Chapter 4

Recording Network Statistics

This chapter describes the most common commands you can use to monitor network performance and generate network statistics. This chapter does not cover all monitoring commands.

The information you gather by performing the tasks in this chapter will help your technical support representative diagnose the problem. Record the information you gather and be prepared to provide it to your service provider.

You can enter any of these commands either from the service console or remotely across a Telnet connection. The output generated by the command appears on your monitor.

This chapter provides sample output; your output will vary. Underlined text in sample output indicates the primary fields you should monitor.

Determine Your Software Version

If you experience problems with your Wireless Router, identify the version of software and the date the software was compiled before contacting your technical support representative. You might also need to supply the router platform.

version

The following example lists the current version of TALnet software as Release 1.2, compiled on February 23, 1996. The router platform is SubSpace 2001 (rev1).

```
TAL> version
TALnet Release 1.2
Copyright 1992-1996 Tetherless Access Limited, All Rights Reserved.
Compiled at TAL: Feb 23 1996, 12:23:05
Router platform: SubSpace 2001 (rev1)
Memory Usage:    Library=42K, Code=393K, Data=37K
```

Determine Uptime

Before you contact your technical support representative, determine the length of time since your Wireless Router was last booted. A very short uptime (for example, a few seconds) might indicate that the system is rebooting unexpectedly.

uptime

In the following example, the Wireless Router has been running for 24 days, 5 hours, and 51 seconds:

```
Router-1> uptime

System up 24 days, 5 hours, 51 seconds.
```

Display Statistics on Devices

Display statistics on all devices on your Wireless Router:

device show all

In the display, note the following information:

- Make sure all configured devices appear in the display. This first column lists the symbolic device name for each device. Your Wireless Router might list the following devices: an Ethernet device, a wireless (radio) device, a watchdog timer, and a device for Point-to-Point Protocol (PPP) connections.
- Make sure the device type for each device is correct. The second column lists the device type: PACKET indicates an Ethernet device, SCC indicates a wireless device, WATCHDOG indicates the watchdog timer, and COM indicates device for PPP connections.

- Make sure the device is associated with the correct interface. The `Owner` column lists the symbolic name for the interface associated with the device.

For example:

```
radio> device show all

Name    Type    Class  Index Flags  Owner    Hardware Address
ether0  PACKET         0     C007   ether0   00:20:0C:10:22:FF
radio0  SCC            0     C007   radio0
wdt0    WATCHDOG       0     8000
```

If you know the symbolic name assigned to a device, you can view information for only that device:

device show *device-name*

For example:

```
radio> device show radio0

Name    Type    Class  Index Flags  Owner    Hardware Address
radio0  SCC            0     C007   radio0
```

Display Statistics about Interfaces

After you display and record statistics about your devices, display statistics about interfaces. First, display statistics about all interfaces on your network:

iface show all

In the display, note the following information:

- Make sure all interfaces are listed. You will probably see all five of the interfaces listed in the first column of this display.
- Make sure the Internet Protocol (IP) addresses for the Ethernet and radio interfaces are correct. The IP address is listed on the first line, and begins with the keyword `address:`.
- Make sure the broadcast address for the Ethernet interface is correct. The broadcast address is the last entry, and begins with the keyword `broadcast`.

For example:

```
radio> iface show all

ether0 Attached to ether0, address: 192.168.180.249 204.94.179.201
       MXU 1500, link encapsulation: Ethernet
       flags 810A trace 0 netmask 0xFFFFFF00 broadcast 204.94.180.255
loopback  Not attached to any device, address: 127.0.0.1
       MXU 1500, link encapsulation: Raw IP
      flags 8002 trace 0 netmask 0xFF000000 broadcast 255.255.255.255
null   Not attached to any device, address: 0.0.0.0
       MXU 1500, link encapsulation: Raw IP
      flags 8002 trace 0 netmask 0xFFFFFFFF broadcast 255.255.255.255
radio0 Attached to radio0, address: 204.94.179.201
       MXU 1500, link encapsulation: TALK
       flags 800A trace 0 netmask 0xFFFFFFFF broadcast 255.255.255.255
tunnel Not attached to any device, address: 0.0.0.0
       MXU 1500, link encapsulation: Raw IP
       flags 8002 trace 0 netmask 0xFFFFFFFF broadcast 255.255.255.255
```

If you know the symbolic name for a specific interface, you can display more in-depth statistics on that interface:

iface show *iface-name* **verbose**

In the display for a single interface, make sure the field with the keyword `ibuf` is not zero (0). This field indicates the current number of buffers available to send or receive information. If you repeat this command over time, the packet and octet counts (indicated by the keywords `pkt` and `oct`) should increase. If they do not increase, no traffic is being sent over the network. For example:

```
radio> iface show ether0 verbose

ether0 Attached to ether0, address: 192.168.180.249 204.94.179.201
       MXU 1500, link encapsulation: Ethernet
       flags 810A trace 0 netmask 0xFFFFFF00 broadcast 204.94.180.255
        sent: pkt 2243 oct 227309 idle 0 disc 3 errs 0 ibuf 4
        recv: pkt 2762 oct 595210 idle 0 disc 0 errs 0 unknown 0
         ibuf 12
```

Monitor the Status of Wireless Neighbors

After you confirm that your devices and interfaces are configured correctly and working properly, display statistics about neighboring Wireless Routers. In other words, display a list of neighbors the local node is capable of transmitting to or receiving from. This is a two-step process:

1. Determine the link-layer addresses of the neighbors by viewing the Address Resolution Protocol (ARP) cache:

   arp

2. Display statistics on the neighbors:

   talk show neighbors

In the following display, locate the IP address of the neighbor you want to monitor. The IP address is listed in the first column of the second portion of the output. Then, locate the link-layer address, listed in the last column. For example, assume you want to monitor the neighboring node at IP address `192.168.182.57`. The ARP cache lists the link-layer address for this IP address as `cc5eb639`.

```
radio> arp

Total number of ARP & RARP packets received...118
Incoming requests for unsupported hardware....0
Incoming length field(s) didn't match types...0
Bad incoming addresses.......................0
Bad ARP or RARP op code......................0
Incoming ARP requests for us..................6
ARP Replies received.........................3
Outgoing ARP requests sent....................3
Incoming RARP requests for us.................5
RARP Replies received........................8
Outgoing RARP requests sent...................14
Outgoing RARP request failures................0

IP addr         Type           Time Q Addr
-------         ----           ---- - ----
255.255.255.255 10 Mb Ethernet 0      ff:ff:ff:ff:ff:ff
192.168.181.3    10 Mb Ethernet 1200  00:20:af:30:01:82
192.168.182.57   TALK           0     cc5eb639
192.168.181.1    10 Mb Ethernet 229   00:00:0c:09:d5:ac
192.168.182.129  TALK           0     cc5eb681
192.168.182.17   TALK           0     cc5eb611
```

Next, display information about the neighbors. Notice that state of the neighbor with this link-layer address (CC5EB639) is Dead. Also notice that three attempts to send packets to this neighbor failed. If a Wireless Router cannot send a packet for 1 second, the attempt fails.

```
radio> talk show neighbors

TALK address: CC 5E B6 09
  Idle state, 3 active neighbors
  0 in transmit queue, 13 queue overflows.
  40 in header cache, sent last hello 12 seconds ago.
  Outgoing transaction ID: 0. Incoming ID: 0.

                              OCTETS            OUTPUT DIALOGS
Neighbor    State      Input    Output   Attempts   Failed  Last Data
---------   --------   -------  -------  --------   -------  --------
CC 5E B6 11  Good        10290     10188       74         0   0+00:19:03
CC 5E B6 39  Dead        56186     52670      490         3   0+00:18:57
CC 5E B6 81  Good          172       544        4         0   0+00:00:55
```

Display the Current Routing Table

If you are having problems sending data to another node, you can make sure that the routing table has the proper information for that node:

route

In the following example, make sure the IP address for the node you are trying to reach appears in the first column, labeled Dest:

```
radio> route

Dest           Len Interface Gateway     Metric  P Timer  Use
192.168.182.57  32  radio0                   0        0    412
192.168.182.17  32  radio0                   0        0     55
192.168.181.255 32  ether0                   1     P  0      0
192.168.182.129 29  radio0  192.168.182.57   1        0     58
192.168.180.0   24  ether0  192.168.181.3    1        0    870
192.168.181.0   24  ether0                   1        0      0
default          0  ether0  192.168.181.1    1     P  0   1380
```

Display Memory Statistics

If you are able to transfer data to another node, but performance seems slower than normal, check the memory usage:

memory

In the following example, check the number of free bytes; this number should be at least 50,000. Also note the number of failures.

```
radio> memory

Memory: 111312 bytes free (of 235456 total), 124144 used,
        24636 requests, 24357 frees, 279 outstanding, 0 failures,
        0 severe shortages, 0 minor shortages,
        0 invalid requests.

Mbufs:  21910 requests, 21869 frees, 41 outstanding, 0 failures,
        15458 grow requests (4523 required additional memory.)

Ibufs:  115699 requests, 115698 frees, 1 outstanding, 0 failures,
        1980 recovered, 0 destroyed,
        39 preallocated: 5 to host, 34 to interfaces.
```