DECrouter 2000
Problem Solving Guide

Order No. AA-JH80A-TN

SUPERSESSION/UPDATE INFORMATION:	This is a new manual
OPERATING SYSTEM AND VERSION:	VAX/VMS V4.4 or later MicroVMS V4.4 or later ULTRIX-32 V1.2 or later ULTRIX-32m V1.2 or later
SOFTWARE VERSION:	DECrouter 2000 V1.0

digital™

First Printing, October 1987

Please use the READER'S COMMENTS form on the last page of this document to give your critical evaluation of this manual. This will assist us in preparing future documentation.

Contents

4 Starting and Dumping the System

5 DECrouter 2000 Hardware Problems

C Using the Remote Console

D Adapter Cable Pin Signals

Index

Figures

Tables

How to Use This Manual

Manual Objectives

This manual describes how to solve problems that may occur with the DECrouter 2000.

Intended Audience

This manual is for network managers who are familiar with networking concepts and DECnet.

The manual assumes that you understand and have some experience of:

- Local Area Networks (LANs)
- Wide Area Networks (WANs)
- Installation of software products on VAX/VMS or ULTRIX systems
- DECnet-VAX (if using a VMS load host)
- DECnet-ULTRIX (if using an ULTRIX load host)

Structure of the Manual

This manual has six chapters and three appendixes.

Chapter 1 is an introduction to problem solving.

Chapter 2 explains how to diagnose problems that may occur when using the DECrouter 2000.

Chapter 3 describes the Loopback Testing facility which can be used to test the DECrouter 2000 and its communication links.

Chapter 4 describes how to start and dump the DECrouter 2000 system.

Chapter 5 describes how to solve hardware-related problems.

Chapter 6 explains how to report problems to DIGITAL.

Appendix A describes the Event Logging facility.

Appendix B describes the TRACE utility.

Appendix C explains how to use the remote console facility.

NOTE

Where *DECrouter* appears in the text, this refers to the DECrouter 2000 node.

Associated Manuals

For more information on the DECrouter 2000, refer to:

- *DECrouter 2000 Installation Procedures*
- *DECrouter 2000 Management Guide*

You may also find the *Routing and Networking Overview* useful in explaining routing concepts and terminology.

All three of these manuals are in the same binder as this manual.

The following provide information about the hardware used with the DECrouter 2000 software:

- *Installing the DEC MicroServer*
- *DEC MicroServer Systems Configuration Card*

If a VAX/VMS system is being used as a load host, you need to be familiar with the following manuals:

- *VAX/VMS Networking Manual*
- *VAX/VMS Network Control Program Reference Manual*

- *Guide to VAX/VMS Software Installation*

If an ULTRIX system is being used as a load host, you need to be familiar with the following manuals:

- *ULTRIX-32 System Manager's Guide* or the *ULTRIX-32m System Manager's Guide*
- The DECnet-ULTRIX documentation set, in particular the *DECnet-ULTRIX Guide to Network Management* and the *DECnet-ULTRIX User's and Programmer's Guide.*

Manual Conventions

<*xxx*>	This one- to three-character symbol indicates that you press a key on the terminal. For example: <RET> indicates the RETURN key <ESC> indicates the ESCAPE key
<CTRL/*x*>	This symbol indicates that you press the CTRL key at the same time as you press another key; for example, <CTRL/C>, <CTRL/Y>, and so on.
Red print	indicates commands and data that you enter.
Italics	indicate variable information.

1
Introduction

This manual contains information to help you solve problems that may occur when using the DECrouter 2000.

Problems can occur for a number of reasons, and Chapter 2, which is a diagnostic map, should help you determine where the problem is. In Chapter 2, left-hand pages identify problems, and right-hand pages diagnose those problems. When you have isolated the problem, Chapter 2 refers you to later chapters in which the various problem-solving tools and utilities are documented.

Before you start to solve problems, you should have the following information about the DECrouter 2000 available.

- DECrouter 2000 node name and node address
- Load host(s) node name and node address

This information, together with information on ports used on the DECrouter 2000, the speeds of lines and the Ethernet hardware address, can be found on the *DEC MicroServer Systems Configuration Card*. This card is attached to the front of the hardware unit.

1.1 Using NCP

During problem solving, you may need to enter NCP commands from a suitable node. This node should be in the same area and on the same Ethernet as your DECrouter 2000, for example your DECrouter 2000's load host.

NOTE

Your load host may be in a different area to your DECrouter 2000. If this is the case, issue the NCP commands from a node in the same area as the DECrouter 2000.

To start NCP on a VAX/VMS host node, issue the following command:

```
$ RUN SYS$SYSTEM:NCP
```

NCP will then display the following prompt:

```
NCP>
```

Enter the NCP command at this prompt. To exit from NCP, type EXIT or <CTRL/Z> at the NCP prompt.

To start NCP on an ULTRIX host node, issue the following command:

```
% ncp
```

ncp will then display the following prompt:

```
ncp>
```

Enter the NCP command at this prompt. To exit from NCP, type exit, quit or <CTRL/D> at the ncp prompt.

NCP commands to be executed on the DECrouter 2000 are issued at a suitable node (for example, the load host) and NCP is told to execute the command on the DECrouter 2000 node.

If you only want to issue one NCP command to the DECrouter 2000, use the TELL prefix:

```
NCP> TELL DECrouter command
```

If you want to issue a series of NCP commands on the DECrouter 2000, first set the executor node to be the DECrouter 2000 as follows:

```
NCP> SET EXECUTOR NODE DECrouter
```

Then issue the NCP commands as required. Refer to the *DECrouter 2000 Management Guide* for full details of the NCP commands you can use with this product. If you have set up a username and password to control access to the DECrouter 2000, you will need to include the username and password when setting executor to the DECrouter 2000. For example:

```
NCP> SET EXECUTOR NODE DECrouter USER username PASSWORD password
```

where *username* and *password* are the access control information.

1.2 Using Event Logging

You should also set up event logging in order to monitor the events generated by both the load host and the DECrouter 2000. Events generated by the load host indicate the progress of loading and dumping operations. Events generated by the DECrouter 2000 indicate failures to forward data, changes in the reachability of nodes as well as problems with communication lines.

To permit a VAX/VMS host node to receive events, issue the following two commands:

```
NCP> SET LOGGING MONITOR STATE ON
$ REPLY/ENABLE=NETWORK
```

These commands turn on logging to OPCOM, the default VAX/VMS monitor. Refer to the *VAX/VMS Networking Manual* for full details of event logging. Note that OPCOM may shorten event messages and may not show the values of counters when these are logged by the DECrouter 2000.

If you want to receive the complete text of messages, you can use the LOGGING CONSOLE component. Refer to the *VAX/VMS Networking Manual*, the *VAX/VMS Network Control Program Reference Manual* and the *DECrouter 2000 Management Guide* for details.

To permit an ULTRIX host node to receive events generated by the DECrouter 2000, issue the following command:

```
ncp> set logging monitor state on
```

This command turns on logging to your local console, the default ULTRIX monitor. Refer to the *DECnet-ULTRIX Guide to Network Management*, the *DECnet-ULTRIX User's and Programmer's Guide*, and the *DECrouter 2000 Management Guide* for full details of the SET LOGGING command.

Unless you specify otherwise, only events related to the permanent database are generated by the DECrouter 2000 and are automatically logged to your load host node. If you want additional events to be logged, you need to specify a sink node to receive the messages.

Issue the following commands to set up sink nodes:

```
NCP> SET EXECUTOR NODE DECrouter
```

where *DECrouter* is the name of the DECrouter 2000, followed by:

```
NCP> SET LOGGING MONITOR SINK NODE node-id KNOWN EVENTS
```

where *node-id* is the node name or node address of the sink node which is to receive the event messages. Events will be logged and displayed on node *node-id* as they occur.

Refer to Appendix A of this manual for full details of event logging, including a list of events logged by the DECrouter 2000.

1.3 Using the SPR System

If you fail to solve the problem by using the diagnostic map and the other problem solving tools in this manual, then you should submit a Software Performance Report (SPR). Chapter 6 gives you full details of how to submit an SPR to DIGITAL.

2
Diagnosing Problems

This chapter explains how to diagnose problems that may occur when using the DECrouter 2000. Isolate the problem by working through each section. If a section does not apply to your problem for example, your DECrouter 2000 is reachable - pass on to the next section that applies to your situation.

If you cannot solve the DECrouter 2000 problem by using the methods suggested in this manual, submit an SPR (Software Performance Report) to DIGITAL. Refer to Chapter 6 for details of how to submit an SPR.

The chapter is designed as a diagnostic map: questions about the problem are on the left-hand pages, and notes relating to the questions are on the right-hand pages.

To diagnose the problem, first find the section relevant to your problem, and then follow the questions down the left-hand pages. When the answer to a question points you to a note, read the specified note which is on the right-hand page. Due to the lay-out being used for this chapter, there are blank pages within the chapter.

The notes give information that will either solve the problem, or suggest which diagnostic tool you ought to use to get further information about the problem.

The chapter is divided into sections as follows:

- DECrouter 2000 unreachable: this can be divided into the following problems:
  - DECrouter 2000 will not load - refer to Section 2.1.1.
  - The software loads correctly, but you cannot communicate with the DECrouter 2000 - refer to Section 2.1.2.

- DECrouter 2000 reachable: this can be divided into the following sections:
  - DECrouter 2000 reachable but cannot communicate with other local nodes - refer to Section 2.2.
  - Circuits fail to come up - refer to Section 2.2.1.
  - Circuits fail due to errors - refer to Section 2.2.2.
  - Remote nodes are unreachable - refer to Section 2.2.3.
- A section describing the counters used for circuits, lines and nodes.

The first step in problem solving is to see if you can communicate with the DECrouter 2000.

Log on to a suitable node, such as the DECrouter 2000's load host, and issue the following command:

```
NCP> TELL DECrouter SHOW EXECUTOR SUMMARY
```

where *DECrouter* is the name of the DECrouter 2000. If this command succeeds, go to Section 2.2. If you cannot communicate with the DECrouter 2000, start problem solving with Section 2.1.

This page deliberately left blank

2.1 DECrouter 2000 Unreachable

This section assumes that you cannot communicate with the DECrouter 2000 from a suitable load host. Check that the correct software is loaded on the DECrouter 2000 and that the hardware is working correctly.

2.1.1 DECrouter 2000 will not Load

Before you begin, make sure that logging of DECnet events 0.3 (Automatic Service) and 0.7 (Aborted Service Request) are enabled on the load host. These messages indicate the progress of a load operation and may provide additional information.

1. Check that the hardware unit is working. Make sure the unit is plugged into the AC supply, that the power indicator is lit (a decimal point on the display) and that the fans are turning.

   Is the DECrouter 2000 powered up and running? ➡ NO See Note 1

   ⬇

   YES

2. Look at the display on the DECrouter 2000 and refer to Section 5.1. Has the DECrouter 2000 successfully completed its self-test procedure? ➡ NO See Note 2

   ⬇

   YES

3. Does the display indicate "E"? ➡ YES See Note 3

   ⬇

   NO

Note 1

If the indicator is not lit or the fans do not turn, refer to Chapter 5.

Note 2

Look at the display on the DECrouter 2000. Check the display against the information in Section 5.1.

Note 3

1. Make sure all connections between the DECrouter 2000 and the Ethernet are securely in place. In particular, check the connection on the DECrouter 2000. Correct any loose connections and try to load the system again. Refer to Chapter 4 for details of how to load the system.
2. Check for a damaged cable between the DECrouter 2000 and the Ethernet transceiver. Replace if necessary and try to boot the system again. Refer to Chapter 4 for details on booting the system.

4. Check that the Ethernet is working correctly by logging in to a load host and entering:

   ```
   NCP> LOOP NODE other-node
   ```

   where *other-node* is the name of a DECnet node on the Ethernet that is beyond the DECrouter 2000. Refer to Chapter 3 for full details on using the LOOP command.

   Does this command fail? ➡ YES See Note 4

   ⬇

   NO

5. Use the *DEC MicroServer Systems Configuration Card* to find out the load hosts. Then try to log in to each one in turn.

   Is the load host running? ➡ NO See Note 5

   ⬇

   YES

6. Is the load host connected to the Ethernet? ➡ NO See Note 6

   ⬇

   YES

7. Check that the load host has a database entry for the DECrouter 2000. For a VAX/VMS load host, issue the following NCP command:

   ```
   NCP> SHOW NODE DECrouter CHARACTERISTICS
   ```

   Check that the load file is listed.

   For an ULTRIX load host, issue the following command:

   ```
   getnode DECrouter
   ```

   and check that the characteristics are correct.

   Is the message "No information available" displayed? ➡ YES See Note 7

   ⬇

   NO

Note 4

You have an Ethernet problem. Take the DECrouter 2000 off the Ethernet and follow your Ethernet fault finding routine. If you still have problems, contact DIGITAL Field Service to repair the hardware unit.

Note 5

If no load hosts are available, wait until one is before reloading the DECrouter 2000. Refer to Chapter 4 for details of how to load the system.

Note 6

The DECrouter 2000 and all its load hosts must be on the same Ethernet. Check that each load host has a physical connection to the Ethernet that the DECrouter 2000 is connected to. Also check all the cables connecting each load host to the Ethernet. Use LOOP NODE commands to check that the load host can communicate with another node by using the Ethernet. Refer to Chapter 3 for details of loopback testing.

Note 7

This message means that this system has not been set up as a load host. Establish the system as a load host following the instructions in the *DECrouter 2000 Installation Procedures*.

8. Check the following database items:

   - Load file specification, which the installation procedure sets to:

     ```
     SYS$SYSROOT:[DECSERVER]ROU010.SYS
     ```

   - Hardware address, as specified on the *DEC MicroServer Systems Configuration Card*
   - DECnet address of the DECrouter 2000
   - Service circuit

   Are these items correct? ➡ NO See Note 8

   ⬇

   YES

9. Is the software image file (ROU010.SYS) present in the directory shown in the load file specification found in the step above? ➡ NO See Note 9

   ⬇

   YES

10. Is the configuration permanent database (ROU*node-name*.SYS) located in a directory indicated by MOM$LOAD? ➡ NO See Note 10

    ⬇

    YES

11. Issue the command:

    ```
    $  SHOW LOGICAL MOM$LOAD/FULL
    ```

    Check that the translation includes the directory in which the configuration file (ROU*nodename*.SYS) is located, and that the logical name is defined in the system table (LNM$SYSTEM_TABLE) with executive mode attributes.

Carry out steps 5 to 11 for any other hosts that can load the DECrouter 2000, then reload the DECrouter 2000 software. Refer to Chapter 4 for details of how to load the system.

Note 8

Correct the information if necessary. You may need to run ROUCONFIG to correct the load host database. Refer to the *DECrouter 2000 Installation Procedures* for details of this procedure.

Note 9

If the software image file is missing, the software has not been installed correctly. Reinstall the software, as described in the *DECrouter 2000 Installation Procedures.*

Note 10

If the configuration permanent database is missing, the DECrouter 2000 has not been configured on this system. Configure the DECrouter 2000, as described in the *DECrouter 2000 Installation Procedures.*

2.1.2 DECrouter 2000 Fails to Respond

This section assumes that you appear to have the correct software on to the DECrouter 2000, but that there are problems communicating with the DECrouter 2000 from a load host.

1. Issue the following command from the load host to test the Ethernet connection between the load host and the DECrouter 2000. Refer to Chapter 3 for full details of loopback testing.

   ```
   NCP>LOOP CIRCUIT Ethernet-circuit-name NODE DECrouter
   ```

   Does this command work? ➡ NO See Note 1

   ⬇

   YES

   Assuming that the load host is in the same area and on the same Ethernet as the DECrouter 2000, issue the following command:

   ```
   NCP>SHOW CIRCUIT Ethernet-circuit-name CHARACTERISTICS
   ```

   Is the DECrouter 2000 the designated router for this Ethernet circuit? ➡ NO See Note 2

   ⬇

   YES

   Check that the node number of the load host is not larger than the MAXIMUM ADDRESS value in the DECrouter 2000's permanent configuration database and that the BUFFER SIZE parameter is the same for both the load host and DECrouter 2000 nodes.

   Is the above information correct? ➡ NO See Note 3

   ⬇

   YES

Note 1

Check the Ethernet connection - refer to Note 3 in Section 2.1.

Note 2

Determine why the designated router cannot communicate with the DECrouter 2000 using the steps given in Section 2.2.3.

Note 3

Correct the parameters as required, and try again. If you still cannot communicate with the DECrouter 2000, take a DUMP from the DECrouter 2000 and submit the dump with an SPR. Refer to Chapter 5 for details of taking a DUMP and to Chapter 6 for details of how to submit an SPR.

4. Ensure that the number of routers on the Ethernet does not exceed the value for MAXIMUM BROADCAST ROUTERS in the DECrouter 2000's permanent configuration database. Ensure that the number of end nodes on the Ethernet does not exceed the value for MAXIMUM BROADCAST NONROUTERS in the DECrouter 2000's permanent configuration database.

   Are the values for the above EXECUTOR parameters correct? ➡ NO See Note 4

   ⬇

   YES

5. Is the event 225.0 being logged at the load host? ➡ YES See Note 5

   ⬇

   NO

6. If you still have problems, take a DUMP from the DECrouter 2000 and submit the dump with an SPR. Refer to Chapter 5 for details of taking a DUMP and to Chapter 6 for details of how to submit an SPR.

Note 4

If the number of routing and end nodes on your Ethernet exceed the values of MAXIMUM BROADCAST ROUTERS and MAXIMUM BROADCAST NONROUTERS, increase the parameter values accordingly.

Note 5

If event 225.0 is logged, there are problems with the DECrouter 2000's configuration. Re-configure the DECrouter 2000's permanent database by running ROUPERM. Refer to the *DECrouter 2000 Management Guide* for details.

2.2 DECrouter 2000 Reachable

This section assumes that you can successfully execute the following command:

```
NCP> TELL DECrouter SHOW EXECUTOR SUMMARY
```

but that the DECrouter 2000 is not working correctly. The problem may be due to the DECrouter 2000 configuration.

You should first check the executor characteristics of the DECrouter 2000. This allows you to check that the correct software has been loaded on to the DECrouter 2000.

Issue the following command from a suitable node, such as the DECrouter 2000's load host:

```
NCP> SET EXECUTOR NODE node-name
```

where *node-name* is the name of the DECrouter 2000.

1. Issue the following command:

   ```
   NCP> SHOW EXECUTOR CHARACTERISTICS
   ```

   The display produced will be similar to that given overleaf:

This page deliberately left blank

```
Node Volatile Characteristics as of  4-AUG-1987 15:41:15

Executor node = 12.99 (WOMBAT)

Identification            = DECrouter 2000 V1.0 BL4
Management version        = V4.2.0
Host                      = 12.98 (KANGA)
NSP version               = V4.1.0
Maximum links             = 512
Delay factor              = 80
Delay weight              = 5
Inactivity timer          = 60
Retransmit factor         = 10
Routing version           = V2.0.0
Type                      = routing IV
Routing timer             = 600
Broadcast routing timer   = 40
Maximum address           = 1023
Maximum cost              = 1022
Maximum hops              = 30
Maximum visits            = 63
Max broadcast nonrouters  = 1022
Max broadcast routers     = 32
Maximum buffers           = 127
Buffer size               = 576
Segment buffer size       = 576
```

The list of characteristics displayed includes the name of the load host that was used to load the DECrouter 2000 software.

Are the characteristics correct? ➡ NO See Note 1

⬇

YES

If you still have problems, you should check that the circuits are working correctly. Refer to Section 2.2.1 for details.

Note 1

If none of the characteristics displayed is correct for the DECrouter 2000, then the DECrouter 2000 permanent database has not been installed, or there is a problem with the load host.

Reinstal the software on to the load host - refer to the *DECrouter 2000 Installation Procedures for details*. Then reload the DECrouter 2000 - refer to Chapter 4 for details of how to load the system.

If some of the characteristics are incorrect, the permanent configuration database has not been loaded. Run the configurator program ROUPERM to modify the database. Refer to the *DECrouter 2000 Management Guide* for details. Then reload the software - refer to Chapter 4 for details.

Check from the load host console that the permanent configuration database has loaded.

You should then check that the DECrouter 2000 has been configured correctly. Check **ALL** the routing parameters, making sure that the node number of the DECrouter 2000 is within the range for the area, and that the value for MAXIMUM BROADCAST ROUTERS is not too small.

Table 2-1 shows the DECrouter 2000 EXECUTOR parameters, their default values and the range of values you can use.

This page deliberately left blank

Table 2-1: EXECUTOR Parameters

Parameter	Default Value	Range
AREA MAXIMUM COST (TYPE=AREA)	1022	1-1022
AREA MAXIMUM HOPS (TYPE=AREA)	30	1-30
BROADCAST ROUTING TIMER	40	1-65535
BUFFER SIZE	576	246-5000
COUNTER TIMER	CLEARED	1-65535, CLEARED
DELAY FACTOR	80	1-255
DELAY WEIGHT	5	1-255
IDENTIFICATION	Software Identification	1-32 characters, CLEARED
INACTIVITY TIMER	60	1-65535
MAXIMUM ADDRESS	1023	1-1023
MAXIMUM AREA (TYPE=AREA)	63	1-63
MAXIMUM BROADCAST NONROUTERS	1022	1-1022
MAXIMUM BROADCAST ROUTERS	32	1-1022
MAXIMUM BUFFERS	500	50-1000
MAXIMUM COST	1022	1-1022
MAXIMUM HOPS	30	1-30
MAXIMUM PATH SPLITS	1	1-4
MAXIMUM VISITS	63	MAXIMUM HOPS - 63
RETRANSMIT FACTOR	10	1-65535
ROUTING TIMER	600	1-65535
SEGMENT BUFFER SIZE	576	246-BUFFER SIZE
TYPE	ROUTING IV	AREA, ROUTING IV

2.2.1 Circuits Fail to come up

This section assumes that the correct software has been installed and the EXECUTOR TYPE is either ROUTING IV or AREA. The following describes how to check that the circuits are working correctly.

1. The first step is to determine the status of the circuits associated with the DECrouter 2000. Issuing the following commands:

```
NCP> SET EXECUTOR NODE DECrouter
NCP> SHOW KNOWN CIRCUITS SUMMARY
```

will produce a display similar to the following:

```
Known Circuit Volatile Summary as of  4-AUG-1987 15:44:32

Circuit          State          Loopback      Adjacent
                                  Name          Node

CO               on                           12.301 (KANGA)
C1               off
ETHERNET         on                           12.16
ETHERNET         on                           12.50 (ROO)
ETHERNET         on                           12.294 (POSSUM)
ETHERNET         on                           12.25 (PLATY)
```

Are all the circuits set to STATE ON? ➡ NO See Note 1

⬇

YES

Note 1

Set all the circuits being used (and their associated lines) to STATE ON. Check that the cables and modems are working correctly. If necessary check the circuits, lines, cables and modems at the remote node.

2. Issue the following command to determine the characteristics for each circuit.

```
NCP> SHOW KNOWN CIRCUIT CHARACTERISTICS
```

The display will be similar to the following:

```
Known Circuit Volatile Characteristics as of  4-AUG-1987 15:42:29

Circuit = CO

Adjacent node              = 12.301 (KANGA)
Block size                 = 576
Cost                       = 5
Hello timer                = 15
Listen timer               = 30
Line                       = LO
Type                       = DDCMP point
Babble timer               = 6000
Transmit timer             = 0
Maximum transmits          = 4
Verification               = disabled


Circuit = ETHERNET

Adjacent node              = 12.255 (ROO)
Designated router          = 12.255 (ROO)
Block size                 = 1498
Cost                       = 4
Maximum routers allowed    = 32
Router priority            = 32
Hello timer                = 15
Listen timer               = 45
Line                       = ETHERNET
Type                       = Ethernet
```

Examine the parameter values displayed.

Have the circuits been correctly configured? ➡ NO See Note 2

⬇

YES

Note 2

Modify the characteristics for each circuit as necessary. Make sure that the circuit type is correct for the line that the circuit is associated with and that the verification is correct. The verification should be the same at each end of the circuit. If the verification is set to ENABLED, check that the correct transmit and receive passwords are being used.

3. Check the counters for the circuit, by issuing the following command:

```
NCP> SHOW KNOWN CIRCUIT COUNTERS
```

The display will list the status of the counters for each circuit; for example:

```
Known Circuit Counters as of  4-AUG-1987 15:47:09

Circuit = CO

        6497  Seconds since last zeroed
        1953  Terminating packets received
        1167  Originating packets sent
           0  Terminating congestion loss
           0  Corruption loss
           0  Transit packets received
           0  Transit packets sent
           0  Transit congestion loss
           1  Circuit down
           0  Initialization failure
      200531  Bytes received
       44768  Bytes sent
        1953  Data blocks received
        1167  Data blocks sent
           0  Data errors inbound
           0  Data errors outbound
           0  Remote reply timeouts
           0  Local reply timeouts
           0  Remote buffer errors
           0  Local buffer errors
           0  Selection intervals elapsed
           0  Selection timeouts
```

This page deliberately left blank

```
Circuit = ETHERNET

        6601  Seconds since last zeroed
       44818  Terminating packets received
        2447  Originating packets sent
           0  Terminating congestion loss
           0  Corruption loss
          54  Transit packets received
          54  Transit packets sent
           0  Transit congestion loss
           0  Circuit down
           0  Initialization failure
           0  User buffer unavailable
     4992970  Bytes received
      480297  Bytes sent
       44885  Data blocks received
        2501  Data blocks sent
```

Are there any unexpected values for the counters? ➡ YES See Note 3

⬇

NO

Note 3

The circuit down value informs you of the number of times the circuit has been up and down. The initialization failure value indicates that there may have been a verification failure. Verification failure may be due to incorrect values being used for buffer size or a password verification failure. Data error values indicate that there may be a noise problem with the line. Check the events logged at the sink node, and modify the circuit parameters if required by running ROUPERM. Refer to the *DECrouter 2000 Management Guide* for details of this procedure.

Refer to Section 2.3.1 for full details of the circuit counters.

4. Once you are sure that the circuits are set up, you should check the lines. Determine the status of the lines associated with the DECrouter 2000 by issuing the following command:

```
NCP> SHOW KNOWN LINES
```

The display will be similar to the following:

```
Known Line Volatile Summary as of  4-AUG-1987 15:44:43

   Line                  State

  LO                     on
  ETHERNET               on
```

This will provide information on the status of the lines being used on the DECrouter 2000.

Are all the lines set to STATE ON? ➡ NO See Note 4

⬇

YES

Note 4

Set all the lines being used to STATE ON.

5. Check that the lines have been configured correctly by issuing the following command:

```
NCP> SHOW KNOWN LINE CHARACTERISTICS
```

The display will be similar to the following:

```
Known Line Volatile Characteristics as of  4-AUG-1987 15:42:55

Line = LO

Device                    = SYN-0
Receive buffers           = 8
Controller                = normal
Duplex                    = full
Protocol                  = DDCMP point
Clock                     = external
Service timer             = 20000
Retransmit timer          = 3000

Line = ETHERNET

Device                    = LNA-0
Receive buffers           = 64
Controller                = normal
Protocol                  = Ethernet
Hardware address          = 08-00-2B-03-71-D8
```

Check that the given protocol for each line is correct and that the parameter values are correct for the configuration. Check that the duplex, clock and controller settings are correct.

Are the lines correctly configured? ➡ NO See Note 5

⬇

YES

Note 5

Check the line configuration information on the load host and correct as necessary. You may need to run the procedure ROUPERM - refer to the *DECrouter 2000 Management Guide* for details.

6. Check the line counters by issuing the following command:

```
NCP> SHOW KNOWN LINE COUNTERS
```

which will display the status of the counters for each line. The display will be similar to the following:

```
Known Line Counters as of  4-AUG-1987 15:47:20

Line = L0

        6522  Seconds since last zeroed
           0  Data errors inbound
           0  Remote process errors
           0  Local process errors

Line = ETHERNET

        6613  Seconds since last zeroed
       45028  Data blocks received
       44533  Multicast blocks received
           2  Receive failure, including:
                Framing error
     5427951  Bytes received
     5226433  Multicast bytes received
        2531  Data blocks sent
        2032  Multicast blocks sent
           1  Blocks sent, multiple collision
           2  Blocks sent, single collision
           1  Blocks sent, initially deferred
      524375  Bytes sent
      495834  Multicast bytes sent
           0  Send failure
           0  Collision detect check failure
           0  Unrecognized frame destination
           0  System buffer unavailable
           0  User buffer unavailable
```

Are there any unexpected values for the counters? ➡ YES See Note 6

⬇

NO

If the DECrouter 2000 is still not working correctly, check that the circuits are not failing due to errors. Refer to Section 2.2.2 for details.

Note 6

The line counters collect information about the number of bytes and data blocks sent and received, local and remote process errors, and the time that has elapsed since the counters were last zeroed. By checking the counters, you may be able to detect certain hardware errors, such as problems with the Ethernet or the modems.

Refer to Section 2.3.2 for full details of the line counters.

2.2.2 Circuits Fail due to Errors

This section covers problems you may have with the circuits.

1. Check the events logged at the node(s) adjacent to the DECrouter 2000. Refer to Appendix A for full details of event logging and a description of the events. You can use the event logging facility to monitor network events.

   Are events being logged at the load host? ➡ NO See Note 1

   ⬇

   YES

2. Do the events logged indicate any problems? ➡ YES See Note 2

   ⬇

   NO

3. If you still have problems, use loopback testing to check that the DDCMP circuit can initialize. Refer to Chapter 3 for full details of loopback testing. Issue the following commands:

   ```
   NCP> SET CIRCUIT circuit-name STATE OFF
   NCP> SET VERIFICATION DISABLED
   NCP> SET LINE line-name CONTROLLER LOOPBACK STATE ON
   NCP> SET CIRCUIT circuit-name STATE ON
   ```

   Does the circuit come up? ➡ NO See Note 3

   ⬇

   YES

4. Repeat the above test with the controller set to NORMAL, and use the loopback connector on the synchronous device. Note that you must use a supported 50-way loopback connector and that you must turn the line on AFTER fitting the connector. Refer to Chapter 3 for full details.

   Does this test fail? ➡ YES See Note 4

   ⬇

   NO

Note 1

If events are not being logged, check that event logging has been turned on at the logging sink node; by default this is the load host. If alternative sink nodes have been specified, check that event logging has been turned on at these nodes and that each node can receive the events.

Check that the logging sink nodes are reachable from the DECrouter 2000 by issuing the following command for each sink node:

```
NCP> TELL DECrouter LOOP NODE event-log-host
```

where *event-log-host* is the name of the node that you want to use to receive the events.

Check that the logging host has sufficient system resources to receive the events.

Note 2

If events occur regularly which state that there is a problem in the DECrouter 2000 passing information between nodes, check the component that the message relates to. Refer to Appendix A for full details of event logging.

Note 3

If the circuit fails to initialize, there is probably a hardware fault. Check the hardware using the loopback tests described in Section 5.3. The circuit is up if it is in STATE ON with no substate. Check for circuit up event messages at the event sink. If there is not a hardware fault, and the circuit still fails to work, submit an SPR - refer to Chapter 6 for details.

Note 4

There is a hardware error. Refer to Chapter 5 for details of how to solve hardware problems.

5. Repeat the loopback test using the local modem.

   Does this test fail? ➡ YES See Note 5

   ⬇

   NO

6. Repeat the loopback test using the remote modem.

   Does this test fail? ➡ YES See Note 6

   ⬇

   NO

If both the modems are working, check that the remote node is working. Refer to Section 2.2.3 for details.

Note 5

If the local modem loopback test fails, but the controller loopback test worked, check the cables to the modem and the local modem.

Note 6

If the local modem is working correctly, but the remote modem loopback test fails, there is a fault on the line between the two modems or in one of the modems.

2.2.3 Remote Nodes Unreachable

This section assumes that the DECrouter 2000 is working correctly, but that you cannot communicate with remote nodes.

1. First check that the DECrouter 2000 has been correctly configured. See Section 2.2 for details.

   Are the DECrouter 2000's characteristics correct? ➡ NO See Note 1

   ⬇

   YES

2. Is the DECrouter 2000 the only router in the area? ➡ NO See Note 2

   ⬇

   YES

3. Is the unreachable node on the same Ethernet as the DECrouter 2000? ➡ YES See Note 3

   ⬇

   NO

4. Is the unreachable node in the same area as the DECrouter 2000? ➡ YES See Note 4

   ⬇

   NO

Note 1

Refer to Note 1 in Section 2.2 for details of how to solve problems with the DECrouter 2000's configuration.

When you check the characteristics for the DECrouter 2000, examine the values for MAXIMUM HOPS, MAXIMUM COST, MAXIMUM ADDRESS and MAXIMUM VISITS. If these values are set too low, the DECrouter 2000 will not be able to communicate with remote nodes. If the DECrouter 2000 is a level 2 (AREA) router, check the value for MAXIMUM AREA, AREA MAXIMUM COST and AREA MAXIMUM HOPS. Refer to the *DECrouter 2000 Management Guide* for full details of these parameters.

Check the event logging facility for events in the range 4.1 to 4.5. Refer to Appendix A for details of the event logging procedure and what the events generated mean.

Note 2

Check that all the routers in the area have been configured correctly. Check the event logging facility for events in the range 4.1 to 4.5 which may indicate what the problem is. Refer to Appendix A for details of the event logging procedure and what the events generated mean.

Note 3

Check that the total number of routers and end nodes on the Ethernet does not exceed the DECrouter 2000's MAXIMUM BROADCAST ROUTERS and MAXIMUM BROADCAST NONROUTERS parameter value respectively. Refer to Table 2-1 for details of these parameter values. Check that these parameter values are correct for all the Ethernet routers in the same area as the DECrouter 2000.

Note 4

Check the path between the DECrouter 2000 and the remote node by using software and hardware loopback tests. Refer to Chapter 3 for details of software loopback tests; refer Section 5.3 for details of hardware loopback tests.

5. Check that the DECrouter 2000 can communicate with the remote area. Issue the following command:

   ```
   NCP> SHOW NODE node-name STATUS
   ```

   Follow the path to the next node from one node to the next until you reach an area router. Then issue the following command:

   ```
   NCP> SHOW KNOWN AREA STATUS
   ```

   to determine if the area is reachable. Repeat the above commands from the remote node if necessary.

   Is the remote area reachable? ➡ NO See Note 5

   ⬇

   YES

If you still cannot communicate with the remote node, there may be a problem with the remote node. Check with the system/network manager for the remote system that the node is running. If the node is running, ask the system/network manager to perform loopback tests from the remote node towards the DECrouter 2000, so the break in the path between the two nodes can be determined.

Note 5

The DECrouter 2000 cannot communicate with the remote node for one of two reasons:

- There is no path available from the nearest level 2 router to the remote area.
- There is no path between the remote level 2 router and the remote node.

You should check the path from the DECrouter 2000 to the remote node by using the software and hardware loopback testing procedures. Refer to Chapter 3 for details of the software loopback tests; refer to Section 5.3 for details of the hardware loopback tests.

2.3 Counters

2.3.1 Circuit Counters

The following counters are common to all circuits:

- Bytes Received

  This counter records data bytes received over the circuit (not including retransmissions).

- Bytes Sent

  This counter records data bytes transmitted over the circuit (not including retransmissions).

- Circuit Down

  Number of times the circuit failed for one of the following reasons:

  - Data link synchronization lost
  - Data link threshold error detected

- Data Blocks Received

  This counter records messages received over the circuit (not including retransmissions).

- Data Blocks Sent

  This counter records messages transmitted over the circuit (not including retransmissions).

- Initialization Failure

  Number of times circuit initialization failed for one of the following reasons:

  - Verification message not received
  - Data link synchronization lost
  - Data link threshold error detected
  - Version skew
  - Adjacent node address too large
  - Adjacent node buffer size too small
  - Invalid verification seed value in received initialization message

- Phase III node did not supply an initialization password
- Unexpected message
- Local node not an area router and adjacent node in different area

- Originating Packets Sent

  Packets generated at this node sent to an adjacent node.

- Seconds Since Last Zeroed

  The elapsed time, in seconds, since the set of circuit counters was last zeroed.

- Terminating Congestion Loss

  Packets discarded because local software was not able to process them fast enough.

- Terminating Packets Received

  Packets received destined for this node.

- Transit Congestion Loss

  Packets intended for forwarding that were discarded because of a local shortage of buffers or full queues.

- Transit Packets Received

  Packets received from adjacent nodes destined for another node in the network.

- Transit Packets Sent

  Packets received from adjacent nodes forwarded to other destinations.

The following counters are kept for DDCMP circuits only:

- Data Errors Inbound

  This counter records occurrences that normally result from data errors on the communications channel inbound to this station. The specific errors associated with this counter are:

  - NAKs Sent Data Field Block Check Error
  - NAKs Sent REP Response

- Data Errors Outbound

  This counter records occurrences which normally result from data errors on the communications channel outbound from this station. The specific errors associated with this counter are:

  - NAKs Received Header Block Check Error
  - NAKs Received Data Field Block Check Error
  - NAKs Received REP Response

- Local Buffer Errors

  This counter records occurrences which normally result from the failure of the user of DDCMP at this station to coordinate properly the supply of receive buffers at this station to data messages supplied by the remote user. The specific errors associated with this counter are:

  - NAKs Sent Buffer Unavailable
  - NAKs Sent Buffer Too Small

- Local Reply Timeouts

  This counter records occurrences which normally result from either:

  1. The loss of communications between stations while this station has data to transmit, or
  2. The choice of an inappropriate value for this station's retransmit timer.

- Remote Buffer Errors

  This counter records occurrences which normally result from the failure of the user of DDCMP at the remote station to properly coordinate the supply of receive buffers at that station to data messages supplied by the user of DDCMP at this station. The specific errors associated with this counter are:

  - NAKs Received Buffer Unavailable
  - NAKs Received Buffer Too Small

- Remote Reply Timeouts

  This counter records occurrences which normally result from either:

  1. The loss of communication between stations while the remote station has data to transmit, or

2. The choice of an inappropriate value for the remote station's retransmit timer.

3. Selection Intervals Elapsed

   This counter is used by half-duplex stations. It can be used as a statistical base when evaluating the number of selection time-outs. This counter records the number of times this station selects the other station.

4. Selection Timeouts

   This counter records occurrences on a half-duplex line which normally result from:

   1. Loss of communication with a remote station, or
   2. Data errors on the communications channel to or from that station, or
   3. The choice of an inappropriate value for this station's Select Timer.

   This counter is maintained only by point-to-point half-duplex stations; it is not maintained by full-duplex point-to-point stations or by multipoint tributary stations. The specific errors associated with this counter are:

   - No Reply To Select - used to record selection intervals in which no transmission was received from the remote station and in which no attempt to transmit could be detected. Specifically, it records expiration of the select timer without receipt of a valid control message or a valid header to a data or maintenance message.
   - Incomplete Reply To Select - used to record selection intervals which were not properly terminated by receipt of a message header with SELECT flag on, during which a transmission was received from the tributary. Specifically, it records expiration of the select timer preceded by

     1. Receipt of a valid control message, or
     2. Receipt of a valid header to a data or maintenance message.

2.3.2 Line Counters

The following counter is kept for all lines:

- Seconds Since Last Zeroed

  The elapsed time, in seconds, since the set of line counters was last zeroed.

The following counters are kept for DDCMP lines:

- Data Errors Inbound

  This counter records occurrences that normally result from data errors on the communications channel inbound to this station. The specific errors associated with this counter are:

  - Header Block Check Errors

- Local Station Errors (Local Process Errors)

  This counter records occurrences most often caused by a fault in this station or by an undetected data error on the channel outbound from this station. The specific error associated with this counter is:

  - NAKs Received Message Header Format Errors

- Remote Station Errors (Remote Process Errors)

  This counter records occurrences most often caused by a fault in a remote station or by an undetected data error on the channel inbound to this station. The specific errors associated with this counter are:

  1. NAKs Received Receive Overrun
  2. NAKs Sent Message Header Format Error
  3. Streaming Tributaries - only used by point-to-point half-duplex stations. It is caused by exceeding the maximum transmission interval without releasing the channel.

The counters listed below are kept for the Ethernet line. Unless otherwise stated, all counters include both normal and multicast traffic. Furthermore, they include information for all protocol types. Blocks received and bytes received counters do not include frames received with errors.

- Blocks Sent, Initially Deferred

  The total number of times that a frame transmission was deferred on its first transmission attempt.

  In conjunction with total frames sent, measures Ethernet contention with no collisions.

- Blocks Sent, Multiple Collisions

  The total number of times that a frame was successfully transmitted on the third or later attempt after normal collisions on previous attempts.

In conjunction with total frames sent, measures Ethernet contention at a level where there are collisions and the backoff algorithm no longer operates efficiently.

- Blocks Sent, Single Collision

  The total number of times that a frame was successfully transmitted on the second attempt after a normal collision on the first attempt.

  In conjunction with total frames sent, measures Ethernet contention at a level where there are collisions but the backoff algorithm still operates efficiently.

NOTE

No single frame is counted in more than one of the above three counters.

- Bytes Received

  The total number of user data bytes successfully received. This does not include Ethernet data link headers. This number is the number of bytes in the Ethernet data field, which includes any padding or length fields when they are enabled. These are bytes from frames that passed hardware filtering.

  When the number of frames received is used to calculate protocol overhead, the overhead plus bytes received provides a measurement of the amount of Ethernet bandwidth (over time) consumed by frames addressed to the local system.

- Bytes Sent

  The total number of user data bytes successfully transmitted. This does not include Ethernet data link headers or data link generated retransmissions. This number is the number of bytes in the Ethernet data field, which includes any padding or length fields when they are enabled.

  When the number of frames sent is used to calculate protocol overhead, the overhead plus bytes sent provides a measurement of the amount of Ethernet bandwidth (over time) consumed by frames sent by the local system.

- Collision Detect Check Failure

  The approximate number of times that collision detect was not sensed after a transmission.

  If this counter contains a number roughly equal to the number of frames sent, either the collision detect circuitry is not working correctly or the test signal is not implemented.

- Data Blocks Received

  The total number of frames successfully received. These are frames that passed hardware filtering.

  Provides a gross measurement of incoming Ethernet usage by the local system. Provides information used to determine the ratio of the error counters to successful transmits.

- Data Blocks Sent

  The total number of frames successfully transmitted. This does not include data link generated retransmissions.

  Provides a gross measurement of outgoing Ethernet usage by the local system. Provides information used to determine the ratio of the error counters to successful transmits.

- Multicast Blocks Received

  The total number of multicast frames successfully received.

  In conjunction with total frames received, provides a gross percentage of the Ethernet usage for multicast frames addressed to this system.

- Multicast Bytes Received

  The total number of multicast data bytes successfully received. This does not include Ethernet data link headers. This number is the number of bytes in the Ethernet data field.

  In conjunction with total bytes received, provides a measurement of the percentage of this system's receive bandwidth (over time) that was consumed by multicast frames addressed to the local system.

- Receive Failures

  The total number of frames received with some data error. Includes only data frames that passed either physical or multicast address comparison. This counter includes failure reasons in the same way as the send failure counter.

  In conjunction with total frames received, provides a measure of data related receive problems. All of the problems reflected in this counter are also captured as events.

  Following are the possible reasons:

  - Block check error
  - Framing error

- Frame too long

- Send Failures

  The total number of times a transmit attempt failed. Each time the counter is incremented, a type of failure is recorded. When the counters are displayed, the list of failures is also displayed. When the counter is set to zero, the list of failures is cleared.

  In conjunction with total frames sent, provides a measure of significant transmit problems. captured as events.

  Following are the possible failures:

  - Excessive collisions
  - Carrier check failed
  - Frame too long
  - Remote failure to defer

- System Buffer Unavailable

  The total number of times no system buffer was available for an incoming frame. The RECEIVE BUFFERS parameter for the line may need to be increased.

- Unrecognized Frame Destination

  The number of times a frame was discarded because there was no portal with the protocol type or multicast address enabled. This includes frames received for the physical address, the broadcast address, or a multicast address.

2.3.3 Node Counters

The following counters are kept for the Executor Node:

- Aged Packet Loss

  This counter records the discarding of received data packets because the visit count exceeds the EXECUTOR MAXIMUM VISITS parameter.

- Maximum Logical Links Active

  This counter records the maximum number of logical links that have been simultaneously active.

- Node Out-of-range Packet Loss

  This counter records the discarding of received data packets because the destination node address is too large.

- Node Unreachable Packet Loss

  This counter records the discarding of received data packets because the destination node is no longer reachable.

- Oversized Packet Loss

  This counter records the discarding of received data packets which are larger than the blocksize that the data link to the next hop can accommodate.

- Packet Format Error

  The number of routing messages received having an invalid format.

- Partial Routing Update Loss

  The number of routing messages received indicating the existence of a reachable node with an address higher than the EXECUTOR MAXIMUM ADDRESS parameter.

- Verification Reject

  The number of invalid exchanges of verification information which occurred while attempting to establish circuits to adjacent nodes.

For all nodes for which counter information is available, the following counters are kept:

- Bytes Received

  The number of user data bytes received including normal, interrupt, connect, accept, reject and disconnect data.

- Bytes Sent

  The number of user data bytes transmitted.

- Connects Received

  The total number of connection requests received.

- Connects Sent

  The total number of connection requests transmitted.

- Messages Received

  The number of End Communications Layer messages received, including detected duplicates.

- Messages Sent

  The number of End Communications Layer messages sent, including retransmissions.

- Received Connect Resource Errors

  The number of received connection requests rejected by the local node.

- Response Timeouts

  The number of timeouts that have occurred in awaiting acknowledgments from the node.

- Seconds Since Last Zeroed

  The elapsed time, in seconds, since the set of node counters was last zeroed.

3
Loopback Testing

3.1 Introduction

Loopback tests are used to check both the DECrouter 2000 hardware and software. There are two kinds of loopback test that you can use to test the DECrouter 2000.

- The LOOP NODE tests are used to verify communication between the DECrouter 2000 and other nodes.
- The LOOP CIRCUIT tests are used to check individual circuits associated with the DECrouter 2000.

During a loopback test, data is sent from a node (for example, the DECrouter 2000) to another network component (for example, another node) and returned to the same node. If the test is successful, the returned data is exactly the same as the original data and the NCP prompt is redisplayed when the test has completed. If the component being tested is not working normally, the returned data is different from the original data, and a message is displayed which indicates that an error has occurred.

The following parameters can be used with the LOOP commands:

- COUNT
- LENGTH
- WITH

The COUNT and LENGTH parameters are used to specify the number of blocks sent during a test and the length of each block sent (in bytes).

The WITH parameter is used to specify the type of binary information sent during the loopback testing. You can specify three types of binary information:

ONES	All binary ones
ZEROS	All binary zeros
MIXED	An alternating sequence of ones and zeros

In the following example, 200 blocks of 20 bytes, each containing all binary ones, are sent over a circuit.

```
NCP> LOOP CIRCUIT circuit-name COUNT 200 LENGTH 20 WITH ONES
```

If you omit the WITH parameter, a combination of ones and zeros (MIXED) is sent. If you omit the COUNT and LENGTH parameters, one block of 40 bytes is sent. For example, the following command sends one block of 40 bytes, containing mixed binary information:

```
NCP> LOOP CIRCUIT circuit-name
```

You can run LOOP CIRCUIT tests over both DDCMP and Ethernet circuits. When you test Ethernet circuits, a separate set of parameters is used. Refer to Section 3.3.3 for details.

3.2 LOOP NODE Tests

The LOOP NODE command is used to test the operation of a logical link between the DECrouter 2000 and other nodes. You can initiate LOOP NODE tests from the DECrouter 2000 to a remote node or from a remote node to the DECrouter 2000. Figure 3-1 shows the path of the test data during a LOOP NODE test.

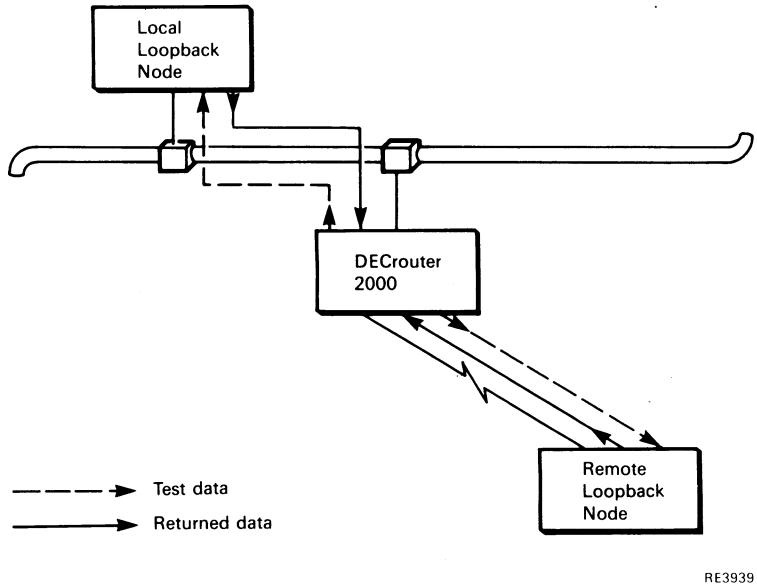

Figure 3-1: LOOP NODE Tests

3.2.1 Checking the Local Nodes

You can check the logical link between the load host (or any other node which is on the same Ethernet as the DECrouter 2000) and the DECrouter 2000 by issuing the following command at the load host:

```
NCP> LOOP NODE DECrouter
```

where *DECrouter* is the name of the DECrouter 2000. If this test fails, check that the load host can communicate with other nodes by repeating the test and looping the data to other nodes on the Ethernet. If these tests are successful, check the DECrouter 2000's Ethernet connection.

3.2.2 Checking Remote Nodes

To check that the DECrouter 2000 can communicate with other nodes in the network, issue the following command:

```
NCP> SET EXECUTOR NODE DECrouter
```

where *DECrouter* is the name of the DECrouter 2000. Check that the lines and circuits to be used for the test are set to ON, then issue the following command:

```
NCP> LOOP NODE node-name
```

where *node-name* is the name of the remote node.

NOTE

This test does not test a specific path from the DECrouter 2000 to the remote node - it simply checks that a path exists.

If this test is successful, repeat the test to nodes progressively further away from the DECrouter 2000 until you discover where the problem is. If you suspect that there is a problem with the DECrouter 2000, refer to Chapter 2.

3.3 LOOP CIRCUIT Tests

The LOOP CIRCUIT command is used to test both the DDCMP and Ethernet circuits.

You can use LOOP CIRCUIT tests to:

- Check the operation of the DECrouter 2000's synchronous communications ports using internal loopback tests.
- Check the operation of the DECrouter 2000's synchronous communications ports using external loopback connectors, or modems in loopback mode.
- Test the DECrouter 2000's Ethernet connection.

To test a DDCMP (synchronous) circuit, test the circuit from each end; that is, from both the local and the remote node (see Section 3.3.2for details of testing DDCMP circuits).

To test an Ethernet circuit, you only have to test the circuit from the local node, as the Ethernet interface loops the test data.

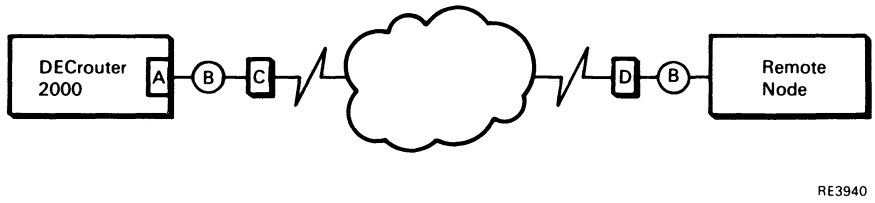

Figure 3-2: LOOP CIRCUIT Tests

In Figure 3-2, test data can be looped back along the path at points A, B, C, and D. Point A is the controller device within the DECrouter 2000; point B is a location on the circuit where a loopback connector can be used; point C is the local modem for the DECrouter 2000; point D is the remote modem. Use the loop switches on the modems to loop the data back from the modem to the source node.

3.3.1 Controller Loopback Tests

This test checks that the communications hardware is working as far as and including the controller. To test the controller, issue the following commands:

```
NCP> SET EXECUTOR NODE DECrouter
NCP> SET LINE line-name STATE OFF
NCP> SET LINE line-name CONTROLLER LOOPBACK STATE ON
NCP> SET CIRCUIT circuit-name STATE ON
NCP> LOOP CIRCUIT circuit-name
```

If this test is successful, but you cannot run loopback tests successfully to remote nodes, check the connections for each of the synchronous ports that you are using.

3.3.2 Circuit Loopback Tests

If the controller test is successful, check each circuit between the DECrouter 2000 and the local modems for each port and synchronous line you are using. Switch the modems to local loopback mode and issue the following command from the load host:

```
NCP> SET EXECUTOR NODE DECrouter
```

where *DECrouter* is the name of the DECrouter 2000. Then issue the following commands to check each line:

```
NCP> SET LINE line-name STATE OFF
NCP> SET LINE line-name CONTROLLER NORMAL STATE ON
NCP> SET CIRCUIT circuit-name STATE ON
NCP> LOOP CIRCUIT circuit-name
```

When you have tested each line in turn and you are satisfied that the circuits are working correctly between the DECrouter 2000 and the local modems, repeat the above tests from the remote node that you are trying to communicate with.

Check the path from the remote node and the remote modem being used to communicate with the DECrouter 2000.

If these tests are successful, then there is probably a fault between the local and the remote modems and you should have the modems checked for: compatibility of speed, cabling, and power supply.

You can use loopback connectors at any point on the physical line to check that the line is working.

You can also use the loopback testers on the remote modems. Use the loopback switch on the modem to loop data arriving at the remote modem back to the DECrouter 2000.

Normally a clock for the test is provided by a connection to a modem. There are two kinds of test for the loopback connectors used with the DECrouter 2000.

- A test at 64Kb which uses a 50-way DEC loopback connector (H3199), connected to the back panel of the DECrouter 2000. When this test is used, the hardware detects the loopback connector, and the line is automatically set to generate an internal clock.

- A test at 19.6Kb which uses an interface specific (RS232, V35 etc) loopback connector on the end of an adapter cable. When this test is used, the loopback connector cannot be detected by the DECrouter 2000 hardware, so the line must be set to the internal clock by issuing the following command:

```
NCP> SET LINE line-name CLOCK INTERNAL STATE ON
```

Once you have completed the test, turn off the clock by issuing the following command:

```
NCP> SET LINE line-name CLOCK EXTERNAL STATE OFF
```

3.3.3 Ethernet Circuit Loopback Tests

To test an Ethernet circuit, issue the following commands from the load host:

```
NCP> SET CIRCUIT circuit-name SERVICE ENABLED STATE ON
NCP> LOOP CIRCUIT circuit-name NODE DECrouter
```

where *DECrouter* is the DECrouter 2000 and *circuit-name* is the name of the load host's Ethernet circuit.

Then issue the following commands which are executed on the DECrouter 2000:

```
NCP> SET EXECUTOR NODE DECrouter
```

where *DECrouter* is the name of the DECrouter 2000, followed by:

```
NCP> LOOP CIRCUIT ETHERNET NODE node-name
```

where *node-name* is the name of the destination node.

When you use an Ethernet circuit for loopback testing, you can use other nodes on the Ethernet to act as assistant nodes for the test. The function of the assistant node is to help transmit and receive loopback data between the load host and the DECrouter 2000 if the DECrouter 2000 cannot be reached directly.

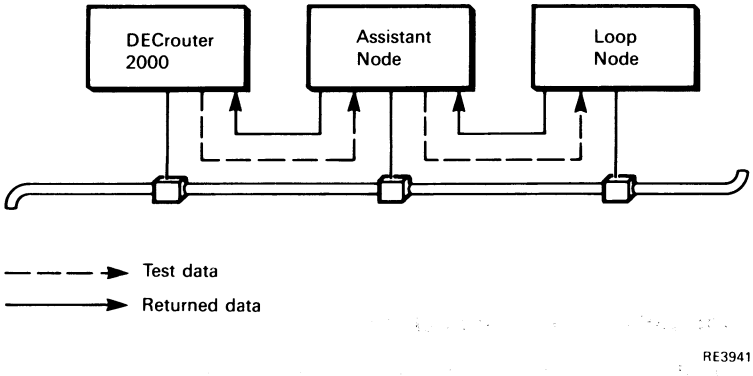

Figure 3-3: Ethernet Loopback Testing

The following LOOP CIRCUIT parameters apply to Ethernet circuits:

- ASSISTANT PHYSICAL ADDRESS
- ASSISTANT NODE
- HELP
- NODE
- PHYSICAL ADDRESS

Use the ASSISTANT PHYSICAL ADDRESS to identify the node, if any, which you want to use as a loopback assistant for passing loopback data between the DECrouter 2000 and the remote node. Alternatively, you can use the ASSISTANT NODE name to identify the loopback assistant node.

If you use either the ASSISTANT PHYSICAL ADDRESS or the ASSISTANT NODE parameter, and do not specify the HELP parameter, you will receive full assistance and will be helped with both receiving and transmitting test loopback data.

If you use ASSISTANT PHYSICAL ADDRESS or PHYSICAL ADDRESS parameter, you will need to know the Ethernet address of the node you are using as a loopback assistant. Determine the Ethernet address of the node by issuing the following commands:

```
NCP> SET EXECUTOR node-name
NCP> SHOW KNOWN LINE CHARACTERISTICS
```

where *node-name* is the name of the node that you want to use as a loopback assistant. The Ethernet address is displayed with the rest of the line characteristics.

The HELP parameter allows you to specify the form of assistance you require. There are three forms of help type:

TRANSMIT — The assistant node relays request data to the remote node which replies directly to the DECrouter 2000.

RECEIVE — The DECrouter 2000 sends request data to the remote node, which relays the reply to the assistant node for transmission to the DECrouter 2000.

FULL — The assistant node relays the request and reply data between the DECrouter 2000 and the remote node.

Use the NODE or PHYSICAL ADDRESS parameter to identify the destination node you want to use for testing the specified Ethernet circuit. For example:

```
NCP> LOOP CIRCUIT ETHERNET PHYSICAL ADDRESS AA-00-04-00-AA-BB -
_    ASSISTANT NODE WOMBAT HELP TRANSMIT
```

will start loopback testing on the Ethernet circuit ETHERNET using node WOMBAT as an assistant node in transmitting the data to the remote node which has the physical address of AA-00-04-00-AA-BB.

4
Starting and Dumping the System

4.1 Booting the System

There are three ways of starting and loading the DECrouter 2000:

- Powering up the hardware by plugging in the power cord.
- Using the remote console.
- Using the DUMP switch.

4.1.1 Plugging in the Power Cord

Normally, you start the system by connecting it to the AC power supply as Figure 4-1 shows.

CAUTION

If you power down the hardware, wait at least three seconds before powering it up again.

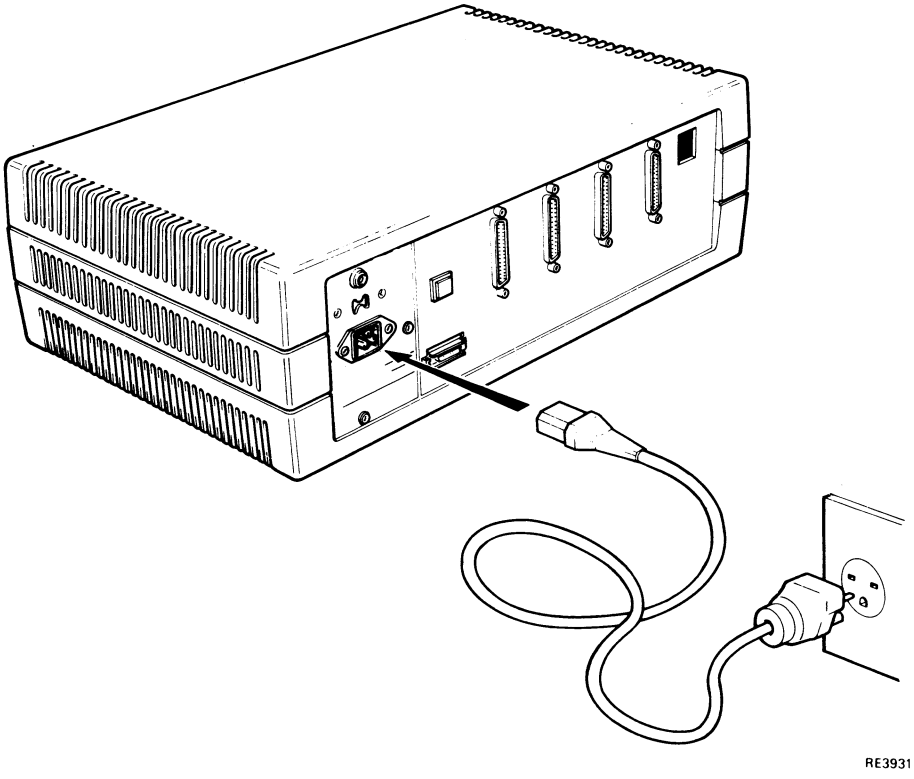

Figure 4-1: Connecting the Power Cord to Start the System

During power up, the DECrouter 2000 carries out an internal test, and asks for a load image from a host node. The display shows the progress of this test and load sequence. See Section 5.1.2 for details of how the DECrouter 2000 uses the display.

4.1.2 Using the Remote Console

The DECrouter 2000's remote console has a BOOT command that you can use to restart the system. So, if the system is already powered up, you follow the sequence in Table 4-1.

Table 4-1: Using the Remote Console to Load the System

	Step	Example
1	Log in to a host system	**VAX/VMS:** Username: *name* Password: *pass* **ULTRIX:** Login: *name* Password: *pass*
2	Connect the terminal as a remote console	**VAX/VMS:** $ RUN SYS$SYSTEM:NCP NCP>CONNECT NODE DECrouter - _ SERVICE PASSWORD ABCDE >>> **ULTRIX:** % ccr -n DECrouter -p abcde >>>
3	Stop the system	>>>HALT
4	Reload the system	>>>BOOT

Appendix C has details of how to use these remote console features.

NOTE

You should use this method of booting the system with care. Any sessions in progress when you issue the command are immediately lost, and no dump of the system status occurs. If you want a record of the system status before the restart occurs, see Section 4.2.1.

4.2 Dumping and Restarting the System

There may be times when you need to dump the current state of a running system, and then restart it. On the DECrouter 2000 there are two ways you can do this:

1. Press the DUMP switch
2. Use the DUMP command from the remote console

4.2.1 Using the DUMP Switch

The simplest way to dump and restart the system is to press the DUMP switch on the rear panel of the DECrouter 2000. Figure 4-2 shows where the switch is.

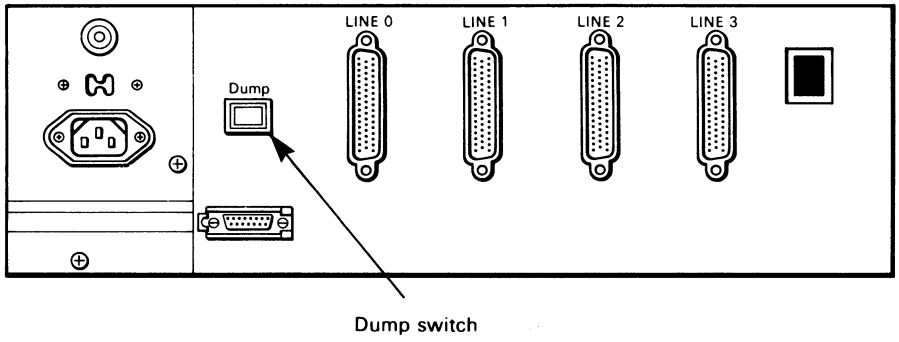

Figure 4-2: The Position of the DUMP Switch on the DECrouter 2000's Rear Panel

When you press this switch, the following occurs:

1. The DECrouter 2000 sends a dump request on the Ethernet and waits for a host node to reply.
2. When a host answers the request, the DECrouter 2000 sends the contents of memory (including the internal error log) to that node. You need about 5000 blocks of disk space (2MB + the size of the microprocessor internal memory) for the dump file.
3. When the dump is complete, the DECrouter 2000 runs its internal test and reboots.

The display on the control panel shows the progress of this procedure. Section 5.1.3 explains the values that appear on the display.

4.2.2 Using the Remote Console

If you are not near to the DECrouter 2000, it may be easier to use the remote console. To do this follow the steps in Table 4-2.

Table 4-2: Using the Remote Console to Dump the System

	Step	Example
1	Log in to a host system	**VAX/VMS:** Username: *name* Password: *pass* **ULTRIX:** Login: *name* Password: *pass*
2	Connect the terminal as a remote console	**VAX/VMS:** $ RUN SYS$SYSTEM:NCP NCP>CONNECT NODE DECrouter - _ SERVICE PASSWORD ABCDE >>> **ULTRIX:** % ccr -n DECrouter -p abcde >>>
3	Stop the system	>>>HALT
4	Dump the system	>>>DUMP

Appendix C has details of how to use the remote console.

The DECrouter 2000 takes the same actions when you enter the DUMP command as it would if the DUMP switch were pressed.

5
DECrouter 2000 Hardware Problems

5.1 The Display Panel on the DECrouter 2000

5.1.1 Position and Use

Each time you power up the DECrouter 2000, it runs an internal test and then loads its software. The unit uses a seven-segment display on the control panel to show how this test and load sequence is progressing.

The DECrouter 2000 also uses this display to:

- Show the progress of a dump sequence
- Report any errors that the internal test detects
- Show that the unit is connected to the electrical supply

Figure 5-1 shows the position of the display on the control panel. The following sections give details of how the DECrouter 2000 uses the display.

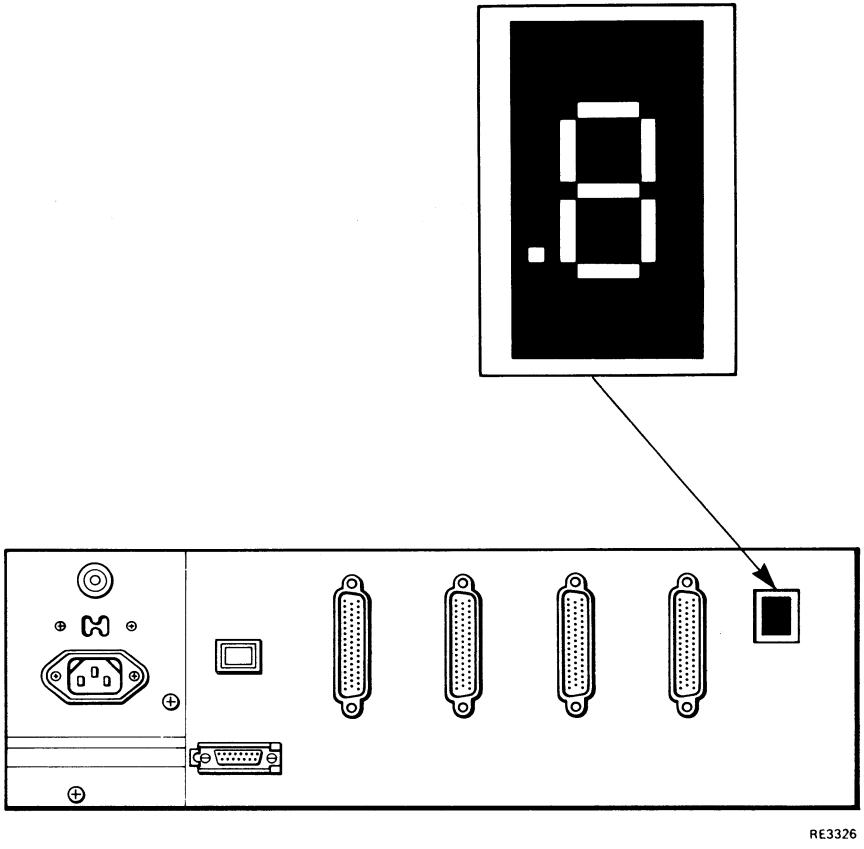

Figure 5-1: Location of the Display on the DECrouter 2000's Rear Panel

5.1.2 Power Up Sequence

On power up, the DECrouter 2000:

- Lights the decimal point of the display
- Tests the display
- Tests the internal circuitry
- Asks for software to be loaded

- Receives the software
- Runs the software

To show these states, the DECrouter 2000 uses different values on the display. Table 5-1 shows the values that can appear during the power up sequence.

Table 5-1: Values that can appear on the Display during Power Up

Display Value	Meaning
.	Power applied; tests are beginning.
Ripple pattern (one segment lights at a time)	DECrouter 2000 and display being tested
1	Waiting to load software. At this point, the DECrouter 2000 has asked a host to provide it with software and is waiting for the host to respond.
2	Loading software. Once a host has responded, the software is loaded across the Ethernet. This value shows that the load is taking place.
3	The load has failed because no load host responded. This means that the DECrouter 2000 will wait a while, and then try to send the message again. When it sends the message, the display reverts to 1.
Circling pattern (two segments light at a time)	Software running normally.

5.1.2.1 Normal Power Up Sequence Without Retry — In a normal load sequence, the display will show the following values:

1 2 4

5.1.2.2 Power Up Sequence Including Retry — If the DECrouter 2000 has to retry, the display shows a different sequence of values:

1 3

The display will alternate between 1 and 3 a load host responds. The display then continues like this:

1 2 4

5.1.3 Dump Sequence

During a dump sequence, the DECrouter 2000:

- Asks for a host to receive the dump
- Carries out the dump
- Enters the power up sequence

Table 5-2 shows the values that can appear on the display during a dump.

Table 5-2: Values That Can Appear on the Display During a Dump

Display Value	Meaning
5	Waiting to dump. At this point, the DECrouter 2000 has asked for a host to accept the dump, and is waiting for a host to respond.
6	Dump in progress. Once a host has responded, the DECrouter 2000 dumps the system's contents to the host across the Ethernet.

As with the power up sequence, the DECrouter 2000 may have to retry when asking for a host to receive the dump. This is reflected in the values that appear on the display, as the following sections show.

5.1.3.1 Normal Dump Sequence Without Retry — In a normal dump sequence, the display will show the following:

5 6

The display will then continue with the power up sequence (with or without retry as necessary).

5.1.3.2 Dump Sequence With Retry — If the DECrouter 2000 has to retry, the display shows a different value:

5

The display shows 5 until a load host responds. Once the dump request has been sent, the display continues as follows:

5 6

The display now continues to show a power up sequence (with or without retry, as necessary).

5.1.4 Dump Failure

If no dump host responds to the dump request, the DECrouter 2000 abandons the dump attempt and reloads the software. So, the display shows '5' and then continues with the power up sequence (with or without retry as necessary).

5.1.5 Halt Codes

To use some remote console commands (such as the loopback tests), the DECrouter 2000 software has to be halted. If the software is halted, the display panel shows the value:

8

5.1.6 Error Codes

The self-test uses the display to report any error that it detects. Table 5-3 shows the values that can appear, and a brief description of what each value means.

Table 5-3: Error Codes That Can Appear on the Display Panel

Display Value	Meaning
F	Fault detected in the DECrouter 2000 - contact DIGITAL Field Service
E	Ethernet connection fault, either within the DECrouter 2000 or between the DECrouter 2000 and the Ethernet

Table 5-3: Error Codes That Can Appear on the Display Panel (Cont.)

Display Value	Meaning
d	Synchronous I/O fault either within the DECrouter 2000 or in the synchronous port (the port itself is only tested if the 50-way loopback connector is attached)
C	The memory used as the system's internal error log has failed - contact DIGITAL Field Service

5.2 Diagnosing DECrouter 2000 Hardware Problems

This section offers guidance on how to diagnose and rectify DECrouter 2000 hardware problems. The problems are dealt with under the following category headings:

- The DECrouter 2000 does not power up (see Section 5.2.1)
- Connecting to the remote modem fails (see Section 5.2.2)
- Faulty synchronous port (see Section 5.2.3)

5.2.1 The DECrouter 2000 Does Not Power Up

If the fans do not turn and the power indicator does not light:

1. Make sure the power cord is correctly connected.
2. Check the fuse in the power plug.
3. Check the DECrouter 2000's circuit breaker.
4. Check the power socket.
5. Check the power cord.

If the fans do not turn (but the power indicator lights), call DIGITAL Field Service because there is a fault with the DECrouter 2000.

If the power indicator does not light (but the fans turn) the system may still work, and all that may be faulty is the power indicator. However, you should contact Field Service to replace the unit as soon as possible.

If the system does not boot, there may be a fault in the DECrouter 2000, and once again you should contact DIGITAL Field Service.

5.2.1.1 Make Sure the Power Cord is Correctly Connected — Check that:

1. The connector is properly seated in the socket on the rear panel.
2. The plug on the other end of the cord is properly connected to the power socket.

5.2.1.2 Check the Fuse in the Power Plug — Change the fuse in the power plug (if the plug has a fuse), and put the plug back in the socket.

5.2.1.3 Check the DECrouter 2000's Circuit Breaker — Reset the breaker as shown in Figure 5-2, and power up the unit again. If the breaker trips every time the power is applied:

1. Make sure the mains voltage selector on the rear panel is correctly set (see Figure 5-3 for your supply voltage).
2. If you have to change the setting on the selector, try to power up the unit again.
3. If the selector is correctly set but the breaker still trips, call DIGITAL Field Service.

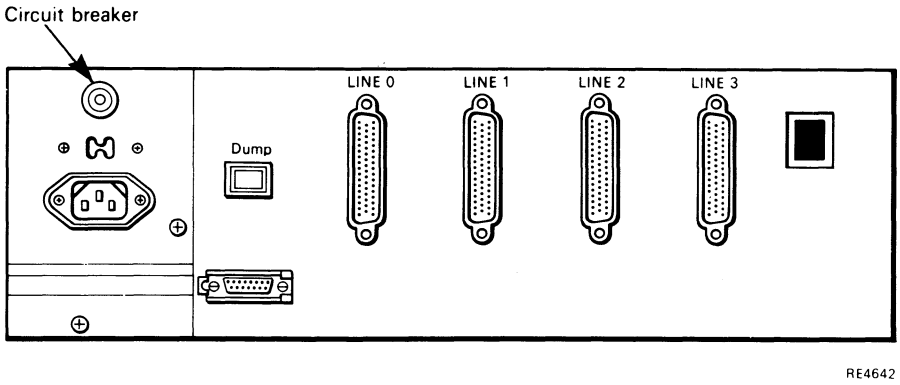

Figure 5-2: Resetting the Circuit Breaker

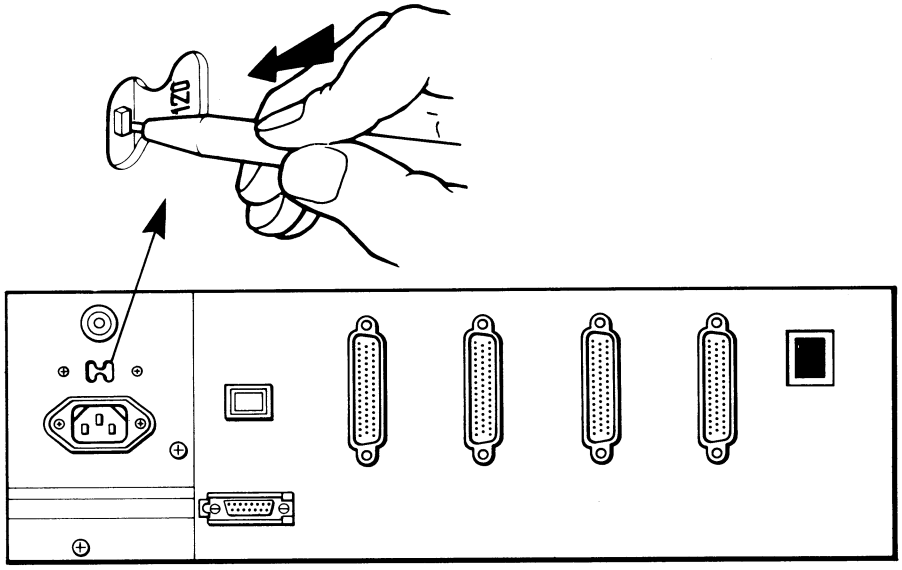

Figure 5-3: Checking the Voltage Selector

5.2.1.4 Check the Power Socket — Plug in something that you know works (such as a lamp or a terminal). If that appliance does not work, use another known good socket for the DECrouter 2000.

5.2.1.5 Check the Power Cord — Use the cord on another piece of equipment (such as a terminal). If that appliance does not work, replace the power cord and try to boot the system again.

If the power cord works, contact DIGITAL Field Service because there is a fault with your DECrouter 2000.

5.2.2 Connecting to the Remote Modem Fails

Work through the following until you correct the fault.

1. Check the dialing information.
2. Check that the local and remote modems are compatible.
3. Check cables.

4. Check the local modem.
5. Check the communications link and the remote modem.

5.2.2.1 Check the Dialing Information — Check with the remote site that you are using the correct dialing information. Make any necessary changes to the *DEC MicroServer Systems Configuration Card* and redial.

5.2.2.2 Check that Local and Remote Modems are Compatible — Check that the modems at your site and at the remote site are configured in a compatible way. If not, reconfigure either set as necessary and redial.

5.2.2.3 Check Cables —

1. Make sure all cables between the DECrouter 2000 and the modem are correctly connected. If you find any loose connections, correct them and then reload the system if necessary.
2. Use loopback tests to check out the modem control lines between the DECrouter 2000 and the modem. Section 5.3shows how to do these tests.
3. Replace any defective cables, and reload the system.

CAUTION

If you change an adapter cable, take care not to hit the dump button.

5.2.2.4 Check the Local Modem —

1. Make sure the modem is turned on.
2. Make sure the communications line is correctly connected to the modem.
3. Test the modem using its self-test facilities (if any). The manufacturer's information tells you what facilities are available and how to use them.
4. Use modem signal loopback tests to make sure the modem is working properly.
5. Correct any fault, or replace the modem, and then reload the system if necessary.

5.2.2.5 Check the Communications Link and the Remote Modem — Use the remote loopback tests explained in Section 5.3 to check the communications link and the remote modem. If the link appears to be faulty, contact its provider. If the remote modem appears to be faulty, contact the remote site and ask them to carry out further fault diagnosis.

5.2.3 Faulty Synchronous Port

The DECrouter 2000's internal test program exercises much of the circuitry in the unit, but it cannot test the synchronous ports themselves because they normally have cables attached to them. However, if a port is faulty, the DECrouter 2000 will detect communication errors when trying to use that line.

If you suspect that a port is faulty:

1. Make sure that the DECrouter 2000 is disconnected from the AC supply.
2. Remove the cables from that port, and attach the 50-way loopback connector (H3199) supplied with the DECrouter 2000 hardware.
3. Power up the DECrouter 2000 and watch the display on the rear panel.

Alternatively, you can use the port loopback test (see Section 5.3 for details).

With the loopback connector attached, the internal test program can now check the port as well as the internal circuitry. If the display shows the value "d", the synchronous port is faulty. If the test program carries on as usual, it's likely that there is no fault on that port. To do any further fault isolation on that line, read the information on loopback testing in Section 5.3.

The DECrouter 2000 has four synchronous ports, and having one faulty port does not necessarily mean that the product cannot be used. For example, if you are using only two lines at speeds less than or equal to 64K bits per second, you can simply use one of the two free ports instead of the faulty one. Note, however, that such reconfiguration should only be used as a temporary measure. Report the fault to DIGITAL Field Service and arrange for the unit's replacement.

5.3 Loopback Testing

There may be times when you cannot use the loopback test facilities offered by the DECrouter 2000 software (see Chapter 3). In these cases, you can use the loopback facilities that the DECrouter 2000 hardware offers. The following sections show what tests are available and how to use them.

5.3.1 Loopback Points

A DECrouter 2000 synchronous link has a number of parts, as Figure 5-4 shows.

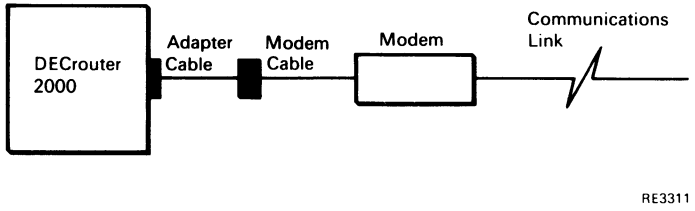

Figure 5-4: The Parts of a Synchronous Communications Link

Loopback connectors can be attached:

- To the socket on the DECrouter 2000
- At the end of the adapter cable
- At the end of the modem cable (if any)

Some modems have loop functions that can provide another two loopback points:

- On the communications link side of the modem
- At the remote modem

By doing a series of tests, each using a different loopback point, you can find out which segment of the line is faulty.

5.3.2 General Test Procedure

Test one line at a time, starting from the synchronous port and working towards the remote modem. Use the tests in this order:

1. Synchronous port
2. Adapter cable
3. Modem cable
4. Local modem
5. Remote modem

In this way, the first test to fail will show the segment that is faulty. For example, if the tests on the DECrouter 2000 port and the adapter cable succeed, but that on the modem cable fails, it is likely that the modem cable is faulty.

Section 5.3.4 explains the test procedure in detail.

5.3.3 The Remote Console's TEST Command

To do the testing, you use the DECrouter 2000's remote console. You need to use just one command (called TEST) to do each loopback test. The detailed procedure in Section 5.3.4 shows you the form of the TEST command to use for each test.

Each time you use TEST, it displays one of the following messages to tell you whether the test was successful.

- If the test is successful:

  Successful Data Loopback on Channel *n*

- If the test fails:

  Data Loopback Test Failed on Channel *n*

In both messages, the appropriate port number appears in place of *n*.

5.3.4 Detailed Procedure

The following sections show how to test a DECrouter 2000 communications link. Work through each section in turn.

5.3.4.1 Connect to the DECrouter 2000 — Attach to the remote console using the following steps:

1. Log in to a host system. This system must be on the same Ethernet as the DECrouter 2000.

2. For VAX/VMS load hosts, start NCP as follows:

   ```
   RUN SYS$SYSTEM:NCP
   ```

   Enter the CONNECT command:

   ```
   CONNECT NODE DECrouter
   ```

If there is a service password to protect the console, use this form of the CONNECT command:

```
CONNECT NODE DECrouter SERVICE PASSWORD password
```

where *password* is the service password.

3. For ULTRIX load hosts, enter the ccr command:

```
% ccr -n DECrouter
```

If a service password is defined to prevent unauthorized use of the console, use the following form of the connect command:

```
% ccr -n DECrouter -p password
```

where *password* is the service password.

When the connection is made, the following prompt appears on your terminal:

```
>>>
```

5.3.4.2 Stop the System Software — The tests need exclusive use of the synchronous ports. So, you need to stop the DECrouter 2000 software before starting the tests.

Enter this command:

```
HALT
```

5.3.4.3 DECrouter 2000 I/O Port Testing — Test the 50-way socket on the DECrouter 2000:

1. Remove the adapter cable from the 50-way socket

CAUTION

When removing the adapter cable, attaching the 50-way loopback, or reattaching the adapter cable, take care not to hit the DUMP button.

2. Attach the 50-way loopback connector (Type H3199)
3. Enter the following command at the console:

   ```
   TEST DATA_LOOPBACK/CHn
   ```

   Replace *n* by the number of the socket (0, 1, 2, or 3).

If the test fails, go to Section 5.3.5. Otherwise, reattach the adapter cable and go on to the next section.

5.3.4.4 Adapter Cable and Modem Cable Testing — These two tests use the same TEST command, but have the loopback connector attached to different points. Follow this procedure:

1. Detach the adapter cable from the modem (or modem cable).
2. Put the appropriate loopback connector on the end of the adapter cable.
3. Enter the following command at the console:

   ```
   TEST DATA_LOOPBACK/CHn
   ```

   Replace *n* by the number of the synchronous line (0, 1, 2, or 3).

If this fails, go to Section 5.3.5. Otherwise go on to test the modem cable (if there is one):

1. Reattach the modem cable to the adapter cable.
2. Detach the modem cable from the modem.
3. Put the appropriate loopback connector on the end of the modem cable.
4. Enter the following command at the console:

   ```
   TEST DATA_LOOPBACK/CHn
   ```

   As before, replace *n* by the number of the synchronous port.

If this fails, go to Section 5.3.5. Otherwise, reconnect the cable to the modem, and go on to the next section.

5.3.4.5 Local Modem Testing — The facilities offered by the local modem determine how you set it up as a loopback point. One of the modem control signals between the modem and the DECrouter 2000 can put the modem into a loopback mode. However,

not all modems recognize this local loop signal, and some need to be manually set to loopback. The modem manufacturer's literature should tell you what loopback facilities are available.

For modems that do recognize the local loop, use the following command:

```
TEST DATA_LOOPBACK/CHn/LOCAL
```

Replace *n* by the number of the synchronous port.

If the test fails, go to Section 5.3.5. Otherwise reset the modem and go on to test the remote link.

NOTE

Some modems have more than one test facility. You can use these as well to do more extensive tests on the modem. Refer to the manufacturer's instructions for information on any further test facilities.

If the modem has a manual test facility:

1. Switch the modem so that it loops back data (refer to the manufacturer's instructions).
2. Enter the command:

   ```
   TEST DATA_LOOPBACK/CHn/MANUAL
   ```

   Replace *n* by the number of the synchronous port.

If the test fails, go to Section 5.3.5. Otherwise, reset the modem and go on to test the remote link.

If your modem has no inbuilt test facilities:

1. Detach the communications link from the modem.
2. Attach a suitable loopback connector to the modem.
3. Enter the following command at the console:

   ```
   TEST DATA_LOOPBACK/CHn
   ```

   Replace *n* with the number of the synchronous port.

If the test fails, go to Section 3.2.2. Otherwise, reconnect the communications link, and go on to the next section.

5.3.4.6 Remote Modem Testing — Finally, you can test the communications link and the remote modem. Before doing this, contact the remote site and check that it is convenient to do the test. Then:

1. Make sure the remote modem is set up for the test (the remote site may have to switch the modem into a loopback mode).
2. Enter the following command at the console:

   ```
   TEST DATA_LOOPBACK/CHn/REMOTE
   ```

   Replace *n* by the number of the synchronous port.

5.3.5 What to do next

If one of the tests has failed, use Table 2-1 to find the cause, and what to do about it.

Table 5-4: Faults Indicated by Loopback Testing

Test that failed	What to do
DECrouter 2000 I/O port	Do one of the following: 1 Use one of the other ports on the DECrouter 2000 (see the Hardware Problem Solving information for details of how to do this). 2 Call DIGITAL Field Service to correct the problem with the DECrouter 2000.
Adapter cable	Replace the adapter cable.
Modem cable	Replace the modem cable.
Local modem	Call the modem's supplier to service the unit, or replace the modem.

Table 5-4: Faults Indicated by Loopback Testing (Cont.)

Test that failed	What to do
Remote modem	Either the remote modem or the intervening communications link is faulty. To isolate the problem, ask the remote site to test the modem. If the modem is working properly, contact the provider of the communications link and report the error. Otherwise, ask the remote site to correct the fault on their modem.

If all the tests pass, and you still have troubles on the link, check the system's configuration (including data speeds). Also, allow for intermittent faults by repeating the tests two or more times. Some conditions (such as electrical storms) cause errors even though all the equipment is working properly. To capture real problems, repeat the test sequence at a later time.

5.4 Summary

Table 2-1 summarizes how to test a communications link.

Table 5-5: Summary of Loopback Testing

Step	Action
Connect the remote console	1 Log in to a load host 2 Connect your terminal as a remote console (see Section C.1)
Halt the DECrouter 2000's software	Enter this command: >>>HALT
Test the I/O port	1 Remove the adapter cable 2 Connect H3199 loopback connector 3 Enter this command: >>>TEST DATA_LOOPBACK/CH*n*

Table 5-5: Summary of Loopback Testing (Cont.)

Step	Action
Test the adapter cable	1 Reconnect adapter cable to I/O port 2 Disconnect other end of adapter cable from modem or modem cable 3 Connect loopback connector to adapter cable 4 Enter this command: >>>TEST DATA_LOOPBACK/CH*n*
Test modem cable	1 Reconnect adapter cable and modem cable 2 Detach modem cable from the modem 3 Connect loopback connector to the modem cable 4 Enter this command: >>>TEST DATA_LOOPBACK/CH*n*
Test the local modem	If the modem has a local loop facility: 1 Reconnect modem cable to the modem 2 Enter this command: >>>TEST DATA_LOOPBACK/CH*n*/LOCAL If the modem has a manual test facility: 1 Reconnect modem cable to the modem 2 Switch the modem so that it loops back data 3 Enter this command: >>>TEST DATA_LOOPBACK/CH*n*/MANUAL

Table 5-5: Summary of Loopback Testing (Cont.)

Step	Action
	If the modem has no test facilities: 1 Reconnect the modem cable to the modem 2 Detach the communications link from the modem 3 Connect a suitable loopback connector to the modem 4 Enter this command: >>>TEST DATA_LOOPBACK/CH*n* 5 Reconnect the communications link to the modem
Test the remote modem	1 Ask for the remote modem to be set up ready to loopback data 2 Enter this command: >>>TEST DATA_LOOPBACK/CH*n*/REMOTE
In all TEST commands, replace *n* by the number of the channel under test.	

6
Problems and System Failure

This chapter describes what to do if the system fails and you believe the failure is caused by the DECrouter 2000 software.

Collect the following information and submit a Software Performance Report (SPR) to DIGITAL, stating that you are using DECrouter 2000 software and give the software version number.

1. Clearly define the problem (one problem on each SPR form).

   State:

   - Whether or not the problem is consistently reproducible and, if so, how to reproduce it.
   - How frequently the problem occurs.
   - Whether there are any factors related to the problem, for example heavy use of the DECrouter 2000 or the use of particular communication devices.

2. Give the priority of the problem.

   - Priority 1

     Major loss of functions. For example, DECrouter 2000 will not boot.

   - Priority 2

     Some loss of functions. For example, performance degradation, data transfer rate falls by a considerable amount.

- Priority 3

  Some impact on the user, manual intervention required. For example, an NCP command is required when there is a circuit failure, to turn the circuit off and back on again.

- Priority 4

  Functions can run with no significant impact on the user, problem can easily be worked around. For example, you may only need to alter the order in which certain NCP commands are entered.

- Priority 5

  No system modifications needed to return to normal functions. For example, you have a suggestion, want some advice, or find a documentation error.

3. Give:

   - Details of your DECrouter 2000's hardware configuration and the ports and speeds that you are using.
   - If you have used a VAX/VMS load host - the version number of any replacement components you may have.

     If, since installing the DECrouter 2000 and as a result of submitting SPRs, you have received replacement components containing .SYS files which you have installed on to your system, use the command ANALYZE/IMAGE on the updated components. Supply the image name, image file identification and the link date and time with the current SPR form.

   - A list of the contents of the permanent database. This consists of a file created by the ROUPERM program. Refer to the *DECrouter 2000 Management Guide* for details of this program. This file is called:

     SYS$SYSROOT:[DECSERVER]ROU*node-name*.SYS

     where *node-name* is the name of the DECrouter 2000.

   - If it is not possible to provide the file called ROU*node-name*.SYS, you should provide the output from the SHOW ALL command. The SHOW ALL command displays the contents of the current configuration in the permanent database. Specify a file name with this command in order to

create a record of the configuration. The SHOW ALL command allows you to check that the current configuration is correct. See the *DECrouter 2000 Management Guide* for information on the SHOW ALL command.

If you have to submit an SPR to DIGITAL, please include the file you have created with the SHOW ALL *file-name* with your SPR. If you have made any changes to the configuration since last booting the DECrouter 2000, you should mention these changes in the SPR.

- A list of the events logged (if any), and the diagnostic code displayed by the DECrouter 2000 (if any) when the problem occurred.
- Any hardware revisions to your system. The ECO level is given a label on the back of the DECrouter 2000.

4. If the system fails, submit a dump. If the DECrouter 2000 has not dumped automatically on system failure, obtain the dump as follows:

   Press the DUMP switch on the rear of the DECrouter 2000. The DECrouter 2000 requests the first available local host to receive the dump. The DECrouter 2000 then up-line dumps the contents of the memory to the host. By checking the event log, you can determine which load host has received the dump.

Copy the contents of the permanent database and any dumps to tape by issuing the following commands:

For VAX/VMS load hosts:

Copy the contents of the permanent database and any dumps obtained to tape by issuing the following commands:

```
$  COPY MOM$LOAD:ROUnode-name.SYS device
$  COPY MOM$LOAD:ROUnode-name.DMP device
```

where *node-name* is the name of the DECrouter 2000 and *device* is the name of the tape drive.

For ULTRIX load hosts:

Copy the contents of the permanent database and any dumps obtained to tape by issuing the following commands:

```
# cp /usr/lib/dnet/rou node-name.sys device
# cp /usr/lib/dnet/rou node-name.dmp device
```

where *node-name* is the name of the DECrouter 2000 and *device* is the name of the tape drive.

Send all the information, together with your SPR form, to the address given on the form.

A
Event Logging

A.1 Introduction

The DECrouter 2000 generates network events which inform you of changes in the status of network components, or of unexpected or unusual events.

Network events are divided into classes as shown in the following table:

Table A-1: Event Classes

Event Class	Description
0	Network Management layer
2	Session Control layer
3	End Communications layer
4	Routing layer
5	Data Link layer
225	Errors in Permanent Database

Within each class, each event has a type number. Thus, Event 5.16 belongs to Class 5 and is type number 16. The DECrouter 2000 always reports events in class 225 to the load host when there are configuration errors. Refer to the description of the SET/CLEAR LOGGING command in the *DECrouter 2000 Management Guide* for information on how to control the logging of other events.

In order to receive event reports, you must:

1. Enable the reception of events on one or more host nodes.

2. Inform the DECrouter 2000 which host nodes it should use to report events.

To permit a VAX/VMS host node to receive and display on a terminal events generated by the DECrouter 2000, issue the following commands:

```
NCP> SET LOGGING MONITOR STATE ON
$ @SYS$SYSTEM:STARTUP OPCOM
$ REPLY/ENABLE=NETWORK
```

These commands turn on logging to OPCOM, the default VAX/VMS monitor. Refer to the *VAX/VMS Networking Manual* for full details of event logging. Note that OPCOM may shorten event messages and may not show the values of counters when these are logged by the DECrouter 2000. If you want to receive the complete text of messages, you can use the LOGGING CONSOLE component. Refer to the *VAX/VMS Networking Manual*, the *VAX/VMS Network Control Program Reference Manual* and the *DECrouter 2000 Management Guide* for details.

To permit an ULTRIX host node to receive events generated by the DECrouter 2000, issue the following command:

```
ncp> set logging monitor state on
```

This command turn on logging to your local console, the default ULTRIX monitor. Refer to the *DECnet-ULTRIX Guide to Network Management*, the *DECnet-ULTRIX User's and Programmer's Guide*, and the *DECrouter 2000 Management Guide* for full details of the SET LOGGING command.

By default, events in class 225 generated by the DECrouter 2000 are automatically logged to your load host node. If you want other events to be logged, you need to specify a sink node to receive the messages.

Issue the following commands to set up sink nodes:

```
NCP> SET EXECUTOR NODE node-id
```

where *node-id* is the name of the DECrouter 2000, followed by:

```
NCP> SET LOGGING MONITOR SINK NODE node-id KNOWN EVENTS
```

where *node-id* is the node name or node address of the sink node which is to receive the event messages.

The event text that appears on your terminal is a standard message text. The message includes the source node (where the event occurred) and a time stamp for the event. For many events, the message includes the component name and type to which the event applies. Additional information may also be displayed, giving the reason for the event and the status of the component. The event text takes the following format:

```
event type class.type, event-text
from node address [(node-name)] dd-mmm-yy hh:mm:ss.ms
component-type, event-qualifiers,...
```

In the following section, the information accompanying each event description includes a brief description of the event parameters that may be displayed for the event.

A.2 Network Management Layer Events

0.0 Event records lost

Events occurred too rapidly for the event logger to collect and buffer them.

0.1 Automatic node counters

A node counter timer expired. The name of the node and the node counters for that node are displayed.

0.5 Node counters zeroed

Node counters were zeroed. The name of the node to which the event applies is displayed. The DECrouter 2000 logs the values of the counters prior to zeroing them.

0.6 Passive Loopback

The software initiated or terminated a passive loopback test on behalf of the adjacent node. The name of the circuit to which the event applies, along with the state of the operation,is displayed.

0.8 Automatic counters

A counter timer for a circuit or line has expired. The name of the component to which the event applies is displayed, along with the counters for that line.

0.9 Counters zeroed

Counters were zeroed for the circuit or line. The name of the component to which the event applies is displayed. The DECrouter 2000 logs the values of the counters prior to zeroing them.

A.3 Session Control Layer Events

2.1 Access control reject

The local node rejected a connection request because of invalid access control information.

Four event parameters are displayed; these are:

- The name and address of the source node
- The type number and identification of the object requesting the connection
- The type number and identification of the object to receive the connection request
- The invalid access control information

A.4 End Communications Layer Events

3.0 Invalid message

NSP received a message that could not be interpreted. There may be a software malfunction in either the local or the remote NSP. One event qualifier is displayed: the NSP message that was invalid. If this event occurs, submit an SPR (refer to Chapter 6 for details). Include with the SPR, information about the network configuration and the system software on the failing node (if identified).

3.1 Invalid flow control

The remote NSP attempted to modify the local flow control value in an invalid manner. There may be a software malfunction in either the local or remote the NSP. One event qualifier is displayed: the current flow control value. If this event occurs,

submit an SPR (refer to Chapter 6 for details). Include with the SPR, information about the network configuration and the system software on the failing node (if identified).

3.2 Database reused

To make the best use of resources, the DECrouter 2000 only keeps node counters for a small number of nodes; this number is usually much less than the total number of nodes in the network. When a connection is made with a new node, the DECrouter 2000 may discard counters for a node to which it no longer has a connection. If this occurs, the counter values for the previous node are logged before they are discarded.

A.5 Routing Layer Events

4.0 Aged packet loss

The packet had visited too many nodes, and so was discarded by the routing protocol. This is a normal occurrence when the network is reconfiguring its routing databases. This can also occur when the DECrouter 2000's EXECUTOR MAXIMUM VISITS parameter has a lower value than the EXECUTOR MAXIMUM HOPS parameter of another node in the area. The name of the circuit to which the event applies is displayed, along with the packet header.

4.1 Node unreachable packet loss

The DECrouter 2000 found that the destination node was unreachable, so the routing protocol discarded the packet. The name of the circuit on which the packet was received is displayed, along with the packet header.

4.2 Node out-of-range packet loss

The destination node number was greater than the maximum node number known to the DECrouter 2000, so the routing protocol discarded the packet. This usually results when a new node is added to the network without increasing the EXECUTOR MAXIMUM ADDRESS value on the DECrouter 2000. The name of the circuit on which the packet arrived is displayed, along with the packet header.

4.3 Oversized packet loss

The packet was too large to forward to the appropriate adjacent node, so was discarded by the routing protocol. Usually, this occurs when the adjacent node's buffer size is too small or when the source node sends a packet that is too large. All routers within the network should use the same size for their buffers. The name of the circuit over which the packet was to be forwarded is displayed, along with the packet header.

4.4 Packet format error

A format error in the packet header caused the routing protocol to discard the packet. This usually results from an error made by the adjacent node in programming the packet format, or it could be due to a circuit error that was not detected by the circuit protocol. If this event occurs frequently, submit an SPR (refer to Chapter 6 for details). The name of the circuit to which the event applies is displayed, along with the first 6 bytes of the packet, displayed in hexadecimal.

4.5 Partial routing update loss

Routing received a routing message that contained node addresses greater than the maximum address known to the local node. Subsequently, information on these nodes was lost. This occurs when the EXECUTOR MAXIMUM ADDRESS value on an adjacent node is increased to accommodate more nodes, but the value on the DECrouter 2000 is not increased. The name of the circuit over which this message was received, the packet header, and the highest node address in the routing update that was lost are displayed.

4.6 Verification reject

An attempt to initialize with another node failed. The local node asked the adjacent node for verification during routing initialization over the circuit, and received an invalid password. Either the DECrouter 2000 expected the wrong NODE RECEIVE PASSWORD, or the adjacent node sent the wrong NODE TRANSMIT PASSWORD. The name of the circuit to which the event applies is displayed, along with the address of the adjacent node that failed to initialize.

4.7 Circuit down, circuit fault

An error has occurred for the circuit. The name of the circuit to which the event applies is displayed, along with the reason for the event. The reason can be one of the following:

- Adjacent node address change

  The adjacent node changed its address unexpectedly.

- Adjacent node address out of range

  The adjacent node's address is greater than the maximum address defined for the DECrouter 2000. This is caused by an incorrectly defined node address or a failure to update the DECrouter 2000's database when a new node is added.

- Adjacent node block size too small

  The circuit block size provided by the adjacent node is too small for normal network operation. The block size may have be set incorrectly at the adjacent node. Check the EXECUTOR BUFFER SIZE and the EXECUTOR SEGMENT SIZE for the adjacent node.

- Adjacent node listener receive timeout

  The node has received no message over the data link within the expected timeout. The remote node may not be running or there may be a fault with the lines and cables.

- Adjacent node listener received invalid data

  A test message sent by the adjacent node contained invalid or corrupted data. This is probably a hardware problem. Check the lines and cables.

- Data errors

  When the line exceeds an error threshold, the line is declared down by the local node's line protocol handler. Check the lines and the modems.

- Dropped by adjacent node

  The adjacent node has broken the circuit connection. Check the adjacent node.

- Invalid verification seed value

  A routing initialization message sent by an adjacent node is not formatted properly. This is probably caused by a remote network software problem.

- Line synchronization lost

  The normal line protocol was restarted or terminated by the adjacent node. Either a line exceeded an error threshold, or network management initiated a line state change. Check the cables connected to the DECrouter 2000, and the remote lines and modems. Factors that can cause a line synchronization error are as follows:

  1. Threshold errors, including more than seven attempts to transmit a message, or seven NAKs received in a row.
  2. Start message received in the ON state (that is, the remote system restarted the line).
  3. Maintenance requested while in the ON state (that is, the remote system tried to perform a maintenance operation such as LOOP CIRCUIT).

- Routing update checksum error

  A routing update packet failed its internal integrity test. Check the line counters for hardware errors.

- Unexpected packet type

  A packet was received out of the normal protocol sequence; for example, the local node received a normal data packet when a verification packet was expected.

- Verification password required from Phase III node

  A required routing initialization password was not specified before an attempt was made to initialize the Phase III node in a Phase IV network.

- Verification receive timeout

  A required verification packet was not received from the adjacent node within the required response time: packets were lost on the line, or a failure occurred at the adjacent node.

- Version skew

  The routing version of the adjacent node is unacceptable to the DECrouter 2000; the adjacent node may have the incorrect software installed. This will occur if you attempt to communicate with a Phase II DECnet node.

4.8 Circuit down

An error has occurred on the circuit. This occurs if a node fails, or is removed from the network. The name of the circuit to which the event applies is displayed, along with the following event qualifiers: the packet header, the reason (as described for event 4.7), and the address of the adjacent node.

4.9 Circuit down, operator initiated

This event may occur if a second router on the Ethernet is removed from the LAN. The name of the circuit to which the event applies is displayed, along with the following event qualifiers: the packet header, the reason (as described for event 4.7), and the address of the adjacent node.

4.10 Circuit up

A node connected to one of the DECrouter 2000's synchronous ports has initialized. The name of the circuit to which the event applies is displayed, along with the address of the newly initialized node. Note that this event does not imply that the node is reachable. Reachability is determined by the higher-level routing algorithms.

4.11 Initialization failure, line fault

A remote node failed to initialize with the local node because of a physical line error. The name of the circuit to which the event applies is displayed, along with the reason for the event (as described for event 4.7).

4.13 Initialization failure, operator fault

A remote node failed to initialize with the local node because of an operator error. An incorrect password could have been used, or the password was not set up. The name of the line to which the event applies is displayed, along with three event qualifiers: the packet header, the reason (as described for event 4.7), and the routing version received from the adjacent node.

4.14 Node reachability change

Because of routing operation, the reachability of a remote node has changed. The name of the node to which the event applies is displayed, along with the new status of the node (reachable or unreachable).

4.15 Adjacency up

The adjacent node on the circuit is initialized. The name of the circuit to which the event applies is displayed, along with the address of the adjacent node.

4.16 Adjacency rejected

The adjacent node on the circuit is not initialized. The name of the circuit to which the event applies is displayed, along with two event qualifiers: the address of the adjacent node and the reason for the event (as described for event 4.7).

4.17 Area reachability change

Routing protocol has changed the reachability of an area. The name of the area to which the event applies is displayed, along with the new status of the area (reachable or unreachable).

4.18 Adjacency down

An error has occurred for an adjacency on the circuit. The name of the circuit to which the event applies is displayed, along with the following event qualifiers: the reason (as described for event 4.7), the packet header, and the address of the adjacent node on the circuit.

4.19 Adjacency down, operator initiated

An adjacency on the circuit is down because of an operator error, or the circuit has been set to STATE OFF. The name of the circuit to which the event applies is displayed, along with the following event qualifiers: the reason (as described for event 4.7), the packet header, and the address of the adjacent node on the circuit.

A.6 Data Link Layer Events

5.0 Locally initiated state change

The circuit state changed because of an operator command. The name of the circuit to which the event applies is displayed, along with two event qualifiers: the old DDCMP state (HALTED, ISTRT (initialized starting the circuit), ASTRT (waiting for acknowledgment before starting the circuit), RUNNING, or MAINTENANCE) and the new DDCMP state.

5.1 Remotely initiated state change

A remote user changed the circuit state. The name of the circuit to which the event applies is displayed, along with two event qualifiers (see event 5.0).

5.2 Protocol restart received in maintenance mode

The remote node restarted normal operation while the DECrouter 2000 had the circuit in maintenance mode. The name of the circuit to which this event occurs is displayed.

5.3 Send error threshold

Too many consecutive data transmission errors occurred. The name of the circuit to which the event applies is displayed, along with the circuit counters for that circuit.

5.4 Receive error threshold

Too many consecutive data reception errors occurred. The name of the circuit to which the event applies is displayed, along with the circuit counters for that circuit.

5.5 Select error threshold

This event only applies to half-duplex circuits. Too many selection errors occurred. The name of the circuit to which the event applies is displayed, along with the circuit counters for that circuit.

5.6 Block header format error

DDCMP received an invalid block header. The name of the circuit to which the event applies is displayed, along with the invalid block header.

5.8 Streaming tributary

This event only applies to half-duplex circuits. A tributary on the circuit is impeding the use of that circuit. The name of the circuit to which the event applies is displayed, along with the tributary address of the received tributary, and the tributary status. Status may be any of the following:

- Streaming
- Continued send after timeout
- Continued send after deselect
- Ended streaming

5.9 Local buffer too small

A local buffer is too small for the received data block. The name of the circuit to which the event applies is displayed, along with the length (in bytes) of the received block, and the buffer.

5.13 Line initialization failure

An initialization failure occurred over the Ethernet line. The name of the line is displayed in the message.

5.14 Send failure on line

A data transmission attempt failed on an Ethernet line. The name of the line to which the event applies is displayed, along with the reason for the failure and the distance. The distance is measured in units of approximately 6 metres, and provides some information as to where on the Ethernet the problem is occurring. Failure reasons can include the following:

- Excessive collisions
- Carrier check failed

- Short or open circuits
- Frame too long
- Remote failure to defer
- Block check error
- Framing error
- Data overrun
- System buffer unavailable
- Unrecognized frame destination

5.15 Receive failed on line

Data was not received on an Ethernet line. The name of the line to which the event applies is displayed, along with two event qualifiers: the reason for the failure (as described in event 5.14), and the Ethernet header, which includes the source and destination node addresses and the protocol type.

5.16 Collision detect check failed on line

A check for collision detection failed on an Ethernet line. The name of the line to which the event applies is displayed.

A.7 Errors in Permanent Database

225.0

This event is reported by the DECrouter 2000 immediately after loading if there are any errors in the Permanent Database Configuration. The event message consists of a series of messages separated by commas, for example:

```
DECnet event 225.0
From node 12.85 (WOMBAT),  3-AUG-1987 18:29:39.24
Error in Permanent Database Configuration, Node with same address as Executor
```

The first message is always:

"Error in Permanent Database Configuration"

The second message indicates the cause of the error and may be:

- Node with same address as Executor

  A node has been defined in the permanent database with the same address as the DECrouter 2000. Therefore, the configuration information for this node will be ignored.

- Unrecognized component

  The configuration file is not compatible with the current version of the DECrouter 2000 software, or is a configuration file for another DEC MicroServer software product (for example, the X25router).

- Invalid file contents

  The configuration file is corrupt and a new file should be created.

If the second message is not one of these, the component that could not be configured as requested is listed. This component could be:

- Executor node
- Node *n.m*
- Line *line-name*
- Circuit *circuit-name*
- Logging console
- Logging file
- Logging monitor
- Object *object-name*

In this case, there will be a a third message containing two error numbers in brackets: [*-n, m*].

The configurator program, ROUPERM, will normally detect configuration errors when the permanent database is defined. In the unlikely event of a configuration error being reported by the DECrouter 2000 when it loads, more information about the error can be obtained by interpreting the first of the two numbers (-n) as follows:

-6 Unrecognized parameter - A parameter is not recognized by the DECrouter 2000.

-8 Unrecognized component - The component (for example a line or circuit) does not exist on the DECrouter 2000.

-9 Invalid identification format - The name of the component is invalid.

-11 Component in wrong state - The operation cannot be performed because the component is in the wrong state.

-15 Resource error - The DECrouter 2000 has insufficient resources to perform the operation.

-16 Invalid parameter value - A parameter has an unacceptable value.

-20 No room for new entry - The DECrouter 2000 has insufficient memory to create the new component.

-22 Parameter not applicable - One of the parameters defined for this component is inapplicable.

-23 Parameter value too long - The value defined for one of the parameters is too long.

-27 Invalid parameter grouping - Conflicting parameters are defined for the component.

-29 Parameter missing - A mandatory parameter was not defined.

The second digit (m) is used by DIGITAL for diagnosis and should be included in the SPR. It is unlikely that you will see these error numbers during normal operation, unless you are trying to use a configuration file intended for another product (for example, an X25router configuration file with a DECrouter 2000 system).

In all other cases, you should submit an SPR and include the full message text, and a description of the intended configuration with your form. Refer to Chapter 6 for details.

B
The TRACE Utility

B.1 How to Use the TRACE Utility

TRACE is a command-driven utility that can be installed on a local VAX/VMS system (this includes your load host), and can be used to monitor, record and analyze the flow of data on the DECrouter 2000.

Note: For full details of the commands used by TRACE, refer to Section B.2.

The relevant data is recorded by TRACE as it passes through the various protocol levels of DECnet, as specified by one or more *tracepoints* -see Section B.1.1. As well as recording the data, TRACE also logs the protocol events that occur.

For DECnet, the following protocol activity can be recorded:

1. ETHERNET
2. DDCMP
3. ROUTING
4. NSP

These types of protocol activity can be monitored from the DECrouter 2000 load host, or from any other VMS node that has TRACE installed.

NOTE

Before you start using the TRACE utility, you should be familiar with the protocols that you want to monitor. Refer to the appropriate *DECnet Functional Specification*, for full details of the protocol you wish to use.

There are two methods of collecting data using the TRACE utility.

- You can collect and analyze the data at the same time as it is traced. This method is known as LIVE tracing, and is described in Section B.1.2.
- You can record unanalyzed data in a file and analyze it later. For this method, you need to run TRACE as a detached process - see Section B.1.3.

NOTE

When using the TRACE utility, data may arrive faster than it can be captured, analyzed or displayed on the screen, so trace records may be lost.

For details of the privileges needed to use TRACE and aspects of security related to the use of the utility, refer to Sections B.1.4 and B.1.5, respectively.

B.1.1 Specifying Tracepoints

Data and/or information is collected at the point where it passes through a layer of the protocol. These data collection points are known as **tracepoints**, and are specified by you when you issue a START command.

Note: If you specify more than one tracepoint in the START command, the tracepoints must all be on the same node.

Each tracepoint is linked to a trace collector. As data passes through the protocol layer, a trace record is created. The tracepoints valid for the DECrouter 2000 are shown in Tables B-1.

Table B-1: DECnet Tracepoint Formats

NODE::ETHERNET

There are no options for use with this tracepoint. Only one channel can be traced.

NODE::DDCMP[.*line-id*]

where *line-id* is the given line name. If the line name is not specified, all currently defined lines are traced.

NODE::NSP

There are no options for use with this tracepoint.

NODE::ROU_ETHERNET

There are no options for use with this tracepoint. Only one channel can be traced.

NODE::ROU_SYNC[.*circuit-name*]

where *circuit-name* is the given circuit name. If the circuit name is not specified, all currently defined circuits are traced.

B.1.2 Performing LIVE Tracing

Initiate data collection by issuing a TRACE/ROUTER command followed by a START command. For example issue the following command to start tracing NSP:

```
$ TRACE/ROUTER

NETTRACE> START/LIVE KANGA::NSP
```

This will turn on the tracepoint specified in the command and analyze the data at your terminal. The following messages are displayed:

```
%NETTRACE-I-CONNECTING, connecting to trace collector...
%NETTRACE-I-CHANSTRACED, n channels are currently being traced.
%NETTRACE-I-PROC_TRPT, processing NSP tracepoint
```

where *n* is the number of lines or channels currently being traced. If new lines or channels are created while tracing is running, these will also be traced. You can use the SHOW command to display the names of the channels being traced. Refer to Section B.2.11 for details.

You can now issue an ANALYZE command to alter the format of the data output. For example:

```
NETTRACE> ANALYZE/DATA=ASCII
```

changes the data output to ASCII format.

When you want to stop tracing, exit from TRACE by typing <CTRL/Z>. This stops tracing, deletes the tracepoints and returns the DCL prompt. If you have set up more then one tracepoint, you cannot turn them off individually, nor can a tracepoint exist while there is no trace collector working. You can issue SHOW commands to display what is being traced. For example:

```
NETTRACE> SHOW TRACE_COLLECTOR */FULL
```

The following information is displayed:

```
                    Tracing on node KANGA:: on date-time

Collector           Collecting to      Tracepoints

BORIS::Tony         Interactive        NSP
                                        NSP-ROUTING
```

"Collector" is the node and process name that is collecting the trace data. The process name is the process where the START/LIVE command was issued.

"Collecting to" shows whether tracing is being done as an interactive process, or whether data is being written to a file.

"Tracepoints" shows the tracepoints and the names of the lines or channels being traced. In this case, tracing is being done on NSP and the channel being traced is NSP-ROUTING.

B.1.3 Running TRACE as a Detached Process

If you have not specified the /LIVE qualifier in the START command, a detached process collects the data and writes it to a file. This local trace collector has a name based on your username taking the format *username*$NETTRACE_*n*, where *n* is an integer.

Initiate trace collection by issuing a TRACE/ROUTER START command. For example:

```
$ TRACE/ROUTER START KANGA::DDCMP
```

The following messages are displayed:

```
%NETTRACE-I-DETSTART, Detached trace collector -
   _SMITH$NETTRACE has been started
%NETTRACE-I-CREATEFILE, File DUA0:[SMITH]TRACE.DAT;1 has been created
%NETTRACE-I-CHANSTRACED, 3 channels are currently being traced
```

This means that tracing has started at the DDCMP tracepoint on the remote node KANGA.

Trace data is written to TRACE.DAT in your current default directory (unless you have specified the /OUTPUT qualifier).

Shared file access allows you to analyze the file at the same time as data is being collected.

As with LIVE tracing, you can issue SHOW commands to display what is being traced. For example:

```
NETTRACE> SHOW TRACE KANGA::*/FULL
```

which displays the following information:

```
                    Tracing on node KANGA on date-time

Collector                Collecting to          Tracepoints

BORIS::SMITH$NETTRACE    DUA0:[SMITH]TRACE.DAT  DDCMP
                                                 SYN-0
                                                 SYN-1
```

"Collector" is the node and process name that is collecting the trace data.

"Collecting to" shows the name of the file that data is being written to.

"Tracepoints" shows the tracepoints and the names of the lines or channels being traced. In this case, tracing is being done at the DDCMP tracepoint on node KANGA over lines SYN-0 and SYN-1.

If you exit TRACE at this point, tracing continues. To stop tracing, issue the STOP command. For example:

```
NETTRACE> STOP
```

Returns the message:

```
%NETTRACE-I-LTCEXIT, Trace collector SMITH$NETTRACE has exited
```

Stopping TRACE deletes the tracepoints. Note that you do not specify a node name in the STOP command; the process you are stopping must be on your local node.

Analyze the file by issuing an ANALYZE command. For example:

```
$ TRACE/ROUTER ANALYZE/OUTPUT=TRACE/DISPLAY=(TIME,EVENT,NAME)
```

will display the following messages:

```
%NETTRACE-I-PROCFILE, Processing file DUA0:[SMITH]TRACE.DAT;1
%NETTRACE-I-CREATEFILE, File DUA0:[SMITH]TRACE.LIS;1 has been created
%NETTRACE-I-PROC_TRPT, Processing DDCMP tracepoint
%NETTRACE-I-EOFILE, End of input reached
```

This command analyzes the file TRACE.DAT and writes the formatted output to TRACE.LIS.

By default, the ANALYZE command starts at the lowest version number of the file that exists. Specify the version number if you wish to start by analyzing a higher version file.

The ANALYZE command takes the file as input, and formats it according to the command qualifiers. See Section B.2.1 for details of these command qualifiers.

B.1.4 Privileges needed to use TRACE

TRACE is installed when the DECrouter 2000 software is loaded with the following privileges:

TMPMBX, SYSNAM, SYSPRV, ALTPRI

The privileges that your process must have to use TRACE depend on the operation you are performing.

- To START or STOP trace you need NETMBX and OPER privilege.
- To SHOW trace you need NETMBX privilege.
- To ANALYZE a trace file, you need no additional privileges, but you need to be able to access the file.

B.1.5 Trace and Security

TRACE allows you to monitor network activity, which means that you can view data as it passes through the protocol that you are tracing. This data may include passwords and other sensitive information. Therefore, you should control access to the TRACE utility by using a password associated with the TRACE object. So, in addition to the privileges required to run the trace utility, you will also need to specify an access control string in a START or SHOW command. Access to the trace object on the node being traced can be controlled by using the SET OBJECT command in the ROUPERM configurator program. Refer to the *DECrouter 2000 Management Guide* for details of this command.

When you issue any command that includes a node specification, you will then need to specify the username and password that were set up for the trace object in the ROUPERM configurator program. For example:

```
NETTRACE> START/LIVE BORIS"fred secret"::NSP
```

will try to connect to the trace object on node BORIS using username fred and password secret.

B.2 TRACE Command Reference

This section contains details of the commands used by the TRACE utility. For ease of reference, the commands have been arranged in alphabetical order (see Sections B.2.1to B.2.14).

To run TRACE, issue the DCL command:

```
$ TRACE/ROUTER
```

This gives you the prompt:

```
NETTRACE>
```

You can now enter any TRACE command. You should note, however, that the following commands - ANALYZE, HELP, SHOW, START and STOP - can also be entered from the standard DCL prompt.

To enter a TRACE command, either type out the command in full (adding optional qualifiers where necessary), or use the default TRACE keypad. You can add new key definitions by using the DEFINE/KEY command (and delete existing definitions via the DELETE/KEY command). DEFINE/KEY allows you to associate a string and set of attributes with a key on the keyboard.

Note: To display a list of the currently defined keys, use the SHOW KEY/ALL command.

For a summary of available TRACE commands, see Table B-2. For guidance on how these commands relate to functions on the default keypad, see Figure B-1.

Table B-2: Summary of TRACE Commands

Command	Description
ANALYZE	Analyzes the data in the file, or allows you to modify the analysis options.
ATTACH	Allows you to switch from one process to another.
BACK	Displays the last screen of analyzed data.
CLEAR	Clears the screen of analyzed data.
DEFINE/KEY	Associates a string and set of attributes with a key on the keyboard.
DELETE/KEY	Deletes the key definitions you have defined.
EXIT	Returns you to the DCL prompt.
HELP	Displays the on-line help about Trace.
NEXT	Displays the next screen of analyzed data.
REFRESH	Redraws the screen.
SHOW	Displays the trace collector and tracepoints or the key definitions.
SPAWN	Creates a subprocess of the current process.
START	Starts the collection of data.
STOP	Stops the collection of data.

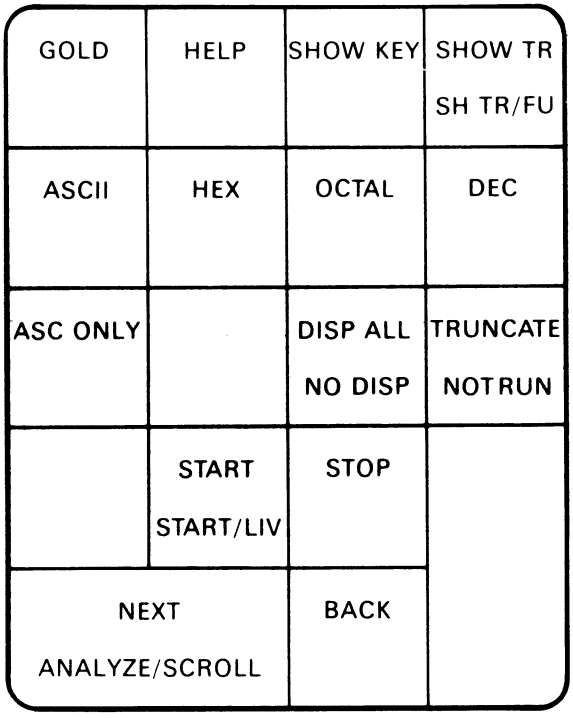

Figure B-1: Default TRACE keypad

Note that in addition to the keys shown on the pad, the file from which the keypad is set up provides definitions for the following keys:

- GOLD NEXT_SCREEN as ANALYZE/SCROLL.
- NEXT_SCREEN as a NEXT screen command.
- PREV_SCREEN as a BACK screen command.
- HELP as a HELP command.
- CTRL/L as a CLEAR screen command.
- CTRL/W as a REFRESH screen command.

B.2.1 The ANALYZE Command

The format of the ANALYZE command is:

```
NETTRACE> ANALYZE [filename] [qualifiers]
```

where *filename* is the name of the file you wish to analyze. The default filename is TRACE.DAT. By default, the ANALYZE command starts at the lowest version of the file that exists and analyzes each version of the file in turn. If you issue the ANALYZE command without specifying a filename, when a file is already being analyzed, the ANALYZE qualifiers are changed without a new file being opened.

Full details of the ANALYZE command qualifiers are given in Section B.2.1.1.

B.2.1.1 ANALYZE Command Qualifiers — The following qualifiers are available for use with the ANALYZE command.

/OUTPUT = *output-filename*

This is the name of the output listing file. If you do not specify /OUTPUT, output is written to your terminal.

/SINCE[= *time*]

Allows you to see data traced since a specified time. Specify the time in standard VMS format.

/BEFORE[= *time*]

Allows you to see data traced before a specified time. Specify the time in standard VMS format.

/PAGE = *number*

Allows you to specify the number of lines generated per page. The default is SYS$LP_LINES = 66.

/WIDTH = *number*

Allows you to specify the number of columns generated per page. If the data is written to a file, the default is 132. If the data is displayed on your terminal, the default is the width of your terminal screen.

/NODATA
/DATA = (*data-type,...*)

Allows you to specify the format of the output, which can be one or more of: HEXADECIMAL, OCTAL, ASCII or DECIMAL. In ASCII output, control characters are represented by " ^character". For example: <CTRL/M> (Carriage Return) is represented by " ^M". The default is HEXADECIMAL.

/[NO]TRUNCATE

Determines whether data that will not all fit on one line is truncated or printed on the next line. The default is /NOTRUNCATE.

/EVENT = (*event-type,...*)

Where event-type is one or more of: TRANSMIT, RECEIVE or ALL. This allows you to select which events should be included in the output. The default is ALL.

/NODISPLAY
/DISPLAY = (*display-option,...*)

This allows you to specify which fields appear in the analyzed output. See Table B-3. The default display options are TIME, EVENT and SIZE.

Table B-3: DISPLAY Qualifier Options

Option	Description
[NO]ALL	Specify ALL if you require all the display options listed below. Use NOALL to cancel a previous ALL specification.
[NO]EVENT	Shows whether this is a received or transmitted data buffer, or some other event.
[NO]FLAG	A flag word in the trace record. Not normally used.
[NO]FUNCTION_CODE	The function code of the operation being traced, normally; this is "Data".
[NO]NAME	Name of the traced line or channel (particularly useful when more than one line or channel is being traced simultaneously).
[NO]SIZE	Shows the total size of the trace record in bytes. This figure includes all protocol headers, but not CRCs or flags.
[NO]STATUS	The status field. Sometimes used to show whether the I/O operation completed successfully.
[NO]TIME	Displays the time at which the record was created (i.e. the time at which data first passed through the protocol level being traced.)

/[NO]REVERSE

If your terminal supports reverse video (and is set to DEC_CRT), specifying /REVERSE causes the received data to be displayed in reverse video, and the titles to be displayed in bold. The default is /REVERSE.

/NOPROTOCOL
/PROTOCOL=(*protocol-identifier*,...)

Where *protocol-identifier* is one or more of the tracepoints. Use /NOPROTOCOL if you want the protocol headers to appear as part of the displayed data (i.e. without being formatted). The default protocols depend on

which tracepoints are being analyzed. /PROTOCOL (which is not normally used) allows you to specify which protocol is displayed. For details of tracepoints and the default protocol identifiers, see Table B-4.

/NOSELECT
/SELECT=(*protocol-identifier*,...)

Where *protocol-identifier* is one or more of the tracepoints. This allows you to specify which protocol headers appear in the output. Any protocol headers that you SELECT are removed from the analyzed data. The default is to display all the protocol headers, see Table B-4.

/TRACE_LEVEL=(*protocol-identifier*,...)

Where *protocol-identifier* is one or more of the tracepoints. If you have specified more than one tracepoint to enable simultaneous tracing of more than one protocol level, the analyzer will only format trace records for one of the protocols.

By default, the analyzer formats the trace records of the first protocol level found in the file. Use /TRACE_LEVEL to specify a protocol level. (The trace records of the protocol you specify will then be formatted.) For details of tracepoints and the default protocol identifiers, see Table B-4.

Table B-4: Default DECnet Protocol Identifiers

Tracepoint name	Default Protocol
DDCMP	DDCMP
ETHERNET	ETHERNET
NSP	NSP
ROU_ETHERNET	ROUTING
ROU_SYNC	ROUTING

/NAME=*channel-name*

Where *channel-name* is the name of the trace records that are to be analyzed. If you specify only part of the channel name, then all records that start with this string will be selected for analysis. Use the /DISPLAY=NAME option to display the channel name in the trace record. /NAME=* will display all the records.

/SAVE_BUFFER_SIZE=*n*

Where *n* is the number of screens of saved lines of data. These lines can be examined by using the NEXT and BACK commands. The default is 30.

/[NO]SCROLL

Determines whether data is typed a page at a time, or continuously. If you specify /NOSCROLL, you have to issue the NEXT or BACK command to display the next screen. /NOSCROLL is not valid if output is to a file. ANALYZE/SCROLL will display data continuously following a NEXT or BACK command. The default is /SCROLL.

B.2.2 The ATTACH Command

The ATTACH command enables you to switch control from your current process to another. You can only specify one of: process-name, /IDENTIFICATION or /PARENT. The format of the command is as follows:

```
NETTRACE> ATTACH [process-name]
```

where *process-name* is the name of the process to which you wish to attach. You cannot connect to a process if:

- The process is your current process.
- The process is not part of your current session.
- The process does not exist.

If you specify a process-name, you cannot use the /IDENTIFICATION or /PARENT qualifier.

B.2.2.1 ATTACH Command Qualifiers — The qualifiers that may be used with the ATTACH command are as follows:

/IDENTIFICATION

Specifies the process identification (PID) of the process that you wish to attach to. If you specify /IDENTIFICATION, you cannot specify /PARENT or *process-name*.

/PARENT

This is only valid if used from a subprocess, and allows you to attach to the parent process. If you specify /PARENT, you cannot specify /IDENTIFICATION or *process-name*.

B.2.3 The BACK Command

This command allows you to move back through analyzed TRACE output one screen at time. (See also the NEXT command - refer to Section B.2.9.)

The format of the command is:

```
NETTRACE> BACK
```

If you are running TRACE as a detached process, the output is directed to a file for later analysis. When the contents of the file are ANALYZEd, the BACK command can be used to turn off the (default) scroll mode and display the data one screen at a time. Each time you enter the command, the previous screenful of data is displayed.

During LIVE tracing, initial use of the BACK command suspends TRACE activity and turns off scrolling. Thereafter, the command functions in the normal way, allowing you to move back through previous screenfuls of data (one screen at a time).

Note: To resume LIVE tracing, first issue the NEXT command, and then type ANALYZE/SCROLL. In this way, all saved records are scrolled through (on screen) before tracing recommences.

B.2.4 The CLEAR Command

The CLEAR command clears the screen of analyzed data. The format of the command is:

```
NETTRACE> CLEAR
```

This command has the same effect as <CTRL/L>.

B.2.5 The DEFINE/KEY Command

The DEFINE/KEY command allows you to define a key on the keyboard with a command. The format of the command is as follows:

```
NETTRACE> DEFINE/KEY key-name equivalence-string
```

where *key-name* is the key you wish to define, and *equivalence-string* specifies the string to be processed when you use the defined key. If the string contains any spaces, the string should be enclosed in quotation marks.

The qualifiers that you can use with the DEFINE/KEY command are shown in Section B.2.5.1.

B.2.5.1 DEFINE/KEY Command Qualifiers — The qualifiers available with the DEFINE/KEY command are as follows:

/[NO]ECHO

Specifies if the command line is echoed after you enter the defined key. You cannot specify both /NOECHO and /NOTERMINATE for a key definition. The default is /ECHO

/[NO]IF_STATE= *state-list*

Specifies a list of states, any one of which must be set in order to enable the specified key definition. For example, using the GOLD key to define a key.

/[NO]LOCK_STATE

Retains the state set by the /SET_STATE qualifier until the /SET_STATE qualifier is used again to alter the state. The default is /NOLOCK_STATE.

/[NO]SET_STATE=*state*

Associates a state with the key being defined. The state name can be any alphanumeric string. You cannot define a key by specifying both /SET_STATE and /TERMINATE.

/[NO]TERMINATE

Determines if the specified command string executes when you enter the key. When you use /NOTERMINATE, you must enter <RET> to execute the command. You cannot define a key by specifying both /SET_STATE and /TERMINATE. The default is /NOTERMINATE.

B.2.6 The DELETE/KEY Command

The DELETE/KEY command allows you to delete a key definition. The format of the command is as follows:

```
NETTRACE> DELETE/KEY key-name
```

where *key-name* is the key you wish to delete. There is only one qualifier for this command which is /STATE.

/[NO]STATE = (*state-name*[,...])

Specifies the name(s) of state(s) for which the specified key definition(s) are to be deleted. If you specify only one state name, you can omit the parentheses. A state name can be any appropriate alphanumeric string. If you omit the /STATE qualifier or use /NOSTATE, key definitions in the current state are deleted.

B.2.7 The EXIT command

The EXIT command allows you to leave TRACE and return to the DCL prompt.

The format of the command is:

```
NETTRACE> EXIT
```

This command has the same effect as <CTRL/Z>.

B.2.8 The HELP Command

This command displays on-line help information about TRACE. The format of the command is:

```
NETTRACE> HELP
```

B.2.9 The NEXT Command

This command allows you to move (forward) through analyzed TRACE output one screen at time. See also the BACK command - refer to Section B.2.3.

The format of the command is:

```
NETTRACE> NEXT
```

If you are running TRACE as a detached process, the output is directed to a file for later analysis. When the contents of the file are ANALYZEd, the NEXT command can be used to turn off the (default) scroll mode and display the data one screen at a time. Each time you enter the command, the next sequential screenful of data is displayed.

During LIVE tracing, the NEXT command has no effect until the BACK command has been used to suspend tracing and work back through the existing output. The NEXT command can then be used in the normal way to display the next sequential screenful of data. Note, however, that once you reach the point where tracing was suspended, use of the NEXT command has the effect of resuming LIVE tracing.

B.2.10 The REFRESH Command

The REFRESH command redraws the screen. The format of the command is:

```
NETTRACE> REFRESH
```

This command has the same effect as <CTRL/W>.

B.2.11 The SHOW Command

The SHOW commands allow you to display the trace collector or the key definitions for your keyboard. To display the trace collector, issue the following command.

```
NETTRACE> SHOW TRACE_COLLECTOR [NODE::] [process-name-string][/[NO]FULL]
```

where *process-name-string* is a string which matches the data collecting process that you want to display. If you omit *process-name-string*, all trace collecting processes are displayed.

If you specify /FULL, the names of the lines or channels being traced are also displayed. If the trace object was installed with a user-name and password, you will need to specify these in the SHOW command. See Section B.1.5 for details.

```
NETTRACE> SHOW KEY [key-name]
```

Displays the key definitions that you have defined using the DEFINE/KEY command. or definitions that have been read from the NETTRACE key definition file. This file is pointed to by the logical name NETTRACE$KEY_INIT. Full details of the SHOW KEY command qualifiers are given in Section B.2.11.1.

B.2.11.1 SHOW KEY Command Qualifiers — The qualifiers available with the SHOW KEY command are as follows:

/ALL

Displays all the key definitions. If you use the /ALL qualifier, do not specify a key name.

/[NO]STATE=*state-name*

Specifies the name of a state table for which the specified key definitions are to be displayed. If you do not specify /STATE, then the key definitions for all states are displayed.

/FULL

Requests that all qualifiers that are associated with a definition are displayed. The default is /NOFULL.

B.2.12 The SPAWN Command

The SPAWN command creates a subprocess of the current process. The format of the command is as follows:

```
NETTRACE> SPAWN [command-string]
```

where *command-string* is the command to be executed by the subprocess. When the command completes, the subprocess terminates and control is returned to the parent process. If you use both the /INPUT qualifier and a command string, commands are obtained from the input files after the specified command string has been executed.

B.2.12.1 SPAWN Command Qualifiers — The qualifiers available with the SPAWN command are as follows:

/INPUT= *file-spec*

Specifies an input file which contains one or more DCL commands to be executed by the spawned subprocess. If you specify a command string with the SPAWN command and an input file with the /INPUT qualifier, the command string is processed before the input file. Once processing of the input file is complete, the subprocess is terminated.

/[NO]LOGICAL_NAMES

Determines whether the system passes logical names and logical name tables to the subprocess. The default is /LOGICAL_NAMES.

/OUTPUT=*file-spec*

Allows you to write the output from the SPAWN command to a specified file.

/PROCESS=*subprocess-name*

Specifies the name of the subprocess to be created. By default, a unique process name is assigned with the same base name as the parent process and a unique number. The default subprocess name format is *username_n*.

/PROMPT=*string*

Specifies the prompt string for DCL to use within the subprocess. By default, SPAWN copies the current prompt from the parent process.

/[NO]SYMBOLS

Determines whether the system passes DCL global and local symbols to the subprocess. The default is /SYMBOLS.

/[NO]WAIT

Controls whether the system waits until the current subprocess is completed before allowing more commands to be issued by the parent process. The /NOWAIT qualifier allows you to issue new commands while the specified subprocess is running. When you use the /NOWAIT qualifier interactively, make sure that you use the /OUTPUT qualifier as well, so output from the subprocess is directed to a file rather than to your terminal. Otherwise, your terminal will be used by more than one process at a time. The default is /WAIT.

B.2.13 The START Command

The format of the START command is different depending on which form of tracing you require.

For LIVE tracing, the format is:

```
NETTRACE> START/LIVE [tracepoint,...] [qualifiers]
```

When TRACE is used as a detached process, however, the format is:

```
NETTRACE> START[/NOLIVE] [tracepoint,...] [qualifiers]
```

The qualifiers that may be used also depend on whether TRACE is being run as a LIVE or a detached process.

The qualifiers available during LIVE tracing are described in Section B.2.13.1, while those used when TRACE is being run as a detached process are given in Section B.2.13.2.

Note: If the trace object was installed with a user-name and password, you will need to specify these in the START command. See Section B.1.5for details.

B.2.13.1 START Qualifiers Used for LIVE Tracing — The qualifiers available when TRACE is used for LIVE tracing are as follows:

/CAPTURE_SIZE=*n*

Specifies the maximum amount of data (in bytes) in each trace record captured. Decreasing the value of this qualifier may reduce the number of lost trace records. The default is 140.

/MAXIMUM_BUFFERS=*n*
/BUFFER_SIZE=*n*

Specifies the number and size (in bytes) of the internal buffers kept by TRACE. Increasing these values may help to avoid trace records being lost. The default for BUFFER_SIZE is 500. The default for MAXIMUM_BUFFERS is 3 if you are tracing a remote node.

/TIME_OUT = *n* (units of 100ms)

Trace records are sent to the collector when a trace buffer is full, or when a timeout occurs. The /TIME_OUT qualifier allows you to specify, in units of 100ms, the rate at which a timeout occurs. The default is 2.

/PAGE = *number*

Allows you to specify the number of lines generated per page. If the data is being output to a file, the default is SYS$LP_LINES-6.

/WIDTH = *number*

Allows you to specify the number of columns generated per page. If the data is written to a file, the default is 132. If the data is displayed on your terminal, the default is the width of your terminal screen.

/NODATA
/DATA = (*data_type,...*)

Allows you to specify the format of the output, which can be one or more of: HEXADECIMAL, OCTAL, ASCII or DECIMAL. In ASCII output, control characters are represented by " ^character". For example: <CTRL/M> (Carriage Return) is represented by " ^M". The default is HEXADECIMAL.

/[NO]TRUNCATE

Determines whether data that will not all fit on one line is truncated or printed on the next line. The default is /NOTRUNCATE.

/EVENT = (*event-type,...*)

Where event-type is one or more of: TRANSMIT, RECEIVE or ALL. This allows you to select which events should be included in the output. The default is ALL.

/NODISPLAY
/DISPLAY = (*display-option,...*)

This allows you to specify which fields appear in the analyzed output. See Table B-5. The default display options are TIME, EVENT and SIZE.

Table B-5: DISPLAY Qualifier Options

Option	Description
[NO]ALL	Specify ALL if you require all the display options listed below. Use NOALL to cancel a previous ALL specification.
[NO]EVENT	Shows whether this is a received or transmitted data buffer, or some other event.
[NO]FLAG	A flag word in the trace record. Not normally used.
[NO]FUNCTION_CODE	The function code of the operation being traced, normally; this is "Data".
[NO]NAME	Name of the traced line or channel (particularly useful when more than one line or channel is being traced simultaneously).
[NO]SIZE	Shows the total size of the trace record in bytes. This figure includes all protocol headers, but not CRCs or flags.
[NO]STATUS	The status field. Sometimes used to show whether the I/O operation completed successfully.
[NO]TIME	Displays the time at which the record was created (i.e. the time at which data first passed through the protocol level being traced.)

/[NO]REVERSE

If your terminal supports reverse video (and is set to DEC_CRT), specifying /REVERSE causes the received data to be displayed in reverse video, and the titles to be displayed in bold. The default is /REVERSE.

/NOPROTOCOL
/PROTOCOL=(*protocol-identifier*,...)

Where *protocol-identifier* is one or more of the tracepoints. Use /NOPROTOCOL if you want the protocol headers to appear as part of the displayed data (i.e. without being formatted). The default protocols depend on

which tracepoints are being analyzed. /PROTOCOL (which is not normally used) allows you to specify which protocol is displayed. For details of tracepoints and the default protocol identifiers - see Table B-6.

/NOSELECT
/SELECT=(*protocol-identifier*,...)

Where *protocol-identifier* is one or more of the tracepoints. This allows you to specify which protocol headers appear in the output. Any protocol headers that you SELECT are removed from the analyzed data. The default is to display all the protocol headers -see Table B-6.

Table B-6: Default DECnet Protocol Identifiers

Tracepoint name	Default Protocol
DDCMP	DDCMP
ETHERNET	ETHERNET
NSP	NSP
ROU_ETHERNET	ROUTING
ROU_SYNC	ROUTING

/NAME=*channel-name*

Where name is the name of the trace records that are to be analyzed. If you specify only part of the channel name, then all records that start with this string will be selected for analysis. Use the /DISPLAY=NAME option to display the channel name in the trace record. /NAME=* will display all the records.

/SAVE_BUFFER_SIZE=*n*

Where buffer size is the number of screens of saved lines of data. These lines can be examined by using the NEXT and BACK commands. The default is 30.

B.2.13.2 START Qualifiers Used for Detached Tracing — The qualifiers available when running TRACE as a detached process are as follows:

/CAPTURE_SIZE=*n*

Specifies the maximum amount of data (in bytes) in each trace record captured.

Decreasing the value of this qualifier may reduce the number of lost trace records. The default is 140.

/MAXIMUM_BUFFERS=*n*
/BUFFER_SIZE=*n*

Specifies the number and size (in bytes) of the internal buffers kept by TRACE. Increasing these values may help to avoid trace records being lost. The default for BUFFER_SIZE is 500. The default for MAXIMUM_BUFFERS is 3 if you are tracing a remote node.

/TIME_OUT=*n* (units of 100ms)

Trace records are sent to the collector when a trace buffer is full, or when a timeout occurs. The /TIME_OUT qualifier allows you to specify, in units of 100ms, the rate at which a timeout occurs. The default is 2.

/PRIORITY=*n*

Specifies the base priority of the detached process. The default is 8.

/OUTPUT=*filename*

Specifies a filename for your output. This file will contain unanalyzed trace records. The default is TRACE.DAT.

/BLOCKS=*n*

Specifies the approximate maximum size (in blocks) of each output file that is created. When the file reaches the maximum size, it is closed and a new version of the file is opened. The default is 200.

/VERSION_LIMIT=*n*

Specifies the maximum number of versions kept of the output file. The default is 10.

/PROCESS_NAME=*process-name*

Allows you to specify an alternative name for the detached process. The default is *username*$NETTRACE.

B.2.14 The STOP Command

The format of the STOP command is:

```
NETTRACE> STOP [process-name]
```

where *process-name* is the collecting process you wish to stop.

To stop LIVE tracing, issue the STOP command without specifying a process name. To stop a detached trace collector, specify the name of the process you wish to stop. The default process name is *username*$NETTRACE.

You can display the names of the trace collectors by using the SHOW command.

B.3 Examples of TRACE Output

B.3.1 NSP Tracepoint Example

The command:

```
$ TRACE/ROUTER START/LIVE WOMBAT::NSP
```

will produce output as follows:

```
-----------+----+-----+(--------------------NSP Frame-----------------)+-------
    Time   |Evnt|Data |Msg               Data  Other Segnum Drc    Flow|Data
hh mm ss cc|    |Size |Type  Dest Source C Numb  Numb D Numb    Irc Cntl|
-----------+----+-----+(-----------------------------------------------)+-------
17:46:19.63|R-> |   38|ConI     0   720   NSP V4.0  Segsize=1459    Nctl| 00 13
17:46:19.71| <-T|    3|ConA   720                                      |
17:46:19.72| <-T|   13|ConC   720 43023   NSP V4.0  Segsize=1461    Nctl| 03 04
17:46:19.74|R-> |   11|LnkS 43023   720             0      1 0      Same|
17:46:19.74| <-T|   11|LnkS   720 43023             1      1 0      Xon |
17:46:19.75|R-> |    7|Oack 43023   720             1                  |
17:46:22.41|R-> |   12|Data 43023   720       0             1           | 14 01
17:46:22.41| <-T|    7|Dack   720 43023       1                         |
17:46:22.42| <-T|   10|Data   720 43023       1             1           | 02
17:46:22.43|R-> |    7|Dack 43023   720       1                         |
17:46:22.43| <-T|   23|Data   720 43023       1             2           | 01 FF
17:46:22.51|R-> |    7|Dack 43023   720       2                         |
17:46:22.52| <-T|   23|Data   720 43023       1             3           | 01 FF
17:46:22.53|R-> |    7|Dack 43023   720       3                         |
17:46:22.54| <-T|   26|Data   720 43023       1             4           | 01 FF
17:46:22.55|R-> |    7|Dack 43023   720       4                         |
17:46:22.55| <-T|   10|Data   720 43023       1             5           | 80
17:46:22.58|R-> |    7|Dack 43023   720       5                         |
17:46:24.92|R-> |    8|DisI 43023   720      Reason = Abort             | 00
17:46:24.92| <-T|    7|DisC   720 43023      Reason = Disc Complete     |

%NETTRACE-I-CONNECTING, connecting to trace collector...
%NETTRACE-I-CHANTRACED, 1 channel is currently being traced
%NETTRACE-I-PROC_TRPT, Processing NSP tracepoint
```

B.3.2 DDCMP Tracepoint Example

The command:

```
$ TRACE/ROUTER START/LIVE WOMBAT::DDCMP.LO
```

will produce the following output:

```
-----------+----+-----+(--------DDCMP Frame--------)+--------------------------
    Time   |Evnt|Data | Typ QS Res Num Ctl Adr Count|Data
hh mm ss cc|    |Size |                             |
-----------+----+-----+(--------------------------)+--------------------------
10:02:05.55|R-> |   21| DAT  S  15  59       1    13| 02 06 A8 33 A8 00 04 B7 0F
10:02:05.58|R-> |    6| ACK  S  16           1     |
10:02:05.65|R-> |   21| DAT  S  16  60       1    13| 02 06 A8 33 A8 00 04 B7 0F
10:02:05.66| <-T|   28| DAT  S  60  17       1    20| 07 55 AA 00 02 00 A8 00 2A
10:02:05.72|R-> |    6| ACK  S  17           1     |
10:02:05.86|R-> |  150| DAT  S  17  61       1   142| 07 19 A8 00 20 00 A0 00 FF
10:02:06.66| <-T|    6| ACK  S  61           1     |
10:02:08.65| <-T|   28| DAT  S  61  18       1    20| 07 55 AA 00 02 00 A8 00 2C
10:02:08.71|R-> |    6| ACK  S  18           1     |
10:02:08.85|R-> |  150| DAT  S  18  62       1   142| 07 19 A8 00 20 00 A0 00 FF
10:02:09.56| <-T|    6| ACK  S  62           1     |
10:02:10.41|R-> |   27| DAT  S  18  63       1    19| 02 06 A8 33 A8 00 60 B7 0F
10:02:10.43| <-T|   21| DAT  S  63  19       1    13| 02 19 A8 06 A8 03 04 06 16
10:02:10.48|R-> |    6| ACK  S  19           1     |
10:02:10.65| <-T|   22| DAT  S  63  20       1    14| 02 19 A8 06 A8 03 38 06 16
10:02:10.70|R-> |    6| ACK  S  20           1     |
10:02:10.73|R-> |   21| DAT  S  20  64       1    13| 02 06 A8 33 A8 00 48 B7 0F
10:02:11.46| <-T|    6| ACK  S  64           1     |

%NETTRACE-I-CONNECTING, connecting to trace collector...
%NETTRACE-I-CHANTRACED, 1 channel is currently being traced
%NETTRACE-I-PROC_TRPT, processing DDCMP tracepoint
```

B.3.3 ETHERNET Tracepoint Example

The command:

```
$ TRACE/ROUTER START/LIVE WOMBAT::ROU_ETHERNET
```

will produce the following output:

```
-----------+----+-----+(-------------------Routing Frame-----------------)+---
    Time   !Evnt!Data !Msg   Dest   Source    R R V I B C PRI     Blk  Hello!Dat
hh mm ss cc!    !Size !Type        Type/addr  Q T Y E K O    VST  Size Timer!
-----------+----+-----+(-----------------------------------------------)+---
17:43:22.06!R-> !  174!RHel        L1R 12.593             126     1498    15!
17:43:22.06!R-> !  174!RHel        L1R 12.593             126     1498    15!
17:43:22.08!R-> !   34!EHel        END 12.77                      1498    15!
17:43:22.08!R-> !   34!EHel        END 12.564                     1498    15!
17:43:22.08!R-> !   34!EHel        END 12.388                     1498    15!
17:43:22.13!R-> !   34!EHel        END 12.550                     1498    15!
17:43:22.14!R-> !   34!EHel        END 12.234                     1498    15!
17:43:22.39!R-> !   34!EHel        END 12.17                      1498    15!
17:43:22.70!R-> !   34!EHel        END 12.365                     1498    15!
17:43:22.73!R-> !   33!EHel        END 12.275                     1484    15!
17:43:22.74!R-> !   33!EHel        END 12.300                     1500    15!
17:43:22.91!R-> !   34!EHel        END 12.91                      1498    15!
17:43:22.96!R-> !   33!EHel        END 12.288                     1484    15!
17:43:23.03!R-> !   34!EHel        END 12.59                      1498    15!
17:43:23.32!R-> !   32!EHel        END 12.221                     1498    15!
17:43:23.44!R-> !   34!EHel        END 12.114                     1498    15!
17:43:23.57!R-> !   34!EHel        END 12.38                      1498    15!
17:43:23.83!R-> !   34!EHel        END 12.394                     1498    15!
17:43:23.88!R-> !   34!EHel        END 12.360                     1498    15!
17:43:24.28!R-> !   34!EHel        END 12.375                     1498    15!
17:43:24.30!R-> !  251!RHel        L1R 12.595              32     1498    15!
17:43:24.47!R-> !   33!EHel        END 12.531                     1500    15!
17:43:24.53!R-> !   34!EHel        END 12.98                      1498    15!
17:43:24.57!R-> ! 1462!Roui           12.583                                 !
17:43:24.59!R-> !  606!Roui           12.583                                 !
17:43:24.69!R-> !   34!EHel        END 12.571                     1498    15!


%NETTRACE-I_CONNECTING, connecting to trace collector...
%NETTRACE-I-CHANTRACED, 1 channel is currently being traced
%NETTRACE-I-PROC_TRPT, Processing ROU_ETHERNET tracepoint
```

B.3.4 Synchronous Circuit Tracepoint Example

The command:

```
$ TRACE/ROUTER START/LIVE WOMBAT::ROU_SYNC
```

will produce the following output:

```
-----------+----+-----+(-------------------Routing Frame-----------------)+---
    Time   |Evnt|Data |Msg   Dest   Source    R R V I B C PRI     Blk  Hello|Dat
hh mm ss cc|    |Size |Type        Type/addr  Q T Y E K O    VST  Size Timer|
-----------+----+-----+(-----------------------------------------------------)+---
17:40:21.51|R-> |    0|
17:40:21.52| <-T|   12|Init        L1R 12.596                      576    15|
17:40:21.52|R-> |   12|Init        L1R 12.596                      576    15|
17:40:21.53| <-T|    0|
17:40:22.57| <-T|    8|Hel             12.596                              |
17:40:22.58|R-> |    8|Hel             12.596                              |
17:40:26.63| <-T|   20|Rou1            12.596                              |
17:40:26.64|R-> |   20|Rou1            12.596                              |
17:40:28.84| <-T|   14|Rou1            12.596                              |
17:40:28.85|R-> |   14|Rou1            12.596                              |
17:40:38.72| <-T|   12|Rou1            12.596                              |
17:40:38.73|R-> |   12|Rou1            12.596                              |

%NETTRACE-I_CONNECTING, connecting to trace collector...
%NETTRACE-I-CHANTRACED, 1 channel is currently being traced
%NETTRACE-I-PROC_TRPT, Processing ROU_SYNC tracepoint
```

B.4 TRACE Messages

TRACE messages are produced either by the TRACE utility or by the detached TRACE collecting process. Messages produced by the detached process are displayed as OPCOM messages. The detached TRACE collector will always produce an OPCOM message just before it exits, giving the reason why it is exiting.

TRACE messages have the format:

```
%NETTRACE-%-string, message
```

where % is a single character indicating the severity of the message and *string* is an abbreviation of the message text.

Messages are listed according to severity. There are four classes:

-F- Fatal messages indicate that the system cannot continue the process and tracing will end.

-E- Error messages indicate that the output or the program result is incorrect, but that the system may continue with tracing.

-W- Warning messages indicate that the command may have performed some, but not all, of your request. You may have to check the command or output.

-I-
-S- Information or success messages indicate that the system has performed your request. This message can appear during or at completion of your request.

A brief description of each message provides the reason for the message, and a possible remedy for the problem if appropriate.

In addition to these messages (described in Sections B4.1 to B4.4), the analyzed trace file may contain entries of the form:

n trace records lost

Trace records are lost when they are being produced more quickly than they can be written to the terminal or to a file (although they are more likely to occur during LIVE tracing). To avoid trace records being lost, use the START command qualifiers as follows:

- Specify a smaller value for CAPTURE_SIZE
- Specify a larger value for BUFFER_SIZE
- Specify a larger value for MAXIMUM_BUFFERS

B.4.1 Fatal messages

%NETTRACE-F-ERROR, unexpected error

An unexpected error has occurred. An accompanying message provides further information.

%NETTRACE-F-INITERROR, error occurred during initialization

An unexpected error has occurred during initialization. An accompanying message provides further information.

%NETTRACE-F-RTLERROR, unexpected error using run time library routine

An unexpected error has occurred using the run time library routines. An accompanying message provides further information.

%NETTRACE-F-SMGERR, unexpected error using screen management routines

An unexpected error has occurred using screen management routines. An accompanying message provides further information.

%NETTRACE-F-SYSIO, unexpected i/o error

An unexpected i/o error has occurred. An accompanying message provides further information.

B.4.2 Error Messages

%NETTRACE-E-CONFAIL, failed to connect to trace collector

TRACE failed to connect to the trace collector object. An accompanying message gives the reason for the failure.

%NETTRACE-E-ERRCRMBX, error creating mailbox *name*

An error occurred creating the mailbox that TRACE uses to communicate with the detached trace collector *name*. An accompanying message gives the reason for the failure.

%NETTRACE-E-ERRDETPROC, error in creating detached process *name*

An error occurred creating the detached trace collector *name*. An accompanying message gives the reason for the failure.

%NETTRACE-E-ERR_FIND_SYMBOL, error finding *symbol*

An error occurred when trying to activate a trace analysis routine. The symbol could not be found in the sharable image.

%NETTRACE-E-ERRSTART, error starting trace

Tracing could not be started. An accompanying message gives the reason for the failure.

%NETTRACE-E-ERRSTOP, error stopping detached process

An error occurred stopping a detached trace collector process. An accompanying message gives the reason for the failure.

%NETTRACE-E-FILEANALYZED, file is already being analyzed

The file you have specified in an ANALYZE command is already being analyzed by another user or process.

%NETTRACE-E-INSFQUO, *quotaname* quota must be at least *n*

The current value for the user's quota *quotaname* is too small for NETTRACE to run. Use AUTHORIZE to increase the quota to at least *n*.

%NETTRACE-E-INVAL_TRPT, invalid tracepoint specified

You have specified an invalid tracepoint. Reenter the command, specifying a correct tracepoint.

%NETTRACE-E-INVRANGE, invalid range *range*

range is not a valid range of values. A valid range is either a single value, or a range of values, or *. For example, 4-5.

%NETTRACE-E-INVREC, invalid record in file

The file being analyzed is either not a valid trace file, or contains corrupted data.

%NETTRACE-E-INVSHOW, invalid SHOW command

An error occurred parsing a SHOW command. An accompanying message provides further information.

%NETTRACE-E-INVSTART, invalid START command

An error occurred parsing a START command. An accompanying message provides further information.

%NETTRACE-E-INV_VALUE, invalid value *value*

The value specified is not a valid number.

%NETTRACE-E-LINKERR, error on link to trace collector

The link to the trace collector failed. An accompanying message gives the reason for the failure.

%NETTRACE-E-LIVEACTIVE, live tracing is already active

You attempted to start tracing while live tracing was already active.

%NETTRACE-E-LTCINITERR, trace collector *name* failed to initialize

The detached trace collector *name* did not initialize correctly. Submit an SPR - refer to Chapter 6 for details.

%NETTRACE-E-MBXERR, unexpected mailbox error

An error occurred on the mailbox that trace uses to communicate with the detached trace collector. An accompanying message gives the reason for the failure.

%NETTRACE-E-MBXREAD, mailbox read timed out

A read operation timed out on the mailbox used between trace and the detached trace collector process. This may happen if the detached trace collector exited abnormally.

%NETTRACE-E-MBXWRITE, mailbox write timed out

A write operation timed out on the mailbox used between trace and the detached trace collector process. This may happen if the detached trace collector exited abnormally.

%NETTRACE-E-NOPARANALYZE, *parameter* is not valid while analyzing a file

You cannot specify *parameter* while analyzing a file.

%NETTRACE-E-NOPARLIVE, *parameter* is not valid while tracing live

You cannot specify *parameter* while live tracing is active.

%NETTRACE-E-OPENERR, error opening *filename*

An error occurred opening file *filename*. An accompanying message gives the reason for the failure.

%NETTRACE-E-PRNOTFOUND, process *process-name* not found -use SHOW command for list

A STOP command failed to find process *process-name*.

%NETTRACE-E-NOT_SHOW_NODE, cannot show another node while tracing live

An attempt was made to show tracing on one node while connected to another node. Trace can only be connected to one node at a time.

%NETTRACE-E-PROTERR, protocol error on link to trace collector

An unexpected protocol error occurred in a message to the trace collector. Submit an SPR - refer to Chapter 6 for details.

%NETTRACE-E-PROT_VERS, unknown protocol version analysis version *n/m*

The protocol version of the analysis routine that was activated is unknown. It may be incompatible with the version of NETTRACE being used.

%NETTRACE-E-READERR, error reading file *filename*

An error occurred reading file *filename*. An accompanying message gives the reason for the failure.

%NETTRACE-E-RMSERR, error using RMS routines

An error occurred using RMS routines to access a file. An accompanying message gives the reason for the failure.

%NETTRACE-E-SHOW_OPTION, specify SHOW TRACE_COLLECTOR or SHOW KEY

To use the SHOW command, you must specify TRACE_COLLECTOR or KEY. See Section B.2.11 for details.

%NETTRACE-E-SHOW_PEND, SHOW command is already pending

A SHOW command was issued with a previous SHOW command still outstanding. Wait for the first command to complete before issuing the next SHOW command.

%NETTRACE-E-STARTPEND, previous START command pending

A previous start command has not completed. Submit an SPR - refer to Chapter 6 for details.

%NETTRACE-E-STOP_RECORD, TRACE_CAPTURE has stopped recording

The trace capture module has stopped recording. This is either due to an error, or because the communications software being traced has been turned off.

%NETTRACE-E-UNKNOWN_TRPT, unknown tracepoint protocol type found

A trace record with an unknown protocol type has been found. Submit an SPR -refer to Chapter 6 for details.

%NETTRACE-E-VERSKEW, version skew detected connecting to trace collector

NETTRACE and the trace collector software are operating incompatible protocol versions. Refer to the product SPDs, and check that the software on the two systems is compatible.

%NETTRACE-E-WRITEERR, error writing file *filename*

An error occurred writing a record to *filename*. An accompanying message gives the reason for the failure.

B.4.3 Warning Messages

%NETTRACE-W-INVTRPT, invalid tracepoint *tracepoint-name*

The syntax of *tracepoint-name* is invalid. For a description of the valid tracepoint syntax, see Tables B-1.

%NETTRACE-W-MISSING, *number* records missing from input files

The trace sequence being analyzed contains a number of missing records. This may occur if a trace file had been deleted.

%NETTRACE-W-NEWSEQ, new trace sequence started

A new file is being analyzed that starts a new sequence of trace records. This sequence does not follow on directly from the previous file that was analyzed.

%NETTRACE-W-NOPARENT, there is no parent to which to attach

The ATTACH/PARENT command was issued from a process that was not a subprocess.

%NETTRACE-W-TRPT_NOT_FOUND, tracepoint *tracepoint-name* not found

Tracepoint *tracepoint-name* is not a valid tracepoint-name.

%NETTRACE-W-VALSET, value of *parameter* set to *value*

A parameter has had its value changed to the value shown. This is because the value requested lies outside the permitted range, or the value chosen for another parameter affects this parameter.

B.4.4 Information and Success Messages

%NETTRACE-I-CHANTRACED, 1 channel is currently being traced
%NETTRACE-I-CHANSTRACED, *n* channels are currently being traced

Tracing has been started and *n* channels are currently being traced. This is the number of lines or channels that match the tracepoint specified. However, if new channels are created they will also be traced. Use SHOW /FULL to obtain a list if channels being traced.

%NETTRACE-I-CONNECTING, connecting to trace collector ...

Your process is trying to connect to the trace collector.

%NETTRACE-I-CREATEFILE, file *filename* has been created

A new output file has been created.

%NETTRACE-I-DETSTART, detached trace collector *process-name* has been started

A detached trace collector process has been started.

%NETTRACE-I-END_SAVED, end of saved trace records

If you have been using the BACK and NEXT commands, this message is displayed when you finish re-examining saved trace records and resume looking at new records.

%NETTRACE-I-EOFILE, end of input reached

There are no more trace records in the file being analyzed.

%NETTRACE-I-FOUNDTRPT, *tracepoint-name* tracepoint found - will be ignored

A tracepoint has been found, but will not be analyzed because another tracepoint type has already been found. Use ANALYZE /TRACE_LEVEL=protocol_identifier to analyze this tracepoint.

%NETTRACE-I-LTCEXIT, trace collector *trace-collector-name* has exited

A trace collector has exited. This will be accompanied by an OPCOM message giving the reason why the process exited.

%NETTRACE-I-PROCFILE, processing file *filename*

The file has been opened and is being analyzed.

%NETTRACE-I-PROCTRPT, processing *name* tracepoint

A tracepoint has been found which is now being analyzed.

%NETTRACE-I-RESSCROLL, resuming scrolling

If you are tracing live, then scrolling automatically resumes when you finish looking at saved records.

%NETTRACE-S-RETURNED, control returned to process *process-name*

This message is returned by the parent process when you log out of the subprocess.

%NETTRACE-I-SHOW_DONE, SHOW command complete - press RETURN to continue

A SHOW command has completed. Press RETURN to remove the current show display and restore the previous display.

%NETTRACE-I-STARTED, tracing has been started

Tracing has been started. The tracepoint specified must match the actual lines being traced exactly for trace data to be collected.

C
Using the Remote Console

The DECrouter 2000 does not have a port dedicated to a console. Instead, it accepts connections over the DECnet console carrier. This means that a terminal connected to a node on the same Ethernet as the DECrouter 2000 can communicate with the DECrouter 2000 as a console terminal.

Using this remote console, you can issue commands to:

- Control the operation of the system
- Get information
- Test the communications lines

The following sections show how to connect a terminal as a console, and describe the features available.

C.1 Connecting to the DECrouter 2000

The first task is to create the logical connection between your terminal and the DECrouter 2000. The way you do this is different for VAX/VMS and ULTRIX systems. Use Section C.1.1 for VAX/VMS systems or Section C.1.2 for ULTRIX systems.

C.1.1 Connecting from a VAX/VMS System

To connect from a load host that runs VAX/VMS:

1. Log in
2. Start NCP:

   ```
   $ RUN SYS$SYSTEM:NCP
   ```

3. Enter the CONNECT command:

   ```
   NCP> CONNECT NODE DECrouter
   ```

   If a service password is defined to prevent unauthorized use of the console, use the following form of the CONNECT command:

   ```
   CONNECT NODE DECrouter SERVICE PASSWORD password
   ```

   where *password* is the service password.

Once the connection is complete, the following prompt appears on your terminal:

```
>>>
```

C.1.2 Connecting from an ULTRIX System

To connect from a load host that runs ULTRIX:

1. Log in
2. Enter the ccr command:

   ```
   % ccr -n DECrouter
   ```

   If a service password is defined to prevent unauthorized use of the console, use the following form of the connect command:

   ```
   % ccr -n DECrouter -p password
   ```

   where *password* is the service password.

When the connection is complete, the following prompt appears on your terminal:

```
>>>
```

C.2 Halting the Software

The HALT command stops the communications software. If the DECrouter 2000 is in the middle of loading software or dumping the system, the HALT command stops these operations. The HALT command can also be used to stop the running software. The form of the command is:

```
>>> HALT
```

Use this command before using any of these commands:

- BOOT
- DUMP
- TEST

C.3 Testing the DECrouter 2000

The TEST command lets you test the synchronous communications ports, their modem control lines, and the adapter cables. Another form of the command lets you continuously test the complete DECrouter 2000.

The first format of the command is:

```
>>> TEST type channel /qualifier
```

where:

type — is one of the words: DATA_LOOPBACK, MODEM_SIGNAL, or CABLE_TYPE. This parameter determines the sort of test you want to do.

channel — defines which of the synchronous lines the test is to run on. You can use one or more of the following:

/CH0
/CH1
/CH2
/CH3
/ALL

/qualifier are qualifiers that can be added to the TEST DATA_LOOPBACK command to specify the sort of test you want to do. You can use one of the following values:

/INTERNAL
/MANUAL
/LOCAL
/REMOTE

The command to test continuously the DECrouter 2000 is:

```
>>> TEST CONTINUOUS
```

The following sections show what each sort of test does, and gives some examples of their use.

C.3.1 Data Loopback Tests

The data loopback tests use a connector in the communications path. This loopback connector joins the transmit and receive paths so that all information sent is returned. The tests send test data along the line and wait for the looped data to return. This is compared with what was transmitted. Differences in the two copies of the data indicate that an error on the path is corrupting the data.

Figure C-1 shows the points where information can be looped back. You use different forms of the TEST DATA_LOOPBACK command for each point, as the figure indicates.

For example, to loopback through the local modem on channel 1, you use either:

```
>>> TEST DATA_LOOPBACK/CH1/LOCAL
```

or

```
>>> TEST DATA_LOOPBACK/CH1/MANUAL
```

Use the first command if your modem recognizes the Local Loop modem signal. Use the second if you have to manually set the modem into loopback mode (for example, by pressing a switch on the modem).

Section 5.3 has more information on how to use these commands to test a communications line.

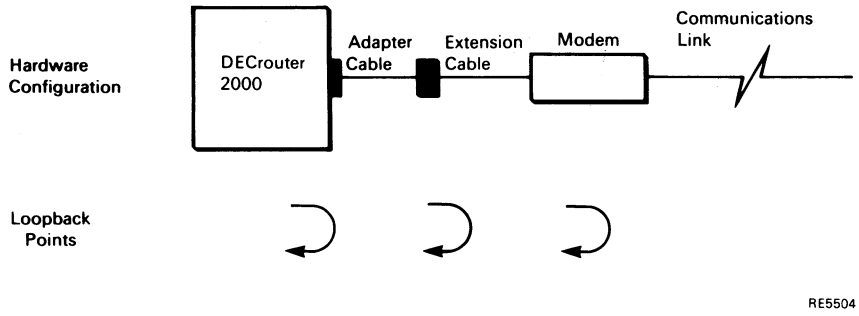

Figure C-1: Loopback Points on a Communications Path

C.3.2 Modem Signal Tests

The second set of tests operate on the modem signals between the DECrouter 2000 and the local modem. The signals can be looped back either at the DECrouter 2000's synchronous port, or at the end of the adapter cable (see Figure C-2).

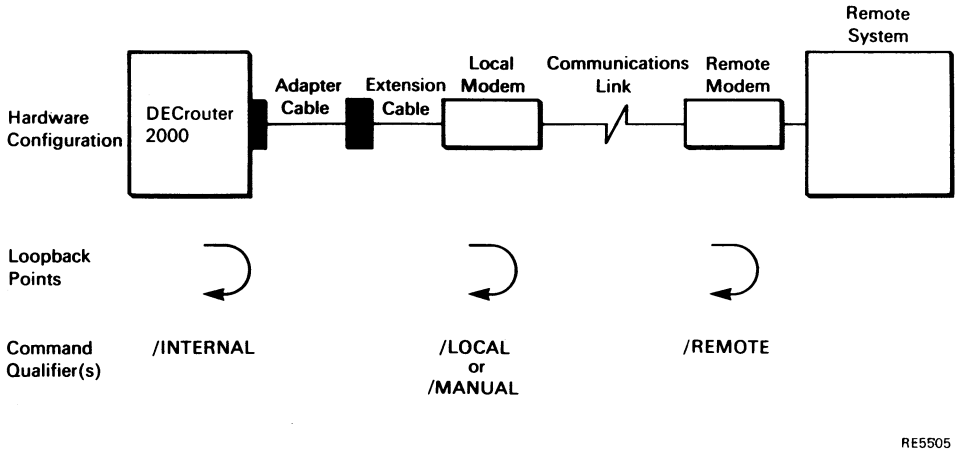

Figure C-2: Loopback Points that can be used with the Modem Signal Test

To do a modem signal test you:

1. Attach a loopback connector to one of the loop points
2. Enter:

   ```
   >>> TEST MODEM_SIGNAL/CHn
   ```

   In the command you use, replace *n* by the number of the line you want to test.
3. If the test is successful, the following message appears on your terminal:

   ```
   Successful Modem Signal Loopback on Channel n
   ```
4. If the test fails, the following message appears on your terminal:

   ```
   Modem Signal Loopback Failed on Channel n
   ```

One use of these tests is to isolate a faulty adapter cable. First, attach the 50-way loopback connector to the DECrouter 2000's synchronous port and run the test. Then repeat the test, but this time with an appropriate loopback connector on the end of the adapter cable. If there is a faulty adapter cable, the first test passes but the second fails.

C.3.3 Cable Type Tests

There is a range of adapter cables available for the DECrouter 2000. These provide easy connection to various interface standards. Each type of cable contains a unique identifier.

Using this test, you can find out the type of cables attached to the DECrouter 2000.

To find out the cable attached to a particular line, make sure there is no loopback connector on the port, and then use:

```
>>> TEST CABLE_TYPE/CHn
```

To find out all the cables attached to the DECrouter 2000, make sure there are no loopback connectors on any port, and then use:

```
>>> TEST CABLE_TYPE/ALL
```

For each cable, the test displays a message like this:

type **Cable on Channel** ***n***

If, however, any port has a 50-way loopback connector on it, the test cannot read the cable type. So, it displays a message like this:

H3199 Loopback on Channel ***n***

If the test cannot determine the type of cable, it displays a message like this:

Unsupported Cable on Channel ***n*****, Cable Code** ***xxxx***

If this message appears for a supported type of cable, check that the cable is correctly connected to the socket, and then try the test again. If the message appears again, the adapter cable is probably faulty.

C.3.4 Soak Testing DECrouter 2000

The DECrouter 2000's internal tests are usually run during power up or when the system is being reloaded. You can, however, run them continuously to soak test an DECrouter 2000. To do this you:

1. Attach any loopback connectors you want to use to the DECrouter 2000's synchronous ports.
2. Enter:

```
>>> TEST CONTINUOUS
```

The unit will now continuously run the internal test program. To reload the system, power down the DECrouter 2000, wait three seconds, and power it up again.

C.4 Getting Information on the DECrouter 2000's Status

Use the SHOW command to display status information. The command is as follows:

```
>>> SHOW STATE
```

The STATE can be one of the following:

- Halted
- Requesting Boot
- Booting
- Deferring Boot
- Requesting Dump
- Dumping
- Deferring Dump
- Running

For example:

Initializing
Software has just loaded, this state is short lived, therefore rarely seen.

Requesting Configuration
Software has requested the configuration permanent database from the load host.

Loading Configuration
The configuration is being down-line loaded.

Running
The configuration has been loaded successfully and the software is running.

Running (Configuration Errors)
There was an error in the configuration. The software is running, though it may not be functioning as expected.

C.5 Reloading the System

You can reload the system with the BOOT command, which causes the system to:

1. Execute the internal test to check the major components in the DECrouter 2000.

2. Issue a load request on the Ethernet.
3. Accept the image from the load host.
4. Read any configuration information from the load host.
5. Start operation.

NOTE

If you need a record of the system's current state, use the DUMP command before using BOOT.

C.6 Dumping the Contents of Memory

Occasionally you may need to dump the contents of the system. The remote console has facilities for doing this, although you would normally use the DECrouter 2000's DUMP switch.

Using the DUMP command at the remote console causes the following to occur:

1. The DECrouter 2000 sends a dump request on the Ethernet and waits for a host node to reply.
2. When a host answers the request, the DECrouter 2000 sends the contents of memory (including the internal error log) to that node.
3. When the dump is complete, the DECrouter 2000 runs its internal test and reboots.

C.7 Console Password

Some of the remote console commands can greatly affect the operation of the DECrouter 2000 (for example, BOOT and DUMP). To help prevent misuse of these commands by unwitting or malicious users, you can set up a password for the remote console. Then, each time anyone wants to attach a terminal as a remote console, they must specify the password (see Section C.1).

C.7.1 Setting a Password

To define a password, use the SET PASSWORD command. The command is as follows:

```
>>>SET PASSWORD password
```

where *password* is up to 16 hexadecimal digits.

For example:

```
>>>SET PASSWORD FEFEFEFEFEFEFEFE
```

The system will ask for confirmation of the password, when you have to enter the password again. When giving this confirmation, the system does not echo the password.

To change the password, simply use this command again.

C.7.2 Changing the Password

As with all passwords, it is worth changing the service password from time to time. This helps to maintain security. To change the password, use the SET PASSWORD command again.

C.7.3 Clearing a Password

To attach the remote console, you have to supply the service password. If you have forgotten what this password is, you can clear the existing password by powering up the DECrouter 2000 as follows:

1. Hold in the DUMP switch.
2. Insert the power cord.
3. Release the DUMP switch when the display shows a value of 1.

Once you've done this, you can set a new password using the SET PASSWORD remote console command.

C.8 Getting Help

To get a listing of the commands, and their formats, use the HELP command:

```
>>> HELP
```

This produces a display like this:

```
Available Commands -
B[oot]
DU[mp]
H[alt]
SE[t] PASSWORD
SH[ow] (PASSWORD,STATE)
TE[st] keyword [/keyword_qualifiers...]
[...] - Optional, (...) - Select One
```

C.9 Detaching the Console

When you have finished using the remote console, press <CTRL/D>. On a VAX/VMS host, the NCP prompt appears on your terminal. On an ULTRIX host, the system command prompt appears.

C.10 Command Summary

Table C-1 shows the format of the remote console commands you can use.

Table C-1: Summary of Remote Console Commands

Command Name		Parameter	Qualifiers
BOOT		None	None
DUMP		None	None
HALT		None	None
HELP	TEST	None	None
SET	PASSWORD	hex-password	None
SHOW	PASSWORD STATE	None	None

Table C-1: Summary of Remote Console Commands (Cont.)

Command Name		Parameter	Qualifiers
TEST	CABLE_TYPE DATA_LOOPBACK MODEM_SIGNAL	None	/ALL /CHn /INTERNAL /LOCAL /MANUAL /REMOTE
TEST	CONTINUOUS	None	None

D
Adapter Cable Pin Signals

D.1 Supported Adapter Cable Pin Signals

This appendix describes the pin signals used in the following:

- RS-422 adapter cable (see Section D.1.1)
- RS-423 adapter cable (see Section D.1.2)
- V.35 adapter cable (see Section D.1.3)
- V.24 adapter cable (see Section D.1.4)
- V.24/RS-232 adapter connector (see Section D.1.5)
- X.21 adapter cable (see Section D.1.6)

In each section, a diagram shows the pin positions at the modem end of the adapter cable and a table lists the signal names. Throughout these tables, (A) or (B) after a signal name refers to wires A and B of a twisted pair. The description of the V.24/RS-232 adapter connector includes both the male and the female connectors.

D.1.1 RS-422 Adapter Cable: 37-Way Pins

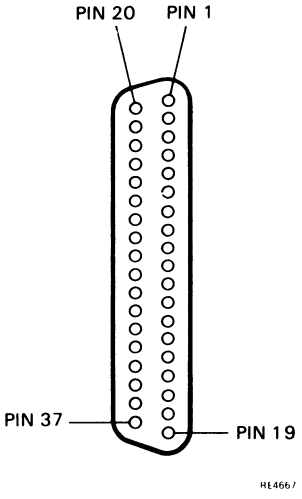

Figure D-1: 37-Way D-Type Connector (Male)

Pin	Signal Name	Pin	Signal Name
4	TX DATA (A)	18	TEST 1
5	TX CLOCK (A)	19	DTE GROUND
6	RX DATA (A)	20	DCE GROUND
7	RTS (A)	22	TX DATA (B)
8	RX CLOCK (A)	23	TX CLOCK (B)
9	CTS (A)	24	RX DATA (B)
10	LOCAL LOOP	25	RTS (B)
11	DSR (A)	26	RX CLOCK (B)
12	DTR (A)	27	CTS (B)
13	DCD (A)	29	DSR (B)
14	REM.LOOP	30	DTR (B)
15	RI	31	DCD (B)
16	SPEED	35	CLOCK (B)
17	CLOCK (A)	37	DTE GROUND

D.1.2 RS-423 Adapter Cable: 37-Way Pins

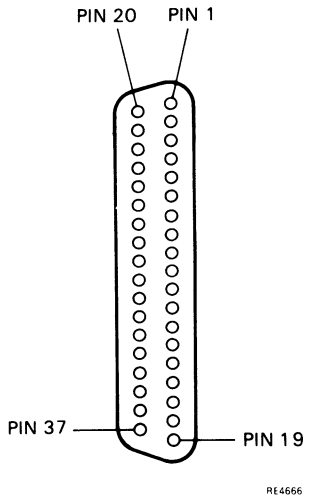

Figure D-2: 37-Way D-Type Connector (Male)

Pin	Signal Name	Pin	Signal Name
4	TX DATA	18	TEST 1
5	TX CLOCK (A)	19	DTE GROUND
6	RX DATA (A)	20	DCE GROUND
7	RTS	22	DTE GROUND
8	RX CLOCK (A)	23	TX CLOCK (B)
9	CTS (A)	24	RX DATA (B)
10	LOCAL LOOP	25	DTE GROUND
11	DSR (A)	26	RX CLOCK (B)
12	DTR	27	CTS (B)
13	DCD (A)	29	DSR (B)
14	REM.LOOP	30	DTE GROUND
15	RI	31	DCD (B)
16	SPEED	35	DTE GROUND
17	CLOCK	37	DTE GROUND

D.1.3 V.35 Adapter Cable: 34-Way Pins

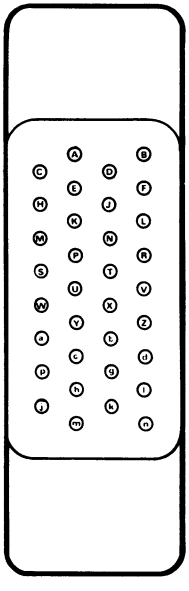

Figure D-3: 34-Way Square Connector (Male)

Pin	Signal Name	Pin	Signal Name
a	TX CLOCK (B)	R	RX DATA (A)
B	DTE GROUND	S	TX DATA (B)
C	RTS	T	RX DATA (B)
D	CTS	U	CLOCK (A)
E	DSR (A)	V	RX CLOCK (A)
F	DCD/I (A)	W	CLOCK (B)
H	DTR	X	RX CLOCK (B)
J	RI	Y	TX CLOCK (A)
P	TX DATA (A)		

D.1.4 V.24 Adapter Cable: 25-Way Pins

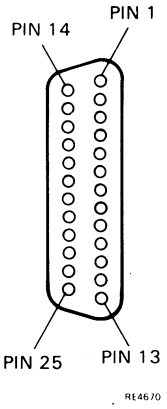

Figure D-4: 25-Way D-Type Connector (Male)

Pin	Signal Name	Pin	Signal Name
2	TX DATA	17	RX CLOCK
3	RX DATA	18	LOCAL LOOP
4	RTS	20	DTR
5	CTS	21	REM.LOOP
6	DSR	22	RI
7	DTE GROUND	23	SPEED SELECT
8	DCD	24	CLOCK
15	TX CLOCK	25	TEST INDICATE

D.1.5 V.24/RS-232-C Adapter Connector: 25-Way Pins

Unlisted pin numbers are not connected.

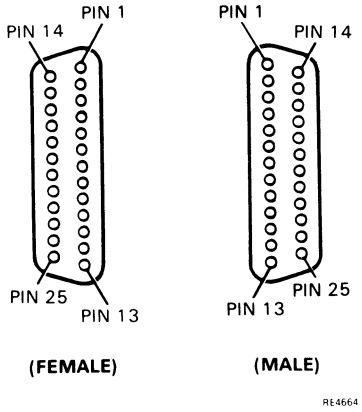

Figure D-5: 25-Way D-Type Connector (Male and Female)

Pin (male)	Signal Name	Pin (female)
2	TX DATA	2
3	RX DATA	3
4	RTS	4
5	CTS	5
6	DSR	6
7	GROUND	7
8	DCD	8
15	TX CLOCK	15
17	RX CLOCK	17
20	DTR	20
22	RI	22
24	CLOCK	24
25	TEST IND	25

D.1.6 X.21 Adapter Cable: 15-Way Pins

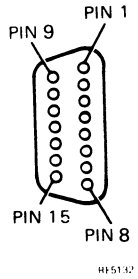

Figure D-6: 15-Way D-Type Connector (Male)

Pin	Signal Name	Pin	Signal Name
2	TX DATA A	9	TX DATA B
3	CONTROL A	10	CONTROL B
4	RX DATA A	11	RX DATA B
5	INDICATION A	12	INDICATION B
6	SIGNAL TIMING A	13	SIGNAL TIMING B
7	BYTE TIMING A	14	BYTE TIMING B
8	DTE GROUND DRAIN WIRE	SHELL	SHIELD BRAID

The DECrouter 2000 needs correct use of the I signal to receive data. To connect the system to interfaces that do not provide these signals, you will need extra, network-specific cabling.

Index

D

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor				Excellent
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

_____

_____

_____

_____

General comments:

_____

_____

_____

_____

Suggestions for improvement:

_____

_____

_____

_____

Name _____ Date _____

Title _____ Department _____

Company _____ Street _____

City _____ State/Country _____ Zip Code _____

digital™

Digital Equipment Company Limited
Networks and Communications Europe
PO Box 121 READING
Berkshire RG2 0TU
England