FDDI Test [6]

The Fiber Distributed Data Interface (FDDI) test detects and isolates faults in the communications link between two Cray Research systems with the FCA-1 (FDDI) adapters installed.

This chapter explains the execution of the FDDI test. It covers the following topics:

- Understanding FDDI
- Getting started with the FDDI test under UNICOS
- Execution examples
- FDDI test menus
- FDDI statistical information
- FDDI test commands
- FDDI test modes
  - Synchronous active-and-passive mode
  - Asynchronous active-and-passive mode
  - IOS software echo mode
  - Ring loopback mode
- Warning situations

6.1 Understanding FDDI

The Fiber Distributed Data Interface (FDDI) connection developed for Cray Research computer systems conforms to the ANSI FDDI specification. It is only available on Cray Research systems with an IOS model E (IOS-E) running UNICOS 8.0 or later.

For additional information on FDDI, see the following documents:

- The `fddi`(4) man page on UNICOS 8.0 (or later)
- *FDDI MAC (Media Access Protocol) Specification (FDDI-MAC)*, document number X3.139-1987, November 5, 1986

- *FDDI PHY (Physical Layer Protocol) Specification (FDDI-PHY)*, document number X3.148-1988, June 30, 1988
- *FDDI PMD (Physical Medium Dependent) Specification (FDDI-PMD)*, document number X3.166-1990, September 28, 1989
- *FDDI SMT (Station Management) Specification (FDDI-SMT)*, document number X3T9.5/84-49, Rev 7.2, June 25, 1992
- *RFC 1390 Transmission of IP and ARP over FDDI networks*, January 1993. D. Katz
- *Logical Link Control Specification (802.2 LLC)*, document number 802.2-1985, July 16, 1984

FDDI is a 100-Mbit/s token-ring network that can be configured to support a sustained transfer rate of approximately 80 Mbit/s (10 Mbyte/s). An FDDI ring consists of a set of stations logically connected as a serial string of stations and media to form a closed loop. Information is transmitted sequentially from one station to the next; each station regenerates and repeats the information.

From the user's perspective, the *frame* is the basic unit of information to and from the network. The maximum frame size on an FDDI network is 4500 bytes, which includes 2 bytes of preamble, 1 byte of start delimiter (SD), 1 byte of frame control (FC), 6 bytes of destination MAC address (DA), 6 bytes of source address (SA), 4478 bytes of information (INFO), 4 bytes of frame check sequence (FCS), and 2 bytes of ending delimiter/frame status (ED/FS). The FC, DA, and SA fields make up what is referred to as the *MAC header*. Because of the way UNICOS implements IP (Internet protocol) through the FCA-1, there are 3 bytes of padding before the MAC header.

When OLNET sends a data buffer to the FDDI network, it must allocate enough space for the pad bytes, the MAC header, and the variable-length INFO field. Stated another way, the buffer must contain 16 bytes (padding + MAC header) with an INFO field from 0 to 4478 bytes in size.

Several types of frames (LLC, SMT, MAC, and so on) are defined by the FDDI standard and can be placed on an FDDI ring. The addressing fields of FDDI frames cannot target a specific logical path (like the header of an NSC message allows you to do). Most of these predefined frame types are received by various daemons for station management, TCP/IP transmission, and so on. For example, while the SMT daemon is receiving SMT frames, no other process can receive SMT frames. However, because each FCA-1 adapter has a different FDDI address, frames can target a specific adapter.

Because other daemons are using up FDDI frames, OLNET needs a way to send its own frame that will be ignored by other daemons. There are eight predefined frame types for this purpose, known as the IMPLEMENTOR frames. OLNET registers with the UNICOS driver software that it wants to receive one of the types of IMPLEMENTOR frames that is coming in on a specific FCA-1 adapter.

Remember, only one application can receive each type of IMPLEMENTOR frame from a specific FCA-1 adapter. Therefore, you cannot execute an OLNET end-to-end test in which both ends of the test try to go through the same FCA-1 adapter. If a mainframe has two or more FCA-1 adapters, you can execute an OLNET end-to-end test in which both ends of the test are on the same machine, but each OLNET must run through a different FCA-1 adapter.

Under the UNICOS operating system, the FDDI character special files, also known as *device nodes*, have the format `/dev/fddi`*n*`/fd`*xx*, where *n* is a 1-digit number, and *xx* is a 2-digit number that represents the logical path. If your site follows the standard device naming conventions for FDDI devices, you can determine the device path name by using the OLNET `DPM` command, which is available from the FDDI Test Initial menu. For more information on the `DPM` command, see Section 6.6, page 206.

6.2 Getting started with the FDDI test under UNICOS

To execute the FDDI test of OLNET, you need to perform the following tasks:

1. Log in to the Cray system(s) on which you intend to run OLNET.
2. Determine the names of the FDDI devices on your system(s).
3. Invoke OLNET.
4. Enter the FDDI test menu.
5. Set up all necessary FDDI test parameters.
6. Execute an FDDI test mode.

When entering commands in OLNET, the case of characters is important only for device names.

6.3 Execution example

This section contains an example of FDDI test execution from one Cray Research system connected to another Cray Research system via the Cray

Research FDDI hardware. The example contains the procedure for testing each part of the connection and then the entire network connection between the two Cray Research systems. Throughout this example, the two Cray Research systems will be referred to as `cloudy` and `cool` (two fictitious machine names).

1. On `cloudy`, enter the following command to execute OLNET:

   ```
   /etc/diag/olnet
   ```

   The following Main menu is displayed:

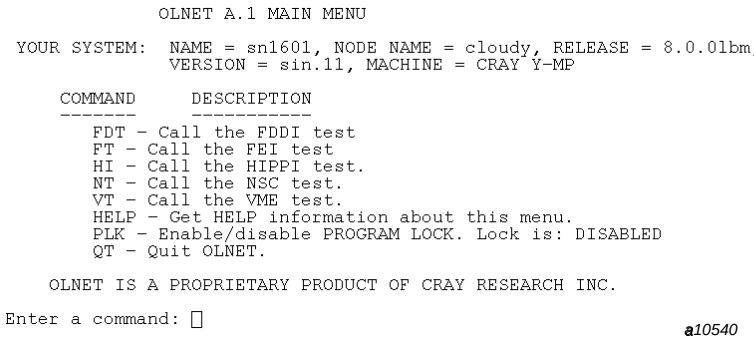

Figure 35. OLNET Main menu

2. From the Main menu, enter `FDT` to select the FDDI test. The FDDI Test Initial menu is displayed:

```
          ********** FDDI TEST INITIAL MENU ***********

   FDDI INITIAL MENU COMMANDS          Current Value(if applicable)
   --------------------------          ----------------------------
    DV - Device path ------------------>  undefined
    DPM  - FDDI Device Path Menu

    HELP - Get HELP information about this menu.

    TMM - Select the OLNET FDDI Test Mode Menu.

    RT - Return to the OLNET Main menu.

  Enter a command:
                                                          a10541
```

Figure 36. FDDI Test Initial menu

3. Select a device path with the DV or DPM command. If you enter DPM, the Device Path menu is displayed:

```
  FDDI device path select menu.


 Select no.   Pathname           status
     1 -   /dev/fddi0/fd00   Device busy
     2 -   /dev/fddi0/fd01   AVAILABLE
     3 -   /dev/fddi0/fd02   AVAILABLE
     4 -   /dev/fddi0/fd03   AVAILABLE
     5 -   /dev/fddi0/fd04   AVAILABLE
     6 -   /dev/fddi0/fd05   Device busy

 Choose one of the following:
  o - Enter a number to select/open a device path.
  o - Enter help.
  o - Press <CR> to exit this routine.

  Enter a command:
                                                  a10542
```

Figure 37. Device Path menu

4. After a valid device path is selected, the FDDI Test Initial menu is updated as shown in the following display:

```
********** FDDI TEST INITIAL MENU ***********

FDDI INITIAL MENU COMMANDS          Current Value(if applicable)
-------------------------          ----------------------------
 DV - Device path ------------------>  /dev/fddi0/fd01
 DPM  - FDDI Device Path Menu

 HELP - Get HELP information about this menu.

 TMM - Select the OLNET FDDI Test Mode Menu.

 RT - Return to the OLNET Main menu.

Enter a command:
```

a10543

Figure 38. Updated FDDI Test Initial menu

5. You are now ready to test this system's FDDI adapter. The first test to be run is the IOS software echo test, which writes data to the IOS software. The IOS software sends the data back immediately.

   Select the FDDI Test Mode menu by entering the `TMM` command. The FDDI Test Mode menu is displayed. Notice that this machine's FDDI address is displayed near the top of the screen and also looked up in the `/etc/ethers` file. If a machine name is associated with this address, it is also displayed.

```
FDDI Test Mode Menu
-------------------
Local address (IEEE): 00:40:a6:00:c8:20 (cloudy-fddi)

Test Parameter Commands          Value
-----------------------          -----
 PC - Pass count --------------->  1
 MP - Messages pass ------------>  10
 ML - Message length ----------->  100
 PT - Pattern type------------->  ADDRESS
 RA - Remote address(Hex) ----->  undefined
 TM - Test mode ---------------->  Active mode
Execute & miscellaneous commands
-------------------------------
 HELP - Get HELP information about this menu.
 EX - Execute: Active mode for FDDI.
 STAT - FDDI Statistical Information Menu
 TR - FDDI driver trace:  DISABLED
 RT - Return to the Initial Menu.
* WARNING - Enter W3 for warning message

Enter a command:
```

a10544

Figure 39. FDDI Test Mode menu

6. To execute the IOS software echo test, you only need to change the value that selects the test to be run. All other default values will work. Enter the `TM` command.

```
The current test mode is: Active mode.
Select one of the following or press <CR> to
leave the current test mode unchanged.

Command        Description
-------        -----------
AM       -----> Active mode
PM       -----> Passive mode
AA       -----> Async active mode
AP       -----> Async passive mode
LBK      -----> Loopback mode
ECHO     -----> IOS software echo mode

 Enter a command:
```

a10545

Figure 40. FDDI test modes

7. Select the IOS software echo test by entering the `ECHO` command.
8. Enter the `EX` command to execute the IOS software echo test. The following display is shown briefly:

```
   OLNET mode ----------> IOS software echo mode
   Current pass count -->      1
   Passes remaining ---->      0

  Tue Sep  7 17:21:00 1993
```

a10546

Figure 41. Executing the IOS software echo test

9. When the IOS software echo test is complete, you see a display similar to the following:

```
Test passes have completed for /dev/fddi0/fd01
IOS software echo mode

Total bytes transmitted =             1000
Total bytes received    =             1000

Elapsed time(HH:MM:SS)  = 00:00:00
       Press <CR> to continue.
```

a10547

Figure 42. IOS software echo test completion message

10. When the IOS software echo test has completed successfully, you can proceed to a test that actually sends data around the FDDI ring. The loopback test places frames on the FDDI ring in which the destination address is the same as the source address. This causes the frame to travel around the entire ring and come back into this machine.

    To execute the loopback test, you only need to change the value that selects the test to be run. All other default values will work. Enter the TM command. The following is displayed:

```
The current test mode is: Active mode.
Select one of the following or press <CR> to
leave the current test mode unchanged.

Command        Description
-------        -----------
AM      -----> Active mode
PM      -----> Passive mode
AA      -----> Async active mode
AP      -----> Async passive mode
LBK     -----> Loopback mode
ECHO    -----> IOS software echo mode

 Enter a command:
```

a10548

Figure 43. FDDI test modes

11. Select the loopback test by entering the LBK command.

12. Enter the EX command to execute the loopback test. The following display is shown briefly. (If you were executing a long-running test, this display would be refreshed every 10 seconds.)

```
    OLNET mode ----------> Loopback mode
    Current pass count -->      1
    Passes remaining ---->      0

   Tue Sep  7 17:17:15 1993
```


a10549

Figure 44. Executing the loopback test

13. When the loopback test is complete, you see a display similar to the following:

```
       Test passes have completed for /dev/fddi0/fd01
       Loopback mode

       Total bytes transmitted =             1000
       Total bytes received    =             1000

       Elapsed time(HH:MM:SS)  = 00:00:00
              Press <CR> to continue.
```


a10550

Figure 45. Loopback test completion message

14. Now that `cloudy` 's loopback capabilities have been checked, repeat steps 1 through 13 on `cool`.

15. You are now ready to execute the end-to-end active-and-passive test between `cloudy` and `cool`. To run this test, you need to tell `cloudy` what `cool` 's FDDI address is, and vice versa. For example, assume that `cloudy` 's FDDI address is `00:40:a6:00:c8:20`, `cloudy` 's machine name on the FDDI ring is `cloudy-fddi`, `cool` 's FDDI address is `00:40:a6:00:d4:e0`, and `cool` 's machine name on the FDDI ring is `cool-fddi`. (The mapping between a machine's FDDI address and its FDDI machine name is contained in the `/etc/ethers` file.)

16. On `cloudy`, load the FDDI address for `cool` in the `Remote address` field. Use the `RA` command from the FDDI Test Mode menu by specifying either the FDDI address or FDDI machine name:

    ```
    ra,00:40:a6:00:d4:e0
    ```

or

```
ra,cool-fddi
```

When the remote address has been set, the `Remote address` field of the FDDI Test Mode menu will be updated to show both the remote FDDI address and, if available, the remote FDDI machine name.

```
FDDI Test Mode Menu
-------------------
Local address (IEEE): 00:40:a6:00:c8:20 (cloudy-fddi)

Test Parameter Commands          Value
-----------------------          -----
 PC - Pass count --------------->  1
 MP - Messages pass ------------>  10
 ML - Message length ----------->  100
 PT - Pattern type------------->  ADDRESS
 RA - Remote address(Hex) ----->  00:40:a6:00:d4:e0 (cool-fddi)
 TM - Test mode ---------------->  Active mode
Execute & miscellaneous commands
--------------------------------
 HELP - Get HELP information about this menu.
 EX - Execute: Active mode for FDDI.
 STAT - FDDI Statistical Information Menu
 TR - FDDI driver trace:  DISABLED
 RT - Return to the Initial Menu.
* WARNING - Enter W3 for warning message

Enter a command:
```

a10551

Figure 46. FDDI Test Mode menu

17. On `cool`, load the FDDI address for `cloudy` in the `Remote address` field. Use the `RA` command from the FDDI Test Mode menu by specifying either the FDDI address or FDDI machine name:

    ```
    ra,00:40:a6:00:c8:20
    ```

    or

    ```
    ra,cloudy-fddi
    ```

    When the remote address has been set, the `Remote address` field of the FDDI Test Mode menu will be updated to show both the remote FDDI address and, if available, the remote FDDI machine name.

```
FDDI Test Mode Menu
-------------------
Local address (IEEE): 00:40:a6:00:d4:e0 (cool-fddi)

Test Parameter Commands          Value
-----------------------          -----
 PC - Pass count --------------->  1
 MP - Messages pass ------------>  10
 ML - Message length ----------->  100
 PT - Pattern type------------->  ADDRESS
 RA - Remote address(Hex) ----->  00:40:a6:00:c8:20 (cloudy-fddi)
 TM - Test mode ---------------->  Active mode
Execute & miscellaneous commands
--------------------------------
 HELP - Get HELP information about this menu.
 EX - Execute: Active mode for FDDI.
 STAT - FDDI Statistical Information Menu
 TR - FDDI driver trace:  DISABLED
 RT - Return to the Initial Menu.
* WARNING - Enter W3 for warning message

Enter a command:
```

a10552

Figure 47. Updated FDDI Test Mode menu

18. For this example, you will set up `cloudy` to be the active side of the transfer and `cool` to be the passive side. Enter the following command to set `cloudy` to active mode:

    ```
    tm,am
    ```

    Enter the following command to set `cool` to passive mode:

    ```
    tm,pm
    ```

19. Always start execution on the passive system first. Enter the EX command on `cool`. The following messages are displayed:

```
Waiting for the first message on /dev/fddi0/fd01
OLNET mode = Passive mode

 Tue Sep  7 17:24:05 1993
```

a10553

Figure 48. Starting execution on the passive system (figure 1)

```
OLNET mode ----------> Passive mode
Current pass count -->      1
Passes remaining ---->      0

Tue Sep  7 17:23:44 1993
```


a10554

Figure 49. Starting execution on the passive system (figure 2)

20. Start execution on the active system. Enter the EX command on `cloudy`. The following message is displayed:

```
OLNET mode ----------> Active mode
Current pass count -->      1
Passes remaining ---->      0

Tue Sep  7 17:23:04 1993
```


a10555

Figure 50. Starting execution on the active system

On successful test completion, the following message is displayed on `cloudy`:

```
Test passes have completed for /dev/fddi0/fd01
Active mode

Total bytes transmitted =            1000
Total bytes received    =            1000

Elapsed time(HH:MM:SS)  = 00:00:00
       Press <CR> to continue.
```


a10556

Figure 51. End-to-end active test completion message

On successful test completion, the following message is displayed on `cool`:

```
Test passes have completed for /dev/fddi0/fd01
Passive mode

Total bytes transmitted =             1000
Total bytes received    =             1000

Elapsed time(HH:MM:SS)  = 00:00:00
       Press <CR> to continue.
```

a10557

Figure 52. End-to-end passive test completion message

6.4 FDDI test menus

After you initialize OLNET and access the Main menu, as described in Section 1.2, page 2, and Section 1.3, page 4, enter **FDT** from the Main menu to display the FDDI Test Initial menu as shown in Figure 53, page 193.

```
********** FDDI TEST INITIAL MENU ***********

FDDI INITIAL MENU COMMANDS          Current Value(if applicable)
-------------------------          ----------------------------
 DV - Device path ------------------>  undefined
 DPM  - FDDI Device Path Menu

 HELP - Get HELP information about this menu.

 TMM - Select the OLNET FDDI Test Mode Menu.

 RT - Return to the OLNET Main menu.

Enter a command:
```

a10558

Figure 53. FDDI test initial menu

If the **TMM** command is entered on the FDDI Test Initial menu, the FDDI Test Mode menu is displayed as shown in Figure 54, page 194.

```
FDDI Test Mode Menu
-------------------
Local address (IEEE): 00:40:a6:00:c8:20 (cloudy-fddi)

Test Parameter Commands          Value
-----------------------          -----
 PC - Pass count --------------->  1
 MP - Messages pass ------------>  10
 ML - Message length ----------->  100
 PT - Pattern type------------->  ADDRESS
 RA - Remote address(Hex) ----->  undefined
 TM - Test mode ---------------->  Active mode
Execute & miscellaneous commands
-------------------------------
 HELP - Get HELP information about this menu.
 EX - Execute: Active mode for FDDI.
 STAT - FDDI Statistical Information Menu
 TR - FDDI driver trace:  DISABLED
 RT - Return to the Initial Menu.
* WARNING - Enter W3 for warning message

Enter a command:
```

a10559

Figure 54. FDDI test mode menu

If the STAT command is entered on the FDDI Test Initial menu, the FDDI Statistical Information menu is displayed as shown in Figure 55, page 194.

```
FDDI (FCA-1) Statistical Information Menu

OPTIONS        DESCRIPTION
-------        -----------
GETULA  -----> Get IEEE Universal Lan Address
GET     -----> Get current driver settings
CDSTATS -----> Clear device statistics
CLSTATS -----> Clear logical path statistics
STATS   -----> Display driver and logical path statistics
DSTRUCT -----> Display device's 'fd_dev' structure
LSTRUCT -----> Display logical path's 'fd_lp' structure
GETVARS -----> Display 'fd_vars' structure
MACNBRS -----> Display MAC neighbor addresses
GET_DAD -----> Display result of duplicate address test
GET_HPC -----> Display 'fdio_hpc_info' structure
ETHERS  -----> Display contents of /etc/ethers file

STYLE   -----> toggle display style to 'show field descriptions'
RT      -----> return to the previous menu.

 Enter a command:
```

a10560

Figure 55. FDDI statistical information menu

6.5 FDDI statistical information

This section describes the statistical information that can be displayed. All but one of the displays is a direct mapping to an available `ioctl`(2) function that can be performed to the FDDI device.

6.5.1 GETULA - Get IEEE universal LAN address

There is one screen for this command.

```
The 'COMM_IOC_GETULA' ioctl function was issued.
This function returns the Universal LAN Address.
Following is the information from the 'netula' structure.

.addr                                              : 00:40:a6:00:c8:20


                Press <CR> to continue.
```

a10561

Figure 56. GETULA screen

6.5.2 GET - Get current driver settings

There are three screens for this command.

```
The 'FDC_GET' ioctl function was issued.
This function returns the device configuration data.
Following is the information from the 'fdio_getset' structure.

.ieee_mac                                                : 00:40:a6:00:c8:20
.fddi_mac                                                : 00:02:65:00:13:04
.ferrno                                                  : 0
.err                                                     : 0
     F_RSP_OK                                            : 0
.des                                                     : 0
.fsw                                                     : 0
.TREQ                                                    : 8
.cc                                                      : 0xb3
     EFCF_CC_SMT                                         : 0x01
     EFCF_CC_LLC                                         : 0x02
     EFCF_CC_MAC                                         : 0x04
     EFCF_CC_IMP                                         : 0x08

    ENTER:  - 'q' to end display.
                - OR -
                - Press <CR> to continue.

  Enter a command:
```

a10562

Figure 57. `Get` screen (1 of 3)

```
     EFCF_CC_IA                                          : 0x10
     EFCF_CC_GA                                          : 0x20
     EFCF_CC_MY                                          : 0x40
     EFCF_CC_MYNSA                                       : 0x80
.padcnt                                                  : 3
.maxwrt                                                  : 10
.maxrd                                                   : 10
.opt                                                     : 0x00
     FDLO_NFRCHK                                         : 0x01
     FDLO_EARLY                                          : 0x02
     FDLO_NERRLOG                                        : 0x04
.rft                                                     : 0x0004
     EFOP_RFT_NONE                                       : 0x0000
     EFOP_RFT_SMT                                        : 0x0001
     EFOP_RFT_LLC                                        : 0x0002
     EFOP_RFT_OLNET                                      : 0x0004
     EFOP_RFT_USCP                                       : 0x0008

    ENTER:  - 'q' to end display.
                - OR -
                - Press <CR> to continue.

  Enter a command:
```

a10563

Figure 58. `Get` screen (2 of 3)

```
     EFOP_RFT_REGT                                  : 0x0010
     EFOP_RFT_IMP_3                                 : 0x0020
     EFOP_RFT_IMP_4                                 : 0x0040
     EFOP_RFT_IMP_5                                 : 0x0080
     EFOP_RFT_IMP_6                                 : 0x0100
     EFOP_RFT_IMP_7                                 : 0x0200
     EFOP_RFT_ALL                                   : 0xffff
.rtmo                                               : 60

               Press <CR> to continue.
```

a10564

Figure 59. `Get` screen (3 of 3)

6.5.3 CDSTATS - Clear device statistics

There is one screen for this command.

```
The 'FDC_CDSTATS' ioctl function was issued.
This function clears the statistics associated with the device you have
chosen.  There is no output from this operation.

              Press <CR> to continue.
```

a10565

Figure 60. CDSTATS screen

6.5.4 CLSTATS - Clear logical path statistics

There is one screen for this command.

```
The 'FDC_CLSTATS' ioctl function was issued.
This function clears the statistics associated with the logical path you have
chosen.  There is no output from this operation.

              Press <CR> to continue.
```

a10566

Figure 61. CLSTATS screen

6.5.5 STATS - Display driver and logical path statistics

There is one screen for this command.

```
The 'FDC_STATS' ioctl function was issued.
This function returns the device and logical path statistics.
Following is the information from the 'commstat' structure.

.index                                                  : 0
.open                                                   : 3
.dev                                                    : 0
.clu                                                    : 1
.iop                                                    : 1
.chan                                                   : 030
.devtype                                                : 10
.physaddr                                               : 0x040a600c820
.dir                                                    : 0
.lpath                                                  : 0
.otime                                                  : 0x44db488004ddef
.pid                                                    : 0

              Press <CR> to continue.
```

a10567

Figure 62. STATS screen

6.5.6 DSTRUCT- Display device's fd_dev structure

There are four screens for this command.

```
The 'FDC_DSTRUCT' ioctl function was issued.
This function returns the device internal data structure.
Following is the information from the 'fd_dev' structure.

.label                                                  : fddiD000
.index                                                  : 0
.iocnum                                                 : 1
.iopnum                                                 : 1
.iopchan                                                : 030
.devtype                                                : 10
.flags                                                  : 0x0c
     FDDF_CONFINGUP                                     : 0x01
     FDDF_CONFINGDN                                     : 0x02
     FDDF_CONFUP                                        : 0x04
     FDDF_DOWNLOADED                                    : 0x08
.smt.UNA.ieee                                           : 00:00:30:00:8e:6f
.smt.UNA.fddi                                           : 00:00:0c:00:71:f6

    ENTER:  - 'q' to end display.
              - OR -
              - Press <CR> to continue.

  Enter a command:
```

a10568

Figure 63. DSTRUCT screen (1 of 4)

```
.smt.DNA.ieee                                          : 00:00:f8:00:00:00
.smt.DNA.fddi                                          : 00:00:1f:00:00:00
.smt.Old_UNA.ieee                                      : 00:00:00:00:00:00
.smt.Old_UNA.fddi                                      : 00:00:00:00:00:00
.smt.Old_DNA.ieee                                      : 00:00:00:00:00:00
.smt.Old_DNA.fddi                                      : 00:00:00:00:00:00
.smt.mac_available                                     : 0
.smt.llc_available                                     : 0
.smt.MAC_DA_Flag                                       : 0
.smt.hi_32_timestamp                                   : 0x4a033f
.mac.ieee                                              : 00:40:a6:00:c8:20
.mac.fddi                                              : 00:02:65:00:13:04
.TREQ.boot                                             : 8
.TREQ.run                                              : 8
.cc.boot                                               : 0xb3
.cc.run                                                : 0xb3
.padcnt.boot                                           : 3

    ENTER:  - 'q' to end display.
              - OR -
              - Press <CR> to continue.

  Enter a command:
```

a10569

Figure 64. DSTRUCT screen (2 of 4)

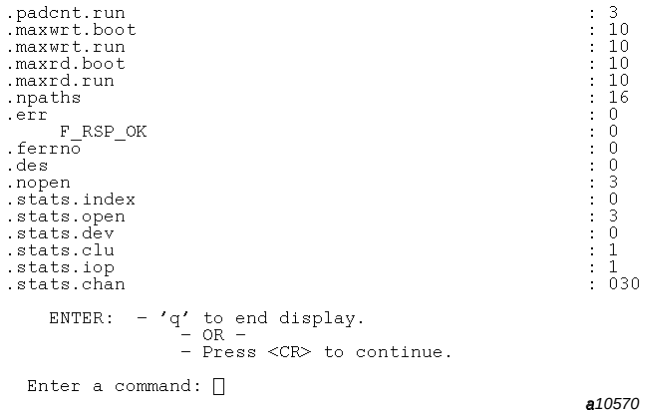

Figure 65. DSTRUCT screen (3 of 4)

```
.stats.devtype                                          : 10
.stats.physaddr                                         : 0x40a600c820
.stats.dir                                              : 0
.stats.lpath                                            : 0
.stats.otime                                            : 0x44db488004ddef
.stats.pid                                              : 0


              Press <CR> to continue.
```

a10571

Figure 66. DSTRUCT screen (4 of 4)

6.5.7 LSTRUCT - Display logical path's `fd_lp` structure

There are three screens for this command.

```
The 'FDC_LSTRUCT' ioctl function was issued.
This function returns the logical path internal data structure.
Following is the information from the 'fd_lp' structure.

.label                                                  : fddiL001
.index                                                  : 1
.dev                                                    : 1
.flags                                                  : 0x04
     FDLF_HALTING                                       : 0x01
     FDLF_CLOSING                                       : 0x02
     FDLF_OPENPATH                                      : 0x04
.rft.boot                                               : 0x00
.rft.run                                                : 0x04
.opt.boot                                               : 0x00
.opt.run                                                : 0x00
.rtmo.boot                                              : 60
.rtmo.run                                               : 60

    ENTER:  - 'q' to end display.
              - OR -
              - Press <CR> to continue.

  Enter a command:
```

a10572

Figure 67. LSTRUCT screen (1 of 3)

```
.rq.na                                                  : 0
.rq.nq                                                  : 0
.wq.na                                                  : 0
.wq.nq                                                  : 0
.err                                                    : 0
     F_RSP_OK                                           : 0
.ferrno                                                 : 0
.des                                                    : 0
.open                                                   : 1
.opid                                                   : 5707
.fsw                                                    : 0
.stats.index                                            : 0
.stats.open                                             : 3
.stats.dev                                              : 0
.stats.clu                                              : 1
.stats.iop                                              : 1
.stats.chan                                             : 030

    ENTER:  - 'q' to end display.
              - OR -
              - Press <CR> to continue.

  Enter a command:
                                                        a10573
```

Figure 68. LSTRUCT screen (2 of 3)

```
.stats.devtype                                          : 10
.stats.physaddr                                         : 0x40a600c820
.stats.dir                                              : 0
.stats.lpath                                            : 0
.stats.otime                                            : 0x44db488004ddef
.stats.pid                                              : 0

              Press <CR> to continue.

                                                        a10574
```

Figure 69. LSTRUCT screen (3 of 3)

6.5.8 GETVARS - Display fd_vars structure

There is one screen for this command.

```
The 'FDC_GETVARS' ioctl function was issued.
This function returns the max number of devices and logical paths.
Following is the information from the 'fd_vars' structure.

.maxdevs                                                : 1
.maxpaths                                               : 16


              Press <CR> to continue.
```

a10575

Figure 70. GETVARS screen

6.5.9 MACNBRS - Display MAC neighbor addresses structure

There is one screen for this command.

```
The 'FDC_GET_MACNBRS' ioctl function was issued.
This function returns this device's ring neighbors.
Following is the information from the 'fdio_mac_neighbors' structure.

.una.ieee                                               : 00:00:30:00:8e:6f
.una.fddi                                               : 00:00:0c:00:71:f6
.dna.ieee                                               : 00:00:f8:00:00:00
.dna.fddi                                               : 00:00:1f:00:00:00


              Press <CR> to continue.
```

a10576

Figure 71. MACNBRS screen

6.5.10 GET_DAD - Display result of duplicate address test

There is one screen for this command.

```
The 'FDC_GET_DAD' ioctl function was issued.
This function returns the results of the duplicate address test.
Following is the information from the 'fdio_dad_results' structure.

.results                                                : 0x02
     FD_DAD_UNKNOWN                                     : 0x01
     FD_DAD_PASS                                        : 0x02
     FD_DAD_FAIL                                        : 0x03


              Press <CR> to continue.
                                                                    a10577
```

Figure 72. GET_DAD screen

6.5.11 GET_HPC - Display fdio_hpc_info structure

There are several screens for this command. Here is the first one:

```
The 'FDC_GET_HPC' ioctl function was issued.
This function returns the information from the HPC.
Following is the information from the 'fdio_hpc_info' structure.

.smt_00[0]                                              : 0xc0
.smt_00[1]                                              : 0x80
.smt_02[0]                                              : 0x00
.smt_02[1]                                              : 0x00
.smt_04[0]                                              : 0x0a
.smt_04[1]                                              : 0x00
.smt_06                                                 : 0x06
.smt_07                                                 : 0x0e
.smt_08                                                 : 0x10
.smt_09                                                 : 0x00
.smt_0A                                                 : 0x00
.smt_0B                                                 : 0x07
.smt_0C[0]                                              : 0x00

    ENTER:  - 'q' to end display.
              - OR -
              - Press <CR> to continue.

  Enter a command:                                          a10578
```

Figure 73. GET_HPC screen

6.5.12 ETHERS - Display contents of /etc/ethers file

There can be many screens for this command depending on the size of your machine's /etc/ethers file.

```
The following table will show the mapping between
the FDDI addresses (in 48-bit ethernet-style) and the
names for those machines, as configured in the file
'/etc/ethers'.

FDDI Address         Name
-----------------    --------------------
ff:ff:ff:ff:ff:ff    broadcast
00:40:a6:00:c8:20    cloudy-fddi
00:40:a6:00:d4:e0    cool-fddi


              Press <CR> to continue.
```

a10579

Figure 74. ETHERS screen

6.5.13 STYLE - Change style of output for display screens

This command acts like a toggle to display either the structure field names or the structure field descriptions. For example, the following shows the GETULA screen with the STYLE command toggled:

```
The 'COMM_IOC_GETULA' ioctl function was issued.
This function returns the Universal LAN Address.
Following is the information from the 'netula' structure.

Universal LAN Address (IEEE format)                 : 00:40:a6:00:c8:20


              Press <CR> to continue.
```

a10580

Figure 75. STYLE screen

6.6 FDDI test commands

This section describes the commands available on the FDDI Test Initial menu and the FDDI Test Mode menu. (This section describes menu execution only. Appendix A, page 267, describes other methods of execution.) FDDI test commands are as follows:

Command	Description
AR	Acknowledgment ratio (required for and applicable to asynchronous active-and-passive mode only). Indicates the number of messages sent by the active system before an acknowledgment message is returned by the passive system. AR is one of the following values:

mm: `1` — Specifies that *mm* messages are sent by the asynchronous active system before the asynchronous passive system returns an acknowledgment message. For example, 100:1 specifies that 100 messages are sent by the asynchronous active system before the asynchronous passive system responds with an acknowledgment message. *mm* is a value in the range 1 through 4096.

mm: `0` — Specifies no return acknowledgment (in effect, a write-only test by the asynchronous active system and a read-only test by the asynchronous passive system). *mm* is a value in the range 1 through 4096.

mm: `RN` — Specifies a random acknowledgment ratio. *mm* indicates the upper range of random values for the acknowledgment ratio and must be a value in the range 1 through 4096.

For example, an acknowledgment ratio of 200:RN specifies that a random number of messages (from 1 through 200) is sent by the asynchronous active system before the asynchronous passive system responds with an acknowledgment message.

The default for AR is 3:1.

```
The current messages/acknowledgement ratio is 3:1

 Select one of the following or press <CR> to
       leave the value unchanged:

     Command      Description
     -------      -----------
      mm:1 ------> mm messages:one acknowledgement.
      mm:0 ------> mm messages:zero acknowledgements.
      mm:RN ------> mm messages:random acknowledgements.

Enter a command:
                                                          a10581
```

Figure 76. Messages/acknowlegment screen

CE Tells OLNET to continue on error. Use the *errorfile* option to specify the file to which error output is written. These options do not appear on the FDDI Test menus. The CE option must be placed between the TMM and EX options in a command-line string. See Section A.2.2, page 270, for more information.

DPM The device path menu (DPM) command allows you to display and dynamically select an FDDI device path (assuming standard FDDI device path naming conventions were used). For more information about device path naming conventions, see fddi(4).

After the DPM command has been executed and if more than one FDDI is connected to your system, a menu option is displayed to choose one of the FDDI connections by selecting a major device path. If only one FDDI is connected to your system (one major path), or you have already selected a major path, a menu containing paths and statuses is displayed. You can select a path from this menu.

```
 FDDI device path select menu.

Select no.   Pathname           status
    1 -   /dev/fddi0/fd00   Device busy
    2 -   /dev/fddi0/fd01   AVAILABLE
    3 -   /dev/fddi0/fd02   AVAILABLE
    4 -   /dev/fddi0/fd03   AVAILABLE
    5 -   /dev/fddi0/fd04   AVAILABLE
    6 -   /dev/fddi0/fd05   Device busy

Choose one of the following:
 o - Enter a number to select/open a device path.
 o - Enter help.
 o - Press <CR> to exit this routine.

 Enter a command:
                                                    a10582
```

Figure 77. FDDI device path select menu

DV Device path name. You must have read/write permission on the device path used by OLNET. Contact your system administrator to obtain these permissions.

```
The current FDDI device path is undefined

Enter a new device path or press <CR>
  to leave the device path unchanged

Following is an example of a device path:
   /dev/fddi0/fd03


 Enter a command:
                                          a10583
```

Figure 78. Device path name screen

errorfile Specifies the file to which error output is written. This option does not appear on the FDDI Test menu. The *errorfile* option must be placed after the **EX** option in a command-line string. See Section A.2.2, page 270, for more information.

EX Executes the test in the test mode specified by the **TM** command. Because the message length and test mode values can be set in any order, when you execute the test OLNET ensures that the message length and test mode values are compatible. If the

combination of message length and test mode values is not compatible, the following message is displayed:

```
 ***********  USER ERROR ***********
Current message length = 4500.
Current test mode      = 'Active mode'.
This is an illegal combination.
For the IOS software echo (ECHO) test, the message
length may be in the range 16 to 4500 bytes.
For all other tests, the range is 16 to 4494 bytes.

            Press <CR> to continue.
```

a10584

Figure 79. Error screen

HELP Gets help for the current menu.

ML, *ml* Message length in bytes. *ml* is a value in the range 16 through 4494 for all test modes except **ECHO**, which allows a range of 16 through 4500. The default for *ml* is 100.

```
The current message length is 100

Select one of the following or press <CR> to
   leave the value unchanged:

  Value             Description
 --------      --------------------------
 16 - 4500       Fixed message length range (for ECHO only).
 16 - 4494       Fixed message length range.
    RN            Random message length

 Enter a command:
```

a10591

Figure 80. Message length screen

MP, *mp* Messages to be generated for each pass. *mp* is a value in the range 1 through 1,000,000. The default for *mp* is 10.

```
The current messages/pass is 10
Select one of the following or press <CR> to
   leave the value unchanged:

 Minimum      Maximum
 -------      -------

     1 <----> 1000000

Enter a command:
```

a10592

Figure 81. Messages/pass screen

PC, *pc* Pass count. *pc* is a value in the range 1 through 1,000,000. The default for *pc* is 1.

```
The current pass count is 1

Choose a value in the following range or
 press <CR> leave the value unchanged:

 Minimum      Maximum
 -------      -------

     1 <-----> 1000000

Enter a command:
```

a10593

Figure 82. Pass count screen

PT, *pt* Pattern type (in 64-bit words). *pt* is one of the following values:

Value	Pattern

AD	Address (default). This sequential address pattern is incremented in each 16-bit parcel of a 64-bit word, as in the following example:
	`000000 000001 000002 000003`
	`000004 000005 000006 000007`

A0	All 1's.
AP	All patterns. A new pattern is generated for each message sent and received. The patterns are

processed in the following order: AD, A0, AZ, S0, SZ, RN, BT.

AZ — All 0's.

BT — Bits. This pattern contains a random number of consecutive 1-bits randomly positioned within a 64-bit word, as in the following example:

```
000001 177770 000000 000000
000000 000000 077770 000000
177777 177777 177600 000000
000000 000000 003777 177700
```

RN — Random. A random pattern is generated for each message sent and received.

S0 — Sliding 1's. This is a 0's data pattern in which a 1-bit is circularly shifted through each 16-bit parcel, as in the following example:

```
000001 000002 000004 000010
000020 000040 000100 000200
```

SZ — Sliding 0's. This is a 1's data pattern in which a 0-bit is circularly shifted through each 16-bit parcel, as in the following example:

```
177776 177775 177773 177767
77757 177737 177677 177577
```

The default for *pt* is AD (address pattern).

For data patterns AP, BT, and RN, OLNET builds a new pattern for each message, thereby requiring extra CPU cycles and possibly reducing the data rate (bytes/second).

```
The current pattern type is ADDRESS

Select one of the following or press <CR> to
leave the current pattern type unchanged.

Command        Description
-------        -----------
RN       -----> RANDOM
AZ       -----> ALL ZEROS
AO       -----> ALL ONES
SO       -----> SLIDING ONES
SZ       -----> SLIDING ZEROS
AD       -----> ADDRESS
BT       -----> BITS
AP       -----> ALL PATTERNS

 Enter a command:
                                                    a10594
```

Figure 83. Pattern type screen

RA, *ra* Remote system's FDDI address or FDDI machine name, as found in the `/etc/ethers` file. If you are using the FDDI address format, the address is similar to an Ethernet 48-bit address where each byte is specified in hexadecimal and separated by a colon.

```
The current remote adapter address is undefined.
Select a new address using either...
- the IEEE 48-bit (Ethernet/Canonical) address form.
- the hostname, as defined in /etc/ethers.
Example: 8:0:4:20:da:2f

Enter a command:
                                                a10595
```

Figure 84. Remote address screen

RT Returns to the previous menu.

TM, *tm* Test mode. *tm* is one of the following values:

AA	Asynchronous active
AM	Synchronous active
AP	Asynchronous passive
PM	Synchronous passive
ECHO	IOS software echo

LBK Loopback

The default for *tm* is AM (synchronous active).

```
The current test mode is: Active mode.
Select one of the following or press <CR> to
leave the current test mode unchanged.

Command        Description
-------        -----------
AM      -----> Active mode
PM      -----> Passive mode
AA      -----> Async active mode
AP      -----> Async passive mode
LBK     -----> Loopback mode
ECHO    -----> IOS software echo mode

 Enter a command:
                                        a10596
```

Figure 85. Test mode screen

TMM Selects the Test Mode menu.

TR Enables or disables the driver trace. The first screen displayed when you turn on a trace file is as follows:

```
  Driver trace is currently  DISABLED
  Select one of the following or press <CR> to
     leave the driver trace unchanged:

   Enter 1 - Set driver trace ENABLED
   Enter 0 - Set driver trace is DISABLED

Enter a command:
                                        a10597
```

Figure 86. Driver trace screen

When you enable a trace file, the following screen is displayed:

```
*** SAVE TRACE INFORMATION ON A FILE ***

 Enter the directory/filename.
 If you want to exit this routine, just enter a <CR>.

Enter the directory/file - >
```


a10598

Figure 87. Trace file enabled screen

W3 Displays a warning message for all OLNET FDDI tests, as follows:

```
        ** WARNING **

  Receiving IMPLEMENTOR frames from other sources
  ------------------------------------------------
  Be careful when interpreting data miscompares reported by the OLNET FDDI
  test suite.  All OLNET FDDI test modes are trying to send and receive
  IMPLEMENTOR frames of a particular type.  IMPLEMENTOR frames on the
  FDDI ring are brought into your machine's IOS because the destination
  address field of the frame is your machine's address.  While your OLNET
  process has a read 'posted', another entity on your FDDI network could
  send your machine IMPLEMENTOR frames of the type OLNET is waiting to
  receive.  Your OLNET process will receive these IMPLEMENTOR frames,
  compare them to what it expected to receive and report a data miscompare.

  Realistically, this situation should not occur because you, as a person
  trying to diagnose an FDDI problem, should know what other processes on
  machines on your FDDI ring might be capable of generating IMPLEMENTOR
  frames of the type OLNET is using.

             Press <CR> to continue.
```

a10599

Figure 88. FDDI warning screen

W4 Displays a warning message for the echo test, as follows:

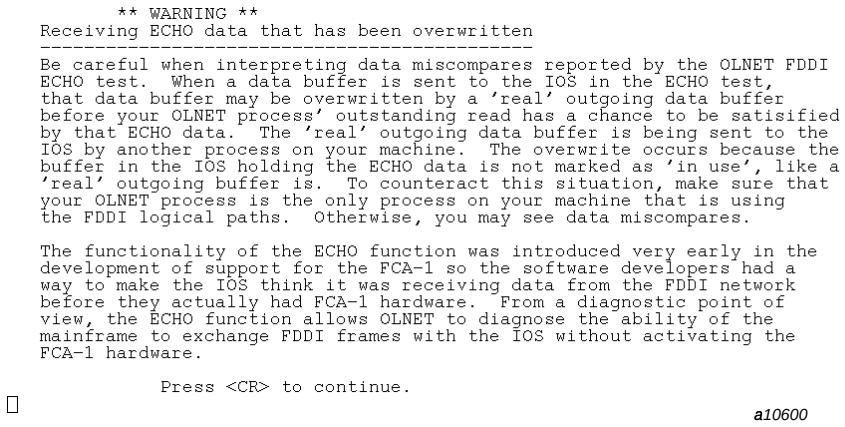

Figure 89. Echo test warning screen

6.7 FDDI test modes

You can execute the FDDI test in any of the following test modes:

- Synchronous active-and-passive mode (active mode is the default)
- Asynchronous active-and-passive mode
- IOS software echo mode
- Ring loopback mode

The following sections describe the execution of each mode.

6.7.1 Synchronous or asynchronous active-and-passive mode

In synchronous active-and-passive mode, one of the Cray Research systems acts as the active side and the other Cray Research system acts as the passive side. The active side generates and sends synchronous messages to the passive side. In response to the active side, the passive side generates and sends messages to the active side.

In asynchronous active-and-passive mode, one of the Cray Research systems acts as the active side and the other Cray Research system acts as the passive side. Unlike synchronous active-and-passive mode, asynchronous mode allows

you to specify a variable number of messages to be sent by the active side before an acknowledgment message is returned by the passive side.

Executing synchronous active-and-passive mode or asynchronous active-and-passive mode between two Cray Research systems tests the following components of the FDDI network:

- LOSP channel(s) of each Cray Research system
- FCA-1 adapter(s) on each Cray Research system
- Fiber-optic media connecting the FCA-1 adapters
- FDDI device driver software
- FDDI IOS driver software

To execute active-and-passive mode, do the following:

1. Initialize OLNET on both Cray Research systems and access the Main menu, as described in Section 1.2, page 2, and Section 1.3, page 4.
2. After you initialize OLNET and access the Main menu, enter `FDT` on each Cray Research system to display the FDDI Test Initial menu (Figure 53, page 193).
3. Set the FDDI device path you want to use on each Cray Research system by using either the `DV` or `DPM` command.
4. Enter `TMM` on each Cray Research system to display the FDDI Test Mode menu (Figure 54, page 194).
5. Designate one Cray Research system as active by setting the test mode to synchronous active (`TM,AM`) or asynchronous active (`TM,AA`). Then, on the other Cray Research system, set the test mode to synchronous passive (`TM,PM`) or asynchronous passive (`TM,AP`).

   The following commands must be set to the same values for each Cray Research system:

`AR`	Acknowledgment ratio (asynchronous active-and-passive mode only)
`ML`	Message length
`MP`	Messages per pass
`PC`	Pass count

PT Pattern type

6. On each of the Cray Research systems, use the RA command to set the FDDI address of the other (remote) Cray Research system.
7. Enter the EX command on the Cray Research system that has been designated as the passive side.
8. Enter the EX command on the Cray Research system that has been designated as the active side.

   Upon test completion, the following message is displayed:

```
Test passes have completed for
test mode
Total bytes transmitted  = n
Total bytes received     = n
Elapsed time(hh:mm:ss)   = hh:mm:ss
Transfer rate            = nbytes/second
```

6.7.2 IOS software echo mode

In IOS software echo mode, the Cray Research system sends FDDI frames to the software running in the IOS, which then sends them back immediately without placing them on the ring. The message length can range from 16 through 4500 bytes.

Executing IOS software echo mode on a Cray Research system tests the following components of the FDDI network:

- LOSP channel(s) of the Cray Research system
- FDDI device driver software
- FDDI IOS driver software

To execute IOS software echo mode, do the following:

1. Initialize OLNET on the Cray Research system and access the Main menu, as described in Section 1.2, page 2, and Section 1.3, page 4.
2. After you initialize OLNET and access the Main menu, enter FDT on the Cray Research system to display the FDDI Test Initial menu (Figure 53, page 193).

3. Set the FDDI device path you wish to use on the Cray Research system by using either the DV or DPM command.
4. Enter TMM on the Cray Research system to display the FDDI Test Mode menu (Figure 54, page 194).
5. Set the test mode to IOS software echo (TM,ECHO).

   The following commands can be adjusted depending on how much data you want to transfer and what the data should look like:

ML	Message length
MP	Messages per pass
PC	Pass count
PT	Pattern type

6. Enter the EX command on the Cray Research system.

Upon test completion, the following message is displayed:

```
Test passes have completed for
test mode
Total bytes transmitted  = n
Total bytes received     = n
Elapsed time(hh:mm:ss)   = hh:mm:ss
Transfer rate            = nbytes/second
```

6.7.3 Loopback mode

In loopback mode, the Cray Research system sends FDDI frames to the FDDI ring with the destination address the same as the source address (which is the FDDI address of the machine being tested). Typically, when a frame is sent around the ring, the machine that put it on the ring will take it off the ring when it sees its own address in the source address field. For the loopback test to work, OLNET must inform the software that it is **not** supposed to discard a frame that comes in with the source address the same as its own address. Rather, the frame should be sent to the process that registered to receive that frame type. For OLNET, the frame is one of the IMPLEMENTOR frame types.

Executing loopback mode on a Cray Research system tests the following components of the FDDI network:

- LOSP channel(s) of the Cray Research system

- FCA-1 adapter(s) on the Cray Research system
- Fiber-optic media connecting the FCA-1 adapter with other FDDI stations
- FDDI device driver software
- FDDI IOS driver software

To execute loopback mode, do the following:

1. Initialize OLNET on the Cray Research system and access the Main menu, as described in Section 1.2, page 2, and Section 1.3, page 4.
2. After you initialize OLNET and access the Main menu, enter `FDT` on the Cray Research system to display the FDDI Test Initial menu (Figure 53, page 193).
3. Set the FDDI device path you wish to use on the Cray Research system by using either the `DV` or `DPM` command.
4. Enter `TMM` on the Cray Research system to display the FDDI Test Mode menu (Figure 54, page 194).
5. Set the test mode to loopback (`TM,LBK`).

   The following commands can be adjusted depending on how much data you want to transfer and what the data should look like:

`ML`	Message length
`MP`	Messages per pass
`PC`	Pass count
`PT`	Pattern type

   OLNET explicitly sets the source and destination address fields of the FDDI frame to the values needed to perform this loopback test. Typically, the destination field is set from the value entered with the `RA` command. If you set a value in the `Remote address` field, OLNET will not change that setting.

   Enter the `EX` command on the Cray Research system.

   Upon test completion, the following message is displayed:

```
Test passes have completed for
test mode
Total bytes transmitted  = n
Total bytes received     = n
Elapsed time(hh:mm:ss)   = hh:mm:ss
Transfer rate            = nbytes/second
```

6.8 Warning situations

This section describes concerns you need to keep in mind when running OLNET because of the way FDDI frames are sent around the network and the way in which those frames are processed by the IOS. The two scenarios described in this section deal with the fact that OLNET may detect a data miscompare between received data and expected data. Typically, a data miscompare would indicate a problem; however, you need to be careful when making this assumption.

6.8.1 Reading unexpected IMPLEMENTOR frames

As described previously, for OLNET to receive FDDI frames from the network, it must register with the FDDI device driver that it wants to receive one of the frame types defined by the FDDI standard. OLNET uses one of the eight IMPLEMENTOR frame types. When frames come in to the IOS, they are forwarded to the logical path that registered to receive them, regardless of their origin. This means that OLNET can receive IMPLEMENTOR frames from any machine on the FDDI network if that machine is capable of generating an IMPLEMENTOR frame of the type for which OLNET is waiting. Therefore, when you execute a FDDI test, OLNET may read an IMPLEMENTOR frame that is not expected.

For example, suppose you are running the LBK test mode and OLNET issues an asynchronous read system call, getting ready to read the data that it is about to send with the write system call. Now assume that another machine on your network sends an IMPLEMENTOR frame (of the type used by OLNET) to your machine. The data from the other machine comes in to the IOS after you post your read and before your write is processed. Your read will complete and OLNET will be returned the IMPLEMENTOR frame from the other machine.

There is no way to prevent this situation. However, the only processes that should be placing IMPLEMENTOR frames (of the type used by OLNET) on the network are other OLNET processes that are running on other Cray Research systems on your FDDI network. So you could inadvertently cause this

situation, but you would also be aware that another OLNET was being run. All FDDI test modes read frames off the FDDI network in the same manner, so this scenario applies to all FDDI test modes.

6.8.2 Reading unexpected echo data

The IOS software loopback test mode uses the `FDC_ECHO` ioctl() function to send data to the IOS that can be read with the `read` () system call. This function was originally intended to provide software developers with a way to read FDDI frames from the IOS as if the frames came in from the network (before the FCA-1 hardware existed).

The channel buffer used for the FCA-1 IOS software is split into a read half, where incoming data is queued, and a write half, where outgoing data is queued. When the `FDC_ECHO` function is processed, data is placed in the write half of the channel buffer. (It is assumed that a read has already been queued to receive the data.) The area of the channel buffer that holds the `FDC_ECHO` data is not allocated explicitly as are the data areas for "real" outgoing write buffers. Therefore, if another process on the mainframe wants to write data to the FDDI network, that process's outgoing data buffer can overwrite the `FDC_ECHO` data buffer before the pending read has time to pick up the data.

For example, suppose you are running the `ECHO` test mode and OLNET issues an asynchronous read system call, getting ready to read the data that it is about to send with the `FDC_ECHO ioctl()` system call. When the IOS receives the data from the `FDC_ECHO` function, the data is placed in an unallocated area of the write half of the channel buffer. Because the IOS knows that the function is an `FDC_ECHO`, it immediately signals the IOS read code that data exists to satisfy the pending read, and then it continues looking for more outgoing data. But, before the read code can retrieve the `FDC_ECHO` data, another process on your machine sends an FDDI frame to the IOS. The outgoing data overwrites the `FDC_ECHO` data because that channel buffer area is not marked as allocated. The asynchronous read from your OLNET completes and returns a data buffer that has another process's outgoing buffer at the beginning. OLNET compares the received buffer with the data sent by the `FDC_ECHO` function and finds a data miscompare.

This situation is similar to the `Reading unexpected IMPLEMENTOR frames` scenario, but this situation is caused by another process that is running on the same machine on which your OLNET is running. The other process can be sending any frame type, not just IMPLEMENTOR frames. The only way to prevent the `FDC_ECHO` function's data buffer from being overwritten is to ensure that your OLNET process is the only active process with an FDDI logical

path open. You must either run OLNET in single-user mode or kill all daemons or processes that have open FDDI logical paths.