Honeywell

LCNI Network Communications Test

SW13-513

LCN Service - 3

LCNI Network Communications Test

SW13-513
9/95

Copyright, Trademarks, and Notices

Revision 01– September 15, 1995

About This Publication

What it covers—This publication provides procedures for running and interpreting results of the LCNI Network Communications Test (Rel 4.3) that tests whether a node is communicating with other nodes through the LCN.

Table of Contents

Table of Contents

GENERAL INFORMATION
Section 1

This section describes procedures and restrictions that apply to the Network Communications Test.

1.1 APPLICATION

The LCNI Network Communications Test is designed to test a node's ability to communicate with other nodes through the LCN. This is a free-standing test, independent of the Hardware Verification Test System (HVTS).

1.2 GENERAL PROCEDURE

The following general procedure must be followed in order to run the Network Communications test program:

A. Load the Network Communications test program into each of the nodes to be tested.

B. Establish the "Clock Master" and "Clock Slave" nodes if clock subsystem monitoring is wished.

C. Establish one node (must have an I/O device) as the Master Error Reporting Module.

D. Synchronize the elapsed time clocks at all test nodes and clear error counts that were accumulated during loading and startup of the individual nodes.

E. Define to the test system the nodes that are to be tested.

F. Start the test.

1.3 HARDWARE REQUIREMENTS

The following minimum hardware configuration is required to use the Network Communications test:

A. LCN cables A and B.

B. Two or more nodes to be tested. Each test node must contain

1. A TDC 3000 processor board.

2. At least 128 k words of memory.

3. An LCNI board connected to LCN cables A and B.

C. Either a Universal Station (or Universal Work Station) or a node equipped with an input-output terminal device with keyboard and printer or CRT. (This is a 300-, 1200- or 9600-baud device that connects to a selected node through the Serial Port interface on its processor board. See the *System Checkout Manual* for details.)

D. A floppy-disk subsystem—This is used for test-module loading and normally is connected to the US or UWS.

E. Operational clock boards in two modules (if clock subsystem monitoring is wished).

1.4 OPERATING RESTRICTIONS

This document applies to Release 4.3 of the LCNI Network Communications Test, which contains features not available with earlier test releases.

The LCNI Network Communications Test is NOT COMPATIBLE WITH RNOS AND SYSTEM SOFTWARE. DO NOT ATTEMPT TO RUN BOTH AT THE SAME TIME.

WARNING

When running this test, just turning off the LCNEs does not isolate a segment, you must also "Kill" the clock signals.

The Network Communications Test was not specifically designed to test the clock subsystem. While the program does MONITOR the operating status of the clock subsystem, it DOES NOT ATTEMPT ERROR RECOVERY IN THE CLOCK SUBSYSTEM AREA. The purpose of monitoring the clock subsystem is to check for interference between the clock-timing signals and the LCN-data signals. It is the responsibility of the test operator to interpret the messages related to the clock subsystem and take whatever steps may be necessary to bring the clock subsystem back to the desired state.

1.5 REFERENCES

The following are other TDC 3000^X publications that may be useful in running this test program:

Publication Title	Publication Number	Binder Title	Binder Number
LCN System Checkout	SW20-510	LCN Installation	3025
Test System Executive	SW13-510	LCN Service - 3	3060-3
Hardware Verification Test System	SW13-511	LCN Service - 3	3060-3
Dual Node Module Service	LC13-510	LCN Service - 2	3060-2
History Module Service	HM13-501	LCN Service - 2	3060-2
Five/Ten-Slot Module Service	LC13-500	LCN Service - 2	3060-2
Universal Station Service	US13-500	LCN Service - 1	3060-1

OPERATING INSTRUCTIONS
Section 2

This section describes how to load and control the Network Communications Test program.

2.1 LOADING AND STARTING THE PROGRAM

NOTE

1. In the following, references to input/output communication between the test operator and the system can be by either a Universal Station's Keyboards and screen or the optional I/O terminal device.
2. Be patient, there are often significant delays during the test-program loading process.
3. Because Network Test and RNOS-based software are incompatible, you must insure that both are not running on the LCN at the same time. Either isolate the test nodes by creating a separate LCN or reset all nodes before starting the loading process.

2.1.1 Keyboard and Display Options

The primary method of Network Communications Test input/output is through the product I/O devices; that is, the Universal Station or Universal Work Station CRT and keyboards. However, the use of an I/O device attached through the processor RS-232C port is still supported. (See Paragraph 2.6 for a summary of keyboard entry equivalencies.)

To request a prompt for operator input, press the following key on the desired keyboard:

- Escape key on the RS-232C keyboard
- Select key on the Operator's or Supervisor's Keyboard
- Escape key or Select key on the Engineer's Keyboard

To signal completion of the input line, press the following key:

- Return key on the RS-232C keyboard
- Enter key on the Operator's or Supervisor's Keyboard
- Return key or Enter key on the Engineer's Keyboard

NOTE

For entries made from the Operator's or Supervisor's Keyboards, use the SMCC/Maintenance keyboard insert, which is in a plastic pocket in the *Service* binder.

2.1.2 Printing Option

If a printer is connected to the US being used as the Master Error Reporting Module, all prompts, operator entry, messages, and alarms can be printed as well as displayed on the CRT. At the Operator's Keyboard, to start the printer, press the PRINT DISP key; to stop printing, press CANCL PRINT. At the Engineer's Keyboard, to start the printer, press CONTROL and P together; to quit printing, press CONTROL and Q together. All text entry is displayed on the CRT as it is typed, but the entered keystrokes are not printed until completion of the input line is signaled by pressing the Return or Enter key.

Printer control entries must be performed between command line entries (i.e., before you press escape/select). Printer control-entry attempts in response to the `yes` prompt are ignored.

2.1.3 Loading Instructions

Follow these steps to load and begin operation of the Network Communications Test.

1. First, to prevent History Modules from automatic reboot of RNOS software, power down all HM disk drives.

2. Load HVTS into a Universal Station to be used to load LCNI into nodes to be tested. (Alternatively, you can use CMTS, which is smaller and thus loads faster.)

   A. When loading from floppy diskettes

      1. Press the RESET button on the Operator's Keyboard. After the hardware and firmware self-tests run, the prompt ">" and the cursor are displayed on the Universal Station Screen.

      2. Press the LOAD button on the Operator's Keyboard. The following prompt is displayed: `N,1,2,3,4,X?`

      3. Insert the first of the two HVTS program-load diskettes (51150670) into a floppy drive, then key in the drive number and press the enter key (using either the Engineer or Operator Keyboard). Wait for the prompt

         ```
         ENTER "L" FOR LOAD OR "D" FOR DUMP
         ```

      4. Enter the capital letter L and press the enter key. You will be prompted when to insert the second diskette (into the same drive). When finished, the loader issues a load completion message to both the screen and the typer, then begins execution of HVTS with display of revision date and revision level.

      5. When HVTS load is complete, remove the HVTS floppy and place the Network Communications test floppy, 51150328, into floppy drive 1.

B. When loading from cartridge disk

1. Press the RESET button on the Operator's Keyboard. After the hardware and firmware self-tests run, the prompt ">" and the cursor are displayed on the Universal Station Screen.

2. Press the LOAD button on the Operator's Keyboard. The following prompt is displayed: `N,1,2,3,4,X?`

3. Insert the system test cartridge disk (51152049) into a drive, then key in the drive number and press the enter key (using either the Engineer or Operator Keyboard). Wait for the prompt

   ```
   HVT,CMT,LCN ?
   ```

4. Enter the capital letter H and press the enter key. When finished, the loader issues a load completion message to both the screen and the typer, then begins execution of HVTS with display of revision date and revision level.

(For more details on the loading and use of HVTS—such as how you can use it to reset nodes remotely—refer to the *Test System Executive,* and the *HVTS Reference Manual.*)

3. RESET all nodes on the LCN except the Universal Station used for node loading.

4. Press the escape/select key at the loading Universal Station. HVTS responds with

   `00:01 NODE nn ALL?` (NODE nn identifies the Universal Station as the HVTS Test Operator Control Station or TOCS)

5. Enter "LOAD n,m,....,t" and press return/enter. (Where "n,m,....,t" represent the node addresses of the nodes to be loaded with the Network Communications Test program.) Shortcut: Enter LOA - to load all nodes on the LCN (except the TOCS).

   If you are loading from cartridge disk, the system will prompt you to select which test system is to be loaded. Enter L and press return/enter.

   A "LOAD request is in progress" message is issued, followed on successful conclusion by a 3-line "Load completed" message.

6. At each node with an output device the following heading appears:

   ```
   LCNI NETWORK COMMUNICATIONS TEST
   DRAWING NUMBER 51150328
   VERSION M.P{.S}
   REVISION DATE DD-MMM-YYYY

   Local node address = AAA
   ```

Where: M= Major release number
P= Point release number
S= Optional preliminary sub-release number
D= Day of the month
MMM= Three-letter month abbreviation
YYYY= Year
AAA= Node's LCN address (in decimal)

On loading, each node is in LISTEN ONLY mode and is set to report alarms to its local output device (whether it exists or not). In this mode, the nodes listen to the LCN and respond to command and test messages but do not send messages on their own. They also respond to commands entered from their keyboards (if present).

7. If the node used for loading is not to be part of the test, go to step 8. If it is to be tested, proceed as follows:

   When loading from floppies—Leave the Network Communications Test floppy in its drive and proceed as above (steps A1 to A4), to load the Net test in the load node. (Do NOT change floppies at step A3; you want the LCNI floppy in the drive.)

   When loading from a cartridge—Leave the test systems cartridge in its drive and proceed as above (steps B1 to B4) to load the Net test in the load node. IMPORTANT: At step B4 enter L (for LCNI)

   Note that you are loading LCNI itself into the load node, thus you will see the LCNI startup message appear on its screen (not the HVTS message).

8. Again press the escape/select key. Now enter NET CLE DNA and wait for the completion message (about 10 seconds). Examine the display to verify that all expected network nodes have responded.

9. If you do not wish to monitor the clock subsystem, skip to step 10. To set up clock subsystem monitoring: Press the I/O-device escape/select key. After the response `Yes?` enter "NET CLM mm CLS ss" and press return/enter. ("mm" and "ss" represent the LCN addresses of the modules containing the "Clock Master" and "Clock Slave," respectively.)

10. You now select the module that is to be used as the Master Reporting Module. This may be the same Universal Station used for loading, but need not be. At the selected module, press the escape/select key. After the response `Yes?`, enter NET CLZ and press return/enter. Wait for the message "Network Commands Complete" (about 5 seconds).

11. Again, press escape/select. Enter NET RUN and wait for the completion message (about 5 seconds).

The system responds with a `Testing Started` message. For explanation of the error messages that may appear during test-program operation, see Section 3 of this document.

The remainder of this section explains details of operator communication with Network Test along with information on the commands that can be used to alter test execution.

2.2 OPERATOR COMMUNICATIONS

2.2.1 Starting the Man-Machine Dialog

Operator communications with Network Test are through an I/O-terminal device that can be connected to any node in the ring. Once the node has been established as the Master Reporting Module (by entry of a NET command line) it also receives all test error messages.

Communication with the program is established by pressing the escape/select key. The program responds with the prompt YES?. At this point the program is ready to accept the operator's command entries.

NOTE

You can change to a different Master Reporting Module at any time just by using a keyboard at the new module to make a NET command line entry.

2.2.2 Command Line Entry

The general form of command line entry is:

command command command . . . return/enter

Note that the Network Communications Test allows multiple commands on the same line.

If an error is made while typing a command line, press the backspace, delete, or back-arrow (depending on keyboard) to delete the immediately preceding character. Press the clear entry key or the "control" key and "x" at the same time to delete the whole line. After line deletion, the program returns to the input mode awaiting another command line.

After each command line, input is completed by pressing the return/enter key. The escape/select key must be pressed again if further command lines are to be entered. Input lines can be up to 80 characters long.

2.2.3 Command Modes

There are two types of command, LOCAL and NETWORK. LOCAL commands affect only the node where entry is made (local node). NETWORK commands affect all nodes in the test ring. (Except when noted otherwise, any program command can be used in either the LOCAL or NETWORK modes.)

The command mode defaults to LOCAL and is changed to NETWORK by entering the command NET at some point in the command line. The NETWORK mode is in effect for the remainder of the input line. The following examples outline the use of the command modes. More detailed examples can be found in Section 2.4 of this document.

2.2.3.1 Example of a LOCAL Command Line

Command Command Command... return/enter

2.2.3.2 Example of a NETWORK Command Line

NET Command Command Command... return/enter

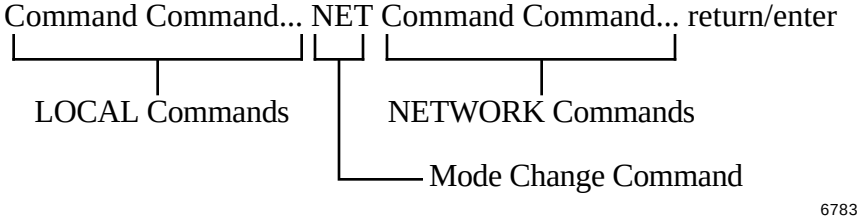
Because the local node is also a member of the network, commands sent to the network also are executed by the local node.

NOTE

1. LOCAL mode is seldom used, because most commands need to affect all nodes on the test ring. The primary exceptions are those commands that require node addresses to be specified.
2. The NET command has the side effect of establishing the local node as the Master Error Reporting Module. In the event that the I/O-terminal device is moved to another node, Network Test error-messages are not redirected until after a NET command is entered at the new node.

2.2.3.3 Example of a Mixed Mode Command Line

A mixed mode-command line is used when only the local node is to be affected by the first part of the command line and the network is to be affected by the second part.



Commands are executed in the order they are typed (from left to right). Because the NETWORK commands are processed last, they override any conflicting LOCAL commands. Additionally, the local node executes the network commands BEFORE they are transmitted to the network. This allows the local node to more efficiently process the incoming replies, as it has already completed the requested operations.

2.2.4 Message Delimiters

Because all nodes report to a single Master Error Reporting Module, it is necessary to separate messages received from one module from those received from a different module. To avoid confusion, messages that can be generated by either a remote module or the local module start with the address of the sending node. Messages that can be generated by only the local module do not include a node indicator.

Node addresses are displayed in either decimal or hexadecimal (base 16) notation. You select the desired number base by use of the DEC and HEX commands. The formats are:

AAA — If DEC addresses are selected
$AA — If HEX addresses are selected

2.2.5 Data Entry Rules

The Network Communications Test's input package is not case-sensitive and input can contain uppercase, lowercase, or a mixture of the two. Each command must be separated from the next by at least one space.

Both hexadecimal (base 16) and decimal numbers are supported. All number entries are assumed to be decimal notation unless preceded by a dollar sign ($), which forces hexadecimal notation for that number only. Example,

15 = decimal 15 or hexadecimal $F
$15 = hexadecimal 15 or decimal 21
$f or $F = hexadecimal $F or decimal 15
f or F = illegal (decimal number contains illegal character)

The alphabetic characters of a hexadecimal number can be entered in upper or lower case.

2.3 PROGRAM COMMANDS

CLE

CLE causes the program to clear the error count to zero and reset the individual time-out counters. Also, the I/O-device output buffer is purged. No program response is generated.

Because the CLE command purges the output buffer, any text currently being printed can be truncated. This is normal and should not be considered an error. Also, a clock error may occur immediately following the CLE command because of a period of inhibited interrupts.

CLM

CLM mm (where mm is an LCN address) places the specified node's clock subsystem in the CLOCK MASTER mode. Several error messages can be generated and sent to the Master Error Reporting Module (see the error descriptions later in this document).

CLS

CLS ss (where ss is an LCN address) places the specified node's clock subsystem in the CLOCK SLAVE mode. Several error messages can be generated and sent to the Master Error Reporting Module (see the error descriptions later in this document).

CLZ

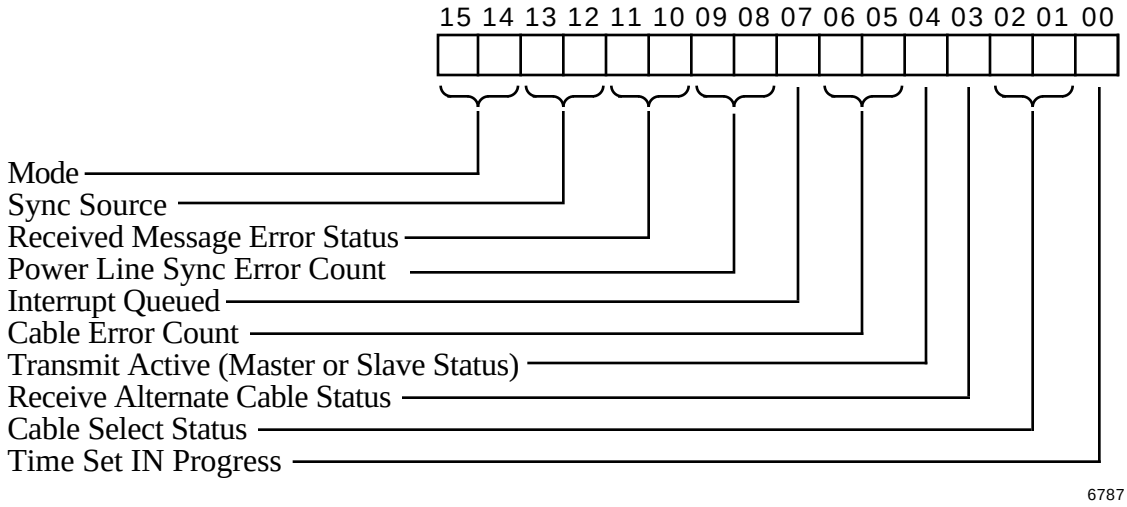
The CLZ command attempts to initialize hardware clocks, but falls back to software time-keeping if hardware clocks are not selected (see CLM/CLS) or are not operational. When used as a Network command (NET CLZ) it synchronizes time of all nodes on the test ring. Several error messages can be generated and sent to the Master Error Reporting Module (see the error descriptions later in this document).

If the network clock source is not functioning, a one-second difference will exist between the node that issued the command and the other nodes in the network.

CST

CST displays the clock-subsystem status-word current value in hexadecimal.

Clock Status Word Format



Explanation of Clock Status Indicators

Mode — 00 = Master Clock source mode is selected
01 = Slave Clock source Mode is selected
10 = Listener Only mode is selected
11 = Local Mode is selected

Sync Source — 00 = Clock is not synchronized
01 = Clock is synchronized to the power line
10 = Clock is synchronized to the received message interrupt

Received Message Error Status — 00 = No Error
01 = Communication Error
10 = Invalid Data Code
11 = Invalid Time Update

Cable Select Status — 00 = Received no time update
01 = Received Cable A time update
10 = Received Cable B time update

DEC

The DEC command causes the node addresses used in messages to be displayed in decimal. No other response is made.

DNA

DNA causes the program to display the address of the node in which it resides. If this command is sent to the network (NET DNA) and the Master Error Reporting Module is in LISTENER mode, it causes automatic configuration of the node address list in each node in the network. See subsection 2.5 for additional details.

DNL

The DNL command causes display of the node list. This list is used to indicate which nodes are available for testing and is also used by the RST command when resetting nodes.

DTE

The DTE command disables the reporting of clock-tick-message errors, and is used to inhibit the reporting of those clock-tick-message errors that do not cause the abandonment of the cable in use. This command is in effect when the program is loaded. See the ETE command for clock-tick enable.

ENA

ENA uses a parameter list of node addresses separated by commas (n,n,n,n) or spaces (n n n) to replace the network configuration list.

If the program is in the LISTEN mode, the ENA command clears the prior node-address list and replaces it with a new list. If the program is not in the LISTEN mode, an error message is displayed (the node-address list remains unchanged) and any commands that follow the ENA command are executed normally, using the previous node-address list. If the program is in the LISTEN mode, the ENA command also clears the individual time-out counters maintained for each node. The command takes the form

ENA parameter_list

Valid node addresses range from 0 to 127 ($7F) inclusive. Node-address subranges (node address-node address) are inclusive and the ending value must be greater than or equal to the starting value. Multiple parameters must be separated by either a comma or space. If multiple parameters are entered, the order is of no importance ("0,1,2,3" = "2,0,3,1"). A node address can appear in the list more than once without consequence ("0,1,2,1" = "0,1,2"). The program maintains the node-address list as a bit map and duplicate addresses simply set a bit that is already set.

Up-to-128 different node addresses can be entered. The node address of the host module does not need to be present in the list and if found, is deleted. The program does not generate a response to the ENA command. For more detailed examples, see Section 2.4 of this document.

ERR

ERR causes the program to display the current decimal error-count.

ETE

This command enables the reporting of clock-tick error messages and is the logical complement to the DTE command.

HEX

The HEX command causes the node addresses used in messages to be displayed in hexadecimal. No other response is made.

NET

NET changes the input mode from LOCAL to NETWORK and can appear only once in the command line. The NET command must be followed by at least one valid network command. The NET command also establishes the Master Error Reporting Module.

NET commands require 5 to 10 seconds to complete. During this time no additional NET commands can be entered. The system displays the message "Network Commands Complete" when NET commands can again be entered.

RCC n

The RCC (Reversed Cable Check) performs a reversed cable check sequence. To perform this check, the system must be locked to a single cable (see the SMA and SMB commands below) and a copy of the program must be loaded in a node containing a K4LCN board or a node containing an updated K2LCN with LCN Firmware revision D or greater. The RCC command must be executed by typing “NET RCC n”. This command initiates the reversed cable check in the remote node and establishes the local node as the Master Error Reporting Module. The reversed cable check consists of sending five messages on the cable currently in use only. After all messages have been transmitted, each node will be asked to verify that all messages were received. If none of the messages were received, the program prints an error message indicating that the node’s cables are most likely reversed. If at least one, but not all of the messages were received, the program prints an error message indicating a recommendation that the system be checked for noise using the “RUN” command. If all messages were received, no message is printed.

Note: If the node from which RCC is initiated is the only node with reversed cables, **all** other nodes will report reversed cables. Also, the node conducting the reversed cable test cannot report errors.

RST

The RST command RESETS ALL MODULES IN THE NODE LIST—including the Master Reporting Module—and must be the only command in the input line. To prevent accidental resets of running systems, the program verifies the operator's intent before performing the reset operation. The operator is asked to repeat the RST command within 15 seconds. If it is not repeated within that time or a command other than RST is entered, the reset operation is not performed.

CAUTION

The RST command resets every module in the node list regardless of its current state or resident software image. Do **not** perform a reset operation unless you are fully aware of the consequences.

RUN

RUN puts the program into the TRANSMIT mode and must be the last command in the input line. If the program is already in the TRANSMIT mode, no action is taken. The program responds with "Testing started" if no errors are encountered.

SAA

SAA places the LCNI in the auto-cable switching mode, with cable A initially assigned as the primary cable. The LCNI firmware continues to use cable A for the specified time (n) or until a fault is detected; it then changes automatically to cable B. No response is generated when the cable is swapped. Note that Cable B continues to be monitored and error messages for it can appear.

The format is SAA n, where n is a numeric with the range of 0-30 minutes (inclusive). If "0" is entered, periodic cable swapping is suppressed.

SAB

SAB places the LCNI in the auto-cable switching mode, with cable B initially assigned as the primary cable. The LCNI firmware continues to use cable B for the specified time (n) or until a fault is detected; it then changes automatically to cable A. No response is generated when the cable is swapped. Note that Cable A continues to be monitored and error messages for it can appear.

The format is SAB n, where n is a numeric with the range of 0-30 minutes (inclusive). If "0" is entered, periodic cable swapping is suppressed.

SCP

<This command is not field-useable and is not supported>

SMA

SMA places the LCNI in the manual cable switching mode with cable A assigned as the primary cable. No response is generated by the program. This is the start-up default.

SMB

SMB places the LCNI in the manual cable switching mode with cable B assigned as the primary cable. No response is generated by the program.

STP

If the current mode is TRANSMIT, the STP command stops the transmission-test messages and puts the program into the LISTEN ONLY mode. If the program is already in the LISTEN ONLY mode, no action is taken. If the previous mode was TRANSMIT, the program responds with the following:

```
Going to listen mode, current error count is XXXXX
```

Where XXXXX is the current error count in decimal.

TIM

TIM causes the program to display the current elapsed time. If the clock is in the network mode, the time is network time. If the clock mode is local, the local-node elapsed time is displayed. Note that the time reported by remote nodes may appear inconsistent from node to node. This is caused by processing delays during the data retrieval. The times reported by the remote nodes are all within 1 second of each other. If the network-clock source is functioning, the time reported by the local node is also within the 1-second window.

The following example illustrates the elapsed time differences:

```
NET TIM entered by operator at node $00

0: Current time is  Day 0  0: 1:15
1: Current time is  Day 0  0: 1:16
2: Current time is  Day 0  0: 1:15
3: Current time is  Day 0  0: 1:16
4: Current time is  Day 0  0: 1:15
```

If the network clock source is not functioning, there is the possibility of an additional 1-second difference between the remote nodes and the node that transmitted the last CLZ command (refer to the CLZ command for further details). The following example illustrates this additional 1-second difference:

```
NET TIM typed by operator at node $00

0: Current time is  Day 0  0: 1:16
1: Current time is  Day 0  0: 1:14
2: Current time is  Day 0  0: 1:15
3: Current time is  Day 0  0: 1:14
4: Current time is  Day 0  0: 1:14
```

2.4 COMMAND STRING EXAMPLES

The command lines in this section are shown in uppercase for clarity only. The input processor is not case-sensitive and characters can be typed in either uppercase, lowercase, or mixed cases (that is, CLE = cle = Cle). This also includes alpha characters within hexadecimal numbers (that is, $A = $a).

The command strings in this section show typical command lines for a 10-node network:

NET CLE ENA 1-10 RUN or NET CLE ENA 1 2 3 4 5 10 9 8 7 6 RUN

These two command lines have identical results. All nodes in the network clear their error counts, enter node addresses 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and switch to the TRANSMIT mode. This line also establishes the local node as the Master Error Reporting Module.

NET ENA 7, 8-10 2 9

This command line instructs all nodes to clear their node-address lists and enter node addresses 2, 7, 8, 9, and 10. Note that the duplication of address 9 (in the 8 through 10 subrange and individually) is legal. The program maintains the node-address list as a bit map and duplicate addresses simply set a bit that is already set. Note that if the program were placed into the TRANSMIT mode, test messages would be sent only to nodes 2, 7, 8, 9, and 10. This line also establishes the local node as the Master Error Reporting Module.

ENA $10-$12 or ENA $10-18 or ENA 16-$12 or ENA 16-18.

A dollar sign ($) preceding a number causes the input routines to interpret the number as a hexadecimal value. All of these command lines instruct the LOCAL NODE to clear the node-address list and enter node addresses 16, 17, and 18 ($10, $11, and $12 hex). Note that the radix (number base) change is only for that number.

STP

This command line instructs the local node to return to the LISTENER mode.

NET STP

This command line instructs all nodes in the network to return to the LISTENER mode. It has no effect on nodes already in the LISTENER mode. This line also establishes the local node as the Master Error Reporting Module.

NET ERR

This command line instructs all nodes in the network to return their current error count. This line also establishes the local node as the Master Error Reporting Module.

NET ERR CLZ CLE TIM ENA 0 , 3 4-9,1 2 RUN

While the legality of this last command line may seem questionable, it is actually a perfectly valid command line and is accepted by the program without error. All nodes in the network (including local node) are instructed to return their current error count and then reset their clocks to zero. All nodes then clear their error counts and return the current elapsed time as they see it. Following this, node addresses 0 through 9 are entered (order is not important) and the nodes are placed in the TRANSMIT mode. This line also establishes the local node as the Master Error Reporting Module.

2.5 AUTOMATIC NETWORK CONFIGURATION

Automatic configuration of all node address lists is accomplished by performing the following actions.

(1) If the network is not in the LISTENER mode, type "NET STP (return/enter)" at the Master Error Reporting Module. (All nodes are in LISTENER mode when first loaded.)

(2) Type "NET DNA (return/enter)" at the Master Error Reporting Module. This causes all network nodes to report their addresses to it. The Master Error Reporting Module displays a list of these addresses and sends it to all network nodes. The message "Network Commands Complete" (approximately 10 seconds after command entry) indicates when additional commands can be entered.

The auto configuration operation is performed only if the Master Error Reporting Module is in the LISTENER mode. At any other time, the list of node addresses is displayed, but no further action is taken.

The auto-configuration operation does not clear the node-address list before entering the addresses of the responding nodes; therefore, if a node has ever responded to the network node-address-display request, its address remains in the list. Then, when the program is placed in the TRANSMIT mode, those nodes that were at one time present but are not now present will fail to respond, causing time-out errors. If this effect is not desired, clear the node-address list before auto configuration. This is most easily accomplished by typing "NET ENA nn (return/enter)" where nn is the address of the Master Error Reporting Module. Auto configuration can then be performed.

Because the ENA command clears the node-address list before adding the list found in its input string, it can be used to override the list generated by auto configuration.

2.6 KEYBOARD ENTRY EQUIVALENCIES

Table 2-1 illustrates keystroke equivalents for the three types of entry devices that can be used with Net Test. Note that there may be two or more alternatives for the same function for each entry-device type.

Table 2-1 — Keystroke Equivalents

Desired Operation	Terminal Keyboard	Operator/ Supervisor Keyboard	Engineer Keyboard
Program Attention	ESC	SELECT	ESC or SELECT
Delete Character	Back Space DEL CTRL-H	Back Arrow Tab Back	Tab Back Cursor Back DEL CHAR
Delete Entire	CTRL-X	CLR ENTR	CLR-ENT CANCEL DEL LINE
End Input	RETURN	ENTER	ENTER RETURN
US/UWS Printer ON	NONE	PRINT DISP	CTRL-P
US/UWS Printer OFF	NONE	CANCL PRINT	CTRL-Q

PROGRAM ERROR MESSAGES
Section 3

This section describes and explains the format and content of the Communication Test Error Messages.

3.1 ERROR MESSAGE HEADER FORMATS

The Net Test is capable of producing numerous error messages. To simplify error location, the error printouts contain a unique header based upon the detecting subsystem. The headers are as follows:

>>> Messages that start with this header pertain to fatal errors usually from the LCN subsystem.

*** Messages that start with this header pertain to recoverable errors from the LCN subsystem.

+++ Messages that start with this header pertain to the clock subsystem.

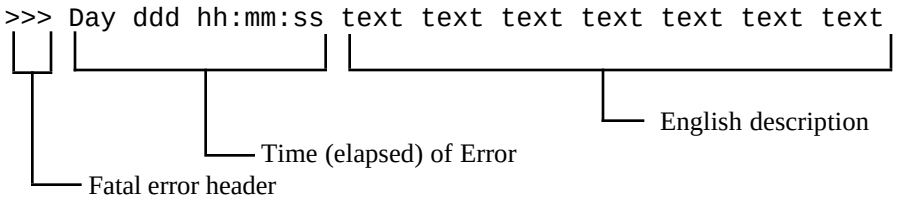
Note that because of line-width restrictions in this document, not all error messages shown exactly match those produced by the program. This exception applies to format only; some errors that require multiple lines in this document will be printed on fewer lines by the program.

3.2 FATAL ERRORS

These errors occur when the program discovers an error that it cannot correct or when recovery would be uncertain. The program places an error code in the node's processor Hexadecimal LED display and, if a terminal is connected to the node's processor serial I/O port, one of the following messages is printed. The program then vectors to the Console Emulator. No message is sent to the Master Error Reporting Module.

3.2.1 Program Detected Fatal Errors (Hex display '-XX')

The following general message structure is used to report these errors:



While this basic structure is used for all fatal error messages, some provide the operator with additional status information. In these cases, the message overflows to additional lines. See the following individual error descriptions for more details.

3.2.1.1 Internal Errors (Hex displays '-00' through '-5F')

The Net Test has internal code to protect against accidental data corruption and fatal programming errors. This error indicates that the program has failed one of these internal checks. An STR should be submitted to the Test and Diagnostic Software group to correct the problem. The STR should contain the error code printed by the program and the number of nodes running in the network at the time of the error. If possible, all program output from the node reporting the error should be neatly folded and turned in with the STR as supporting documentation.

```
>>> Day ddd hh:mm:ss Internal Error...  Error Code = xxxx
```

Internal error code in decimal

6784

3.2.1.2 Clock Command Register Busy (Hex display '-62')

Before issuing a command to the clock subsystem, the clock driver checks the busy bit of the clock-command register. If the busy bit is not cleared within 60 msec., the driver reports this error. If this error is printed, it indicates that the clock subsystem was at one time functional and has since failed. If the clock subsystem fails during startup a runtime error is reported (see PASCAL runtime errors later in this document).

```
>>> Day ddd hh:mm:ss Clock did not come ready within 60 msec.
Vectoring to Console Emulator
```

3.2.1.3 Hardware Status Error (Hex display '-80')

This error is discovered by the LCNI driver during LCNI operation. This error is reported if the Self-Test or Transaction-Error bits are set in the LCNI hardware-status register (Register Base + $0). The error is reported in the following form:

```
>>> Day ddd hh:mm:ss Fatal LCNI Hardware Status Error. Status = $SSSS
```

Hardware status word

6784

3.2.1.4 Configuration Status Error (Hex display '-81')

This error is discovered by the LCNI driver during LCNI operation. This error is reported if the LCNI fails to set the Off-Line bit in the configuration-status register (Register Base + $2) when placed in the Off-Line mode. The error is reported in the following form:

```
>>> Day ddd hh:mm:ss Fatal LCNI Configuration Status Error.
      Status = $SSSS
```

Configuration status word

6784

3.2.1.5 Operational Status Error (Hex display '-82')

Operational-status errors are discovered by the LCNI-driver interrupt-service routines. There are two variations of this error, depending on the interrupt vector being serviced. If the error is discovered when the board-level vector (vector 0) is being serviced, the error is caused by finding any of the following bits set in the operational-status register (Register Base + $A):

IOCB Interrupt Flag (Bit 15); Illegal Command (Bit 13); processor Abort Flag (Bit 12) ; LCN Address Parity Error (Bit 9); Integrity Fail (Bit 6); Reserved Bits (8, 5, 4, 3)

If one of the IOCB vectors (vectors 1, 2, or 3) is being serviced, the error is caused by finding any of the following conditions:

IOCB Interrupt Flag (Bit 15) reset; CCB Format Error (Bit 13) set; IOCB Format Error (Bit 12) set; Reserved Bit 8 set; Integrity Fail (Bit 6) set; Reserved Bit 5 set; Reserved Bit 4 set; Reserved Bit 3 set

The error message takes the following form:

```
>>> Day ddd hh:mm:ss Fatal LCNI Operational Status Error.
      Status = $SSSS $VVVV
```

Vector Number

Operational-status word

6784

3.2.1.6 No LCNI Found Error (Hex display '-86')

This error is found by the LCNI driver during initialize. The processor firmware builds a list of operational LCNI boards during self-test. The list is ordered by slot number, starting at the lowest and progressing to the highest. The Net Test uses the first entry of this list to determine the position of the LCNI. While the LCNI is usually located in slot 2 immediately above the processor, the program will operate with the LCNI located in any valid card slot. If the list of operational LCNIs built by the processor firmware during self-test is empty or contains an illegal slot-number value, the program prints this error and vectors to the Console Emulator.

```
>>> Day ddd hh:mm:ss No Operational LCNI could be found
```

3.2.1.7 Command Register Busy Error (Hex display '-87')

This error is detected by the LCNI driver if the command register-busy flag in the LCNI command register (Register Base + $8) does not clear within 15 msec. of the previous command.

```
>>> Day ddd hh:mm:ss The LCNI Command Register did not come ready
within 15 msec. after the previous command $CCCC
```

Command Register

6784

3.2.2 Pascal Runtime Detected Errors (Hex display '1XX')

The Pascal runtime support package contains various checks to protect against fatal programming errors. If a Pascal runtime-detected error occurs, it indicates that a low-level (undetected by the program) internal error has been detected. An STR should be submitted to the Test and Diagnostic Software group to correct the problem. The STR should contain the error code printed by the program and the number of nodes running in the network at the time of the error. If possible, all print-outs from the node reporting the error should be neatly folded and turned in with the STR as supporting documentation. This information will aid in the resolution of the problem. After printing the error message, the program exits to the Console Emulator.

Following are the Hex displays and messages associated with this error class. (AAAAAAAA represents the program counter at the time of error detection.)

```
Hex Display '100'   ARRAY SUBSCRIPT OUT OF BOUNDS.  PC = AAAAAAAA

Hex Display '101'   SUBRANGE LIMITS EXCEEDED.  PC = AAAAAAAA

Hex Display '102'   REFERENCE ATTEMPTED THROUGH NIL POINTER.  PC = AAAAAAAA

Hex Display '103'   CASE SELECTOR MATCHES NO LABEL.  PC = AAAAAAAA

Hex Display '104'   ERROR DURING FILE I/O.  PC = AAAAAAAA

Hex Display '105'   ATTEMPT TO READ PAST END OF FILE.  PC = AAAAAAAA

Hex Display '106'   ERROR DURING WRITE TO FILE.  PC = AAAAAAA

Hex Display '107'   ERROR DURING READ FROM FILE.  PC = AAAAAA

Hex Display '108'   NEGATIVE FIELD WIDTH DURING TEXT WRITE.  PC = AAAAAAAA

Hex Display '109'   INVALID INTEGER READ FROM TEXT.  PC = AAAAAAAA

Hex Display '10A'   ERROR DURING INTERRUPT DRIVEN I/O.  PC = AAAAAAAA

Hex Display '10-'   ILLEGAL LOGICAL FILE NUMBER.  PC = AAAAAA

Hex Display '10C'   ATTEMPT TO RELEASE NONEXISTENT TIMER BLOCK. PC = AAAAAAAA

Hex Display '10 '   FILE LINK ERROR.  PC = AAAAAAAA

Hex Display '10E'   INTERRUPT VECTOR ERROR.  PC = AAAAAAAA
```

3.2.2.1 Clock Failure (Hex display '10F')

On startup, the clock subsystem is placed into the LOCAL mode. This message indicates that the clock busy-status bit was still set 60 msec. after the command was issued. The runtime package treats this as a fatal error and vectors to the Console Emulator. This error generally indicates very old clock firmware.

```
>>> Clock subsystem has failed.  Vectoring to Console Emulator
```

3.2.3 Exception Errors (Hex display '-1XX')

Each of the 68000 exception vectors is directed to an exception-processing subroutine. The detection of an exception can be caused by either a software error or a hardware error. If the most probable cause is software, an STR should be submitted to the Test and Diagnostic Software group for correction of the problem. The STR should contain the error code printed by the program and the number of nodes running in the network at the time of the error. If possible, all program output from the node reporting the error should be neatly folded and turned in with the STR as supporting documentation. This information will aid in the resolution of the problem. After printing the error message, the program exits to the Console Emulator. To attempt recovery, type GO 82000.

Following is a list of the hexadecimal displays with each associated error message and most probable cause.

Hex Display '-101' `RECEIVED UNEXPECTED BUS ERROR EXCEPTION.`
Generally caused by software.

Hex Display '-102' `RECEIVED UNEXPECTED ADDRESS ERROR EXCEPTION.`
Generally caused by software.

Hex Display '-103' `RECEIVED UNEXPECTED ILLEGAL INSTRUCTION EXCEPTION.`
Generally caused by software.

Hex Display '-104' `RECEIVED UNEXPECTED DIVIDE BY ZERO EXCEPTION.`
Caused by software.

Hex Display '-105' `RECEIVED UNEXPECTED CHK INSTRUCTION EXCEPTION.`
Caused by software.

Hex Display '-106' `RECEIVED UNEXPECTED TRAPV INSTRUCTION EXCEPTION.`
Caused by software.

Hex Display '-107' `RECEIVED UNEXPECTED PRIVILEGE EXCEPTION.`
Caused by software

Hex Display '-108' `RECEIVED UNEXPECTED TRACE EXCEPTION.`
Caused by software

Hex Display '-109' `RECEIVED UNEXPECTED LINE 1010 EMULATOR EXCEPTION.`
Caused by software.

Hex Display '-10A' `RECEIVED UNEXPECTED LINE 1111 EMULATOR EXCEPTION.`
Caused by software.

Hex Display '-10-' `RECEIVED UNEXPECTED UNASSIGNED (30-3B) EXCEPTION.`
Caused by hardware.

Hex Display '-10C' `RECEIVED UNEXPECTED UNINITIALIZED INTERRUPT VECTOR EXCEPTION.`
Caused by hardware.

Hex Display '-10 ' `RECEIVED UNEXPECTED UNASSIGNED (40-5F) EXCEPTION.`
Caused by hardware.

Hex Display '-10E' `RECEIVED UNEXPECTED SPURIOUS INTERRUPT EXCEPTION.`
Caused by hardware.

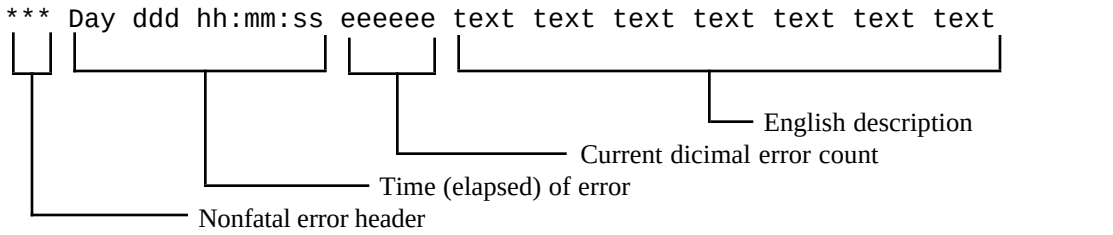
Hex Display '-10F' `RECEIVED UNEXPECTED UNASSIGNED (64-7F) EXCEPTION.`
Caused by hardware.

Hex Display '-110' `RECEIVED UNEXPECTED TRAP3 - TRAP15 EXCEPTION.`
Generally caused by hardware.

Hex Display '-111' `RECEIVED UNEXPECTED UNASSIGNED (C0-1FF) EXCEPTION.`
Caused by hardware.

3.3 RECOVERABLE LCN ERRORS

Recoverable LCN errors are not fatal and the program execution continues without operator intervention. The following general message structure is used to report these errors to the Master Error Reporting Module:

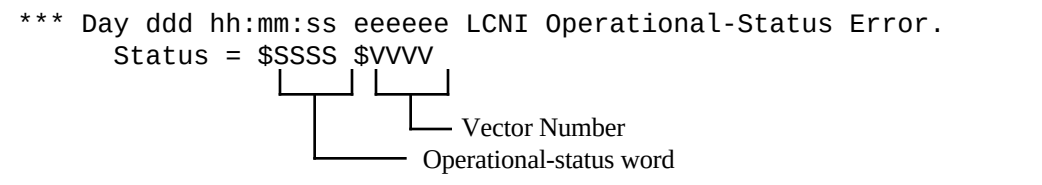


6785

While this basic structure is used for all nonfatal error messages, some provide the operator with additional status information. In these cases, the message overflows to additional lines. See the following individual error descriptions for more details.

3.3.1 Operational Status Error

Operational-status errors are discovered by the LCNI driver interrupt-service routines. This error is caused by finding the Stolen Token bit (bit 7) set in the operational-status register (Register Base + $A). The following form is used to report the error:



6785

3.3.2 Alone In The Ring Error

This error is detected by the LCNI driver board-level interrupt-service routine. It indicates that when an attempt was made to pass the token, no other nodes responded. Because this error indicates a ring collapse, it may not be reported to the Master Error Reporting Module. The error count is incremented in any case. The message for this error is as follows:

```
*** Day ddd hh:mm:ss eeeeee  Alone In The Ring.
```

3.3.3 Media-Dead Error

This error is detected by the LCNI driver board-level interrupt-service routine. It indicates that the LCNI has detected silence on both cables and the token ring has collapsed. The program automatically begins attempts to re-establish the token ring. Attempts are made at the rate of 20-per-second and no limit is placed on the number of retry attempts. When the network is down, the program continues to perform commands typed at the local keyboard. Commands directed to the network are executed by the local node but no attempts are made to transmit them to the network. Network commands should be retyped after the network has been re-established, as they are not saved. It should be noted that collapse of the network is very likely to result in a loss of messages (including errors being sent to the Master Error Reporting Module). The message for this error is as follows:

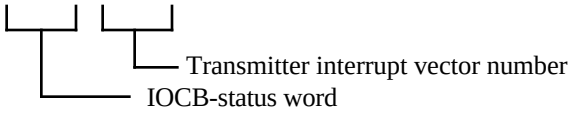
```
*** Day ddd hh:mm:ss eeeeee  Both Media Dead.
```

3.3.4 IOCB-Status Error

IOCB-status errors are discovered by the LCNI-driver IOCB interrupt-service routines. There are two variations of this error, depending on the interrupt vector being serviced. If the error is discovered when servicing the Transmit IOCB chain (vector 1), the error is caused by finding the Transmit-Error bit (bit 14) set in the transmit IOCB-status word (IOCB + $A).

These errors are reported in the following format:

```
*** Day ddd hh:mm:ss eeeeee LCNI IOCB Status Error.
Status = $SSSS $0001
```



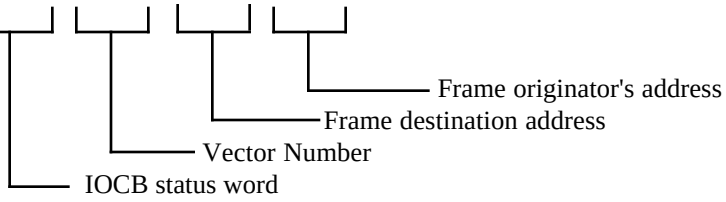
6785

If the error is detected while servicing either of the Receive-IOCB chains (vectors 2 and 3), the error is caused by finding any of the following Receive-IOCB status-word (IOCB + $A) bits set:

- FCS Error (Bit 14)
- Framing Error (Bit 13)
- Oversize Frame Error (Bit 12)
- Receiver Overrun Error (Bit 11)
- Manchester Decoder Error (Bit 10)
- Bits 9 through 1

Messages received with IOCB status-word errors are not processed and will cause a time out at some node in the network. The message format for these errors is as follows:

```
*** Day ddd hh:mm:ss eeeeee LCNI IOCB Status Error.
Status = $SSSS $VVVV $DDDD $OOOO
```



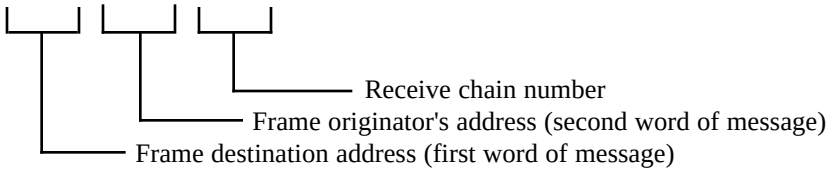
6785

Note that the frame-originator address is contained in the lower byte of this word. If bit 15 of this word is set, it indicates that the message was sent to receive-chain #2. If bit 15 is reset, it indicates that the message was sent to receive-chain #1.

3.3.5 Undersized-Frame Error

This error is detected by the Executive while processing an incoming message. It indicates that the received message was too small to be a valid test or command message. The most common source of these undersized frames is the periodic state messages sent by the processor firmware when it is in charge of a node. These messages are not processed and if sent by another Net Test image, cause a time out at the sending node. The error is reported using the following format:

```
*** Day ddd hh:mm:ss eeeeee Received undersized frame.
Header was $DDDD $OOOO $CCCC
```



6785

Note that the format used by the processor firmware does not match the format used by the Net Test. Because of this, the meanings of the various fields differ. In all cases, however, the first two hex words printed are the first two words of the received frame. If the frame was sent by another Net Test image, the second word contains the originator's address in the lower byte. If bit 15 of this word is set, it indicates that the message was sent to receive-chain #2. If bit 15 is reset, it indicates that the message was sent to receive-chain #1.

3.3.6 Response Time-Out Error

When the Executive sends a message to another node in the network, it starts a five-second timer. If the remote node fails to acknowledge reception of the message within this five-second period, the Executive prints this error message. Note that the loss of either the original message or the acknowledge message will cause this error. The format of the message is as follows:

```
*** Day ddd hh:mm:ss eeeeee Node AAA failed to respond.
```

6785

If a node fails to respond to five messages in a row, the program stops transmitting test messages to that node and the following addition is made to the time-out message above:

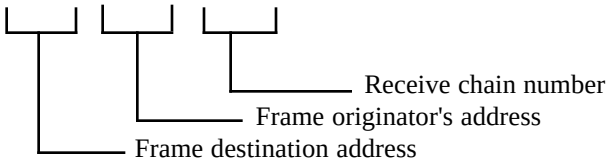
```
Node AAA has reached the message time-out error limit of five.  No
further test messages will be sent to this node.
```

The CLE command resets all the individual time-out counters and resets the overall error count. Alternately, the ENA command can be used to re-enter the list of node addresses, as the clearing of the node-address list also resets the time-out counters. This alternate method does not clear the overall error count, and can be executed in only the LISTEN mode.

3.3.7 Incorrect Receive Chain Error

Each message transmitted by the program contains a flag that indicates the intended receive chain. If the message arrives on the wrong receive chain, the Executive reports an Incorrect Receive Chain Error. The following format is used:

```
*** Day ddd hh:mm:ss eeeeee A message was received on the wrong
receive chain.
Header was $DDDD $OOOO $CCCC
```



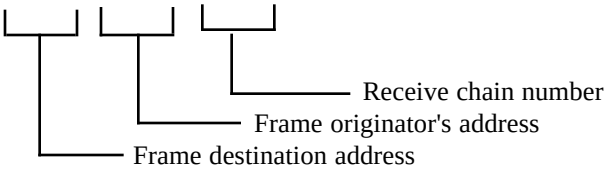
6786

Note that the frame-originator address is contained in the lower byte of this word. If bit 15 of this word is set, it indicates that the message was sent to receive-chain #2. If bit 15 is reset, it indicates that the message was sent to receive-chain #1.

3.3.8 Incorrect Checksum Error

Each message transmitted through the network carries a software-generated checksum that is generated by the transmitting node. This checksum is recalculated at the receiving node and compared to the checksum contained in the message. If it is incorrect, the Executive reports an Invalid Checksum Error. The following format is used:

```
*** Day ddd hh:mm:ss eeeeee A message was received with an incorrect
software checksum.
Header was $DDDD $OOOO $CCCC
```



6786

Note that the frame-originator address is contained in the lower byte of this word. If bit 15 of this word is set, it indicates that the message was sent to receive-chain #2. If bit 15 is reset, it indicates that the message was sent to receive-chain #1.

3.4 CLOCK SUBSYSTEM ERRORS

The Network Test was not designed to test the clock subsystem; therefore, although the program DOES MONITOR the operating status of the clock subsystem, it DOES NOT ATTEMPT ERROR RECOVERY IN THE CLOCK SUBSYSTEM AREA. The purpose of monitoring the clock subsystem is to check for interference between the clock-timing signals and the LCN-data signals. Clock error messages are displayed at the Master Error Reporting Module, but clock errors are not included in the error count printed by the ERR command. It is the responsibility of the test operator to interpret the messages related to the clock subsystem and take whatever steps may be necessary to bring the clock subsystem back to its original state.

3.4.1 Clock Time-Set Message

When the clock driver detects a time-set operation, this message appears:

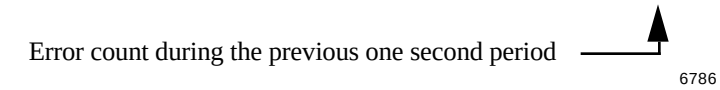
```
+++ Clock time-set performed, new time is -- Day ddd hh:mm:ss
```

Time-set operations are a result of the CLZ command and are printed at every node that detected the time-set operation. No operator action is required as this is not an error.

3.4.2 Clock Power Line Sync Error

When the clock subsystem is using the power line as a synchronization source, missing or displaced line interrupts cause this message to be printed. No operator action is required for this error. The format of the message is as follows:

```
+++ Day ddd hh:mm:ss Clock power line sync errors, count = X
```



3.4.3 Clock Sync Source Change

If three errors are accumulated against the current synchronization source within the previous one-second period, the clock subsystem changes to a back-up sync source. The message indicates that a change in sync source has been made and indicates the new source. The format of the message appears below:

```
+++ Day ddd hh:mm:ss Clock sync change, running without sync
                or
+++ Day ddd hh:mm:ss Clock sync change, running with power line sync
                or
+++ Day ddd hh:mm:ss Clock sync change, running with cable sync
                or
+++ Day ddd hh:mm:ss Clock sync change, running with illegal source
```

The operator action for this error is dependent on the previous sync source. If the previous source was "cable sync," typing CLZ should put the clock source back to "cable sync," assuming that the conditions were only temporary. If this action fails to bring the clock subsystem back to "cable sync," there is most likely a hardware error somewhere in the network clock. If the previous sync source was other than "cable sync," there is currently no error recovery procedure. Clock sync-source changes often cause a change in the clock subsystem-operating mode. For further information consult the description of the "Clock Mode Change" error.

3.4.4 Clock Mode Change

Any time the clock subsystem changes operating modes, this message is printed to indicate the new mode. If the previous clock mode had been LISTENER and the clock source changed from "cable sync" to "power-line sync," the clock subsystem would change to the LOCAL mode, having lost sync with the network. Not all mode changes are errors. The CLM, CLS, and CLZ commands are all capable of causing a mode change.

The text of the message is shown below:

```
+++ Day ddd hh:mm:ss Clock mode change, mode is master
                or
+++ Day ddd hh:mm:ss Clock mode change, mode is slave
                or
+++ Day ddd hh:mm:ss Clock mode change, mode is listener
                or
+++ Day ddd hh:mm:ss Clock mode change, mode is local
```

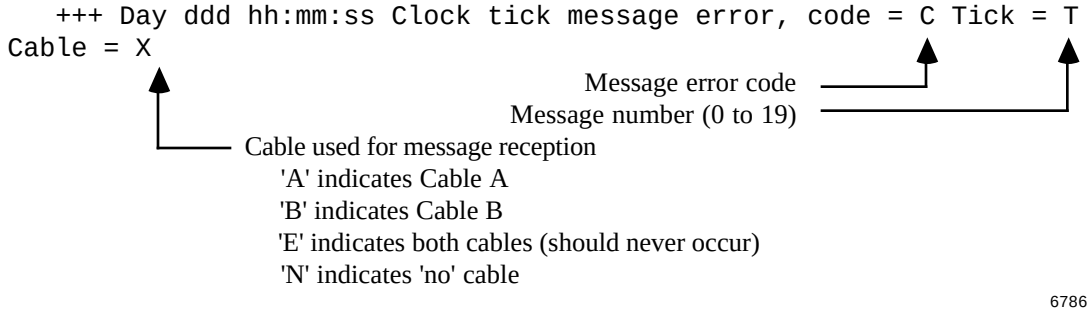
3.4.5 Clock Interrupt Overlap

If the clock driver fails to complete the processing of a clock interrupt before the clock subsystem issues its next interrupt, a bit is set in the clock-status word. The presence of that bit causes the following message:

```
+++ Day ddd hh:mm:ss Clock interrupt overlap error.
```

3.4.6 Clock Tick Message Error

Every 50 msec. (20 times/second), the network-clock source transmits the time information. If an error is discovered in the incoming time message, the following is printed:



CROSSED CABLE TESTING
Section 4

This section describes the crossed cable detection procedure used in systems that have only K2LCN or K4LCN boards installed in the modules (that is, no CSR boards driving the 12.5 kHz clock).

4.1 TEST PROCEDURE

In an LCN system (or segment of a system) that has only modules with K2LCN or K4LCN boards installed, it is impossible for the CLKS test in HVTS to detect crossed LCN cables because it is not possible to disable one or the other of the clock transmitters.

To detect crossed LCN cables, use the following procedure.

1. Load the LCNI test.

2. Establish a Master Error Reporting node and put the test in run mode.

3. Lock all nodes to Cable A using the command NET SMA.

4. Let the test run for a few minutes to establish a baseline of error reporting.

5. On a node other than the Master Error Reporting Node, remove the B cable by disconnecting the “tee” from the LCN I/O board. If the LCN cables are not crossed, no errors will occur. If errors do occur, nodes that report loss of communications are on the wrong LCN cable.

6. Replace the “tee” and use a NET CLE command to clear the errors.

7. Lock all nodes to Cable B using the command NET SMB.

8. On a node other than the Master Error Reporting node, remove the A cable by disconnecting the “tee” from the LCN I/O board. If the LCN cables are not crossed, no error will occur. If the cables are reversed, the same group of nodes (as in step 5) will report loss of communications.

   By knowing the physical routing of the cables (consult the system topology map), it is possible to isolate where the cables have been switched and if there is more than one location that the cables are switched.

   Be aware that the node from which the cables are removed can itself be listening to the wrong cable.

READER COMMENTS

Honeywell IAC Automation College welcomes your comments and suggestions to improve future editions of this and other publications.

You can communicate your thoughts to us by fax, mail, or toll-free telephone call. We would like to acknowledge your comments; please include your complet name and address

BY FAX: Use this form; and fax to us at (602) 313-4108

BY TELEPHONE: In the U.S.A. use our toll-free number 1*800-822-7673 (available in the 48 contiguous states except Arizona; in Arizona dial 1-602-313-5558).

BY MAIL: Use this form; detach, fold, tape closed, and mail to us.

Title of Publication: **LCNI Network Communications Test** Issue Date: **9/95**

Publication Number: **SW13-513**

Writer: **C. Phillips**

COMMENTS: _____

_____

_____

_____

_____

_____

_____

_____

RECOMMENDATIONS:_____

_____

_____

_____

_____

_____

_____

_____

NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE _____ FAX _____

(If returning by mail, please tape closed; Postal regulations prohibit use of staples.)

Communications concerning technical publications should be directed to:

Automation College
Industrial Automation and Control
Honeywell Inc.
2820 West Kelton Lane
Phoenix, Arizona 85023-3028

FOLD FOLD

From: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE USA

BUSINESS REPLY MAIL

FIRST CLASS **PERMIT NO. 4332** PHOENIX, ARIZONA

POSTAGE WILL BE PAID BY ....

Honeywell

Industrial Automation and Control
2820 West Kelton Lane
Phoenix, Arizona 85023-3028

Attention: Manager, Quality

FOLD FOLD

Cut Along Line

Additional Comments:

Honeywell

Industrial Automation and Control
Honeywell Inc.
16404 North Black Canyon Highway
Phoenix, Arizona 85023-3033

Helping You Control Your World