understands the encoding used by the signal coming over the line. Every network type uses some sort of physical protocol that encodes the actual data into a string of ones and zeros. As long as the device on the other end of the wire uses the same protocols as the router interface, this line should read **up**. However, even if you have a working circuit plugged into your router interface, if the routers on either end are using different line protocols, the status here would read **down**. A usable circuit must have both an "up" circuit and protocol agreement. If your circuit is up, but your protocol is down, your configuration is probably wrong at one end or the other. The line protocol is set in the "encapsulation" section (described in point 8).

The interface reports the sort of hardware this router is using for this connection 3. This name is frequently some obscure Cisco internal part number or name, but it might include a description of the part. Unfortunately, there is no publicly accessible master list of these parts, but if you are familiar with the technology, you might be able to glean some useful information from the hardware description. In this example, we have the Fast Ethernet port's MAC address.

You can enter whatever you like in the **Description** field 4. While this might seem pointless for a usual SOHO network with just one interface, it can be very helpful to enter a plain-English descriptive name here when your router has multiple interfaces of each type. On serial circuits such as T1s or DS3s, I recommend putting the telco circuit ID in this field. (We'll discuss circuit IDs in [Chapter 4](#).)

Every configured interface has some basic TCP/IP configuration information 5, such as an IP address and netmask. (We'll see how to configure this later this chapter.)

Next, we have some basic information about the physical protocol spoken on this interface 6. The

standard MTU (Maximum Transmission Unit) for most devices on the Internet is 1,500 bytes, as shown here; if yours is different, someone set it that way for some particular reason. Changing this value will increase the fragmentation of packets that pass through your network and is generally inadvisable. If your MTU isn't 1,500 bytes on an Ethernet or T1 circuit, that's almost certainly your problem. Big circuits such as DS3s and OC178s have their own proper MTU values, and mucking with them will cause all manner of difficulties.

The BW value is the total bandwidth of this interface ⑦. This can be vital information if you're trying to determine why your router seems to be slow. If your router can handle 10,000 kilobits per second (like this interface), and you try to cram twice that amount through your network, you're going to have serious problems.

Finally, the encapsulation is the logical protocol used for this interface ⑧. In the most basic sense, this tells the router what sort of network you're attached to. In this example, we're using ARPA encapsulation, which is used for all Ethernet interfaces. (Serial links have more options, as we'll discuss in [Chapter 4](#).) This is where the "line protocol" discussed earlier is set. If your line is up, but your line protocol is down, this is probably the setting you need to change. See [Chapter 4](#) for the common encapsulation types.

Configuring Interfaces

You enter configuration information for a single interface in configure mode (as discussed in [Chapter 2](#)), but to do so, you must specify the interface that configuration applies to. After you're in configuration mode, enter the interface name.

```
router# conf t
router(config)# int faste2/0
router(●config-if)# no shut
```

The `config-if` (Cisco-ese for "configure interface") label indicates that configuration changes you make will only apply to the single interface you specified ●.

Every interface is shut off by default. To activate an interface, use the `no shut` configuration option. If your interface doesn't work, and you think everything is configured correctly, you probably forgot to turn it on. To disable an interface, enter `shut`.

To leave configuration mode, enter CTRL-Z.

- [download Chinese Sun](#)
- [The Far Pavilions pdf, azw \(kindle\), epub](#)
- [download online Saviors and Survivors: Darfur, Politics, and the War on Terror for free](#)
- [read Microbiology PreTest Self-Assessment and Review \(14th Edition\)](#)

- <http://dadhoc.com/lib/Sex--Power--and-Slavery.pdf>
- <http://econtact.webschaefer.com/?books/New-Media-New-Methods--The-Academic-Turn-from-Literacy-to-Electracy--New-Media-Theory-.pdf>
- <http://damianfoster.com/books/Black-Hawk-Down--A-Story-of-Modern-War.pdf>
- <http://ramazotti.ru/library/Chasing-the-Phoenix--Darger-and-Surplus--Book-2-.pdf>