Chapter 3

Configuring the TALnet Software

Each Wireless Router comes with a preinstalled configuration file, called *talnet.cfg*. This configuration file includes required and recommended commands. However, you must modify these commands so that they are appropriate for your network. Depending on your level of user authorization, you can modify these commands in the following ways:

- Make permanent changes by editing the configuration file. (If user privileges have been set, you must have Admin privileges.)
- Make temporary changes that last until the router is restarted. (If user privileges have been set, you must have Operator privileges.)

Note In general, you should always modify the configuration by editing the configuration file.

This chapter describes how to modify the configuration file. The arguments for most commands in this chapter are based on the information in the configuration worksheet in Chapter 2. For more information about the commands, refer to Appendix A, "TALnet Command Reference."

3.1 Sample Configuration File

The following is a configuration file similar to the file that ships with your Wireless Router. A pound sign (#) indicates a comment line. Some commands have been commented out because they are not required, but provide useful examples.

```
#
#
# TALnet Configuration File
# Copyright 1995-1996, Tetherless Access Limited -- All rights reserved.
#

#
# Hostname (Comprised of letters upper or lower case, digits 0-9 and
# hyphen -)
hostname TAL
```

```
#
# SNMP Configuration

# Community String
snmp community add public read-only

# System Contact
snmp syscontact John Doe

# System Location
snmp syslocation TAL

#
# Set up TALtalk wireless subsystem device and interface
# For one radio, use these:

talk radio radio1 address 0x7D10FFFF model S config scc
talk radio radio1 channel one config number 5 MW 1 PN 5
talk radio radio1 ip-address 192.168.0.1

# For 2 radios, comment out the above three lines
# Then the following commands should be uncommented:

# talk radio radio1 address 0x0000ABCD model S config scc 0x380 5 1 1
# talk radio radio1 channel one config number 5 MW 1 PN 5
# talk radio radio1 ip-address 192.168.0.1
# talk radio radio2 address 0xABCD0000 model L config scc 0x3E8 7 3 3
# talk radio radio2 channel one config number 5 MW 1 PN 5
# talk radio radio2 ip-address 192.168.0.2

# Link TALtalk to RIP with this command
# talk radio radio1 auto-neighbor on

#
# Authentication examples

#
# Table authentication: explicit allow would deny all others
#
# talk radio radio1 neighbor allow 0x0000ABCD

#
# Key authentication: key creation turns this on
#
# key create 023-59ST

#
# Set up other devices

#      Watchdog timer:
device make watchdog wdt0
device config wdt0 enable
```

```
#      Ethernet over packet driver:
device make packet pkt0 0x65
device config pkt0 up enable

#      Serial port:
#
# device make com com2 0x2F8 3
# device config com2 cts rtson dtron rxfifo 1024 modem speed 9600 up enable

#
# Set up wired interfaces

#     Ethernet over packet driver:
iface make ether0 device pkt0 address 192.168.0.1 encapsulation ether
iface config ether0 netmask 255.255.255.0 mxu 1500 broadcast 192.168.0.255
  up

#     Serial port:
#
# iface make ppp0 address 192.168.1.1 mxu 1500 encapsulation ppp

#
# Optimize TCP connections to/from this router
tcp mss 1460
tcp window 2920

#
# IP Packet Filter Examples
#
# ip filter ether0 deny in tcp 192.168.180.0/24 192.168.181.0/24
# ip filter ether0 permit in * 192.168.180.0/24 192.168.181.0/24
# ip filter radio1 permit in icmp 192.168.180.0/24 !192.168.181.0/24
# ip filter radio1 deny in tcpsyn * 192.168.180.0/24:23

#
# Set up Domain Name Service (DNS) information

# Nameserver
#
# domain addserver 198.41.0.4

# DNS suffix MUST end in dot
#
# domain suffix yourdomain.com.

#
# Static routing example

# Set up a default route on Ethernet if RIP router not available
route addprivate default ether0 192.168.0.2

# Setup initial routes for remote wireless router(s) that are neighbors
#
# route add 192.168.2.1 radio1
```

```
#
# Dynamic routing example (RIP-1 & RIP-2)

# Publish to remote wireless router(s)
#
# rip add 192.168.2.1 rip2

# Publish on the local ethernet
#
# rip add 192.168.0.255 rip2

#
# Add users

# Authorization flags:
#
# keyword  Description
# =======  ===========
# read     Read files
# create   Create files
# write    Overwrite & delete files
# operator System operator - can reboot and reconfigure
# ppp      Can set up PPP
# console  Can login over the console
# Netlogin Can login over the network (ftp & telnet)
# Admin    System Admin - execute all commands
#                         Also sets operator, read, create, and write
#
# user create operator authorization Netlogin console operator password
# faux  root /
# user create Admin authorization Admin Netlogin console password
# super root /
# user create pppuser authorization ppp iface ppp0 password pppassword

#
# Start servers

start snmp
start ftp
start telnet
start echo
start discard
#
# start rip
```

3.2 Configuration Overview

The following are the basic steps for configuring your TALnet software. Some of the tasks are required; others are recommended to fine-tune your network. Each task is described in greater detail later in this chapter.

1. Log on to the system—Required when user authorization is set.
2. Open the configuration file—Required.
3. Specify global network parameters—Required.
4. Configure radio parameters—Required.
5. Configure Ethernet connections—Required for connections to an Ethernet LAN.
6. Locate the Domain Name System (DNS) server—Recommended.
7. Set up routing—Required.
8. Define authorized users—Recommended.
9. Set up Internet Protocol (IP) packet filters—Optional.
10. Modify the Address Resolution Protocol (ARP) cache—Optional.
11. Configure PPP connections—Optional.
12. Load and execute the new configuration file, and log out of the system—Required.

For information on monitoring your network, refer to Chapter 4, “Monitoring and Maintaining Your Network.”

Note This chapter does not describe every possible configuration command, nor does it describe all variations of each configuration command. For more-detailed information about TALnet commands, refer to Appendix A, “TALnet Command Reference.”

3.3 Logging On to the System When User Authorization is Set

The very first time you log on to the Wireless Router, no authorized users have been defined so you are not prompted for a user name or password. Once authorized users are defined, you must log on to the Wireless Router as described in the following procedure. Follow these steps to log on to the Wireless Router through the service console:

1. At the `user`: prompt, enter your username as it is defined in the *talnet.cfg* configuration file, then press Return.
2. At the `password:` prompt, enter the password as it is defined in the configuration file and press Return.

The TALnet prompt (`TAL>`) appears and your screen looks similar to the following example:

```
TALnet Release 2.0
Copyright 1992-1996 Tetherless Access Limited, All Rights Reserved.
Compiled at TAL: Jun 26 1996, 15:24:55
Router Platform: SubSpace 2001 (rev1)
Installed Features: Encryption, TALK, PPP, IP Routing.

Reading default config file d:\tal/talnet.cfg
wd0 is enabled.
Talk auto-neighbor mode: on
RIP started with interval 30000


Console login required.

user: console
password:
TALnet Release 2.0
Copyright 1992-1996 Tetherless Access Limited, All Rights Reserved.
Compiled at TAL: Jun 26 1996, 15:24:55
Router Platform: SubSpace 2001 (rev1)
Installed Features: Encryption, TALK, PPP, IP Routing.

TAL>
```

3.4 Opening the Configuration File

The first time you modify the configuration file, you must do so through the service console. You can make subsequent revisions either locally through the service console or remotely by using the File Transfer Protocol (FTP). If you have defined any users with the **user** command, you should assign at least one of them Admin and console privileges. Only users who have Admin and console privileges can modify the configuration file directly from the service console.

Caution Make sure you create a backup of the configuration file before you make changes. If you maintain several different routers, you might want to keep a directory on your computer with copies of each configuration file.

After you open the configuration file using one of the methods described in this section, you are ready to modify the configuration file. The rest of this chapter describes basic configuration tasks. The last section in this chapter describes how to load the new configuration file onto the router, execute the changes to that file, and log out of the Wireless Router.

3.4.1 Opening the File from the Console

If you modify the configuration file from the console, you make the changes directly on the router. Follow these steps to open and edit the configuration file directly from the service console:

1 Ensure that you are in the directory D:\TAL:

```
TAL> cd D:\TAL
```

2 Set up the TALnet software so that you can stop routing:

```
TAL> copy start.net stop.net
```

3 Exit the TALnet software:

```
TAL> exit
Are you sure? (y/n)? y
```

4 Change to the directory that contains the configuration file:

```
c:\> cd \tal
```

5 Make a backup of the *talnet.cfg* file:

```
c:\tal> copy talnet.cfg *.001
```

6 Open the configuration file using the **Xvi** editor supplied with your router:

```
c:\tal> xvi talnet.cfg
```

Note **Xvi** commands are similar to UNIX **vi** commands. Appendix C, “Using Xvi,” provides a brief description of the **Xvi** editor.

During software configuration, you might issue a command that creates a new session. Table 3-1 describes basic key sequences you can use on the service console during these sessions.

Table 3-1 Service Console Key Sequences

Function	Key Sequence
Refresh the screen.	**Ctrl-l**
Switch between active sessions.	**Ctrl-w**
Return to the main session but leave the current session running.	**Ctrl-z**
Close the current session and return to the main session.	**Ctrl-c**

To view currently active sessions, issue the **session** command without an argument. To switch to an existing session, issue the **session** *number* command or enter **Ctrl-w**.

3.4.2 Opening the File from a Remote System

After you have modified the configuration file the first time, you can make subsequent revisions remotely if you have been assigned Netlogin privileges using the **user** command. To open the file remotely, copy the configuration file to your personal computer or Administrative host using FTP. Follow these steps to copy, open, and edit the configuration file:

1 Establish an FTP connection to the router:

```
> ftp router
```

The *router* argument is either the DNS name or the IP address of any valid interface connected to the router.

2 When prompted, enter your username and password as defined in the configuration file.

3 Select ASCII transfer mode:

```
ftp> ascii
```

4 Turn hash mark printing on; hash marks appear on the screen to indicate progress during a file transfer:

```
ftp> hash
```

5 Change to the directory that contains the configuration file:

```
ftp> cd c:\tal
```

6 Copy the configuration file to your system:

```
ftp> get talnet.cfg
```

7 Open the configuration file with the text editor of your choice.

Note You might need to close your FTP session before editing the configuration file. To close the session, use the **quit** command.

During software configuration, you might issue a command that creates a new session. Table 3-2 describes basic key sequences you can use when you connect to the router using Telnet.

Table 3-2 Telnet Connection Key Sequences

Function	Key Sequence
Return to the main session but leave the current session running.	**Ctrl-z**
Close the current session and return to the main session.	**Ctrl-c**

To view currently active sessions, issue the **session** command. To switch to an existing session, issue the **session** *number* command.

3.5 Specifying Global Parameters

Specify the following global parameters:

- Hostname
- System contact (for SNMP queries)
- System location (for SNMP queries)

3.5.1 Defining a Hostname

The hostname is the name of your Wireless Router. The default hostname is TAL.The name you specify is used in system prompts and should be the same as the DNS name for your router.

To set the hostname, use the **hostname** command:

hostname *name*

Where:

- *name*—Is the hostname, and can contain uppercase and lowercase letters, numbers, and hyphens, but cannot contain spaces. The hostname is case sensitive and should not begin with a number.

3.5.2 Identifying the System Contact

In order for the Simple Network Management Protocol (SNMP) agent to return system contact information on queries from management stations, you must identify the person responsible for the router. To do this, use the **snmp syscontact** command:

snmp syscontact *name*

Where:

- *name*—Sets the value of the SNMP variable *system.sysContact.0* to the specified string. You can use uppercase and lowercase letters, numbers, and spaces.

To find out the currently defined system contact, issue the **snmp syscontact** command without an argument at a service console or through a remote Telnet connection.

3.5.3 Identifying the System Location

In order for the SNMP agent to return system location information to queries from management stations, you must identify the physical location of the router. To do this, use the **snmp syslocation** command:

snmp syslocation *description*

Where:

- *description*—Sets the value of the SNMP variable *system.sysLocation.0* to the specified string. You can use uppercase and lowercase letters, numbers, and spaces.

To find out the currently defined system location, issue the **snmp syslocation** command without an argument at a service console or through a remote Telnet connection.

3.6 Configuring Radio Parameters

The router subsystem communicates with the wireless subsystem through a synchronous serial interface called the wireless interface. You must create and configure the wireless interface of each radio you connect to the router subsystem using the TALtalk link-layer protocol. After you create and configure the interface, you must configure the channel on which each radio operates. Each channel operates on a specific frequency and power level, and uses a specific pseudorandom noise (PN) code. Follow these steps for each radio connected to the router subsystem:

1. Configure the wireless interface.
2. Configure the radio channel.
3. Set the IP address for the wireless interface.
4. As needed, modify the power level on a per-link basis.

The following subsections describe these tasks in greater detail. The last subsection includes examples of these tasks. Issue the **talk show radio** command at a service console or through a remote Telnet connection to determine currently configured values.

3.6.1 Configuring the Wireless Interface

The wireless interface connects to your router through a synchronous serial device called the synchronous serial communications controller (SCC). To create and configure the wireless interface, you must assign a symbolic name to the interface, associate the name with the TALtalk link-layer address for the device, and initialize the device.

To create and configure the wireless interface for a radio, use the **talk radio address** command:

talk radio *iface-name* **address** *talk-address* [**model** {**L** | **S**}] **config scc** [*I/O-base IRQ receive-DMA transmit-DMA*]

Where:

- *iface-name*—Assigns a symbolic name to the interface. This interface name is used in other **talk** commands and in **talk show** output. The name can include uppercase and lowercase letters, numbers, and hyphens, and should indicate the type of interface and unit number; for example, *radio1* or *radio2*.
- **address** *talk-address*—Associates the symbolic interface name with the synchronous serial device. The address is the TALtalk link-layer address, which is usually assigned by TAL or your service provider. The address is an 8-digit hexadecimal number preceded by the characters *0x*. For example, *0x41544D31* and *0x00000001* are valid TALtalk addresses.
- **model** {**L** | **S**}—Specifies the band in which the radio operates. L-band radios operate in the 902- to 928-MHz frequency range. S-band radios operate in the 2400- to 2483.5-MHz frequency range. If you do not specify the **model** keyword, the router assumes that you are using the L-band.
- **config scc**—Initializes the SCC device used by the router to communicate with the wireless interface. You only need to enter the arguments *I/O-base*, *IRQ*, *receive-DMA*, and *transmit-DMA* if you are configuring two wireless interfaces. The values of these arguments are predefined for you in the *talnet.cfg* configuration file and should not be changed.

3.6.2 Configuring the Channel

After you have created and configured the wireless interface, you must configure the following parameters on the radio channel. Each Wireless Router currently can support two wireless interfaces; if you are attaching two radios to your router, you must configure one channel for each radio. Refer to Chapter 2 for more information about transmission frequencies, power levels, and PN codes.

- Transmission frequency—Select the frequency on which you are going to transmit data.
- Transmission power level—Select the strength of the radio transmission.
- PN code—Select one of eight PN codes, also called spreading codes, to encode network data for transmission.

To configure a radio channel, use the **talk radio channel** command:

talk radio *iface-name* **channel** *channel-name* **config number** *channel-num* **MW** *power* **PN** *code*

Where:

- *iface-name*—Represents the symbolic name of the wireless interface you are configuring. This name is the one you assigned with the **talk radio address** command.
- **channel** *channel-name*—Assigns a symbolic name to the channel.
- **number** *channel-num*—Configures the channel and frequency on which the radio will operate. The *channel-num* value is an integer between 1 and 9 on the L-band radio and 1 and 15 on the S-band radio. It must be the same for all radios within your wireless network. See Chapter 2 for available channel numbers and frequencies.

Caution Channels on L-band radios overlap; do not select channel numbers that are immediately adjacent to each other. The link analyses and performance tests you carried out during hardware installation should help you determine the appropriate channels.

- **MW** *power*—Configures the power level at which the radio will transmit. The *power* value should be the lowest possible setting that provides reliable data transmission, and should be determined during your link analysis. Specify the power in milliwatts (mW). See Chapter 2 for available power settings.
- **PN** *code*—Specifies one of eight PN codes the radio will use to encode data for network transmission. The *code* value can be an integer between 1 and 8; all radios in your network must use the same code.

3.6.3 Defining the IP Address

You must define an IP address for each wireless interface to which you attach a radio.

Note If you are configuring two radios, you must define the IP address of the first radio before beginning configuration of the second radio with the **talk radio address** command.

To define the IP address, use the **talk radio** command:

talk radio *iface-name* **ip-address** *ip-addr*

Where:

- *iface-name*—Represents the symbolic name of the wireless interface you are configuring. This name is the one you assigned with the **talk radio address** command.
- **ip-address** *ip-addr*—Binds the IP address to the wireless interface. The IP address is expressed in four-part dotted decimal format, for example, 192.168.185.79.

3.6.4 Modifying the Power on a Per-Link Basis

The previous section described how to configure the transmission frequency, power level, and PN code for a wireless interface. Any packet that goes out the specified wireless interface uses these settings. The transmission frequency and PN code must be the same for all links that must communicate in a network. However, you can adjust the power level on a per-link basis. For example, if you perform a link analysis and discover that one Wireless Router in the network cannot hear another Wireless Router, but all other links in the network are fine, you can increase the power level between only those two Wireless Routers.

To increase the power level on a per-link basis, use the **talk radio neighbor MW** command:

talk radio *iface-name* **neighbor** *talk-address* **MW** *power*

Where:

- *iface-name*—Represents the symbolic name of the wireless interface you are configuring. This name is the one you assigned with the **talk radio address** command.
- **neighbor** *talk-address*—Identifies the link on which you are changing the power. The *talk-address* argument is the TALtalk link-layer address of the neighboring transmitter.
- **MW** *power*—Configures the power level at which the radio will transmit. Specify the power in milliwatts (mW). See Chapter 2 for available power settings.

3.6.5 Radio Configuration Examples

The following example configures the wireless interface `radio1`:

```
talk radio radio1 address 0x7D10FFFF model S config scc
talk radio radio1 channel one config number 5 MW 1 PN 5
talk radio radio1 ip-address 192.168.0.1
talk radio radio1 neighbor 0x11223344 MW 200
```

The first line assigns the symbolic name `radio1` to the interface, specifies that the address of that interface is `7D10FFFF`, that the radio is an S-band model, and enables the SCC device.

The second line specifies that the channel for interface `radio1` is called `one`, the channel number is `5` (using a transmission frequency of 2427.863 MHz), the default power level is `1` mW, and the PN code is `5`. All other radios on the network should use channel 5 and a PN code of 5.

The third line defines the IP address `192.168.0.1` for the wireless interface `radio1`.

The fourth line changes the power to the neighbor with the address `11223344` to `200` mW.

If you have a second radio attached to your Wireless Router, configure it in a similar manner.

3.7 Configuring Your Ethernet Connection

If you are going to connect your Wireless Router to an Ethernet LAN, you must configure the Wireless Router to accept Ethernet connections. This consists of configuring the device and the interface.

To configure the device and interface, you must perform the following general steps:

1. Identify and configure the hardware device type.
2. Associate an interface to a specific device and configure the interface.

The following subsections describe these tasks in greater detail. The last subsection includes an example of creating devices and interfaces.

3.7.1 Configuring the Device

Devices refer to the interface at the hardware or link-layer level. Interfaces refer to the interface at the network protocol level. The packet device provides a connection between the TALnet protocol stack and a hardware device. This release of the TALnet software only supports a Class 1 Ethernet device.

To create and configure a device, follow these steps:

1. Create a device by identifying the hardware interface and how that interface is going to be used. Assign a symbolic name to the device; this name is used in other commands.

   device make packet *device-name software-interrupt-number*

   The parameters you specify vary depending on the type of device you create. Refer to the **device** command description in Appendix A, “TALnet Command Reference.”

2. Configure whether and how the device handles different kinds of information. For example, you can set the speed at which the device should run, how the device handles interrupts, or set up tracing on the device. Use the **device config** command:

   device config *device-name* **up enable**

To remove an existing device, use the **device kill** command. To display information about a device, use the **device show** command at a service console or through a remote Telnet connection.

3.7.2 Configuring the Interface

After you create a device, you must create and configure the Ethernet interface. Follow these steps:

1. Create the interface, associate it with a device using the **iface make** command, specify the encapsulation method, and assign an IP address to that interface.

   iface make *iface-name* **device** *device-name* [**address** *ip-addr*] **encapsulation** *encap-name*

Note You can assign more than one IP address to your Ethernet interface by repeating the **address** *ip-addr* keyword and argument in the **iface make** subcommand. Refer to Section 3.7.3 for special considerations and instructions.

2. Configure the interface using the **iface config** command:

   iface config *iface-name* [*global-config...*] [*encap-specific...*]

   The options you specify vary depending on the type of interface you are configuring. Refer to the **iface** command description in Appendix A, "TALnet Command Reference," for more information.

To remove an existing interface, use the **iface kill** command. To display information about an interface, issue the **iface show** command at a service console or through a remote Telnet connection.

3.7.3 Assigning Multiple IP Addresses

You can assign more than one IP address to your Ethernet interface. This is useful if you plan to run more than one subnetwork on the same Ethernet interface but do not have another router doing the forwarding.

Note Increasing the number of IP addresses on the interface increases the number of computations the router must make for every datagram it receives—this might result in slower performance. Also, you can only specify one subnet mask for all addresses on the Ethernet port. You cannot specify subnet masks for each address you assign.

To assign more than one IP address, repeat the **address** *ip-addr* keyword and argument in the **iface mak**e subcommand:

iface make *iface-name* **device** *device-name* **address** *ip-addr* [**address** *ip-addr*]
encapsulation *encap-name*

3.7.4 Ethernet Configuration Examples

This section provide examples of the Ethernet configuration for the following scenarios:

- An Ethernet interface with one IP address.
- An Ethernet interface with two IP addresses.

Ethernet Interface with One IP Address

The following example configures an Ethernet Class 1 device (`pkt0`) and associates it to an Ethernet interface (`ether0`):

```
device make packet pkt0 0x65
device config pkt0 up enable
iface make ether0 device pkt0 address 192.168.0.1 encapsulation ether
iface config ether0 netmask 255.255.255.0 mxu 1500 broadcast 192.168.0.255
  up
```

The `device make` line creates the device and assigns the symbolic name of `pkt0`. The line also specifies a software interrupt number of `0x65`.

The `device config` line marks device `pkt0` as running, and enables the processing of incoming data.

The `iface make` line associates the interface `ether0` with the device `pkt0`, assigns an IP address of `192.168.0.1`, and specifies Ethernet encapsulation.

The `iface config` line specifies a broadcast address of `192.168.0.255`, which broadcasts messages to all hosts on the local network, and a subnet mask address of `255.255.255.0`. The line also specifies that datagrams no larger than `1500` bytes can be transmitted or received on the interface. Finally, the `up` keyword enables the interface.

Ethernet Interface with Two IP Addresses

The following example illustrates an Ethernet interface (`ether0`) with two IP addresses (`192.168.180.231` and `192.168.181.231`). Both addresses have the same subnet mask (`255.255.255.0`):

```
iface make ether0 device pkt0 address 192.168.180.231 address
192.168.181.231 encapsulation ether
iface config ether0 netmask 255.255.255.0 up
```

3.8 Defining the Location of the DNS Server

As described in Chapter 2, DNS is a distributed database system that associates 32-bit IP addresses with easily recognizable hostnames. Higher-layer protocols such as Telnet use hostnames to identify network devices (hosts). The router and other network devices must be able to associate hostnames with IP addresses to communicate with other IP devices.

To take advantage of DNS, you must perform the following tasks:

1 Identify a nearby server you want to handle DNS requests using the **domain addserver** command. You can identify additional servers by repeating the command.

  domain addserver *ip-addr*

  Where:

  - *ip-addr*—Specifies the IP address of the nearby server in four-part dotted decimal format.

2 Specify the default domain name for the router you are configuring using the **domain suffix** command. Any hostname that does not contain a domain name (that is, any name without a dot) will have the dot and the specified domain name appended to it before being sent to the DNS resolver for translation to an IP address.

domain suffix *suffix*

Where:

- *suffix*—Is the domain name ending with a dot (.).

The following example identifies the server at IP address `198.41.0.4` and specifies `yourdomain.com` as the domain name:

```
domain addserver 198.41.0.4
domain suffix yourdomain.com.
```

To display the currently defined suffix, enter the **domain suffix** command without an argument at a service console or through a remote Telnet connection. To display a list of name servers, enter the **domain list** command at a service console or through a remote Telnet connection.

3.9 Setting Up Routing

You can use static or dynamic routing algorithms to perform routing functions.

If you use static routing, you must create routing tables before you begin routing. These routing tables do not change unless you manually change them. Use static routing only if network traffic is relatively predictable and network design is relatively simple (for example, if there is only one path to the destination, and that path will not change). Perform the following tasks if you use static routing:

1 Create a routing table.

2 Add a default route to the routing table.

3 Make sure RIP is not enabled on your router.

If you use dynamic routing, each router in your network dynamically maintains a routing table. The Routing Information Protocol (RIP) is an interior gateway protocol that enables dynamic routing. Perform the following tasks to use dynamic routing:

1 Manually add a default route to the routing table.

2 Identify associated networks (networks from which your router receives routing update messages).

3 Optionally, enable automatic wireless neighbor updates.

4 Implement wireless neighbor security (also optional).

5 Make sure RIP is enabled on your router.

The following subsections explain these tasks.

3.9.1 Creating a Static Routing Table

To manually add entries to a routing table, use the **route add** command. This command adds the entry to the table and sends the information to other routing tables in the network.

route add *dest-ip-addr*[/*bits*] *iface* [*gateway* [*metric*]]

Where:

- *dest-ip-addr*—Indicates the target IP address expressed in four-part dotted decimal format.
- /*bits*—Is a slash followed by an integer indicating the number of effective subnet mask bits.
- *iface*—Is the symbolic name of the interface through which the datagram is routed.
- *gateway*—Indicates the target router, and can be an IP address expressed in four-part dotted decimal format, or **none**. Use the **none** keyword when you do not want to specify a target router but do want to specify a hop count.
- *metric*—Is an integer indicating the hop count to the destination.

3.9.2 Adding a Default Route to a Routing Table

Whether you are using static or dynamic routing, you should manually enter a default route. This default route is used if no other entry in the routing table is valid.

To add a default route that is not propagated to other routing tables in the network, use the **route addprivate** command:

route addprivate default *iface* [*gateway* [*metric*]]

To add a default route that is propagated to other routing tables in the network, use the **route add** command:

route add default *iface* [*gateway* [*metric*]]

The following descriptions apply to arguments in both commands:

- **default**—Specifies a default route.
- *iface*—Is the symbolic name that indicates the interface through which the datagram is routed.
- *gateway*—Indicates the target router, and can be an IP address expressed in four-part dotted decimal format, or **none**. Use the **none** keyword when you do not want to specify a target router but do want to specify a hop count.
- *metric*—Is an integer indicating the hop count to the destination.

3.9.3 Identifying Associated Networks

To use dynamic routing on the Ethernet interface, or on the wireless interface if you do not enable automatic neighbor updates, you must specify at least one network or host to which routing updates will be sent. A routing process listens to updates from other routers on these networks and broadcasts its own routing information on those same networks.

To specify associated networks, add each network to the RIP neighbor table using the **rip add** command:

rip add *hostid*

Where:

- *hostid*—Indicates the neighbor, and is an IP address expressed in four-part dotted decimal format. On an Ethernet LAN, the host ID is usually the broadcast address; in other words, the host portion is all ones.

You can also specify the type of neighbor you are adding and the UDP port number for the RIP daemon. If you do not specify a neighbor type, the default is RIP Version 2 (RIP-2). If you do not specify a port number, the default is 520. Refer to the **rip** command description in Appendix A, “TALnet Command Reference.”

To display the current RIP neighbor table, use the **rip status** command at a service console or through a remote Telnet connection.

3.9.4 Enabling Automatic Neighbor Updates

If you are using dynamic routing, you can enable automatic neighbor updates. If this feature is enabled, every time your Wireless Router discovers a neighboring Wireless Router on a wireless link, it automatically adds the neighboring router to the RIP neighbor table. Likewise, if your Wireless Router does not hear a neighboring router for at least two minutes, the neighboring router is removed from the RIP neighbor table.

Automatic neighbor updates are disabled by default. To determine whether the feature is enabled or disabled, use the **talk radio** *iface-name* **auto-neighbor** command without any arguments at a service console or through a remote Telnet connection.

To enable or disable the auto-neighbor feature, use the **talk radio auto-neighbor** command:

talk radio *iface-name* **auto-neighbor** [**on** | **off**]

Where:

- **on**—Enables automatic neighbor updates.
- **off**—Disables automatic neighbor updates.

3.9.5 Enabling Wireless Neighbor Security

Wireless neighbor security protects data from unauthorized access; setting it up is optional. You can enhance wireless neighbor security through one or both of the following processes:

- Neighbor address identification—Explicitly allows neighboring routers to contact a particular router based on their address.

  Neighbor address identification verifies the legitimacy of all router contacts, confirming that the connection request is received from a pre-authorized (known) source.

- Key authentication—Authenticates the keys provided by neighboring routers.

  Key authentication is a process that verifies the router’s identity by checking its key (as defined in the *talnet.cfg* file during configuration), and upon successful verification, authorizes access.

Note The key authentication feature must be ordered separately. You can use the **version** command to see if it was included in your software—if it is included, you see the word “Encryption” in the list of included features. Contact your technical support representative to order the key authentication feature.

Although both neighbor address identification and key authentication are optional, we recommend that you implement both processes in your router’s configuration file to obtain optimal security against unauthorized access.

Neighbor Address Identification

Neighbor address identification is the process of comparing the router attempting contact with a list of authorized routers in the configuration file, *talnet.cfg*.

To explicitly allow a neighbor to contact your router, use the **talk radio neighbor allow** command:

talk radio *iface-name* **neighbor allow** *talk-address*

Where:

- *iface-name*—Is the symbolic name assigned by the **talk radio address** command.
- **allow**—Indicates that the router you are configuring should allow the specified neighbor to send it information.
- *talk-address*—Is the TALtalk link-layer address of the neighboring transmitter.

You must enter **allow neighbor** commands in the *talnet.cfg* configuration file to ensure that the router is set to perform neighbor address identification. If you do not explicitly allow neighbors in the configuration file, the router will bypass neighbor address identification security and allow all neighbors access.

Note The first time you issue the **talk radio neighbor allow** command, you must enter it in the *talnet.cfg* file, *not* at the service console. You can enter subsequent **talk radio neighbor allow** commands at the console. (Remember that any commands you issue at the service console are not retained when the system is rebooted.)

When you want to disallow a neighbor during the current session, enter the **talk radio neighbor disallow** command at the console:

talk radio *iface-name* **neighbor disallow** *talk-address*

Where:

- *iface-name*—Is the symbolic name assigned by the **talk radio address** command.
- **disallow**—Indicates that the router you are configuring should not receive information sent by the specified neighbor.
- *talk-address*—Is the TALtalk link-layer address of the neighboring transmitter.

To permanently disallow a neighbor, delete the **talk radio neighbor allow** command for that neighbor from the *talnet.cfg* file.

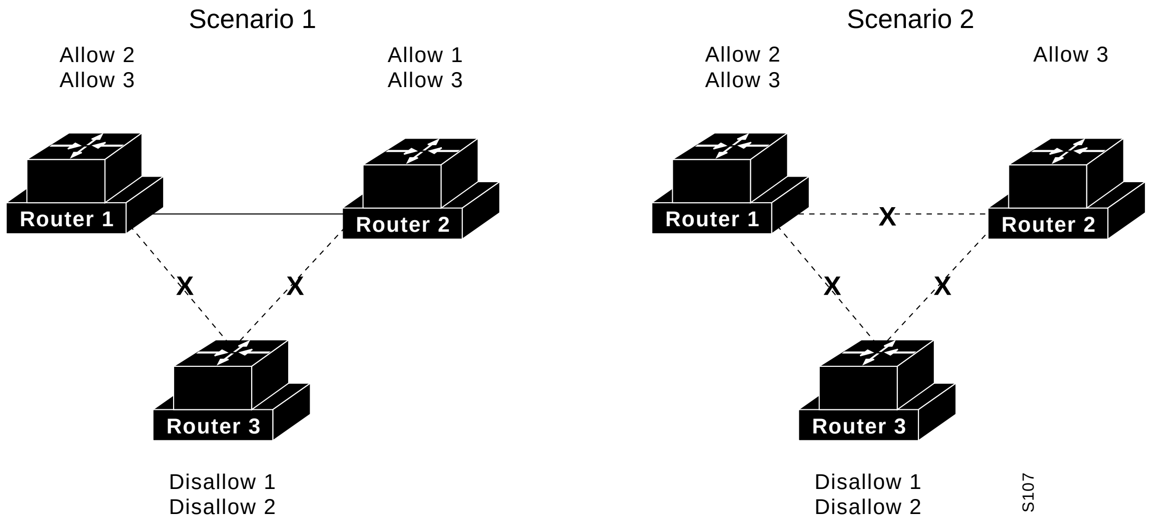
Note If no other **talk radio neighbor allow** commands currently exist in the configuration file, then the neighbor must be explicitly disallowed. This is accomplished by including the **talk radio neighbor disallow** command (including the neighbor's address) in the *talnet.cfg* file.

To resume contact with a disallowed router, you can issue the **talk radio neighbor allow** command or, to make the change permanent, you can remove the **talk radio neighbor disallow** command from the *talnet.cfg* file.

To display the current table of explicitly allowed neighboring routers, use the **talk show allowfilters** command. The command **talk show neighbors verbose** will also display all allowed neighbors.

The following figure illustrates the outcomes of two allow and disallow scenarios.

Figure 3-1 Allow and Disallow Scenarios



- **Scenario 1: Disallows override allows**—Routers 1 and 2 both allow router 3, but router 3 disallows 1 and 2. The "disallows" overrides the "allows" so that no data can be sent to or received from router 3. Routers 1 and 2 can send data to and receive data from each other because they allow each other.
- **Scenario 2: The absence of an allow is the same as a disallow**—No data is being sent or received between any of the routers. As in scenario 1, routers 1 and 2 both allow router 3, but router 3 disallows 1 and 2 so no data is sent to or received from 3. Router 1 allows router 2, but since router 2 does not explicitly allow router 1, this overrides the allow so that no data can be sent or received between router 2 and router 1.

Key Authentication

Key authentication uses a special key to verify a router's identity. If you do not create a key, the router does not use key authentication when establishing links with other routers. To permanently establish, change, or delete key authentication, you enter commands directly in the *talnet.cfg* configuration file rather than at the service console. The first time you enable key authentication on your router, you need to establish the key in the *talnet.cfg* file. You should add this key to the configuration files of all routers in your network that must communicate with each other.

If you have Admin privileges, you can enable key authentication using the **key create** command:

key create *8-byte-key*

Where:

- *8-byte-key*—Represents the value or character string you want to use as your key. The key should be eight characters (you may use alphanumeric characters and hyphens).

Note All routers within a network must be configured with the same 8-byte key. Once you establish a key in the *talnet.cfg* configuration file, key authentication is assumed for all routers in the network.

You can determine whether a router is using key authentication by looking for the following message that appears in the boot up messages when TALnet is running:

```
Authentication key has been created
```

If you have Admin privileges, you can temporarily change key authentication for newly acquired routers using the **key config** command. Only use the **key config** command at the service console; to permanently change key authentication for an entire network, you must change the **key create** command in the *talnet.cfg* file.

key config *8-byte-key*

Where:

- *8-byte-key*—Represents the value or character string you want to use as your new key. The key should be eight characters (you may use alphanumeric characters and hyphens).

Caution A router must have the same key as other nodes on its local network. If you change a router's key in the configuration file and authenticated neighbors within the same network are not also changed, the router will become isolated on the network.

If you have Admin privileges, you can temporarily delete key authentication for newly acquired routers using the **key delete** command. Only use the **key delete** command at the service console; to permanently delete key authentication for an entire network, you must delete the **key create** command in the *talnet.cfg* file. Unless you create a new key after the old key is disabled, this router will not use key authentication when establishing new link access with other routers.

key delete *8-byte-key*

Where:

- *8-byte-key*—Represents the value or character string you no longer want to use as a key. The key should be eight characters (you may use alphanumeric characters and hyphens).

Note Changes or deletions made to key authentication at the service console are for the current session only. If you have not removed the **key create** command line from the *talnet.cfg* configuration file, when the router is rebooted it resumes key authentication with the entered key. To make permanent changes to key authentication, you must make your changes directly in the configuration file.

3.9.6 Enabling or Disabling RIP

To enable RIP, use the **start rip** command. To disable RIP, use the **stop rip** command.

3.10 Defining User Authorization

You can provide users with different levels of access to the Wireless Router by setting up a database of users. In particular, you can define the level of changes the user is authorized to make, whether the user can connect at the service console or remotely, the interface on which the user can connect using PPP, and the root directory.

To add a new user, enter the **user create** command.

user create *name* [**authorization** *flag* [*flag...*]] [**iface** *iface*] [**password** *pswd*] [**root** *dir*]

After you have added a user, you can append options for that user with the **user config** command:

user config *name* [**authorization** *flag* [*flag...*]] [**iface** *iface*] [**password** *pswd*] [**root** *dir*]

The following descriptions apply to arguments in both commands:

- *name*—Is the username of the user you are adding.
- **authorization** *flag*—Establishes the level of access privileges. All TALnet commands (and some subcommands) require different authorization levels. The argument *flag* represents one or more of the following levels of authorization for users:
  - **Read**—Allows the user to read any file on the Wireless Router (for example, the *talnet.cfg* configuration file, or executable files [**.exe*]). This command also allows the user to obtain information about the disk (using the **dir** command, for example). When you issue this command, it adds the Read authorization bit (0x001).
  - **Write**—Allows the user to modify files. When you issue this command, it adds the Write authorization bit (0x004).

— **Create**—Allows the user to create new files or directories. When you issue this command, it adds the Create authorization bit (0x002).

— **Operator**—Allows the user to reboot the router and interactively modify the configuration for an active session without permanently changing the *talnet.cfg* configuration file. This command does *not* allow the user to type **exit** to get back to the router's native operating system. When you issue this command, it adds the Operator authorization bit (0x010).

— **Admin**—Allows the user to permanently change the configuration file and to enter the **exit** command to return to the router's native operating system. When you issue this command, it adds the Admin authorization bit (0x200). However, when you display the current users with the **user show** command, the authorization bit is listed as 0x217 because the system automatically appends the Operator, Read, Write, and Create privileges to the Admin privilege.

— **PPP**—Allows the user to connect to the Wireless Router using the Point-to-Point Protocol (PPP). You must also specify the interface on which the user can make PPP connections. When you issue this command, it adds the PPP authorization bit (0x20).

— **Console**—Allows the user to connect to the router from a service console. When you issue this command, it adds the Console authorization bit (0x040).

— **Netlogin**—Allows the user to connect to the router from a Telnet or FTP session. You must also specify the root directory for FTP transfers. When you issue this command, it adds the Netlogin authorization bit (0x100).

You can assign a combination of privileges to a single user; for example, you could assign Admin, PPP, and Console privileges to the same user. Alternatively, you can assign privileges with a bitmask of the desired privileges. We recommend that you define at least one user with Admin and Console privileges. If you do not assign a user with Console privileges, anyone can access the router from a service console without being prompted for a username or password.

- **iface** *iface*—Specifies a particular interface. Specify the interface if you are going allow the user to make PPP connections.
- **password** *pswd*—Specifies a user password.
- **root** *dir*—Indicates the user's root directory for FTP transfers. Specify the root directory whenever you assign Netlogin privileges.

In the following example, the user `operator` can make changes to the configuration interactively, without modifying the configuration file and rebooting the Wireless Router because the user has Operator privileges. The user's password is `faux`.

```
user create operator authorization Netlogin console operator password faux
root /
```

In the following example, the user `admin` has Admin privileges. To connect to the Wireless Router through a service console, the user `admin` must issue a password, `super`. If any users are defined, only users with Console privileges can access the Wireless Router through a service console.

```
user create Admin authorization Admin Netlogin console password super
root /
```

To display the user database, enter the **user show** command (you must have Admin privileges) at a service console or through a remote Telnet connection.

3.11 Establishing IP Packet Filters

Packet filtering helps control packet movement through the network. Such control can help limit network traffic and restrict network use by certain users or devices.

A packet filter is a sequential collection of permit and deny conditions that apply to addresses. When the router receives a packet, it checks that packet against any filters. If you define any packet filters on an interface, all packets are checked against the filters. If a packet does not match any of the conditions specified, it is implicitly denied. Therefore, you must make sure you establish filters for all packets you want to accept. Because the router searches sequentially through packet filters, make sure you define more specific filters first.

To filter packets, use the **ip filter** command:

ip filter *iface-name* {**deny** | **permit**} {**in** | **out**} *type src dst*

Where:

- *iface-name*—Is the symbolic name of the interface on which you want to apply the filter. This name should match the symbolic name assigned to that interface with the **iface create** or **iface make** command.
- {**deny** | **permit**}—Indicates whether you want to reject (deny) or accept (permit) the specified packet.
- {**in** | **out**}—Indicates whether you want to filter incoming or outgoing packets.
- type—Specifies the type of packet you want to filter. Section 3.11.1 describes this argument in greater detail.
- src—Allows you to determine how to apply filters on the source address of the packet. Section 3.11.2 describes this argument in greater detail.
- dst—Allows you to determine how to apply filters based on the destination address of the packet. Section 3.11.2 describes this argument in greater detail.

3.11.1 Filtering Packets Based on Packet Type

You can specify the type of packet you want to filter. You can filter any of the following types of packets:

- To filter all packets, enter an asterisk (*).
- To filter only Internet Control Message Protocol (ICMP) datagrams, specify **icmp**.

- To filter only ICMP redirect datagrams, specify **icmprd**. An ICMP redirect indicates that a datagram should have been sent to a different router.
- To filter all ICMP datagrams except redirect datagrams, specify **icmpxrd**.
- To filter all TCP packets, specify **tcp**.
- To filter only TCP SYNs, specify **tcpsyn**. A SYN is a synchronize sequence number flag in the TCP header, and specifies the first byte of data in the current message. This byte of data initiates a connection to a router. Filter on SYNs when you want to prevent a host from initiating a connection to the router. In general, you filter on incoming SYNs.
- To filter all TCP packets except TCP SYNs, specify **tcpxsyn**.
- To filter all UDP datagrams, specify **udp**.

3.11.2 Filtering Packets Based on Source or Destination Addresses

You must specify how you want to filter based on the source and destination addresses of a packet. You can use any of the following options:

- *—Applies the filter to all addresses.
- *ip-addr*—Applies the filter to the specified address.
- **!** *ip-addr*—Applies the filter to all but the specified address. If this is used with a deny condition, all addresses but the specified address are denied. If this is used with a permit condition, all addresses but the specified address are permitted.
- *ip-addr*/*bits*—Applies the filter to all addresses in which the specified number of bits matches the most significant bits in the specified address.
- **:** *loport* {+|– *hiport*}—Applies the filter to the specified port or port range. The plus sign (+) specifies any port number greater than or equal to the one entered in the previous field. The minus sign (–) specifies a range of port numbers, and must be followed by the high port number. You can only use this when filtering TCP or UDP packets. This is useful for denying only certain connections; for example, Telnet or FTP sessions. You can specify filtering based on ports and filtering based on an IP address.

3.11.3 Packet-Filtering Examples

The following example applies filters on interface `ether0`. The first line denies all incoming TCP packets with a source address within the `192.168.180` subnet and a destination address within the `192.168.181` subnet. The second line includes a wildcard (*) that permits all traffic with these source and destination addresses. The result is that all traffic with these addresses are permitted *except* TCP packets. Because filtering is sequential, you must list the more specific filter first. Once the router encounters the filter that permits all packets, it stops looking for filters with that source and destination address. Because filtering uses an implicit deny condition, all other packets coming in on interface `ether0` are denied.

```
ip filter ether0 deny in tcp 192.168.180.0/24 192.168.181.0/24
ip filter ether0 permit in * 192.168.180.0/24 192.168.181.0/24
```

The following example permits all incoming ICMP packets on interface `ether0` with a source address within the `192.168.180` subnet and any destination address *except* those within the `192.168.181` subnet. Because packet filtering includes an implicit deny condition, all other packets are denied.

```
ip filter ether0 permit in icmp 192.168.180.0/24 !192.168.181.0/24
```

The following example denies all incoming connection requests, as specified with the `tcpsyn` keyword, from any source address to the `192.168.180.0` network on the Telnet port (`:23`). It does not deny traffic on established connections.

```
ip filter ether0 deny in tcpsyn * 192.168.180.0/24:23
```

3.12 Modifying the ARP Cache

In earlier sections, we discussed how DNS maps names to 32-bit IP addresses. However, data links such as Ethernet or the radio use different schemes for physical addresses. For example, Ethernet uses 48-bit addresses to determine a frame's destination interface and the TALtalk address uses 32-bit addresses. Ethernet addresses are assigned by the manufacturer. TALtalk addresses are assigned by the site coordinator, TAL, or a central authority.

The TALnet software uses ARP and RARP to map between the IP address at the network layer and the physical address at the link layer. When a router wants to send a datagram, it sends out a broadcast message called an *ARP request* to every other router on the network. This message asks the router with that IP address to send back its physical address. All routers on the network receive the ARP request, but only the router with the matching IP address responds. This router returns an *ARP reply* that contains the IP address and the corresponding physical address.

To limit the number of broadcasts sent over the network, ARP maintains a cache of recently acquired IP-to-physical address mappings. Before sending out an ARP request, a router checks the ARP cache. Whenever an ARP request is sent, all routers on the network update the ARP cache with the sending router's addresses. The sending router also adds the mapping sent in the ARP reply. In addition, whenever you add a new router to your network, you can broadcast its addresses to all other routers on the network.

The TALnet software also supports proxy ARP, a technique in which one router, usually a gateway, answers ARP requests intended for another by supplying its own physical address. This answering router acts as a proxy agent; the frame is sent to the proxy agent, which then assumes responsibility for routing the frame to the correct destination. In this way, proxy ARP allows you to use a single IP network address with multiple physical networks.

To manually add an entry to the ARP cache, use the **arp add** command. Use this command when a router does not respond correctly to an ARP request.

arp add *hostid* {**ether** | **talk**} *link-layer-addr*

To establish a router as a proxy agent, use the **arp publish** command. This command adds the entry to the ARP cache and specifies that the proxy router should respond to ARP requests intended for the routers for which it acts.

arp publish *hostid* {**ether** | **talk**} *link-layer-addr*

The following descriptions apply to arguments in both commands:

- *hostid*—Specifies the DNS name or IP address in four-part dotted decimal format.
- **ether**—Specifies an Ethernet link-layer address.
- **talk**—Specifies a TALtalk link-layer address.
- *link-layer-addr*—Specifies the physical address of the interface.

3.13 Configuring Support for PPP Connections

PPP is a link-layer protocol that provides a method for transmitting datagrams over serial point-to-point links. Using PPP, you can run TCP/IP applications including Telnet and FTP over serial lines. PPP allows you to connect through a Wireless Router to a network with the same functionality as a PC directly attached to a local network.

Using PPP with a TAL Wireless Router, you can make two kinds of connections:

- Remote connections—A PC can dial in to the router across a high-speed modem (at least 9600 baud) and standard telephone lines. If the cable you are using to make PPP connections supports hardware flow control, make sure that hardware flow control is enabled on the modem.
- Direct connections—A PC or router can connect directly to the router using an RS-232 cable.

The following sections provide a brief overview of how PPP works, then describe how to configure your router to make remote or direct connections.

Note The TALnet software does not yet provide capabilities for dialing out.

3.13.1 Overview of PPP

PPP consists of three components:

- A method of encapsulating datagrams on a serial link: either an asynchronous link with 8 bits of data and no parity, or bit-oriented synchronous links.
- A link control protocol (LCP) to establish, configure, and optionally test the data-link connection. This allows each end to negotiate various options.
- A family of network control protocols (NCPs) specific to different network-layer protocols. The NCP for IP is called IPCP, and allows each end of the link to specify if it can handle header compression, exchange IP addresses, and perform other IP-only options.

When you make a PPP connection, the following occurs to establish communications between the two nodes:

1. The originating end of the link sends LCP link establishment frames that open the connection and negotiate configuration parameters.
2. The originating end of the link sends LCP link-quality reporting (LQR) frames to determine the quality of the link.

3 The originating end of the link sends NCP frames to choose and configure network-layer protocols.

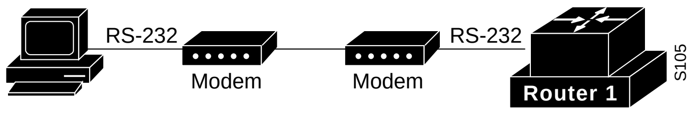
At this point, datagrams can be sent over link. LCP closes the link by sending link termination frames. This usually occurs when you explicitly request that the link be closed, but LCP can close the link at other times, for example, when an inactivity timer expires.

3.13.2 Configuring Support for Remote PPP Connections

You can connect your PC to a Wireless Router remotely using modems and a standard telephone line. Figure 3-2 illustrates this kind of connection.

Note You must have ppp privileges as defined with the **user** command in order to connect remotely via PPP.

Figure 3-2 Remote PPP Connection



Follow these steps to configure your Wireless Router to accept remote PPP connections:

1 Create a device that provides an asynchronous serial connection:

device make com *device-name* **0x2f8 3**

Where:

- *device-name*—Is a symbolic name for the device; for example, *com2*.
- **Ox2f8**—Is a number that tells the Wireless Router how to address the device.
- **3**—Is a number that tells the Wireless Router on which interrupt request line to expect interrupts for the device.

2 Configure the device to support PPP connections:

device config *dev-name* **cts rtson dtron rxfifo** *bytes* **modem speed** *bit-rate*

Where:

- *dev–name*—Is the symbolic name you assigned with the **device make com** subcommand.
- **cts**—Enables outgoing hardware flow control. The Wireless Router checks the modem for a Clear-to-Send (CTS) signal, which indicates that the router can send data to the modem.
- **rtson**—Enables incoming hardware flow control. The Wireless Router sends a Request to Send (RTS) signal to the modem, indicating that the router can receive data.
- **dtron**—Activates the DTR flag, which tells the modem that the Wireless Router is ready to accept calls.

- **rxfifo** *bytes*—Sets the size of the transmit/receive buffer, in bytes. The default is 1024 bytes.
- **modem**—Indicates that a modem is attached to the device.
- **speed** *bit-rate*—Sets the speed at which the device should run. The speed you set should be as high as the modem supports. Note that if you use hardware flow control, you can usually use higher speeds.

3 If necessary, configure the Wireless Router to reinitialize the modem if the router reboots. This might be necessary if your modem does not have enough memory to save its own configuration and you want to ensure that the modem resets correctly.

device config *device-name* **init "***string***"**

Where:

- *device–name*—Is the symbolic name you assigned with the **device make com** subcommand.
- *string*—Is any valid combination of modem configuration commands. The commands you can use vary depending on your modem. You can also include the following special characters:

  \p—To pause for one second.
  \r—To send a carriage return.
  \\—To send a backslash.
  \"—To send a quotation mark.

Note If you configure the Wireless Router to reinitialize the modem, you must place this command line before the **device config** command with the **enable** keyword. You must configure all other PPP device parameters *before* this command.

---

4 Enable the device:

device config *dev-name* **up enable**

Where:

- *dev–name*—Is the symbolic name you assigned with the **device make com** subcommand.
- **up**—Allows the wireless router to transmit or receive data through this device and presents a login prompt to the user when the modem detects a valid connection.
- **enable**— Turns the physical device on, making it ready to transmit or receive data.

5 Create an interface and specify the method of encapsulation:

iface make *iface-name* **address** *ip-addr* **mxu** *size* **encapsulation ppp**

Where:

- *iface-name*—Assigns a symbolic name to the interface. This name can include uppercase and lowercase letters, hyphens, and numbers, and should indicate the type of interface and the unit; for example, ppp0.

- **address** *ip-addr*—Specifies the IP address to be used for the interface.
- **mxu** *size*—Sets the maximum size of datagrams that can be received on the interface. This is the size without any PPP encapsulation. We recommend that you use 1500 bytes.
- **encapsulation ppp**—Specifies PPP encapsulation.

6 Create a list of users who can make PPP connections:

user create *name* **authorization ppp iface** *iface-name* **password** *pswd*

Where:

- *name*—Is the username of the user.
- **authorization ppp**—Specifies that the user is authorized to make PPP connections.
- **iface** *iface-name*—Defines the interface on which the user can make PPP connections. The argument *iface-name* is the symbolic name you defined in the **iface make** command.
- **password** *pswd*—Specifies a user password.

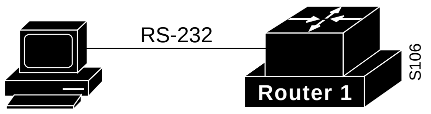
Note Before you make a PPP connection through a modem, you must configure that modem to use auto-answer mode and to match the speed and other parameters of the Wireless Router. You can configure the modem directly from the Wireless Router; see the description of the **tip** command in Appendix A, “TALnet Command Reference.”

3.13.3 Configuring Support for Direct PPP Connections

You can also connect your PC or another router directly to a Wireless Router. You might make this kind of connection if you want a dedicated PPP connection to the Wireless Router. This kind of connection might actually use a modem, but because the connection is dedicated, the software does not consider the modem. Because you are configuring a dedicated direct connection, you do not need to define users.

Figure 3-3 illustrates a PC directly connected to a Wireless Router.

Figure 3-3 Direct PPP Connection



The following process describes how to configure your Wireless Router to support direct PPP connections. Because the connection is dedicated and direct, the configuration commands are slightly different from those you use to configure a remote PPP connection.

1 Create a device that provides an asynchronous serial connection:

   device make com *device-name* **0x2f8 3**

   Where:

   - *device-name*—Is a symbolic name for the device; for example, com2.
   - **0x2f8**—Is a number that tells the Wireless Router how to address the device.
   - **3**—Is a number that tells the Wireless Router on which interrupt request line to expect interrupts for the device.

2 Configure the device to support PPP connections:

   device config *dev-name* [**cts rtson**] **rxfifo** *bytes* **speed** *bit-rate* **enable**

   Where:

   - *device-name*—Is the symbolic name you assigned with the **device make com** subcommand.
   - **cts**—Enables outgoing hardware flow control. The Wireless Router checks the remote device for a Clear-to-Send (CTS) signal, which indicates that the router can send data to the modem. (Do not use this keyword if your cable does not support the CTS signal.)
   - **rtson**—Enables incoming hardware flow control. The Wireless Router sends a Request to Send (RTS) signal to the modem, which indicates that the router can receive data. (Do not use this keyword if your cable does not support the RTS signal.)
   - **rxfifo** *bytes*—Sets the size in bytes of the transmit/receive buffer. The default is 1024.
   - **speed** *bit-rate*—Sets the speed at which the device should run. The speed you set should be as high as the device you are connecting to supports. Note that if you use hardware flow control, you can usually use higher speeds.
   - **enable**—Turns the physical device on, making it ready to transmit or receive data.

3 Create an interface, associate it with the device, and specify the method of encapsulation:

   iface make *iface-name* **device** *device-name* **address** *ip-addr* **mxu** *size* **encapsulation ppp up**

   Where:

   - *iface-name*—Assigns a symbolic name to the interface. This name can include uppercase and lowercase letters, hyphens, and numbers, and should indicate the type of interface and the unit; for example, *ppp0*.
   - **device** *device-name*—Associates the interface with the device, specified by the argument device-name. This argument is the symbolic name you assigned with the **device make** command.
   - **address** *ip-addr*—Specifies the IP address to be used for the interface.
   - **mxu** *size*—Sets the maximum size of datagrams that can be received on the interface. This is the size without any PPP encapsulation. We recommend that you use 1500 bytes.

- **encapsulation ppp**—Specifies PPP encapsulation.
- **up**—Marks the device as up and starts the link-layer protocol.

3.13.4 PPP Configuration Example

The following example configures the Wireless Router to accept remote PPP connections:

```
device make com com2 0x2F8 3
device config com2 cts rtson dtron rxfifo 1024 modem speed 9600 up enable
iface make ppp0 address 192.168.1.1 mxu 1500 encapsulation ppp
user create pppuser authorization ppp iface ppp0 password ppp
```

The `device make` line creates the device and assigns the symbolic name of `com2`. The line also specifies the standard I/O base of `0x2f8`, and an interrupt request line of `3`.

The `device config` line configures device `com2` to listen to CTS signals from the modem and send RTS signals to the modem. The `dtron` keyword tells the modem that the router is ready to accept calls. The transmit/receive buffer size is set to `1024` bytes. The modem keyword indicates that a modem is attached to the device. Finally, the speed is set to 9600 bps, the device is marked as running, and the `enable` keyword turns the physical device on, making it ready to transmit or receive data.

The `iface make` line creates the interface `ppp0`, assigns an IP address of `192.168.1.1`, sets maximum size of datagrams to `1500` bytes, and specifies PPP encapsulation.

The `user create` line defines a user with the username `pppuser`, who can make PPP connections. The user must make these connections on the interface `ppp0`, and password for the user is `ppp`.

3.14 Reloading the Configuration File

After you modify the configuration file, you must reload and execute that file. The steps for doing this differ depending on whether you edited the file locally through the service console or remotely using FTP.

Caution Make sure you create a backup of the original configuration file before you replace it with the one you modified.

3.14.1 Reloading the File from the Console

After you have modified the configuration file according to the sections earlier in this chapter, follow these steps to save the file and execute the changes. These steps assume you are still in the Xvi editor.

1 Save the changes to the configuration file:

```
<ESC>:wq
```

Note Saving the changes might take several minutes. A C: prompt will appear when the process is complete.

2 Run the *start.bat* script to execute the changes:

```
c:\tal> start
```

3 Verify the changes by checking the new configuration. Issue a command to show the current status of something you changed. The following are some suggestions:

Check that the interfaces you configured exist:

```
TAL> iface show all
```

Check the system contact:

```
TAL> snmp syscontact
```

4 Exit the system in one of the following ways:

- If this is the first time you have logged onto your system, enter exit:

```
TAL> exit
```

- If you have previously logged onto your system and defined users, log out of the Wireless Router:

```
TAL> logout
```

Note It is very important that you log out of the Wireless Router whenever you have completed your activities on that router. If you are logged in through the service console via a modem and disconnect before logging out, other users can access TALnet over a modem without logging in.

3.14.2 Reloading the File from a Remote System

After you have modified the configuration file according to the sections earlier in this chapter, follow these steps to save the file and reload it.

1 Establish an FTP connection to the router:

```
> ftp router
```

The *router* argument is either the DNS name or IP address of the router.

2 When prompted, enter your username and password as defined in the configuration file.

3 Change to the *tmp* directory on the D: drive:

```
ftp> cd d:\tmp
```

4 Select ASCII transfer mode:

```
ftp> ascii
```

5 Turn hash mark printing on; hash marks (#) appear on the screen to indicate progress during a file transfer:

```
ftp> hash
```

6 Using FTP, copy the new configuration file back to the D: drive on the router under a new, temporary name such as *temp.cfg*. Remember that eventually you will be overwriting the original configuration file, so make sure you have backed up the original file.

```
ftp> put talnet.cfg temp.cfg
```

7 Close the FTP session:

```
ftp> quit
```

8 Open a Telnet session to the router:

```
> telnet router
```

The *router* argument is either the DNS name or IP address of the router.

9 When prompted, enter your username and password as defined in the configuration file.

10 Change to the TAL directory on the C: drive:

```
TAL> cd c:\tal
```

11 Move the new configuration file to the TAL directory on the C: drive:

```
TAL> copy d:\tmp\temp.cfg c:\tal\temp.cfg
```

12 Rename the existing *talnet.cfg* file to *talnet.old*:

```
TAL> rename talnet.cfg talnet.old
```

13 Rename the new configuration file *temp.cfg* to *talnet.cfg*:

```
TAL: rename temp.cfg talnet.cfg
```

14 Reload the configuration file by restarting the system; this also ends the Telnet session:

```
TAL> reboot
```

After about 2 minutes, verify that you loaded the correct configuration file. Follow these steps:

1 Open a new Telnet session to the router:

```
> telnet router
```

The *router* argument is either the DNS name or IP address of the router.

2 View the current configuration file:

```
TAL> view c:\tal\talnet.cfg
```

Entering the **view** command displays the file on your screen. To scroll down, enter **Ctrl-V**. To scroll up, enter **Ctrl-U**.

3. Close the file you are viewing:

   ```
   q
   ```

4. Log out of the TALnet software and close the Telnet connection:

   ```
   TAL> logout
   ```