Honeywell

Understand the TPS Network Software Environment

L61520

LCN

TotalPlant

Notices and Trademarks

This module supports **TotalPlant** Solution (TPS) system network.

TPS is the evolution of TDC 3000X.

Honeywell Inc.

Industrial Automation and Control

Automation College

2820 West Kelton Lane

Phoenix, AZ 85053-3028

1-800 852-3211

Table of Contents

Figures

TABLES

MODULE INTRODUCTION

The purpose of this module is to give you an understanding of the Honeywell Real-time Network Operating System (RNOS), and the Software Environment. The topics included in this module are the Node Administrator, Complex Command Handler, Work Management, File Manager, Communication Manager, and Base Application Software.

THE RNOS CONCEPT

The Honeywell Realtime Network Operating System (RNOS) has been built to run on the Motorola 68000/68020/68040 microprocessors. RNOS is built on top of MTOS-68K kernel (Multi-Tasking Operating System for the M68000/M68020/68040 processors) to support the TPS Network system requirements. MTOS is a trademark of Industrial Programming, Inc.

To the user of the TPS Network and the related software, the functionality of RNOS is transparent. There should never be any reason for the user to access the RNOS files or to view or modify the RNOS program in any manner. Any problem the user suspects with the operating system should immediately be directed to Honeywell Technical Assistance Center (TAC).

NODE STRUCTURE

A node is a layered set of functions.

The *hardware layer* consists of the components and firmware in the node. This layer is accessed when power is applied to the node. An important component of the hardware is the LCN interface, consisting of the LCN board and LCN paddle board.

The *software environment* consists of an MTOS kernel (multi-tasking operating system), on top of which is RNOS (real-time network operating system):

- MTOS is an operating system created by Industrial Programming, Inc. It works with the Motorola series of 68020/68040 processors to provide RAM memory management and device driver interfaces.
- RNOS, a Honeywell operating system specific to the LCN, consists of a set of application interfaces, including those for parameter access, file transfer, and network queries.

The Honeywell *Base Application Software* (BAS) consists of all Honeywell-provided applications unique to each node type.

The *user application layer* consists of all the user-created or provided applications and database, such as points, schematics, CL programs, and external load modules.

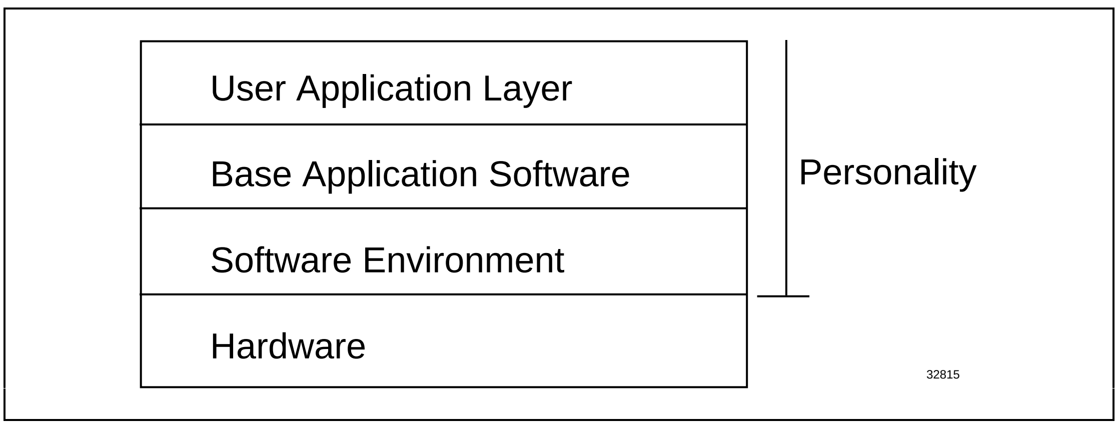


Figure 1 - TPS Network Node Structure

DATABASE STRUCTURE

The TPS Network uses a distributed database concept. Each physical node can contain one or more databases. The transport of data to and from the database is provided by the software environment.

Data for transport is accessed by application functions called data owners. The data owners respond upon request by the software environment data transport mechanism. Data owners exist because each application must organize its data primarily for its own use; therefore, conversion of that data to the common transport form becomes the application's responsibility. The data user then converts the data it obtains through the software environment to a format required for its own use.

The Software Environment provides a universal format for all data on the LCN. The applications provide their own formatting for the user's view of the data.

In redundant nodes, the synchronization between databases is performed by application software (sometimes assisted by hardware).

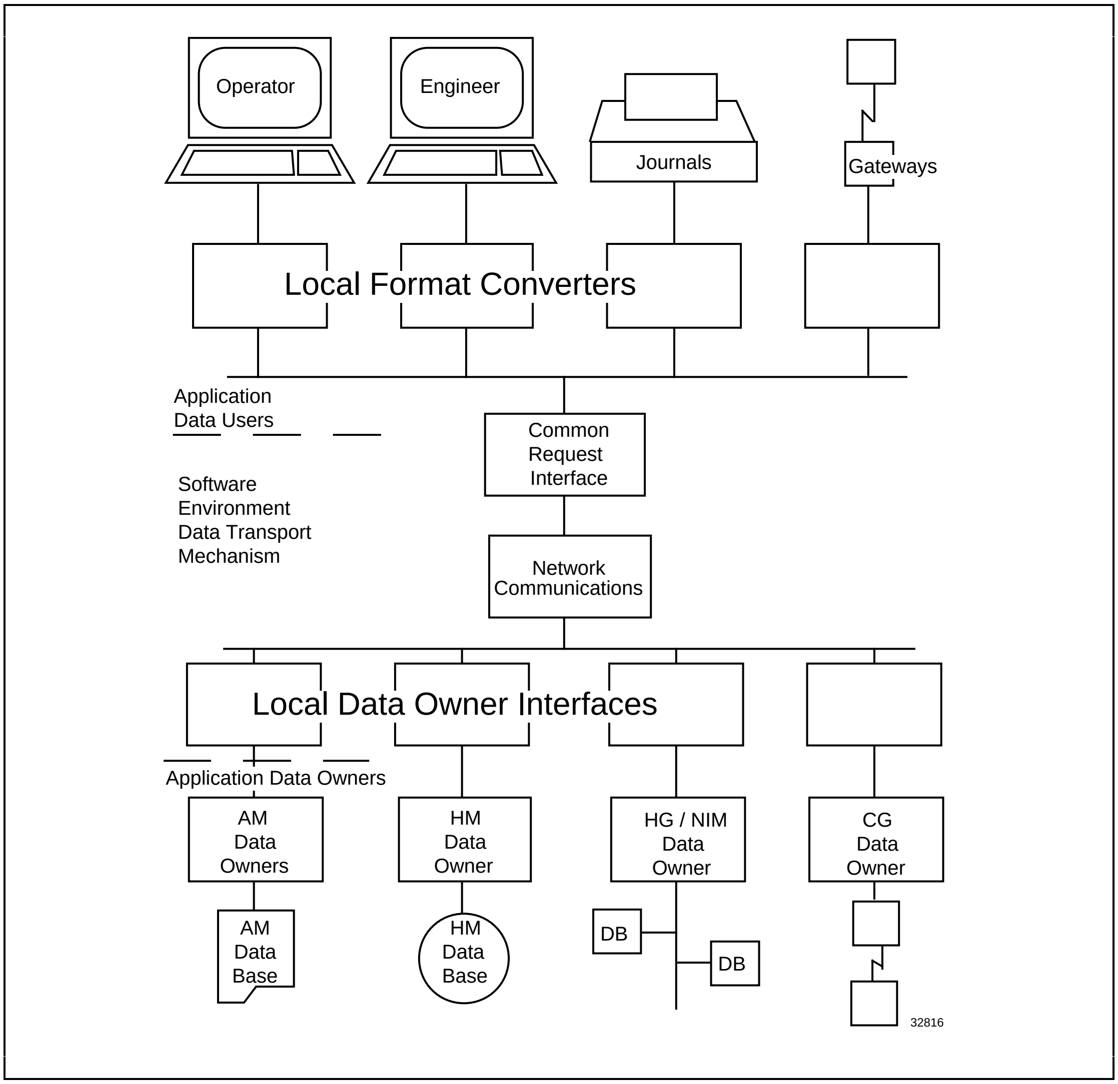


Figure 2 - TPS Network Database Management

THE SOFTWARE ENVIRONMENT

The software environment includes the following components:

1. The MTOS underlying operating system
2. A set of local support utilities
   - Boot Loader
   - Memory Dump
   - Heap Management
   - Critical Region Management
   - Task Scheduling
   - Exception Management
   - Local File Management
3. A set of I/O Drivers
   - Memory Drivers
   - Clock Drivers
   - Display Drivers
   - Keyboard input and output Drivers
   - Printer Drivers
   - Data Hiway Drivers
   - SCSI Drivers
   - Process Network Interface Driver
   - LCN Driver
4. Application Network Communication Mechanisms
   - Work Management
   - File Management
   - Parameter Access
   - Queries
   - Name Conversion
   - The Complex Command Handler
   - Event Distribution
5. Diagnostic Tools
   - On-Process Test
   - Performance Instrumentation
6. Network Administrator
   - Node Administrator
   - Clock Management
   - Communication Management
   - Transaction Management

Not all elements of each component exist in every node [for example, query requestors exist in Universal Stations (USs), Global User Stations (GUSs), and Computer Gateways (CGs); query servers exist in History Modules (HMs), Hiway Gateways (HGs), Network/Interface Modules (NIMs), and Application Modules (AMs)].

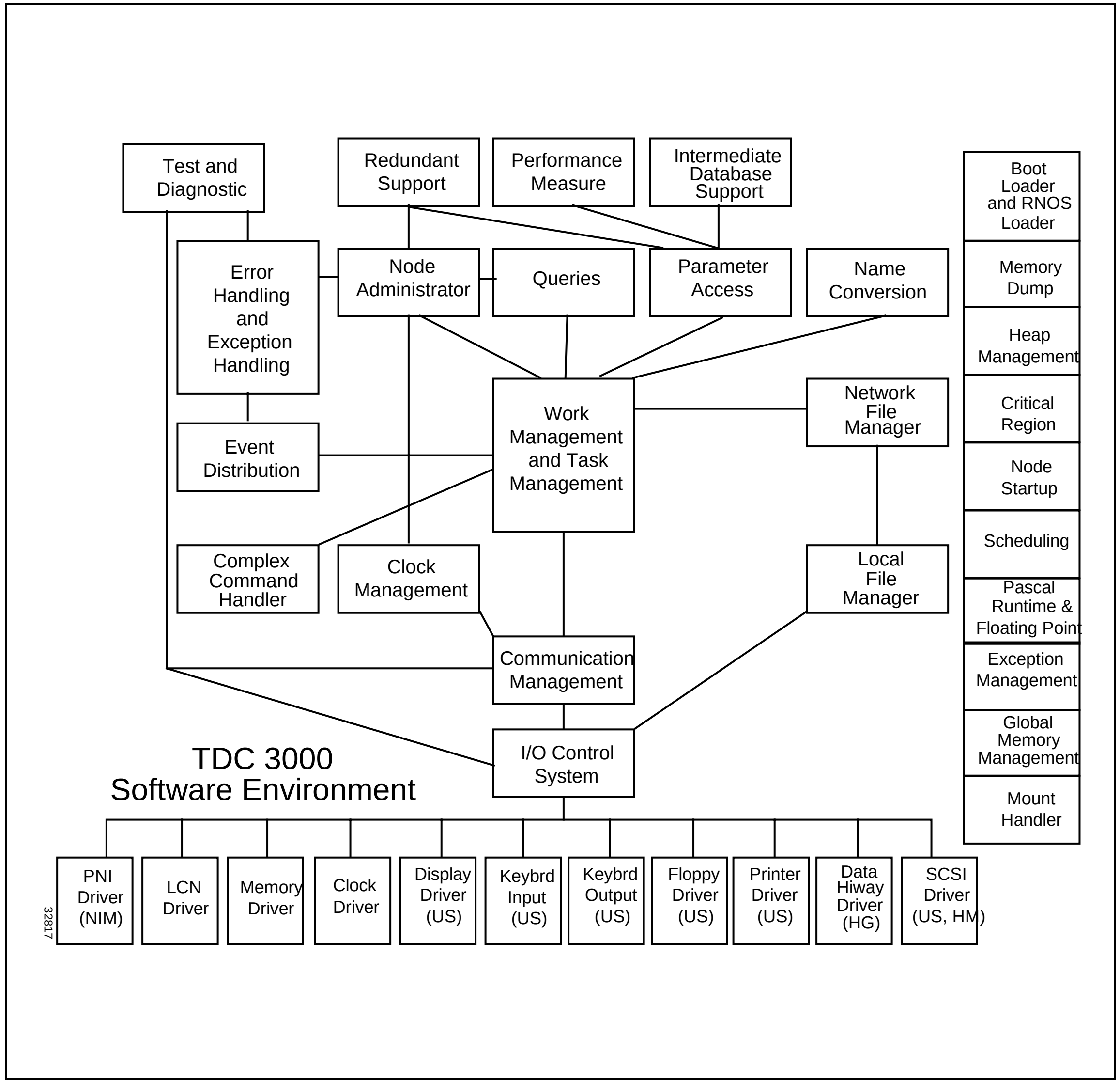


Figure 3 - Software Environment

COMMUNICATION LAYERING

Applications may initiate network communications by calling any of the following Software Environment functions:

Parameter access	To access individual parameters or lists of parameters from entities
Query	To identify entities in a node or unit or having certain characteristics in common, and to access historical data
File transfer	To access files or records of files
Name conversion	To convert names to internal IDs for optimum access at a later time
Error handler	To report errors and initiate error handling
On-Process Tests	To perform background testing, memory cleansing, and integrity checks
Command handler	To receive immediate acknowledgement that a request has been received and the processing initiated
Event distribution	To annunciate and historize events

In addition, on-process test runs periodically and initiates network communication to report errors or initiate test.

All communication interfaces ultimately call on work management to perform communications.

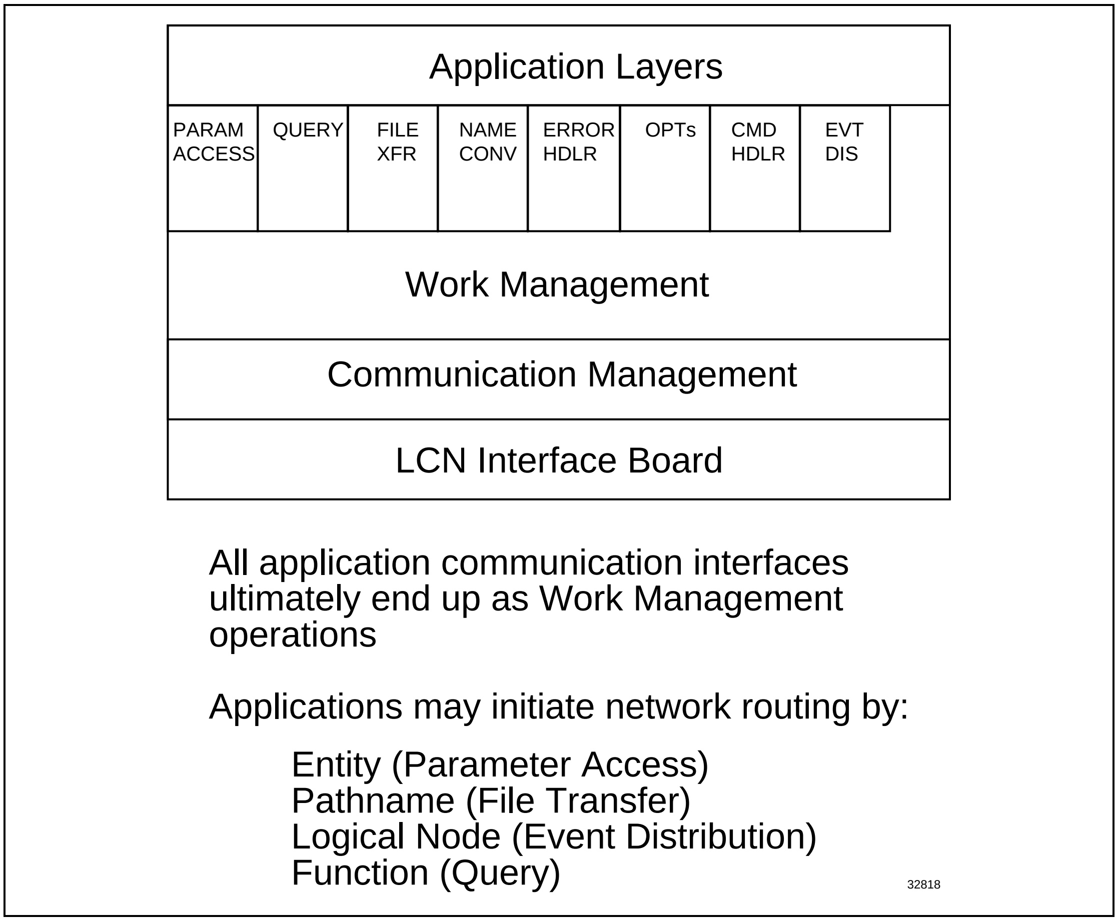


Figure 4 - Communication Layering

LOGICAL ADDRESSING

The software personalities and databases in the TPS Network physical nodes are gathered into these functional groupings:

- Function Set
- Data Realm Qualifier
- Logical Node Instance
- Function Set Qualifier
- Logical Node Alias
- Job Identifier

Function Sets (Fs)

A Function Set is a set of tasks and routines within a physical node interacting with each other to perform a well-defined function.
Examples:

- Control
- Data Access
- Area_Manager
- Console_Manager

Data Realm Qualifier (DRQ)

The data realm qualifier is the specific data upon which a function set operates.
Examples:

Function Set	Data Realm Qualifier
Area_Manager (in a Universal Station)	Area_03.
Control (in an Application Module)	Unit_01.

Logical Node (LN)

A logical node is defined by a Function Set and a Data Realm. In the majority of cases, a logical node is completely contained within a physical node. The major categories for logical nodes are

- Processor Status Data Point ($PRSTSn) for node n
- AM or CG for unit n
- US or GUS for area n
- HG for hiway n
- NIM for network n

Logical Node Instance (LNI)

A logical node instance is the result of assigning a function set (FS) a scope of work (DRQ) within a physical node:

LNI = FS + DRQ

In this way, the same function set can operate in several logical node instances by assigning different scopes of work. For example, in a single AM, there can be the following logical node instances:

Control.Unit_03
Control.Unit_04

Function Set Qualifier (FSQ)

A function set qualifier is used to define different addressing groups of function sets. For example, redundant nodes will each have an instance of the same logical node. These can be differentiated by applying the function set qualifiers of Primary and Secondary:

Primary.Control.Unit_03

Some logical nodes have multiple instances (such as the Area_Manager, existing in each Universal Station), and are communicated to in groups by using a group name determined by the function set qualifier:

Console_02.Area_Manager.Area03

Logical Node Alias (LNA)

Logical node aliases are logical node instances with different function set qualifiers assigned. The LCN on which the message is to be routed must also be specified:

LNA = LCN# + FSQ + LNI = LCN# + FSQ + FS + DRQ

Examples:

LCN1.Primary.Control.Unit_01
LCN1.Console_02.Area_Manager.Area_03

Job Identifiers

When messages are sent on the system, they are sent to job queues. A job queue is addressed by a Job ID, consisting of the Logical Node Alias plus a Job Type (JT) that specifies the function to be performed by the logical node instance:

Job ID = LNA + JT = LCN# + FSQ + FS + DRQ + JT

Examples:

Job ID = LCN1.Primary.Control.Unit_01.Data_Access
Job ID = LCN1.Console_02.Area_Manager.Area_03.Command

Job queues exist only within physical nodes. To get to the physical node, the full logical name portion of the Job ID is converted to a number between 1 and 8064, called the LCN logical address.

Summary:

Job Type (JT) = Function performed by Logical Node (LN).
Job Identifier = Logical Node Alias (LNA) + Job Type (JT).

LCN +	**FSQ +**	**FS +**	**DRQ +**	**JT**
Local	Console_02	Area_Manager	Area_01	Command
Local	Primary	PPR_Control	Unit_03	Data Access

FS (spans the FS column)

LNI (spans the FS and DRQ columns)

Job Identifier (spans all columns)

The Job Identifier can be seen at the Detailed Module Errors display from System Maintenance Control Center (SMCC).

Job Identifiers (Continued)

```
                                                          18 Sep 96 17:11:52
DETAILED MODULE ERRORS

                    MODULE NUMBER : [19]          PRINT       [$P1]
                    Node Type           Application Module
                    Node State           Running

        Error Entity Type : Node Amin Status Broadcast Timeout
        Node   :  4
        Error Descriptor Size  : 11
        Error Descriptor Type : Communication Error
        Error Occurrence Time:  04 Jan 83 11:39:46:050
        Error Descriptor Info
        Task Name  : $$SWATCHDOG
        Second  Party  :  4
        Second  Party  Status  :  Timeout
        Job  ID
                                              LCN  :  0
                                              FSQ  :  0
                                        FS    :  30
          DRQ   :  0
          Job Type : 5

 For Information On Functions And Options Displayed On This Menu,
 Position The Cursor On the Desired Target And Press HELP.
```

35352

MESSAGING

Messages can be sent on the LCN in three ways:

1. To a single node (referred to as Physical Node-to-Physical Node or Point-to-Point messaging).
2. To all nodes (referred to as a Broadcast). This uses a physical node-type of information frame with a broadcast bit set in the frame.
3. To a subset of nodes (referred to as a Multicast message) using logical addressing.

Logical Addressing

Within each node's RAM memory there exists a Logical Address Filter Table. It is 1008 bytes long (8064 bits). Each bit position corresponds to a logical node instance within the physical node. This filter table is used to control which messages are accepted by the physical node when logical addressing is used.

Example of a Multicast

1. A Hiway Gateway (HG) point goes into alarm.

2. The HG needs to notify all USs or GUSs that have that point's unit assigned to their Area Databases (This is the logical node instance for the US/GUS.).

3. If the HG uses a PN to PN method of communicating, it has to send individual messages to each US.

   If the HG uses a multicast method of communicating, it only sends one message since only those nodes with the corresponding logical node instance will accept the message.

4. Each LCN interface board of each LCN node receives each message.

5. When a node receives a message with a logical address, it looks in its filter table to see if the appropriate bit position is set.

   (Each node has a logical address filter table in its memory in which the bits were set when the node was started up.)

6. If the filter table bit position corresponds to what is in the frame, the node accepts the message; otherwise, the node doesn't accept the message.

   In this way, a multicast can be used to prevent the node from sending more than one message.

WORK MANAGEMENT

Definitions

Term	Definition
Producer	A task that causes work to be done by another task. It does this by "producing" a work request and submitting it to Work Management.
Work Request	A specific request made by a producer for a specific job.
Job	A category of work.
Consumer	A service task that processes a work request supplied by Work Management.
Reviewer	A task (usually the original producer of the work request) that reviews the changed contents of a message after it has been consumed by a service task.
Work Management	Accepts work requests from producers and schedules processing by suitable consumers.

Communication Calls

Communications from the user's view, whether the user is an application or a higher level software environment function, is encapsulated in three calls:

- produce
- consume
- review

Communication Process

1. The initiator of a network message does a “produce” that is handled by Work Management.

2. Work Management examines the request to see if the receiving queue is local to the node:

If receiving queue is...	Then...
Local to the Node	Work Management passes the message to the local queue and initiates the task that can respond to the work request.
Not Local to the Node	Work Management passes the message to Communication Management, which converts the Job ID to an LCN logical address and then passes it to the LCN driver.

3. Communication Management breaks the message into packets.

4. The LCN driver incorporates the packets into LCN-sized frames and transmits them, one frame per token pass, on the LCN.

5. Assume the message is a multicast type message. Any node with the appropriate logical address set in its logical address filter table accepts the packets and passes them to the LCN driver.

6. The LCN driver passes the packets to the Communication Manager on the receiving end, who puts the message packets back together and turns the reconstituted message back over to Work Management.

7. Work Management then puts the message in the assigned job queues and initiates the service task that can respond to the work request.

8. The service task “consumes” the message.

9. When the service task has serviced the message, if a “review” has been requested, the reply message is passed back through Work Management and Communication Management to the requesting task review queue.

10. The requester makes a “review” call and either receives the reply or waits until Work Management triggers it when the reply comes in.

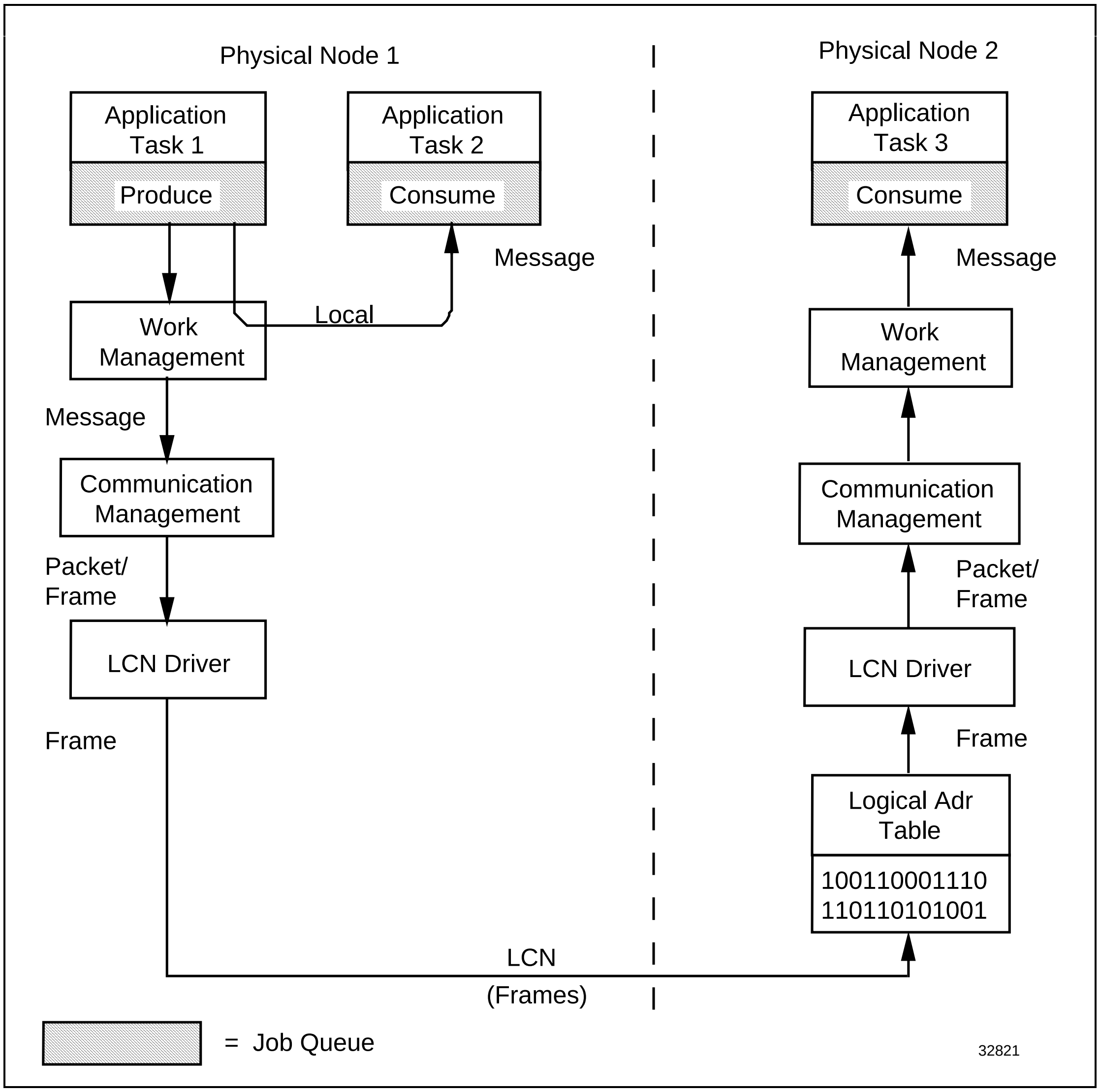


Figure 5 - Work Management

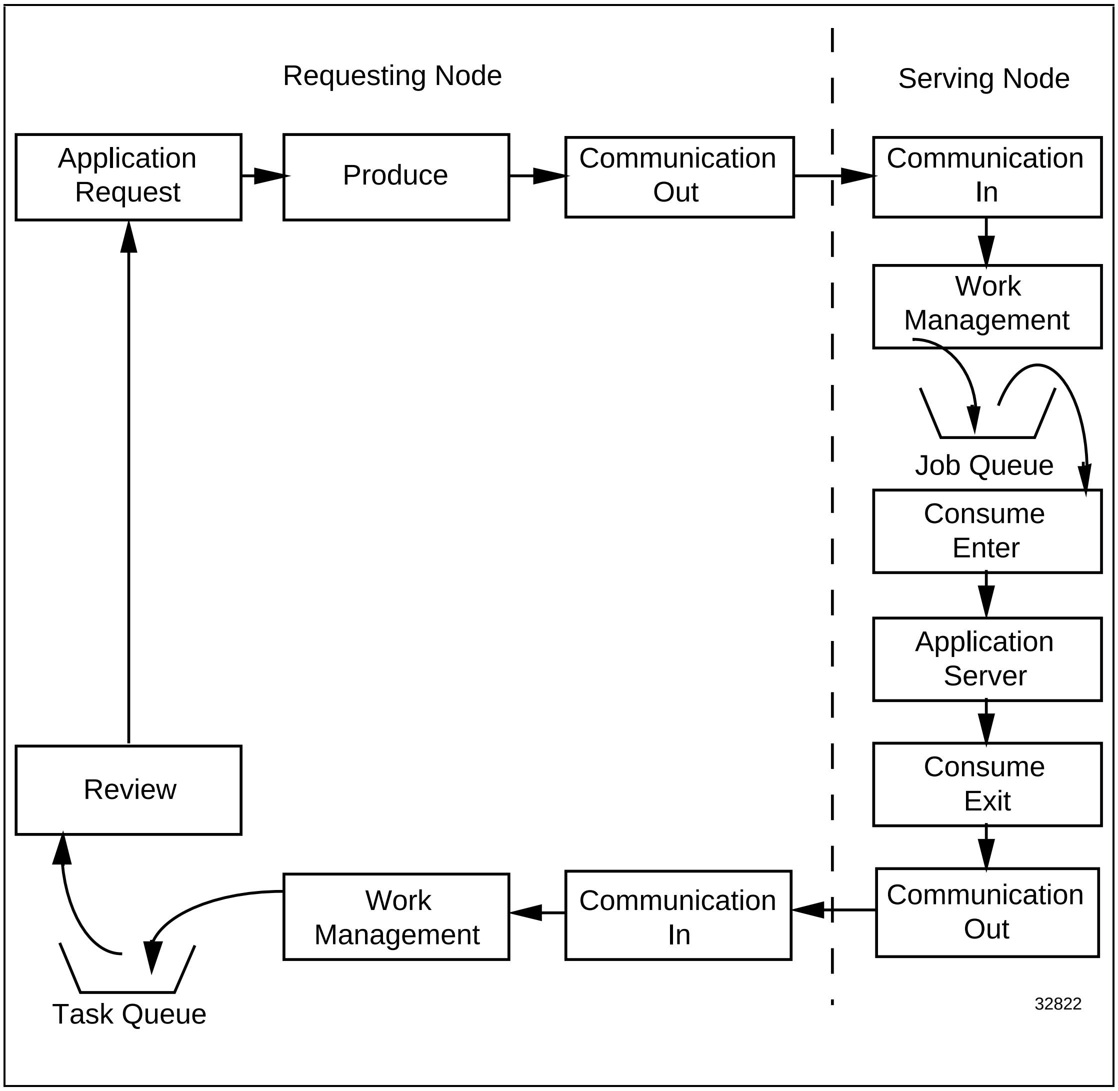


Figure 6 - Network Communications

NODE ADMINISTRATOR

Once every 30 seconds, the Node Administrator broadcasts a message to all nodes giving its status and the state of all its logical nodes. On a change of state, it broadcasts the change immediately. In either case, it does not wait for acknowledgements. The Node Administrator expects to get a message from every other node every 30 seconds. When the Node Administrator in a node misses messages from another node, communication between the two nodes may not be reliable.

If three or more nodes miss four Node Administrator broadcasts in a row from another node, the node becomes isolated from those nodes that missed its broadcast. In R320 and later, a node can become isolated if another single node misses six of its consecutive Node Administrator broadcasts from another node. To re-establish communication between two isolated nodes, one of them must be shut down and reloaded.

Each application can request state-change notification by providing the Node Administrator with a job queue in which to put them. A query mechanism also exists for anyone to find out node states.

The Node Administrator is also responsible for starting up the node function sets, keeping track of the location and revision of the Network Configuration File (NCF), doing board type checks, and keeping the network status synchronized.

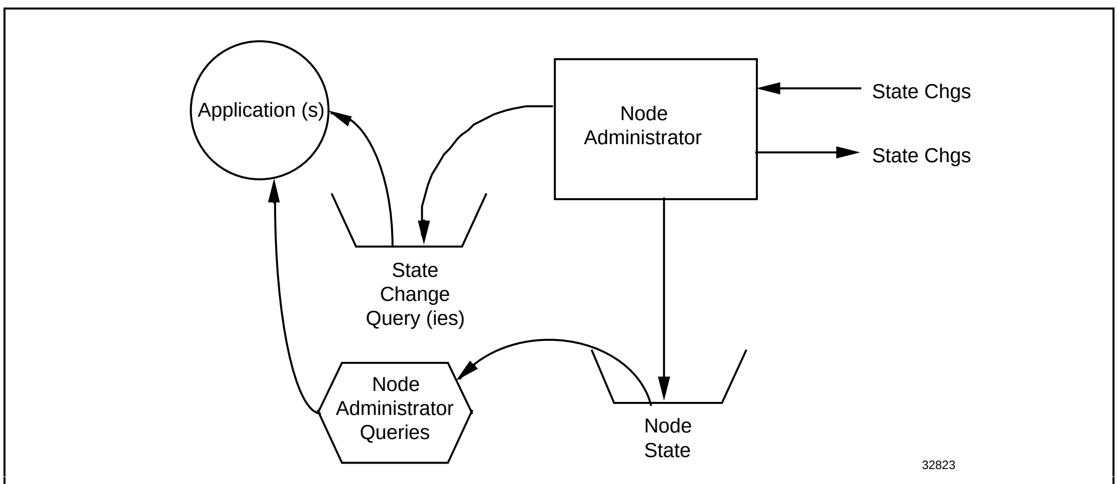


Figure 7 - Node Administrator

COMPLEX COMMAND HANDLER

A complex command is a command that requires that the Requestor first be told that the command has been accepted and then be notified that the command has been completed.

The Complex Command Handler (CCH) provides this dual-response capability to a single request. This capability is required when the actual function may take too much time for the requesting application to wait for completion, but the application must know that the request has been accepted before it can continue.

After the request has been generated by the requesting node, the Complex Command Handler acknowledges that the server node has accepted the request. The Complex Command Handler performs this service by putting a dispatcher between the requestor and the server in the server's node. Once the request is completed, the service task sends the reply.

Thus, the dispatcher acknowledges the acceptance of the request after putting it in the server's queue, and the server sends the completion message. The requestor and the server must agree on the addressing of the message.

The requestor sends the message to the CCH in the server node.

The server defines to the dispatcher the message placement.

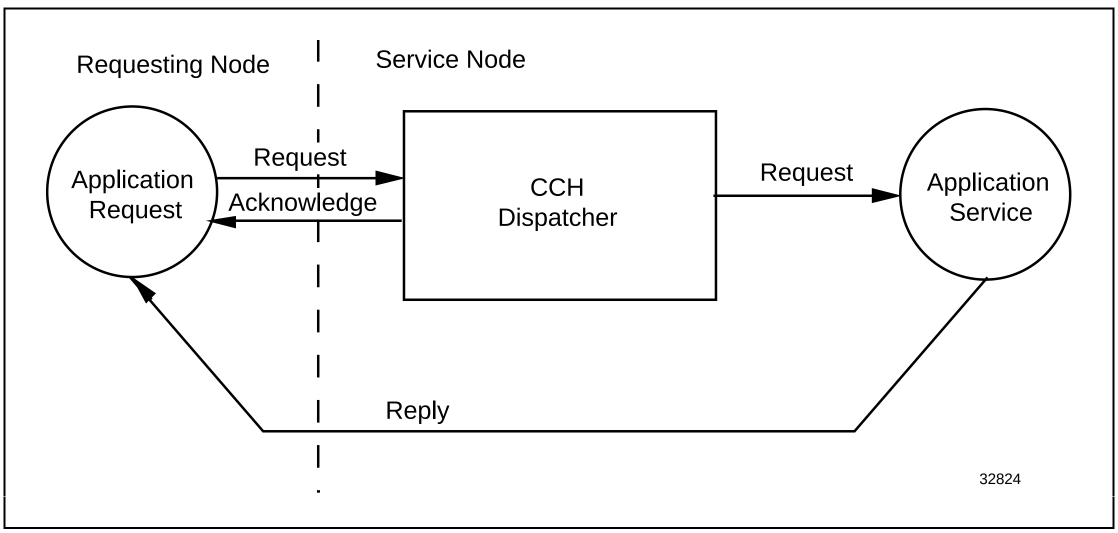


Figure 8 - Complex Command Handler

ERROR HANDLING

Error handling is initiated by applications, by the On-Process Test (OPT), and by components of the software environment. Error handling consists of alarming and journalizing the error and possibly crashing the node.

A local error journal is kept for SMCC analysis of any node crashes. When a node crashes, if it is able, it sends a message out on the LCN broadcasting that it is going down.

Fault analysis is performed on the HM by a function called On-Process Analysis (OPA) using data sent by the error handlers in all the nodes. The data contains a description of the last 32 error blocks in memory. The result is usually a recommendation to the operator.

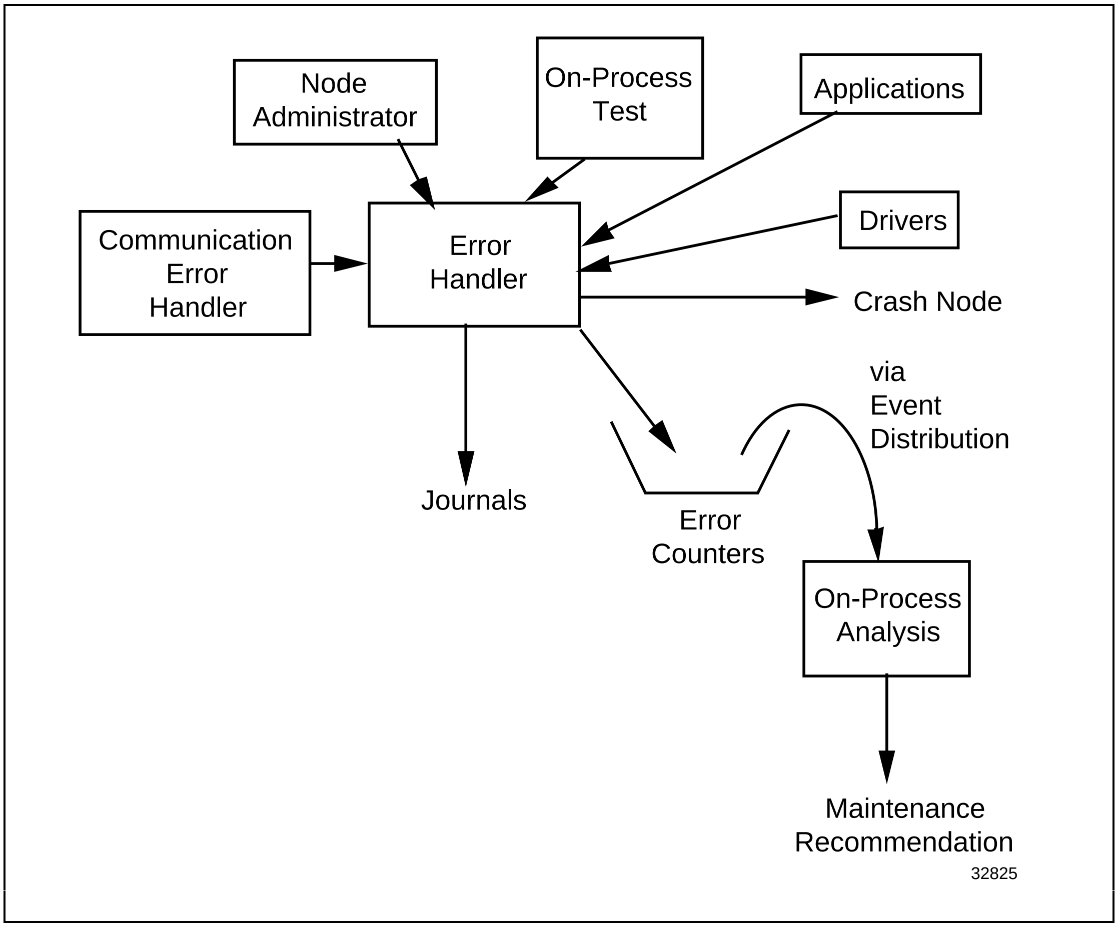


Figure 9 - Error Handling

FILE MANAGER

The file management system consists of four tasks:

Task	Description
Network File Manager	Exists in every node. Allows the node to read files into memory or write memory into files.
File Manager I/O	Allows remote access to files in a node. Found in those nodes with local file systems (such as the US and the HM).
Local File Manager	Manages files within the node or on other devices. Found in those nodes with local file systems (such as the US and the HM). The US has a memory-resident file system and also handles the removable media file system. The HM has its file system on the hard drive(s).
Cyclic File Manager	Resides on the HM. Is used for the journals and history files. The journal and history files are files that "wrap around" when they get full.

The file manager

- Creates, deletes, opens, closes, reads, writes, and copies files around the network.
- Handles output devices (printers) as if they were files.
- Finds files addressed by their logical address.
- Handles redundant hard drives.
- Keeps track of nodes with open files so that it can close them on node failures.
- Synchronizes console global Logical Device IDs (LDIDs).
- Mounts requests for removable media.
- Performs network searches for volumes and files.

EVENT DISTRIBUTION

Events are sent to one or both of the following places:

- Universal Station's (US) or Global User Station's (GUS) Real Time Journal (RTJ), or
- History Module's (HM) historical journals.

Event Distribution sends out a message to one or both of the above receivers and waits for acknowledgements to be returned. The transaction manager keeps track of all the acknowledgements that are returned.

If all the expected acknowledgements are not returned or the transaction manager times out, the events are passed to another task. Along with the events, the transaction manager also sends information as to which receivers did not return acknowledgements.

This task then sends the message point-to-point to those receivers that have missed the events. This is tried three times. If the task is unsuccessful in the three attempts, the event is "lost." In the case of the RTJ, the [Event Recovery] target appears on the US or GUS.

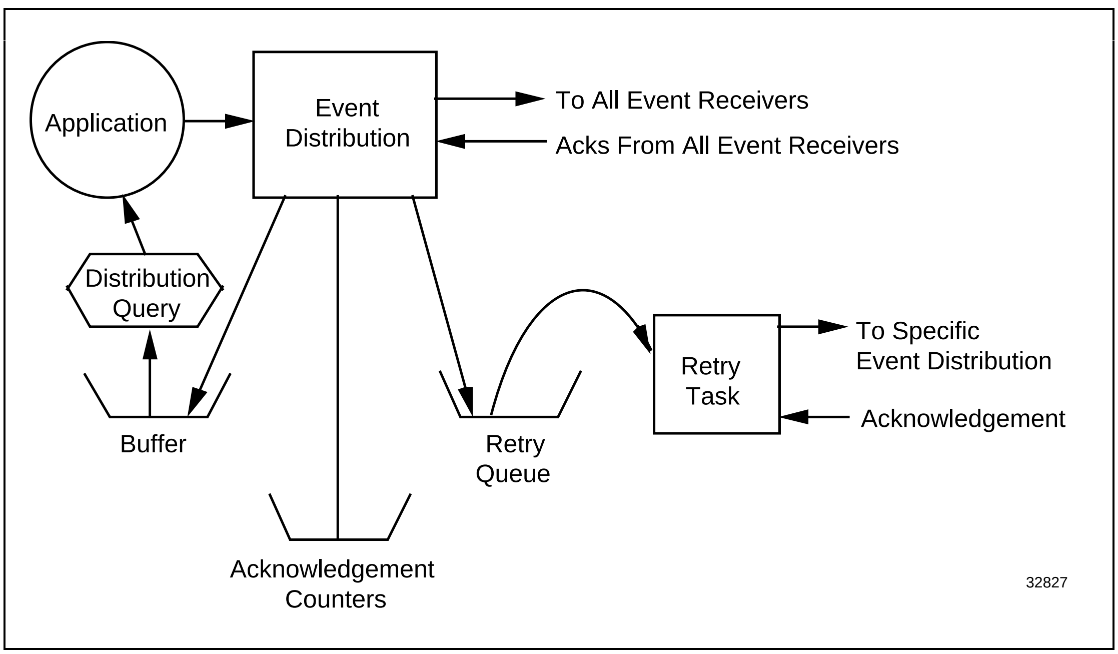


Figure 10 - Event Distribution

DATA ACCESS

The Data Access subsystem has these three parts:

1. Parameter Access
2. Name Conversion
3. Queries

Parameter Access

Parameter access obtains or stores data across the LCN without the requester having to know where the data physically resides or its form. Parameter access does this for either one parameter at a time or a list of up to 1000 parameters. Parameter access provides the utilities to build the list. The list and the values that are cached in the requesting node as a result of processing the request are called the Intermediate Database (IDB). Parameter access splits the list into messages per logical node, with a maximum of 200 parameters per list.

Sequence for a List Request

The sequence for a list (IDB) request is as follows:

1. Create the IDB.

2. Enter the items into the IDB, one or several at a time. An item consists of the point.parameter.qualifier and, on a store request, includes the access key and the value to be stored.

3. Initiate the collection/storage of the parameters. This can be done with wait, no-wait, or alert when complete. On a wait, the requester is frozen until completion. On no-wait, the requester must poll for completion. On an alert, a message is sent to a requestor-specified queue and the requestor must sooner or later wait on that queue.

4. The requestor may get the data out of the IDB after it has been collected in two ways:
   - it could request that the data be stored in its own buffers as part of the collect request, or
   - it can ask for the items one by one using the get-item call.

5. The user may use the IDB to collect or store data as many times as desired. When finished with the IDB, the user must delete it or lose the memory associated with it.

Parameter Access Features

Some of the features of parameter access are

- Macro parameter names can be used in a request, such that a collection of parameters is returned. This is called a “collection set.” An example of this would be the macro parameter “group” that returns all the parameters needed to support the Group display.

- Each parameter access request and service task handles one request at a time. For parallelism, multiple copies of those tasks are used.

- Data owner interfaces to parameter access can be through a procedure call (such as the AM) or by separate tasks (such as the HG or NIM). In the HG and NIM, separate channels are used for US/GUS and other requests (AM and HM) so that an overload of US/GUS requests does not interfere with tasks such as control and history collection.

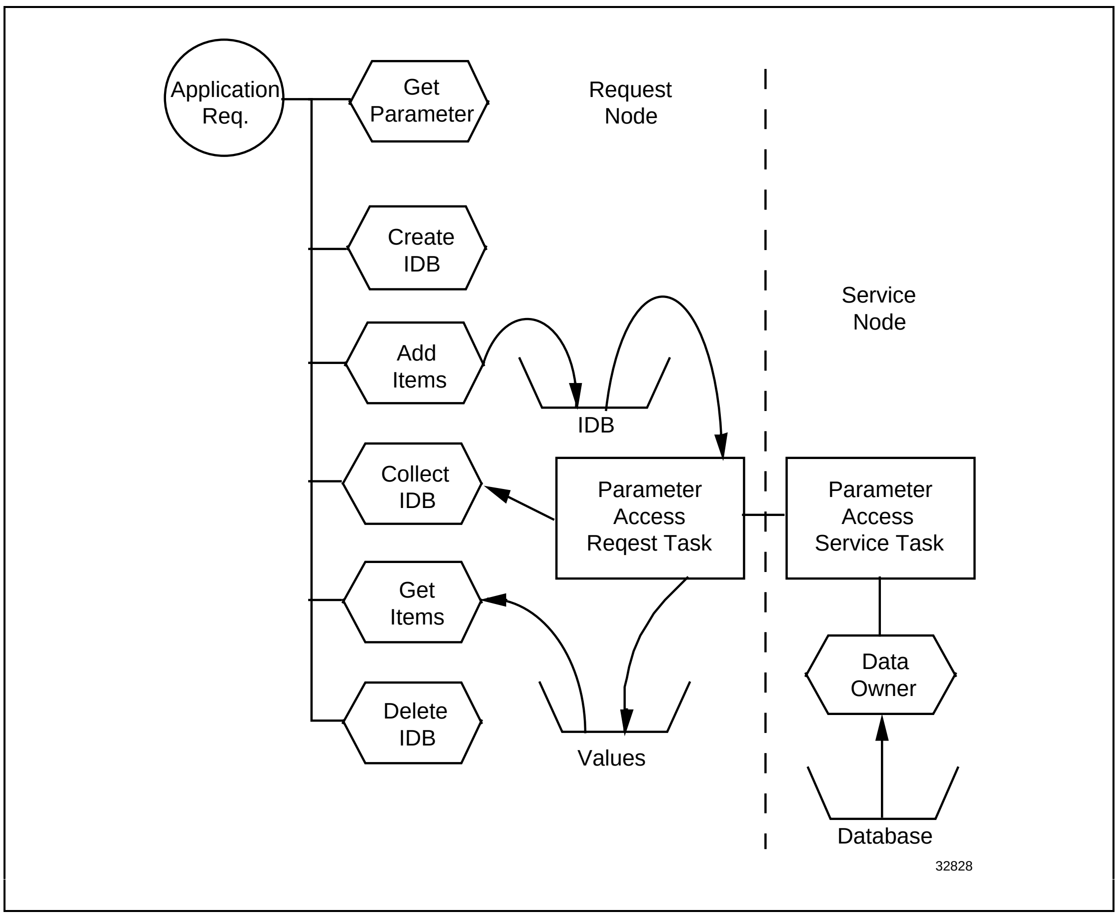


Figure 11 - Parameter Access

Name Conversion

Name conversion converts an external name, such as an entity name (for example, FIC21301), to an internal routing code form called the internal identifier. Specifically with entity name conversion, the name conversion element of Data Access broadcasts the name on the network to all the other nodes, getting them to reply with either the ID or "can't find entity" response. Name conversion also checks for duplicate tagnames and advises the requestor if it cannot search all of the nodes.

Entity name conversion always searches the local name table within the initiating node first. Most reserved point names are cached locally as a performance enhancement.

For nodes that must initiate events, a local access mechanism to the name table is provided.

Names converted by Data Access include:

- entity names
- parameter names
- enumeration names
- parameter list names

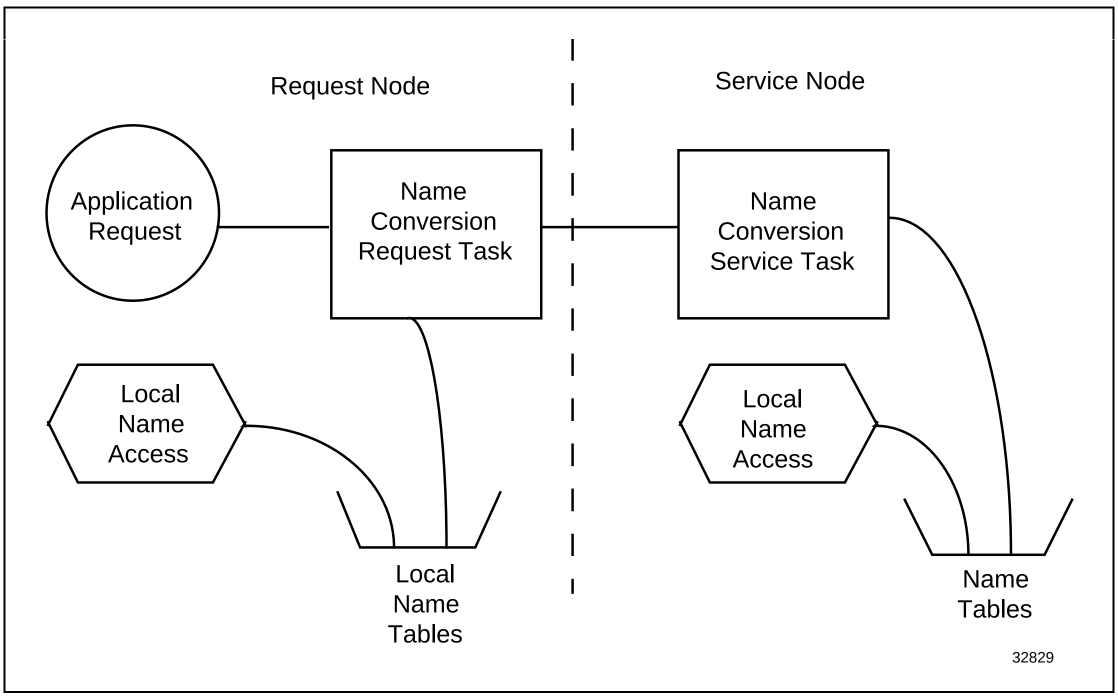


Figure 12 - Entity Name Conversion

The internal form of the entity uses 8 bytes of storage to describe an entity with a unique number. The fields of the internal entity are shown in following table.

Table 1 Internal Identifier Fields

Internal Form Field	Parameter
Entity type	ENT_TYPE
Unit	UNIT or UNITNAME
LCN	CRB (0 = local LCN)
Function set	FUNC_SET
Data realm	DATA_RLM
Revision	SERIAL
Local routing code (2 bytes)	LRC

Note that the LCN-function set-data portion of the entity ID is used as a communication address to route Data Access requests around the LCN. Once the request is in the right node, the local routing code gets it to the right entity.

Whenever an entity is deleted, the storage location that it occupied in the database is cleared out by placing blanks in the name and zeros in every field of the internal ID except the revision. The revision field is incremented by one so that the next entity that occupies that storage location will not get the same internal entity ID even if it uses the same entity type and unit, etc.

Entity Internal Identifiers

Entity	assoc	network routing				local routing	
type	unit	LCN	FS	DR	Rev	base no	sub

Components:

Entity type — An internal key created by combining the user-build type and owning node type.

assoc unit — 1..101 (system point)

network routing

LCN = 0..32 (control room bus, 0=local bus)
FS = 1..63 (owning function set)
DR = 0..n (0=local node, n=depends on funct_set type)

Rev — 0..255 (revision, changed in the data access table whenever the point is deleted)

local routing

base no = Locator of data in data realm data access defined for process inputs.

sub = Subscript, optional field based on entity type, augments base no. in locating data.

Queries

There are three types of queries supported by the Local Control Network:

History queries These queries return time slices of history for a point**.**parameter or a list of point**.**parameters; used for trends, logs, and archiving data.

Parameter queries This query function returns the data form of the parameters on a particular point. Control Language (CL) programs and schematics use the query upon compile.

Point queries This query function provides lists of entities and their parameters on request. This query can get a list per node, per unit, or per process connected box. It can also select a subset of entities of a unit based on the value or the existence of some particular parameter. It uses the tables provided by the name conversion to get the entity names and uses parameter access to get the other associated parameters it must check.

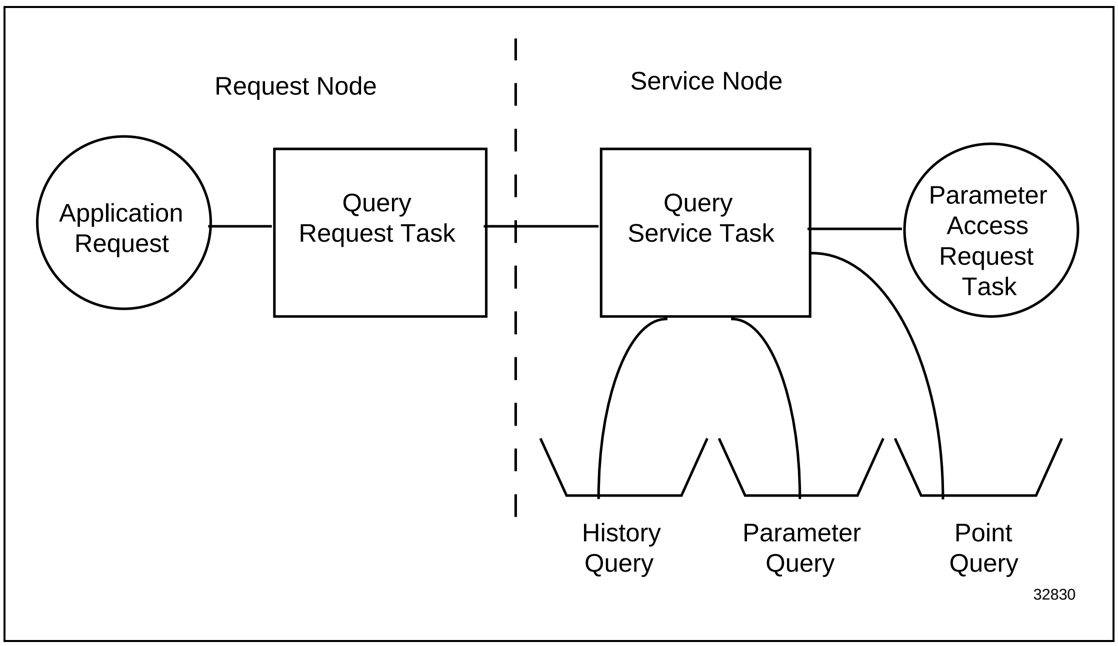


Figure 13 - Queries

ON-PROCESS ANALYSIS

The On-Process Analysis (OPA) function in the HM receives fault reports from the system error handler or on-process test in the LCN nodes through the event distribution mechanism. It analyzes the error and decides whether to make a maintenance recommendation based on the number of similar errors over a given time period.

If maintenance recommendations should be made, OPA sends an event to the Real Time Journal on the Universal Station. This is also a function of event distribution.

Maintenance recommendations are made based on replacing the Optimal Replacement Unit (ORU), which is a board or peripheral device.

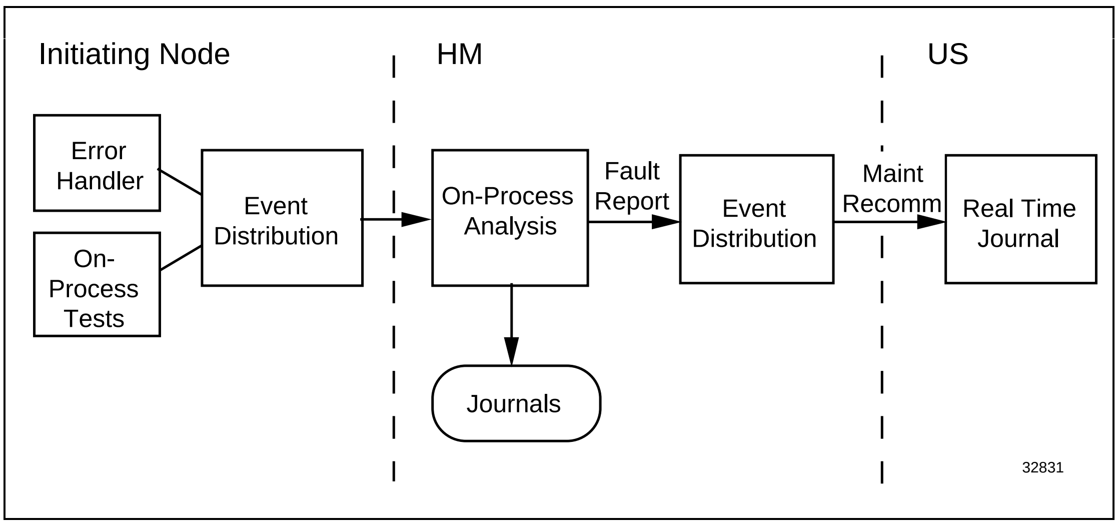


Figure 14 - On-Process Analysis

QUALITY AND LOGIC TEST

The Quality and Logic Test (QLT) is loaded by the firmware loader upon successful completion of the firmware self-test. The firmware self-test is performed upon node power up. The QLT software resides in a single load image that is independent of the node type. The QLT is initiated by a node load of a node in the “power on” state.

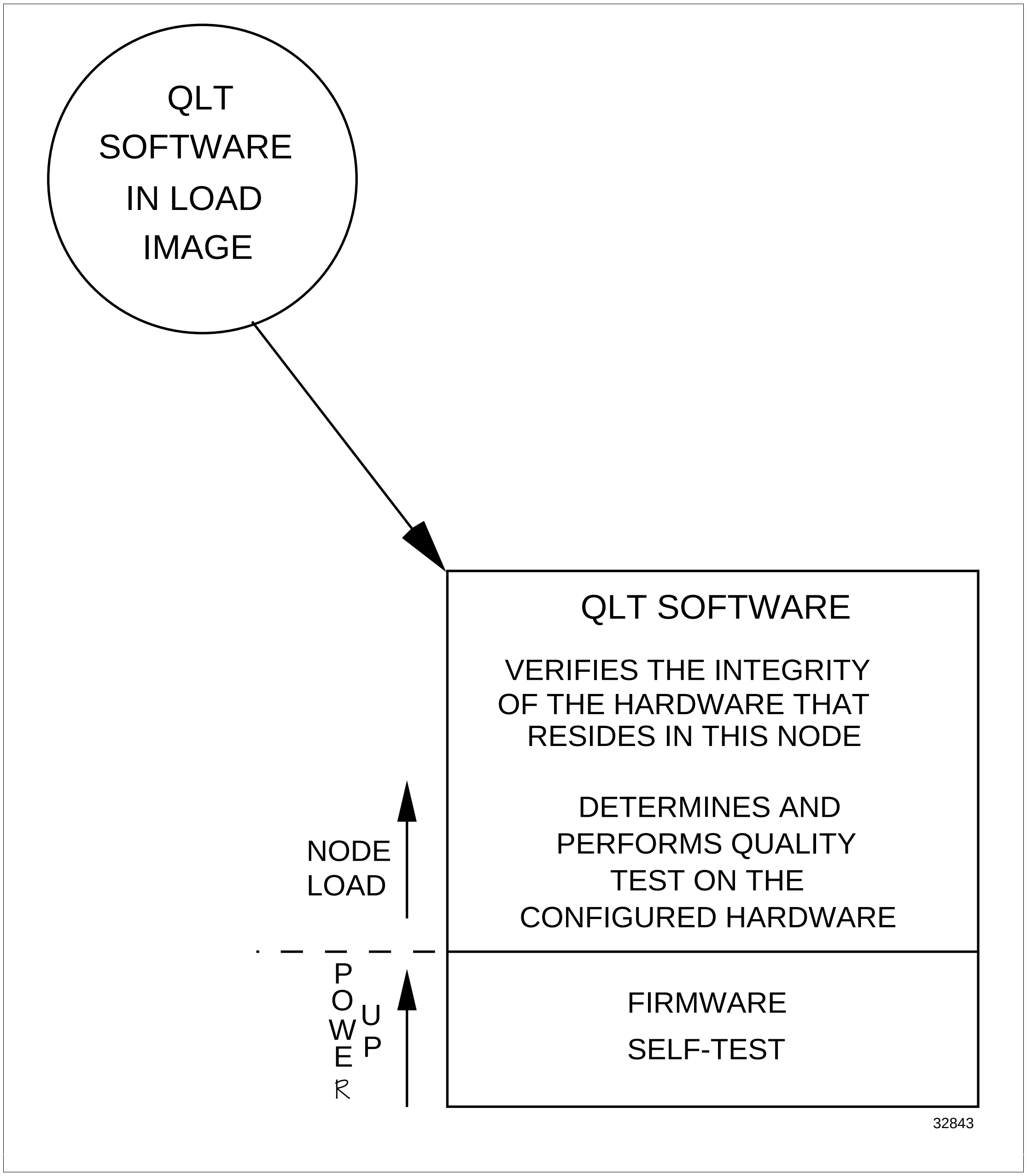


Figure 15 – Quality and Logic Test

The QLT

- Verifies the integrity of the hardware that resides in a given node. This is accomplished by loading and executing the QLT during the startup of any node, to qualify the node for a personality load.

- Determines the hardware configuration of the node and performs a quality test on the configured hardware.

- Completes the Module Configuration Table to include the firmware revision and the node's hardware operational status.

There are certain critical tests that must pass the QLT or the loading of the node will be halted.

If a node has been shut down by a command given from another node, it will remain in the "qualified" state. The QLT will run only on nodes that have been reset or on which power has been cycled. Whether or not the QLT runs again on a "qualified" node upon subsequent node load depends on the type of load initiated. If you load a "qualified" node from another node, this will not reset the node; consequently, the QLTs will not be rerun. If you load a "qualified" US using its LOAD button, this forces a reset of the node and in this case, the QLT runs.

ON-PROCESS TESTS

On-Process Tests (OPTs) are resident in each on-process physical node of a TPS Network, and perform the following functions:

- Background testing of the LCNI board
- Periodic scrubbing of memory to eliminate soft errors
- Load image integrity check
- Memory and hard drive EDAC test

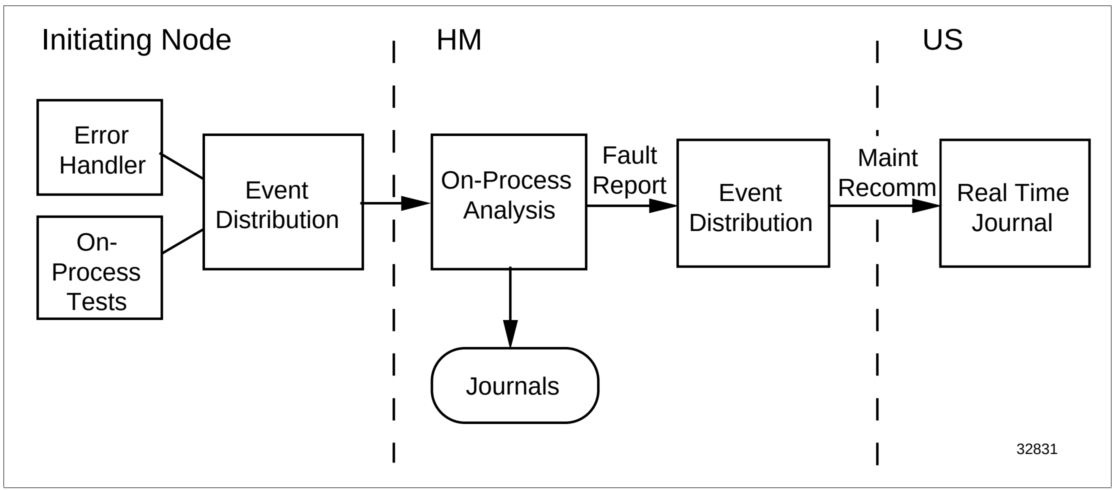


Figure 16 – On-Process Tests

The OPT reports all detected errors to the System Error Handler, which reports the errors to the System Error Journal and the On-Process Analysis software for subsequent analysis.

LCNI INTEGRITY TEST

This test verifies the ability of the LCNI in each node of the system to detect and report message frames containing an incorrect Frame Check Sequence:

- Each physical node sends a test frame to the OPT function in all other nodes.

  The test frame contains a software-generated incorrect Cyclic Redundancy Check (CRC).

- Each node checks the test frame for correctness and responds with a positive or negative acknowledgement to the OPT from which it received the test frame.

- If a node acknowledges the test frame, it indicates that the receiving node cannot detect CRC errors, and this condition is reported.

MEMORY SCRUB

In large memory arrays such as those used in the TPS Network network boards, changes can occur in the contents of individual bit cells. The memory scrub portion of the OPT uses Error Detection and Correction (EDAC) to correct words that contain single-bit errors.

LOAD IMAGE INTEGRITY CHECK

This test runs every 5 seconds to determine if a problem exists with the fixed memory that might be caused by a software or hardware error.

MEMORY AND HARD DRIVE EDAC TEST

The memory EDAC test is run every 5 minutes on the memory boards to ensure that the boards can detect and correct single-bit errors. The hard drive test runs every 30 seconds and reads one test sector that has a bad EDAC and verifies that the controller detects and correctly reports the error.

BASE APPLICATION SOFTWARE

Each node of the TPS Network is loaded with a base application software. The components of the software loaded to each of the nodes depend on the function of the node being loaded. The following sections list the software loaded to each of the nodes on the system:

Application Module

Prefetch/poststore

Every 0.5 second, the stores for the previous cycle and the fetches for the next cycle are initiated through data access.

Half-second cycle

Processing is done on a 0.5-second cycle. All points are scheduled into some 0.5-second cycle of the minute.

Fast/Slow/Background/ IPP Processors

The AM runs four processors:

- Fast Processor—for more critical points,
- Slow Processor—for slower cycle points,
- Background Processor—for asynchronous functions,
- Internetwork Point Processor (IPP)—for cross-network functions through a Network Gateway.

Non-preemption of CL

For each processor (except the background processor), the CL routine runs as an integral part of the process and cannot be interrupted by the processor.

Unit Partitioning

The AM maintains a database for each unit. Work can be partitioned among several AMs by assigning each the responsibility for a unit or set of units.

Other Base Software

- Checkpointing by unit
- Alarm initiation by unit
- Redundancy (HMPU or K4LCN processors only)

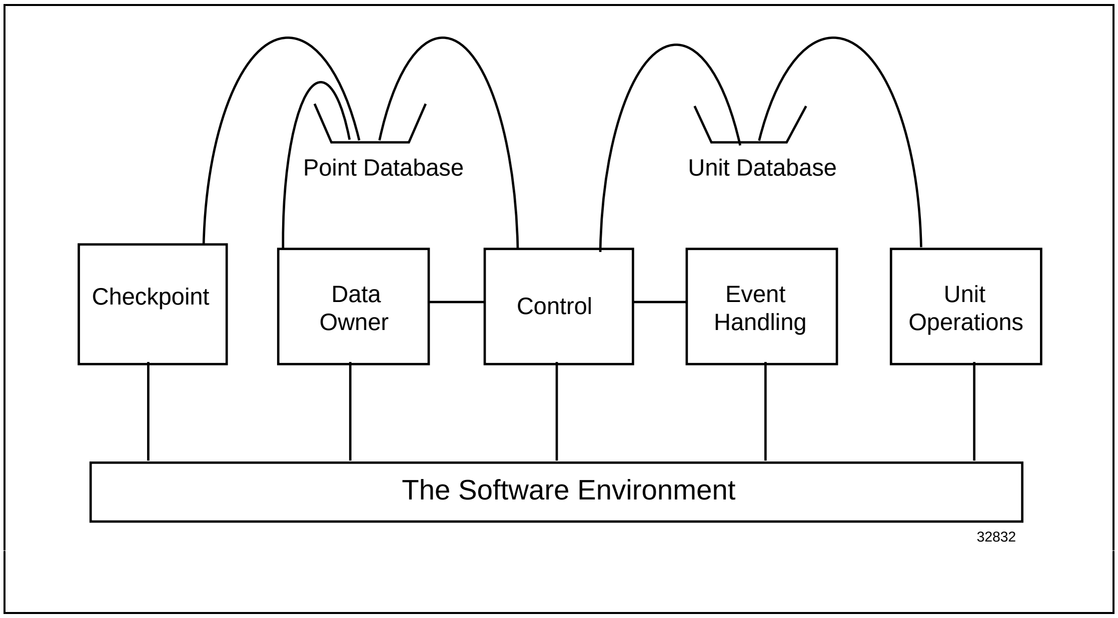


Figure 17 - Application Module

Computer Gateway

Advance Control Interface Data Points	Provides a method of initiating programs in the computer periodically, on event, or by operator demand. In addition, all stores to the LCN must be authorized by the operator enabling the point for the associated computer program. A data list can be associated with the point to allow collection of data when the computer program is initiated.
Calculated Results Data Points	Provides a storage place for computer data. Custom Data Segments (CDSs) may be defined for these points. Advanced Control Programs (ACPs) can use these points for Reads.
Operator Messages	The send message mechanism is part of the base software.
File Transfer	The ability to deal with volumes and files on the HM: catalog, read, write, and delete user files.
Other Base Software Functions	– Checkpointing – Link Configuration

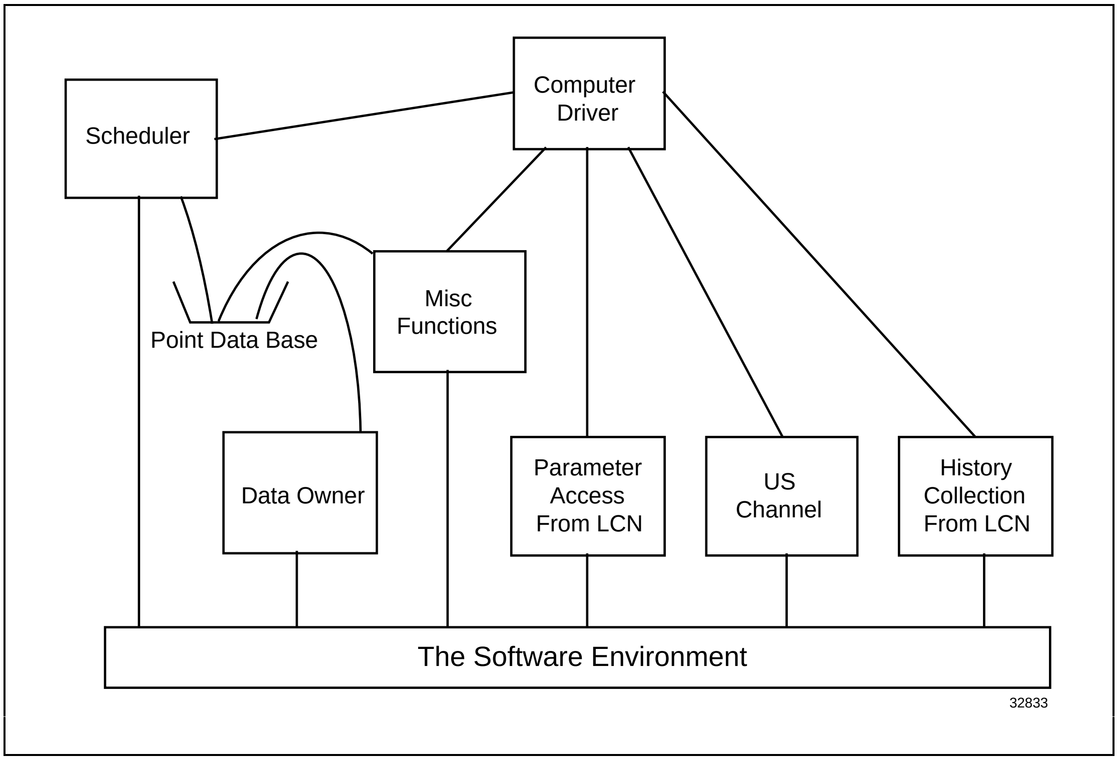


Figure 18 - Computer Gateway

Hiway Gateway

Database Split

The database is split between the controllers and the Hiway Gateway. The HG contains the extended data needed to make the points used by the LCN. The HG is considered the data owner for all data on the Hiway.

Priority Scheme

The Hiway Gateway supports the network parameter priority scheme as follows:

1. Regulatory Control requests from the operator.
2. AM/CG Control.
3. Display invocation and history collection.
4. Display and trend pen updates.
5. Engineering functions such as point building, SMCC, checkpointing, and queries.

Detect and Report Events

Event detection and reporting is done as a part of the base software.

Hiway Security

Hiway security, where by the HG personality runs hiway address diagnostics as a part of the base software.

Other Base Software Functions:

Checkpointing	Event-initiated process triggering
Redundancy	Maintains primary/secondary synchronization.
Unit Operations	Maintains unit alarm status.
Application Parameter Access	Provides data access and data owner functionality.
Trend Access	Accesses trend card history from Basic Controllers.
Event Handling	Gathers, packages, and initiates events.
Command Handling	Processes box and Hiway commands.

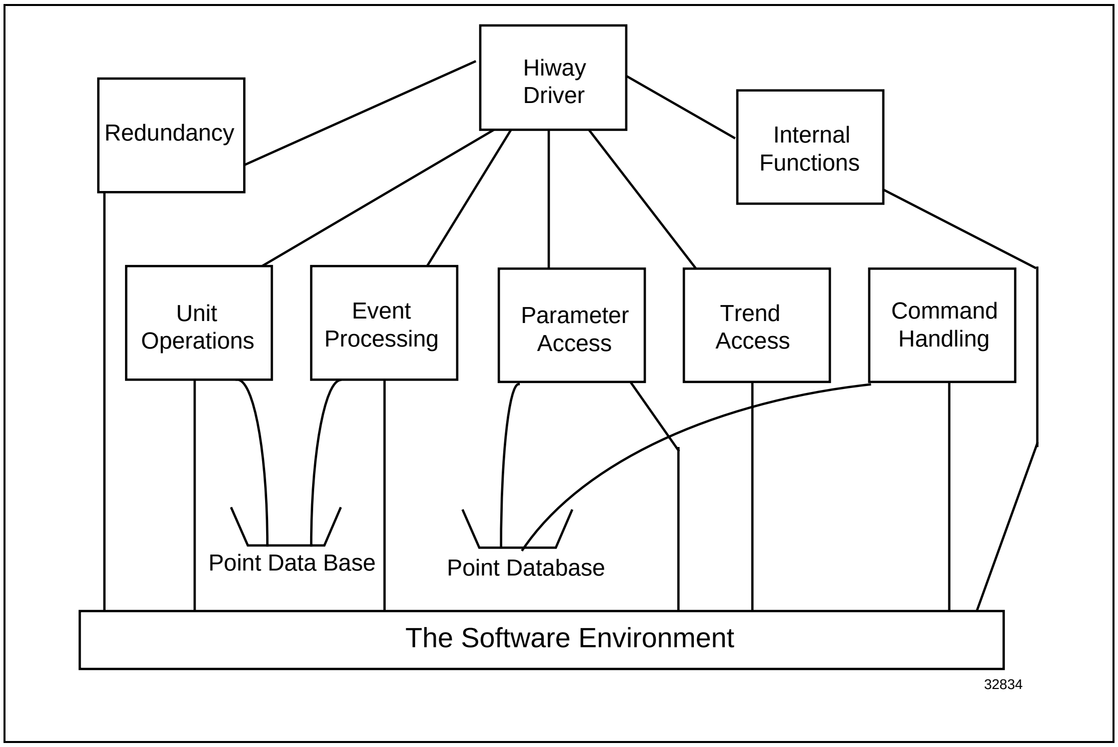


Figure 19 - Hiway Gateway

Network Interface Module

System Interface to the Universal Control Network	The UCN uses the OSI 802.4 physical layer standard and the 802.2 link layer standard. The application layer is Honeywell proprietary.
Functionality	Converts UCN format to LCN and vice versa; provides isolation between the two. Provides routing for events from the UCN and requests to the UCN. Provides an extension of the database in the controllers to fit the LCN point concept. Is the data owner representative to the LCN of all UCN data. Enforces the LCN access request priorities on the UCN (with support from the UCN boxes).
Structure	The NIM has the following software components: **Redundancy** Maintains primary/secondary synchronization. **Unit Operations** Maintains unit alarm status. **Application Parameter Access** Provides parameter access and data owner functionality. **Event Handling** Gathers, packages and initiates events. **Command Handling** Processes box commands. **UCN Security** Performs periodic UCN validation.
Base Application Software	**Message Conversion** Converts LCN messages to UCN messages and vice versa. **Scatter/Gather** Parses an LCN request to the various UCN nodes; gathers responses to report back to the LCN. **Revision Isolation** The NIM is required to tolerate more than one software revision on the UCN at a time in the event of software upgrades.
NIM priorities	1. Operator changes, event processing, regulatory control 2. Requests from AM, CG, HM 3. Display invocation 4. Display updates 5. Checkpointing, point building, engineering tasks

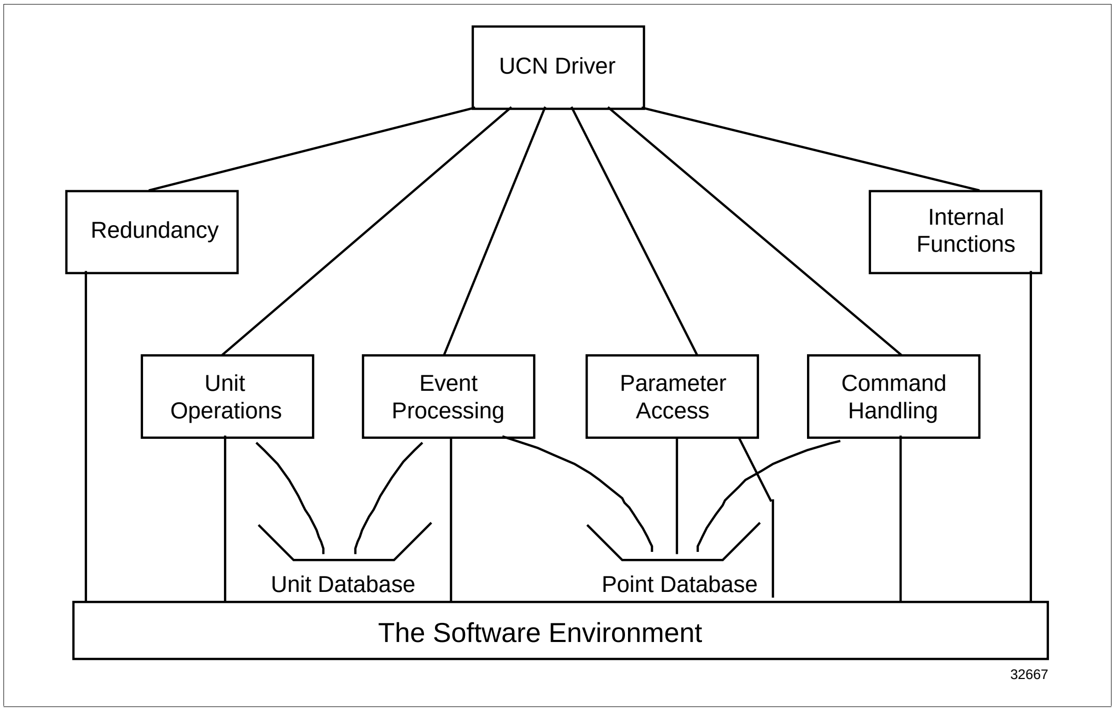


Figure 20 - Network Interface Module

History Module

History Storage and Retrieval	Process history is stored for retrieval by the operator trend functions. Event history retrieval supports filtering by event type or by point.
System File Storage	The file system consists of up to 15 volumes per physical disk. Each volume can be divided into 63 directories. Each volume can have a total of up to 9995 files. The system dictates the definition of a single volume (the local volume); the rest are user-defined.

Other Base Software:

On process Analysis	Analyzes error reports to see if a maintenance recommendation should be made.
Checkpoint control	Schedules checkpointing of the nodes assigned and of units within nodes.
Cyclic file manager	Maintains the identity of the first and last records on a cyclic file.

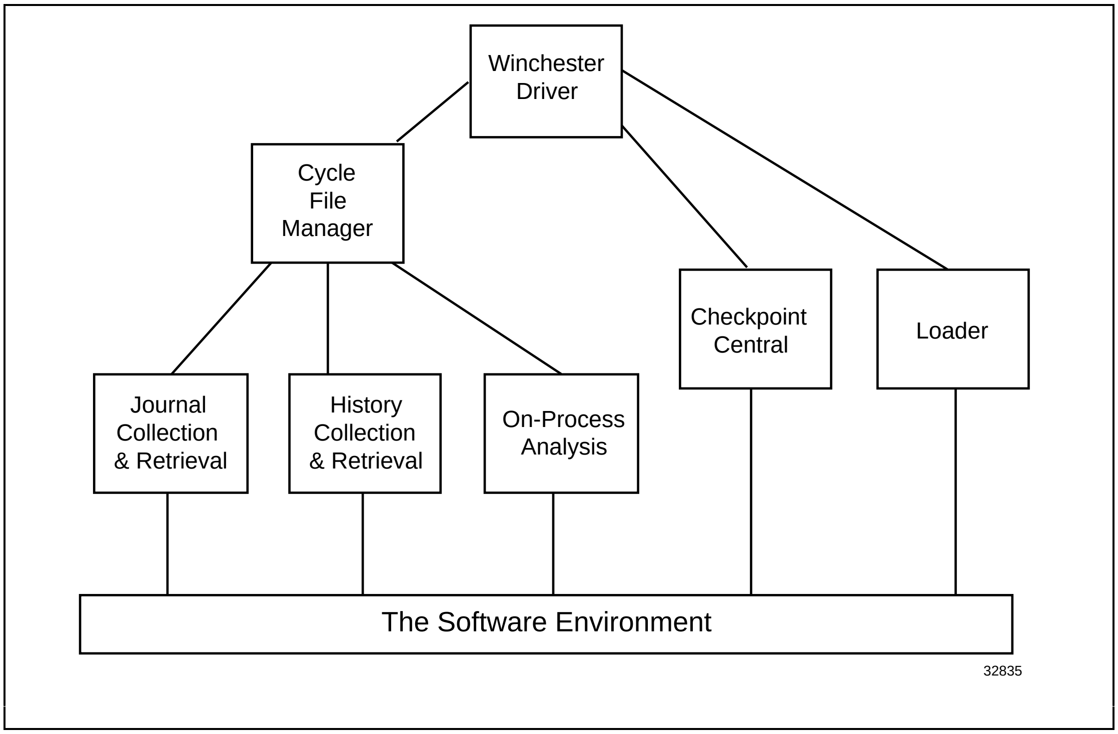


Figure 21 - History Module

Universal Station or Global User Station

Area Manager
This function coordinates the status of all Universal Stations on the same area.

Console Manager
This function coordinates the status of all Universal Stations assigned to the same console.

Kernel
This set of functions handles keyboard entry, screen output, network data access, and US scheduling activities.

Standard Displays
This set of abstracts provides the standard set of screen functions supplied with the system.

Report Manager
This function coordinates, schedules, builds, and outputs logs and reports on a console.

Real Time Journal Manager
This function processes the events received from the system, routes them to their output device, and formats them for output.

Alarm Manager
Maintains the current alarm list in the US.

Schematic Interpreter
Interprets the abstracts, builds the display, and initiates output through the screen primitives.

Screen Primitives
These modules control the drawing of the screen.

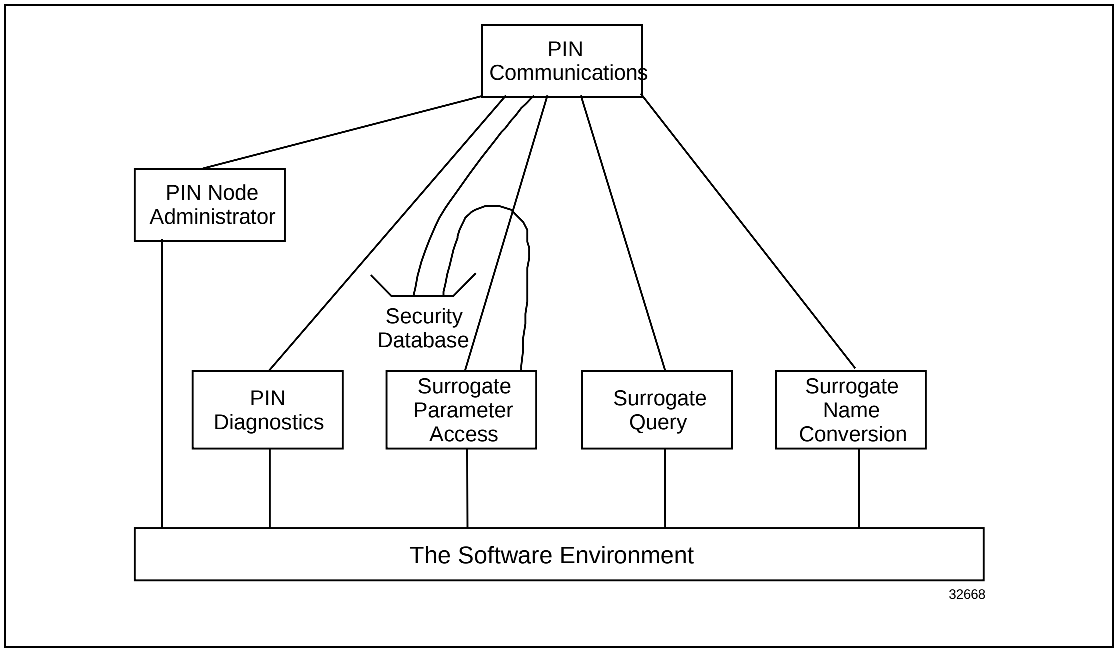


Figure 22 - Universal Station or Global User Station

Network Gateway

The NG connects multiple LCNs together over the plant information network. This is a redundant 10 Mb broadband coax. network using the OSI 802.4 physical layer and 802.2 link layer standards

Inbound Security

The local engineer defines which remote LCNs the local LCN will communicate with, which of the functions described below are allowed, and what local databases may be affected by remote LCNs.

Parameter

Parameter Access to any customer-built database on the LCN is possible, if authorized. These parameters may be displayed or used in AM control algorithms.

File Transfer

File transfer to/from volumes on the network device of the LCN is possible, if authorized. This allows copying of files between LCNs.

Structure

The NG has these software components:

PIN Node Administration
Maintains status of nodes on the Plant Information Network (PIN).

Surrogate Parameter Access
Acts as agent for remote requests.

Surrogate Query
Acts as agent for remote requests.

Surrogate Name Conversion
Acts as agent for remote requests.

Base Application Software:

Revision Isolation
Message structure conversion between revisions is handled by the NG for messages on the current revision and one revision back.

Data conversion
Enumerations across LCNs are all treated as self-defining. That is, the ASCII value as well as the enumeration is passed on any transfer.

Surrogate Requesters
The NG on the remote LCN acts as a surrogate for the requestor, taking care of the scatter/gather of data and multiple acknowledgements that may be necessary to complete the request.

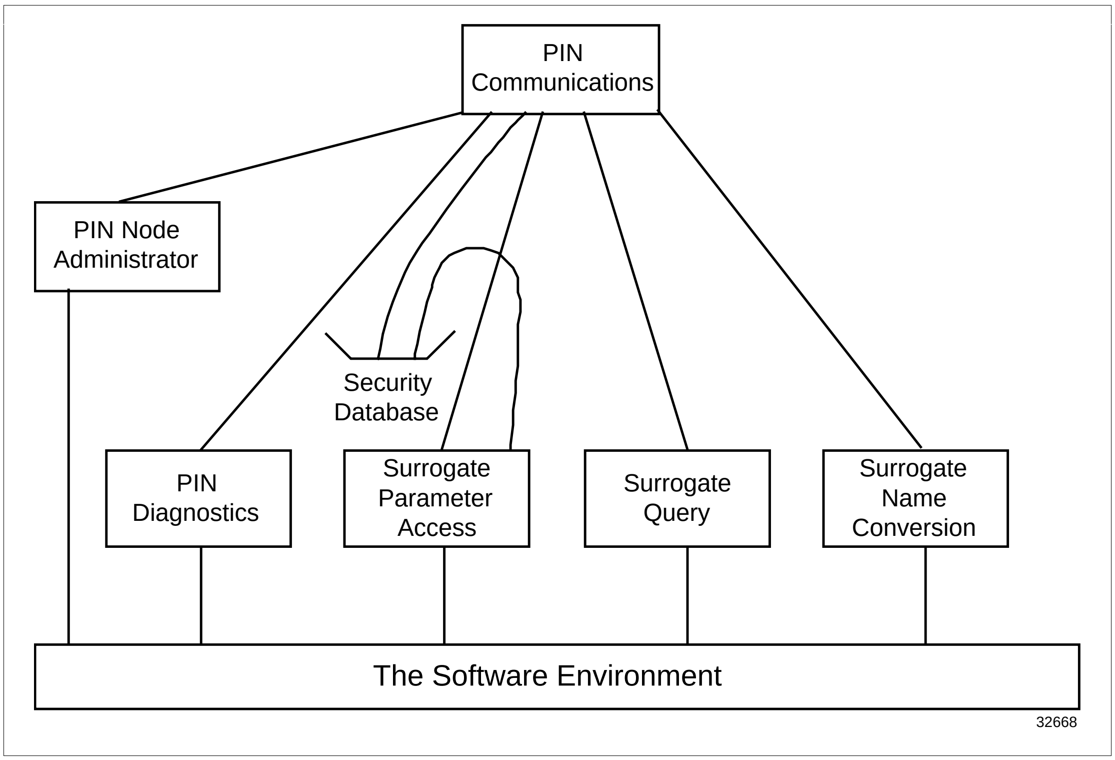


Figure 23 - Network Gateway

LAB TIME

Use your US.

Take with you:

- This module

LAB EXERCISE 1 – NODE ADMINISTRATOR

1. Power off Universal Station A and leave the power switch in the off position.

   Monitor the status of Universal Station A from Universal Station B.

   After 2 minutes, Universal Station A will be isolated by the rest of the LCN nodes because they have failed to receive its Node Administrator Broadcast messages.

   Look at the error block for any other running node on the LCN. The error entity type should indicate “Node Administrator Status Broadcast Timeout” against Universal Station A.

2. Rename the current NCF.CF file in the &ASY directory to BNCF.CF.

   Copy a different NCF into the &ASY directory.

   Shutdown and attempt to load a Universal Station.

   After the NCF revision error occurs, call up the error block or the System Error journal for that US. The error should indicate $$NODE_ADMIN CRASH.

End of Lab Exercise 1

LAB EXERCISE 2—QLT

Lab Instructions

1. Reset US:

    a. Cycle power on a Universal Station.

    b. Watch the board LEDs light as they go through their firmware self-tests.

    c. Initiate a load at the US using the LOAD button. Watch for the QLT message.

2. Shut down Universal Station A using the SHUTDOWN target from Universal Station B.

    a. Notice that Universal Station A goes to the Qualified state.

    b. Initiate a load to Universal Station A from Universal Station B.

    c. Notice that the QLT message is not displayed but that the Load in Progress message comes up immediately.

End of Lab Exercise 2

LAB EXERCISE 3—
DATA ACCESS - NAME CONVERSION

Lab Instructions

1. Build a custom AM point named IDTEST### (where ### = your partition number).

2. Create a schematic called TEST### in your student directory S###. See your course manager for the correct unit for your point. Add value IDTEST###.NAME.

3. Compile the schematic. This binds the internal entity ID to the .DO file.

4. Copy TEST###.DO to directory PICT and call up the schematic. The point name IDTEST### should be indicated.

5. Delete IDTEST###. The schematic should indicate @@@ or !!! signs.

6. Rebuild IDTEST### with the exact same configuration; however, it will have a different internal ID than the original point.

7. The schematic should indicate @@@ or !!!! signs.

8. Recompile the schematic; this reestablishes the internal connection between the schematic and the point.

9. If on Release 510 or later software, clear the screen from the System Menu.

10. Call up the schematic. The point name should once again be displayed.

11. Delete IDTEST### and delete TEST###.DO from PICT.

End of Lab Exercise 3

LAB EXERCISE 4—
DATA ACCESS – INTERNAL ENTITY IDENTIFIER

Lab Instructions

1. From schematic PERFMENU, call up the DATACHNG display.
2. Enter the following point.parameters into the DATACHNG ports:

   TIC21###.ENT_TYPE

   TIC21###.UNIT

   TIC21###.CRB

   TIC21###.FUNC_SET

   TIC21###.DATA_RLM

   TIC21###.SERIAL

   TIC21###.LRC

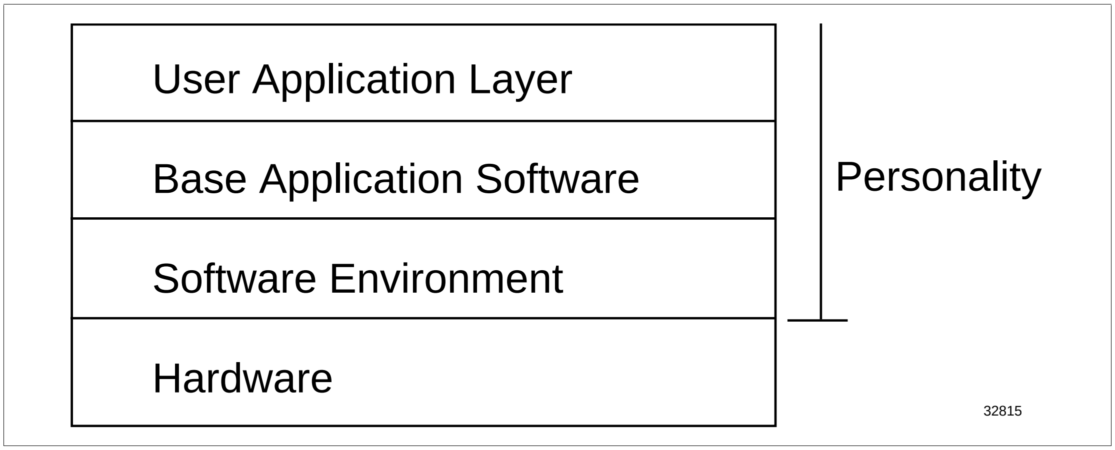
3. Compare the values of these parameters with the internal ID parameters listed in Table 1.

End of Lab Exercise 4

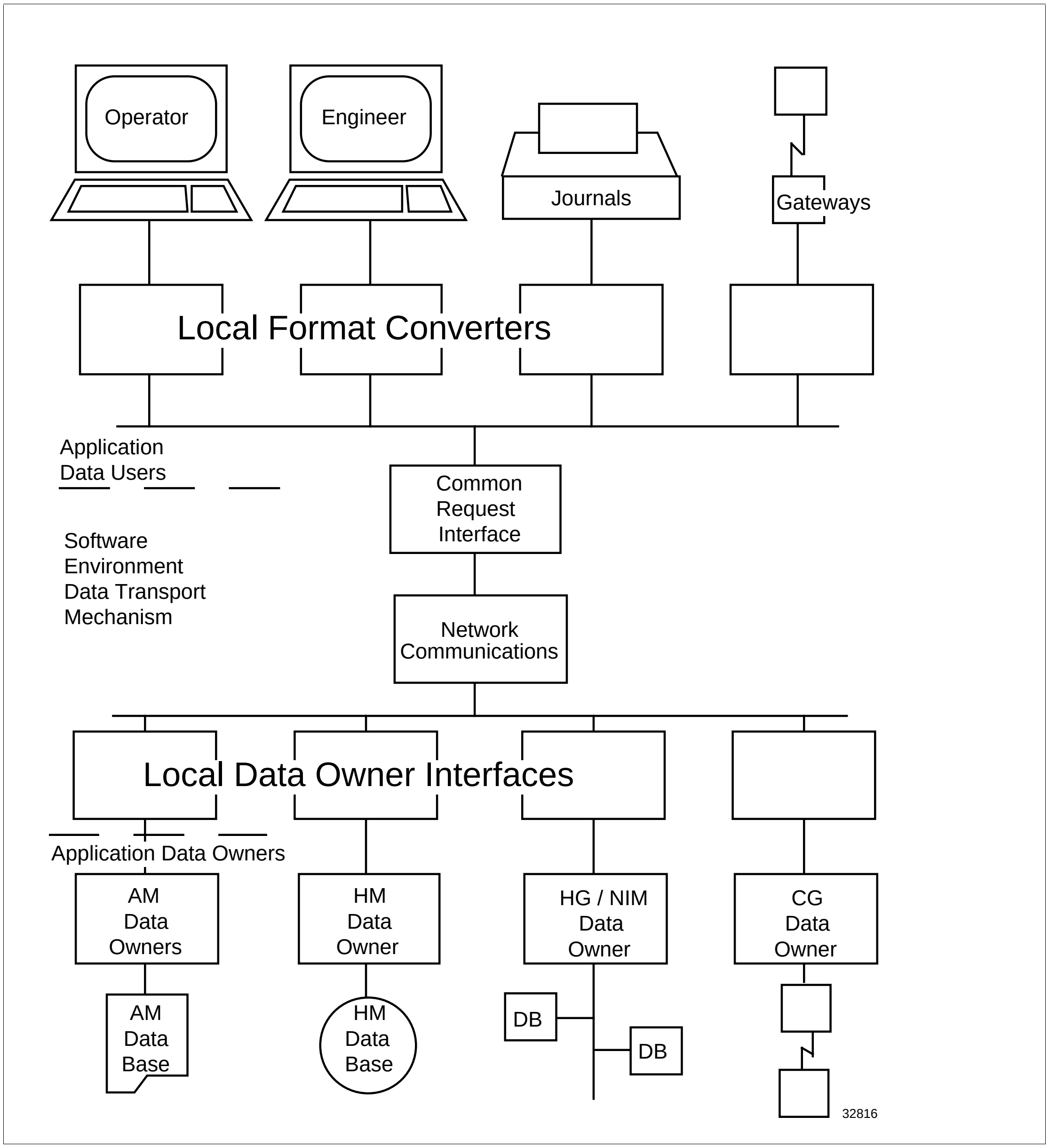
APPENDIX

Lecture Notes

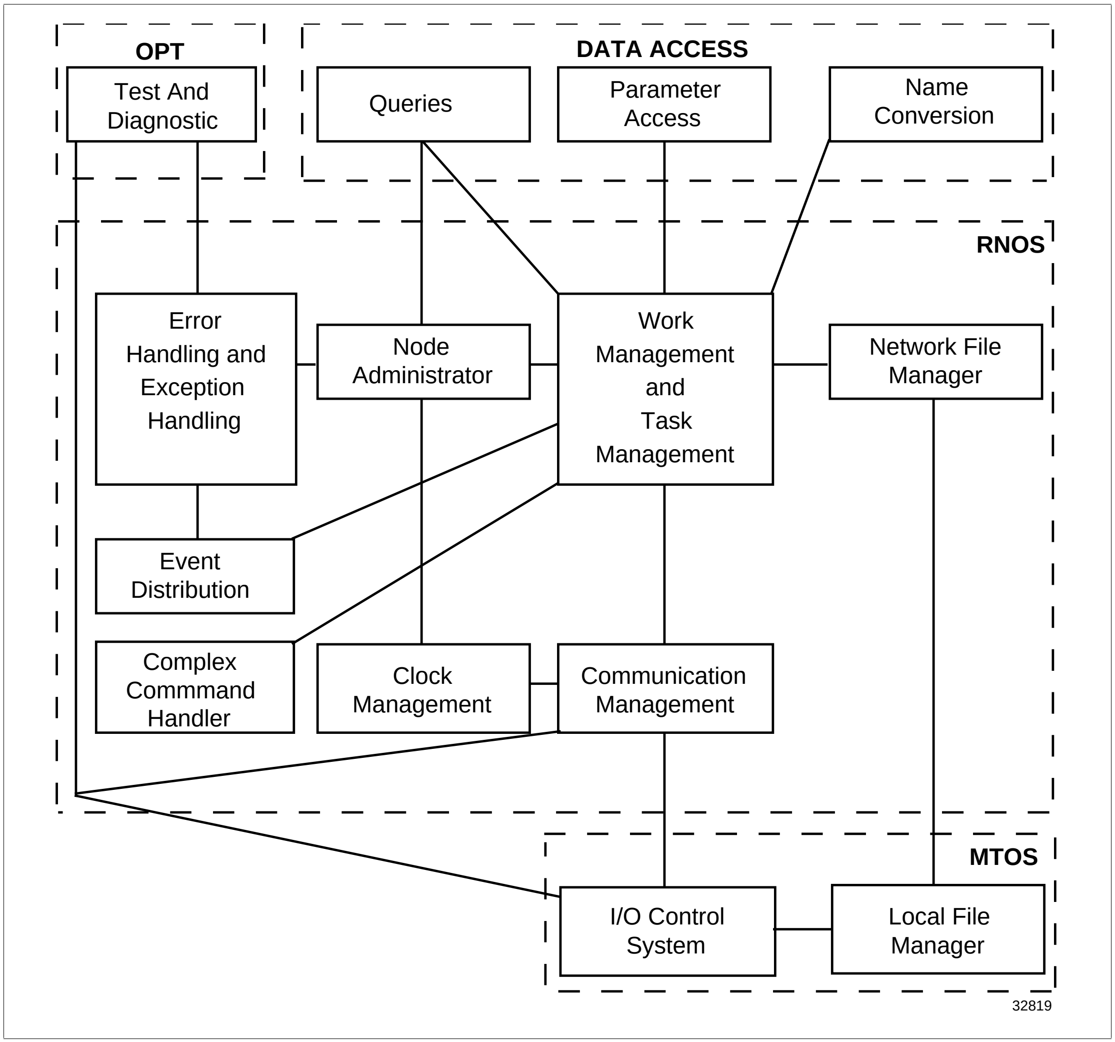
TPS Network Node Structure



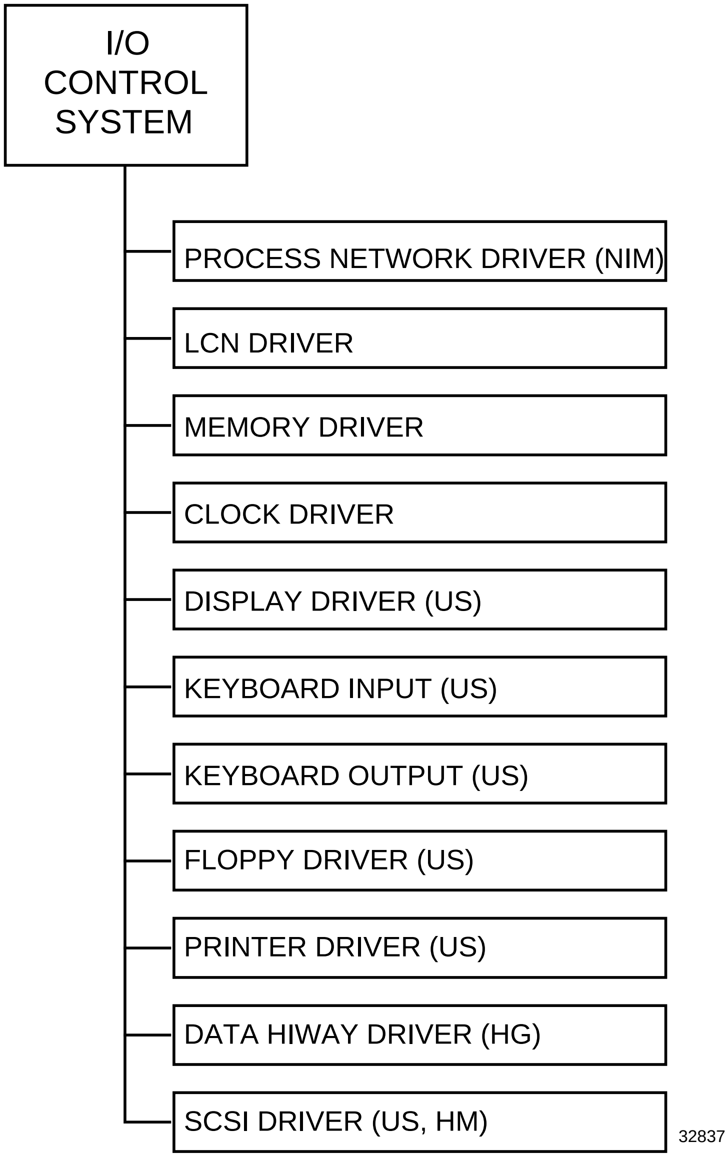
Database Management in TPS Network



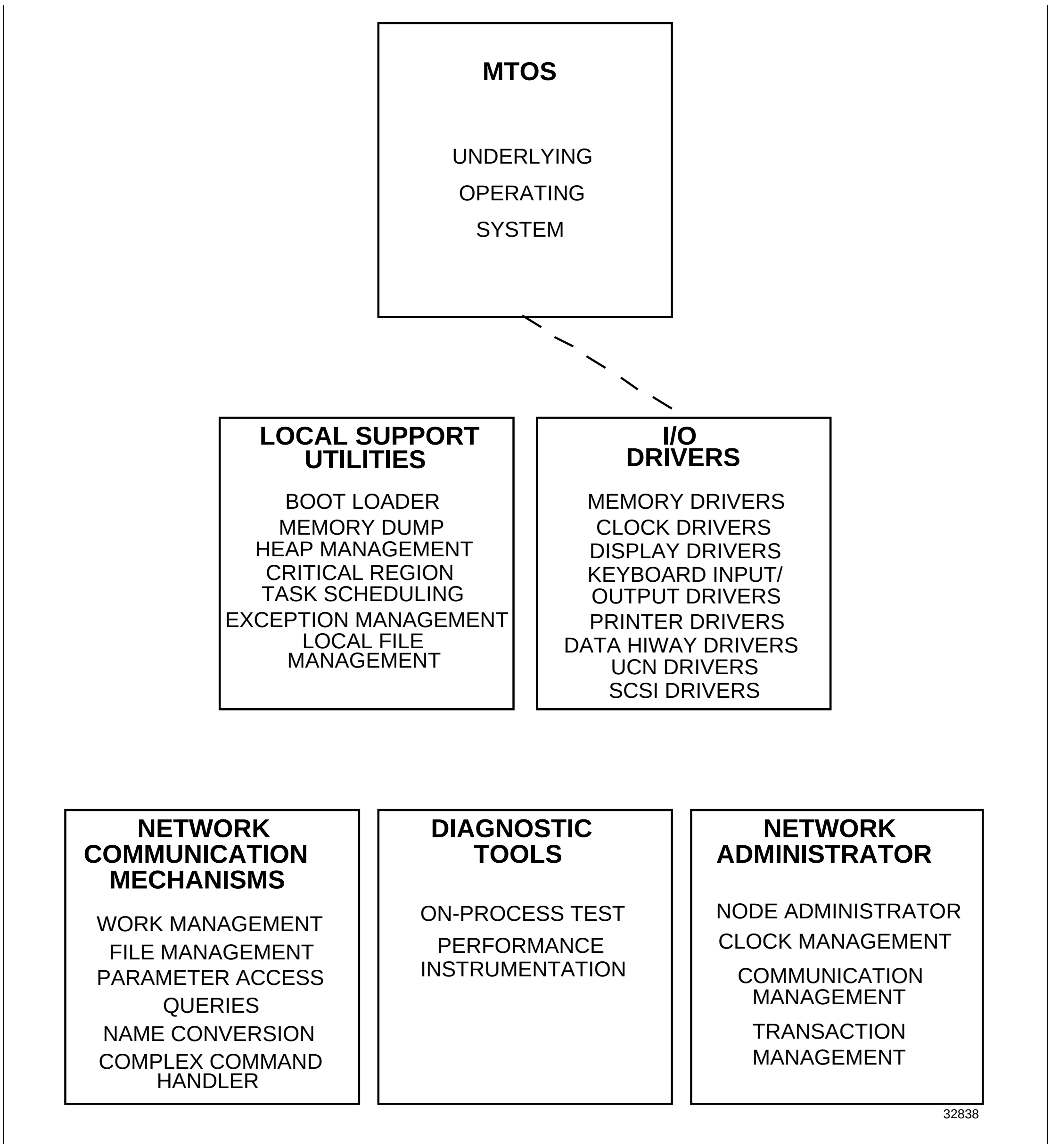
TPS Network Software Environment



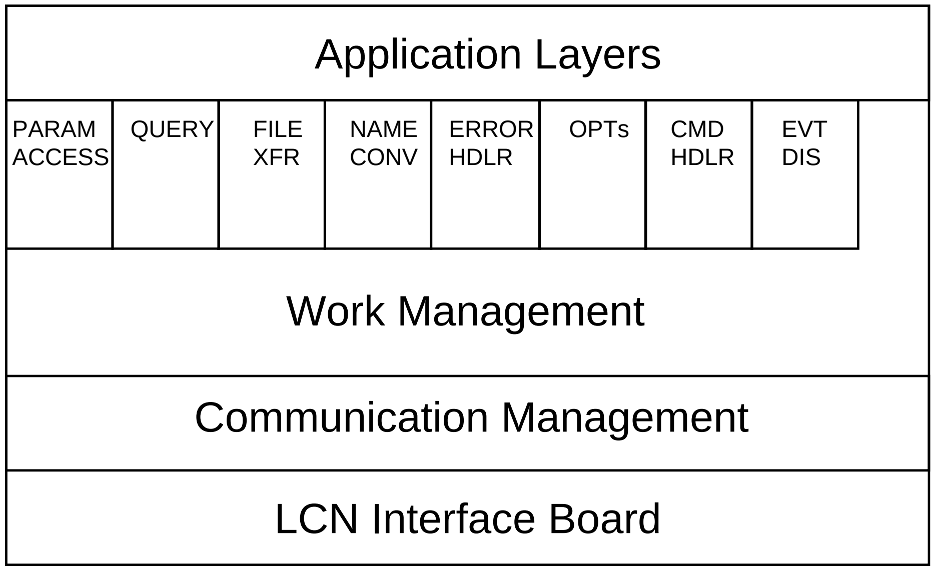
MTOS I/O Control System



Software Environment Components



Communication Layering



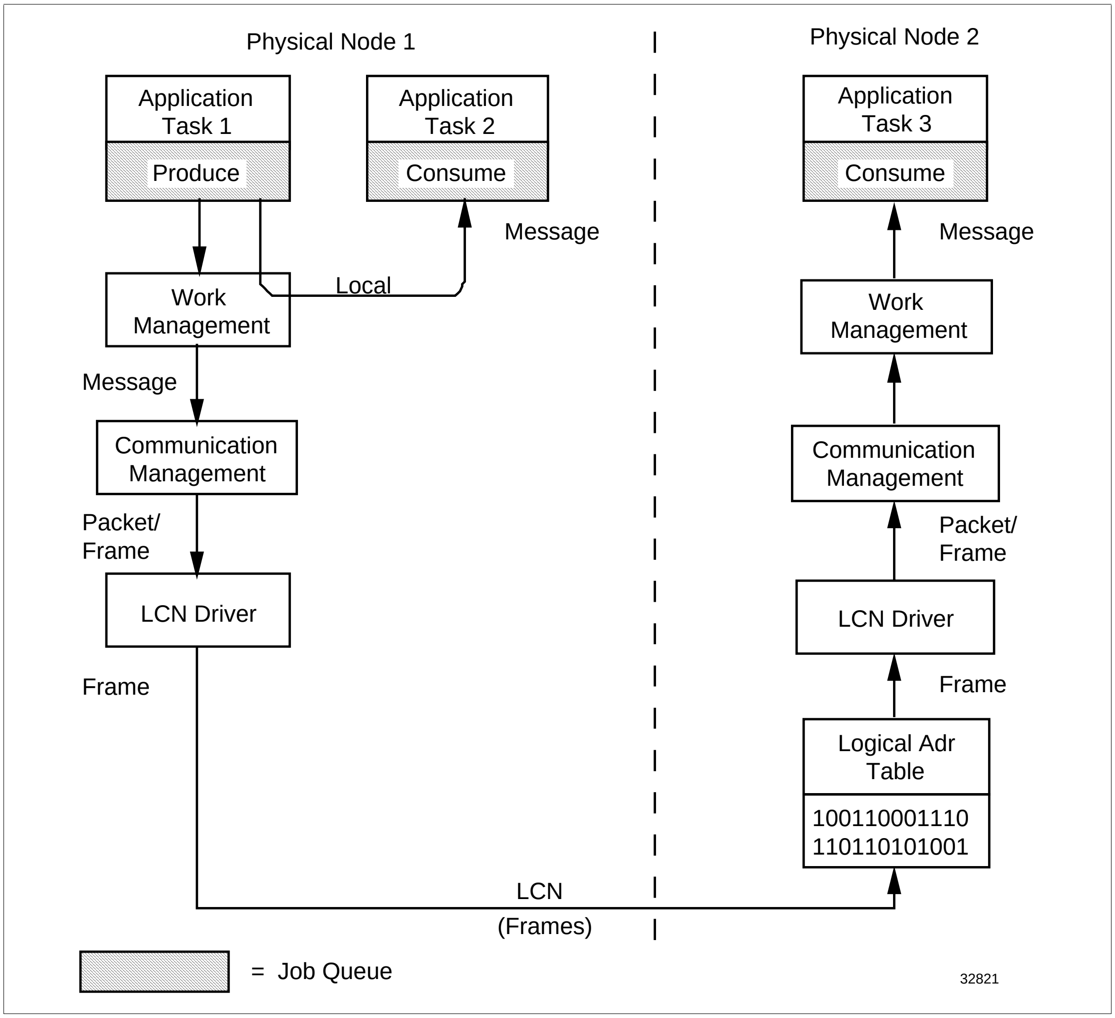
All application communication interfaces ultimately end up as Work Management operations

Applications may initiate network routing by:

- Entity (Parameter Access)
- Pathname (File Transfer)
- Logical Node (Event Distribution)
- Function (Query)

32818

Work Management

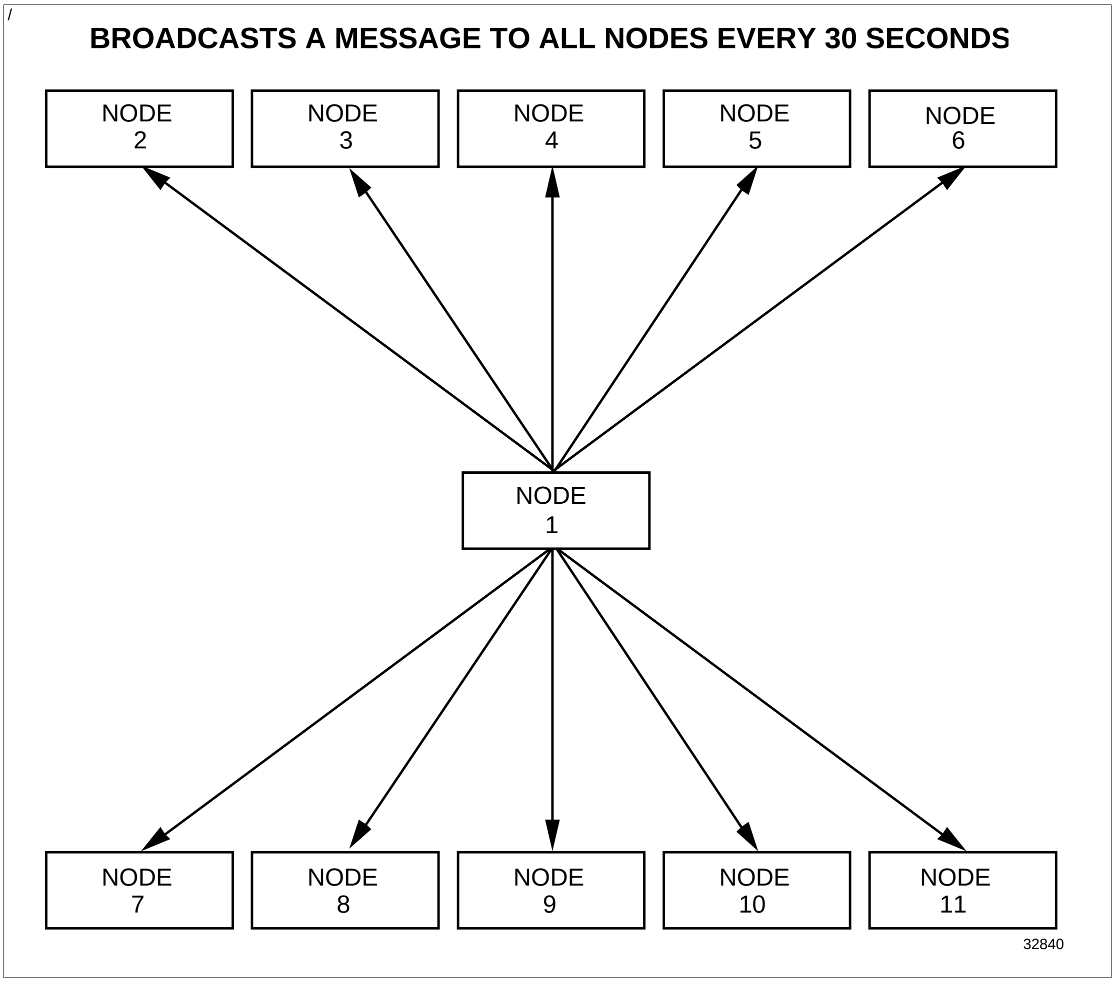


Node Administrator

Functions:

- Watchdog functions
- Function Set startup
- Notification of physical/logical node state change
- Switchover and shutdown for redundant pairs
- System time change notification
- NCF time stamp revision
- Maintains transaction list
- Maintains status of nodes on the network

Node Administrator

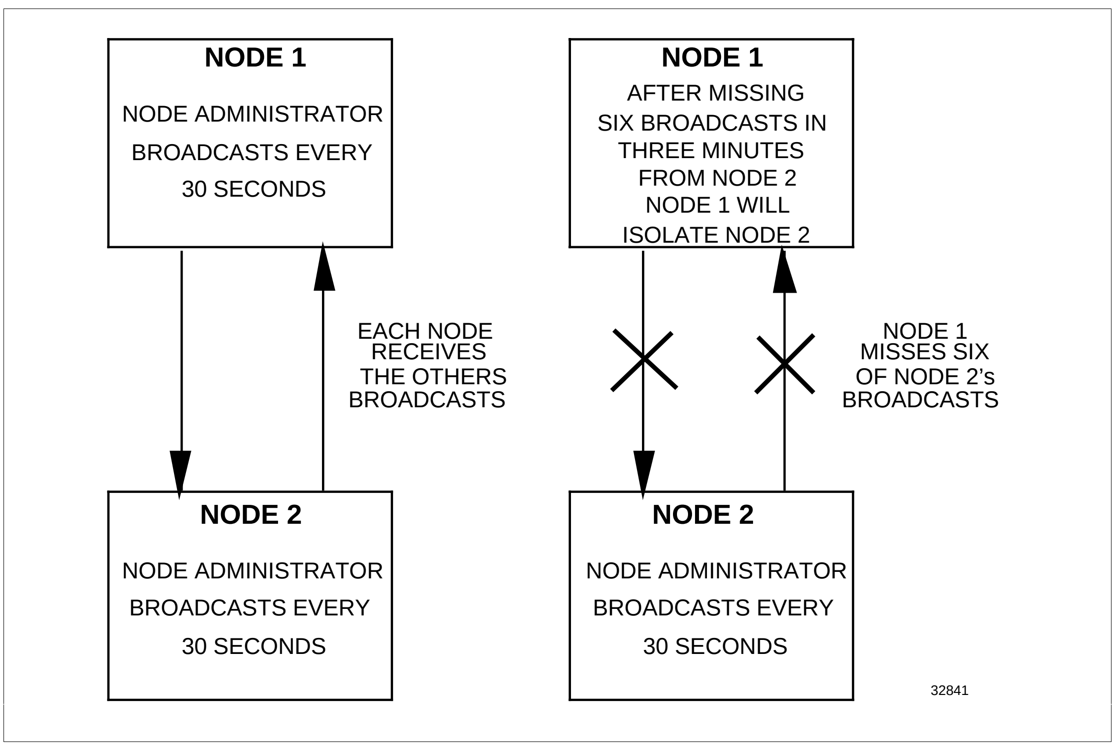


Node Administrator

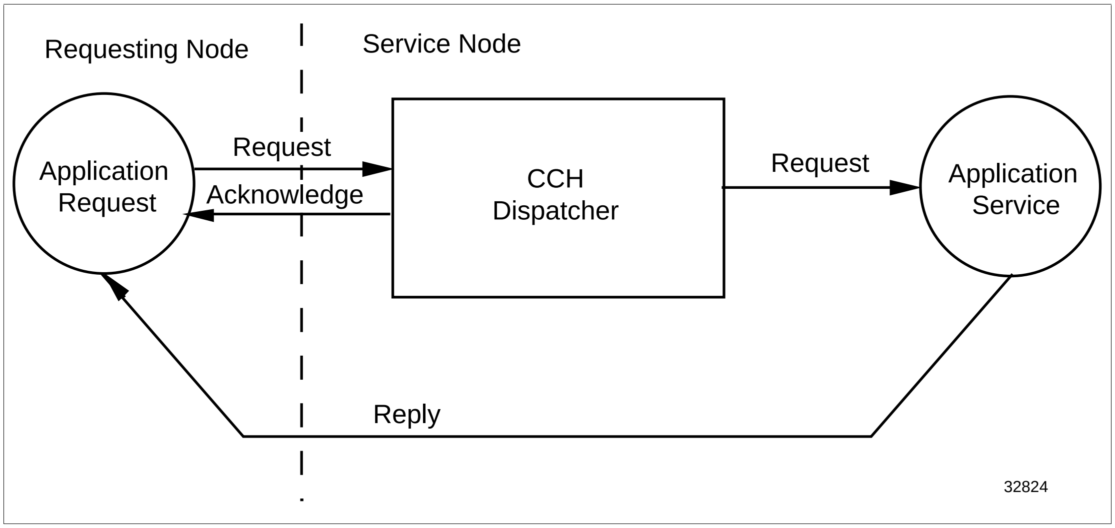
- Each node administrator expects to hear the Administrator Broadcast from each node every 30 seconds.

- If three or more nodes miss four node administrator broadcasts in two minutes from another node, they will “vote” to isolate that node.

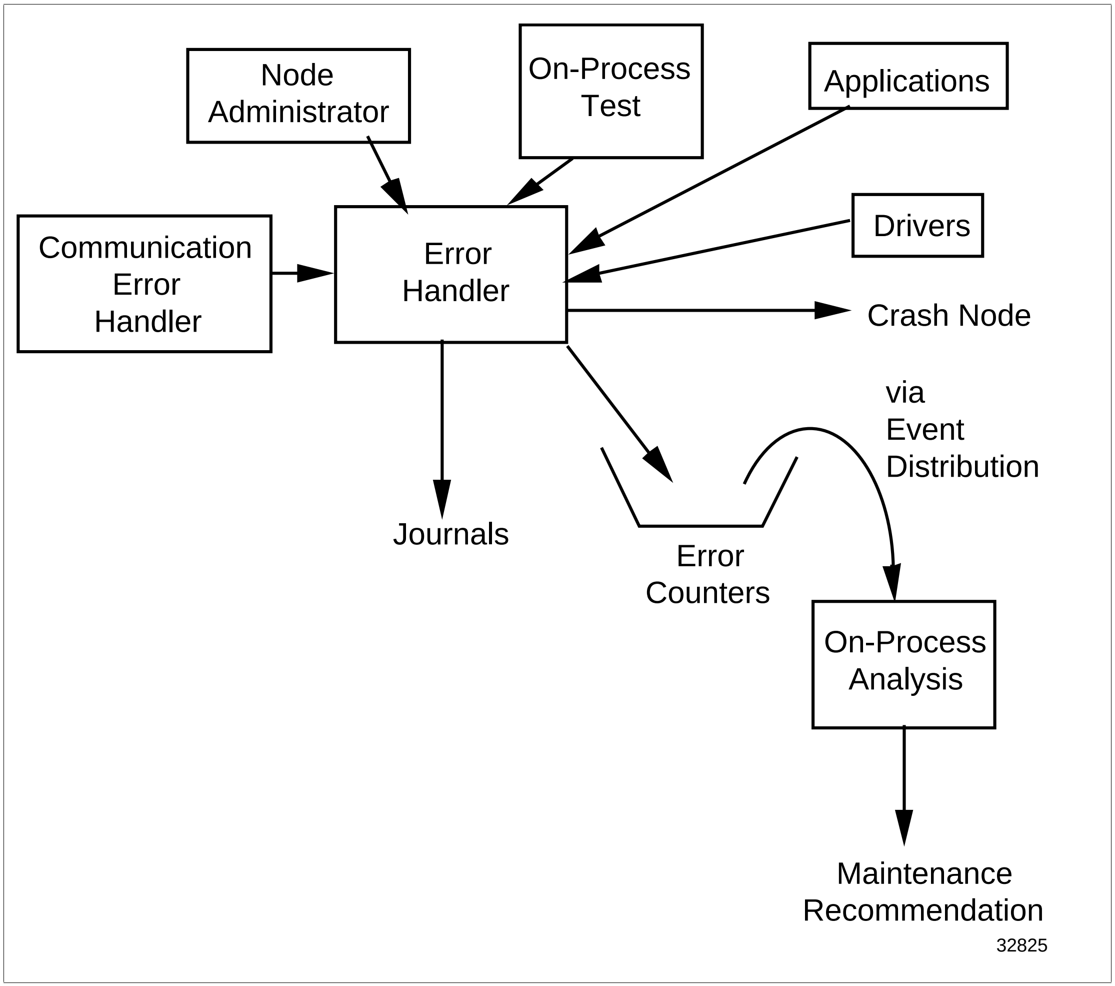
- In R320 and later, if a single node administrator misses six broadcasts in three minutes from another node, it will no longer communicate with that node and it will “isolate” it.



Complex Command Handler



Error Handling

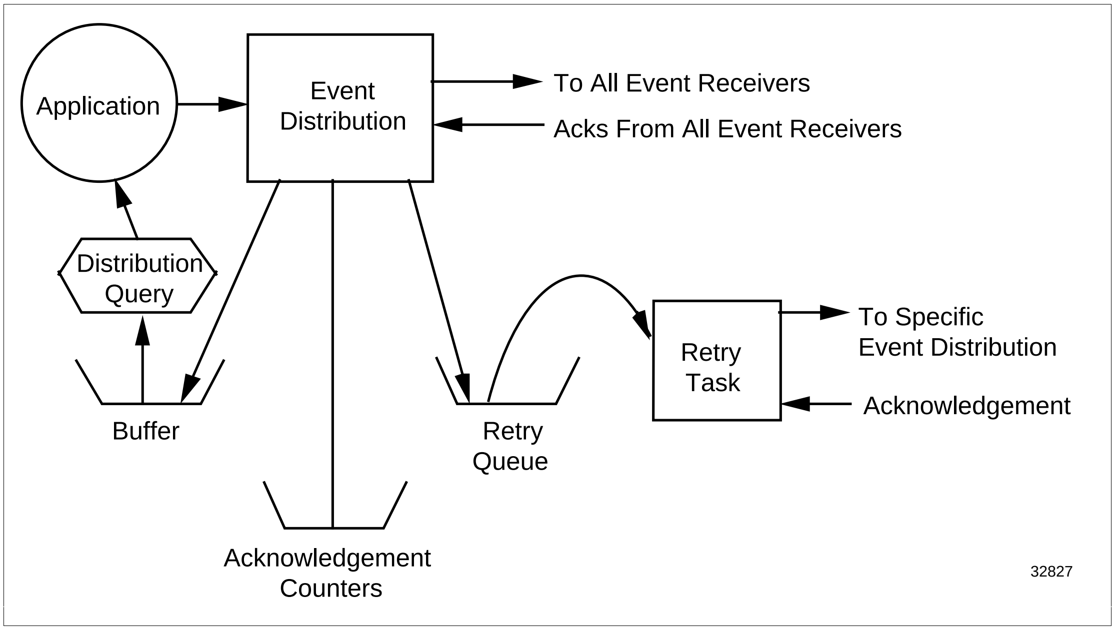


File Manager

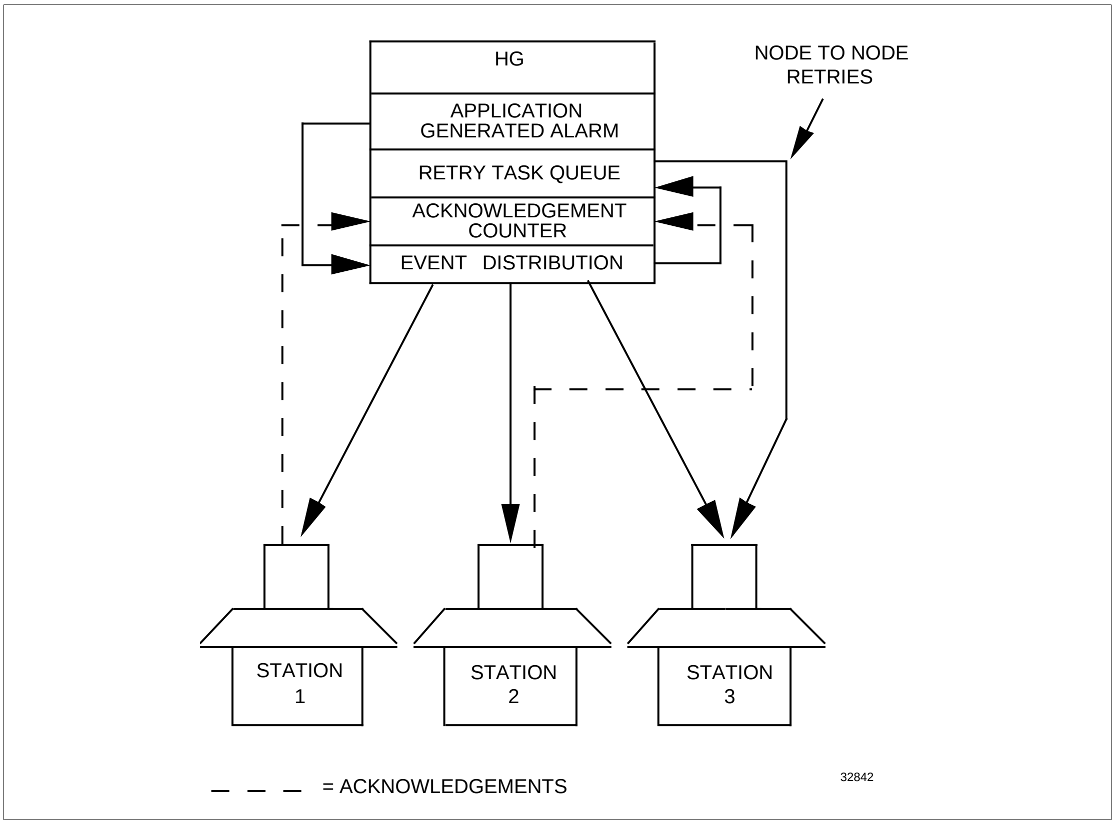
Functions:

- Basic file and directory operations
- Maintains device statuses
- Coordinates access to devices on a physical node
- Locates the file system devices/volumes across the network.
- Transports I/O requests across the network
- Handles disk redundancy on file operations
- Closes file/device upon device or node failure
- Controls output access to RTJ printing functions
- Supports access to contiguous and linked files

Event Distribution

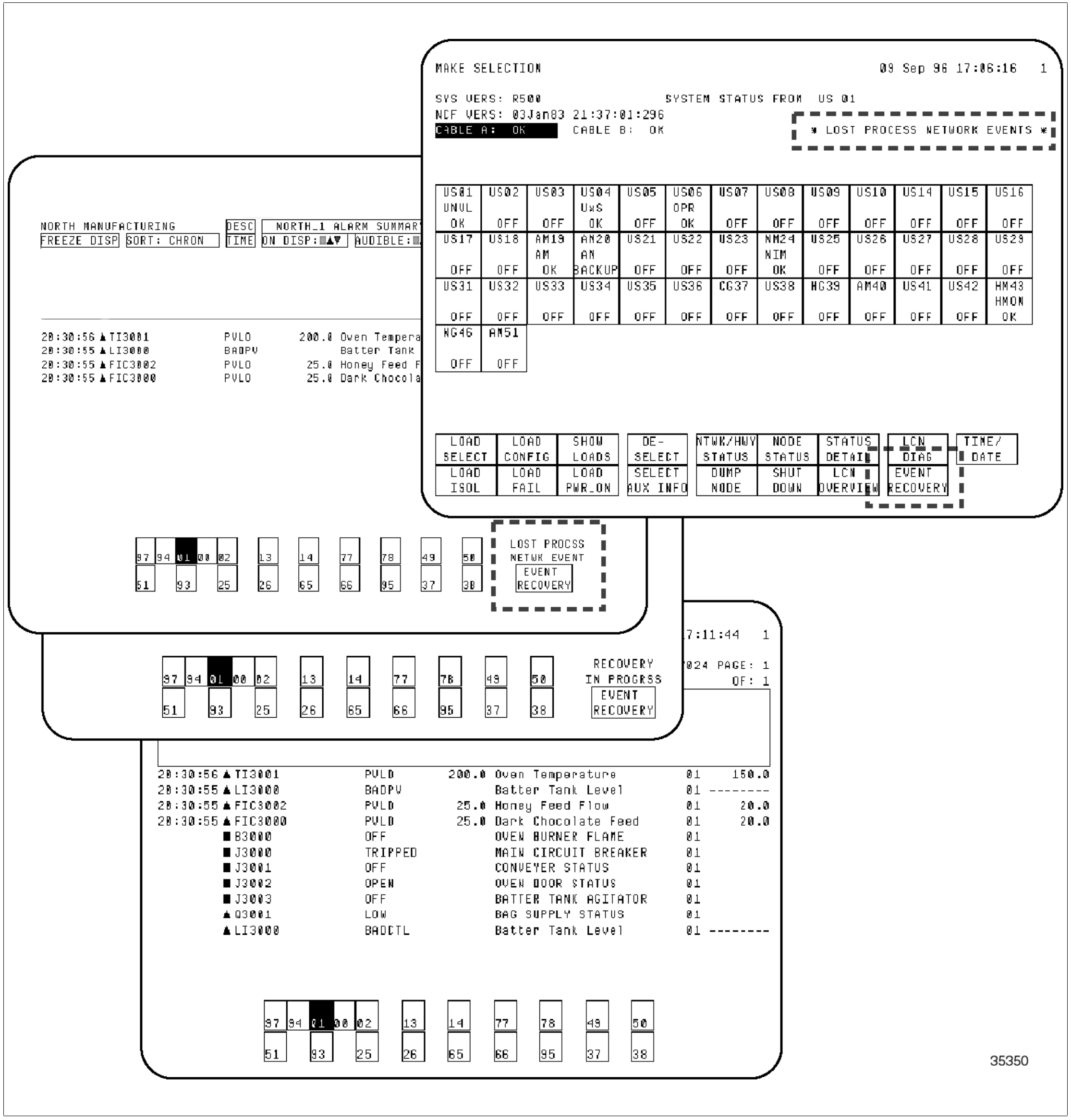


Event Distribution



- If an acknowledgement is not received from node(s) that should have received the event, up to three retries are initiated.
- The first retry is initiated after five seconds, the second retry is 10 seconds after the first, and the third retry is 15 seconds after the second retry.
- If after three retries there is still no acknowledgement, a command is given that brings up the EVENT RECOVERY target.

Event Recovery



Parameter Access

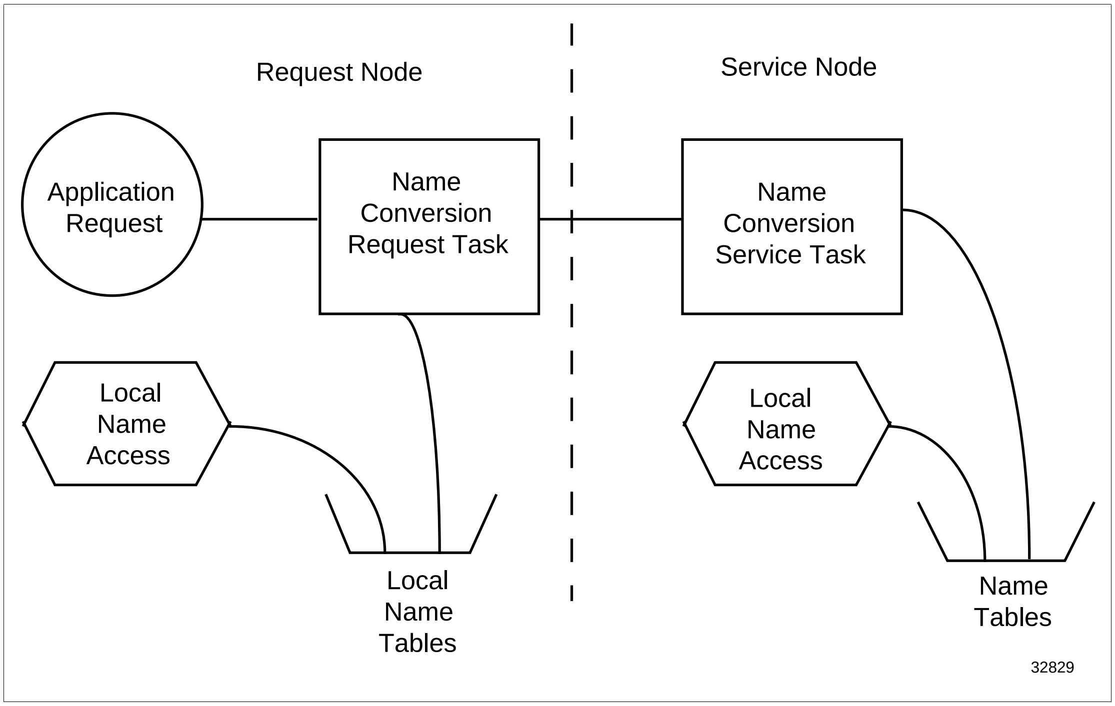
Features:

- Creates intermediate databases (IDBs)
- Deletes IDBs
- Uses Collection Sets to access parameter data

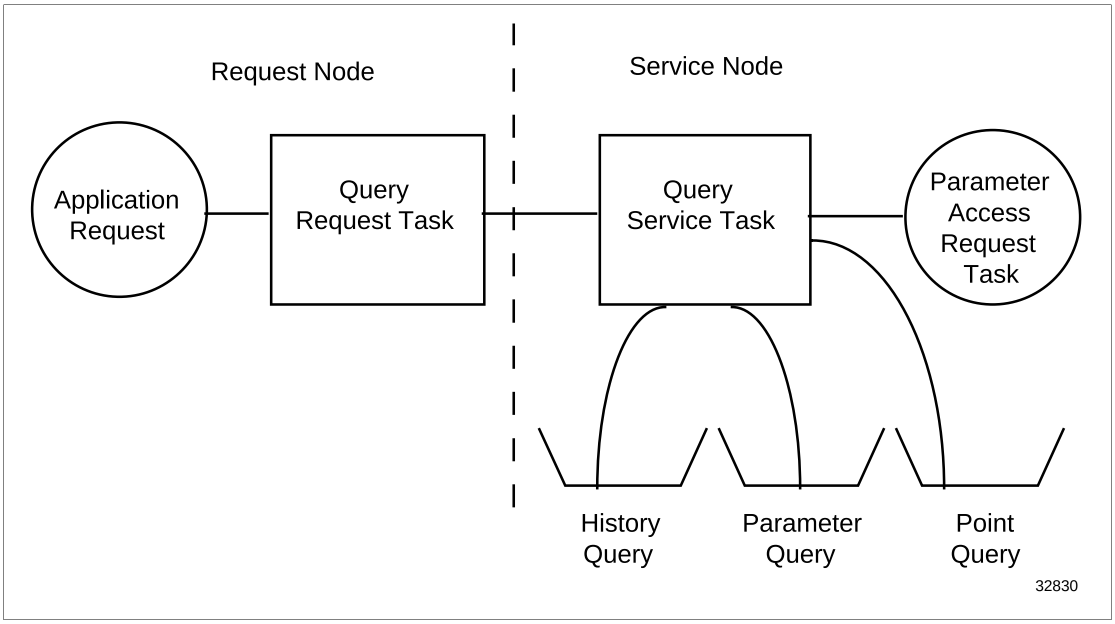
What is an IDB?

An IDB is a listing of parameters created by the requesting node to be accessed from a serving node. These lists are created when needed and deleted upon parameter request completion.

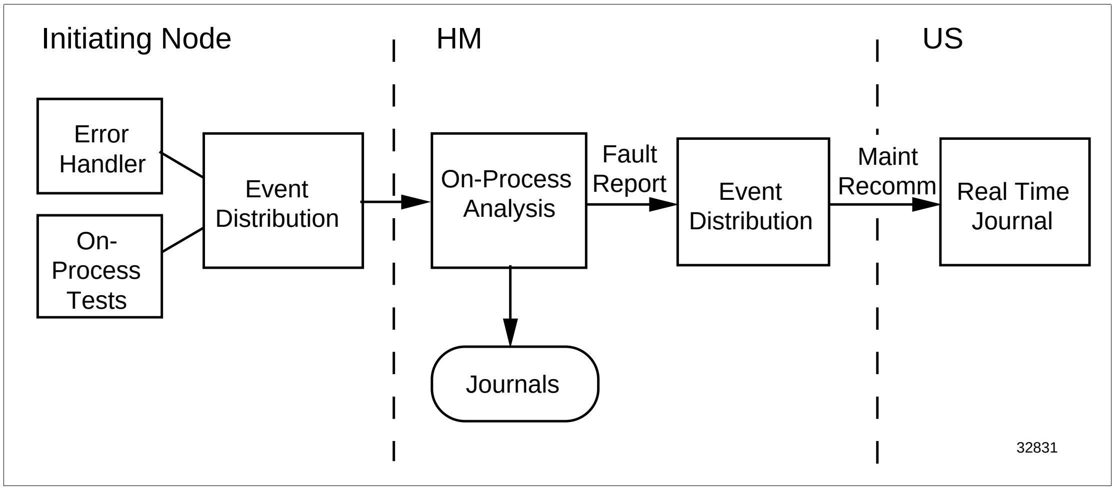
Entity Name Conversion



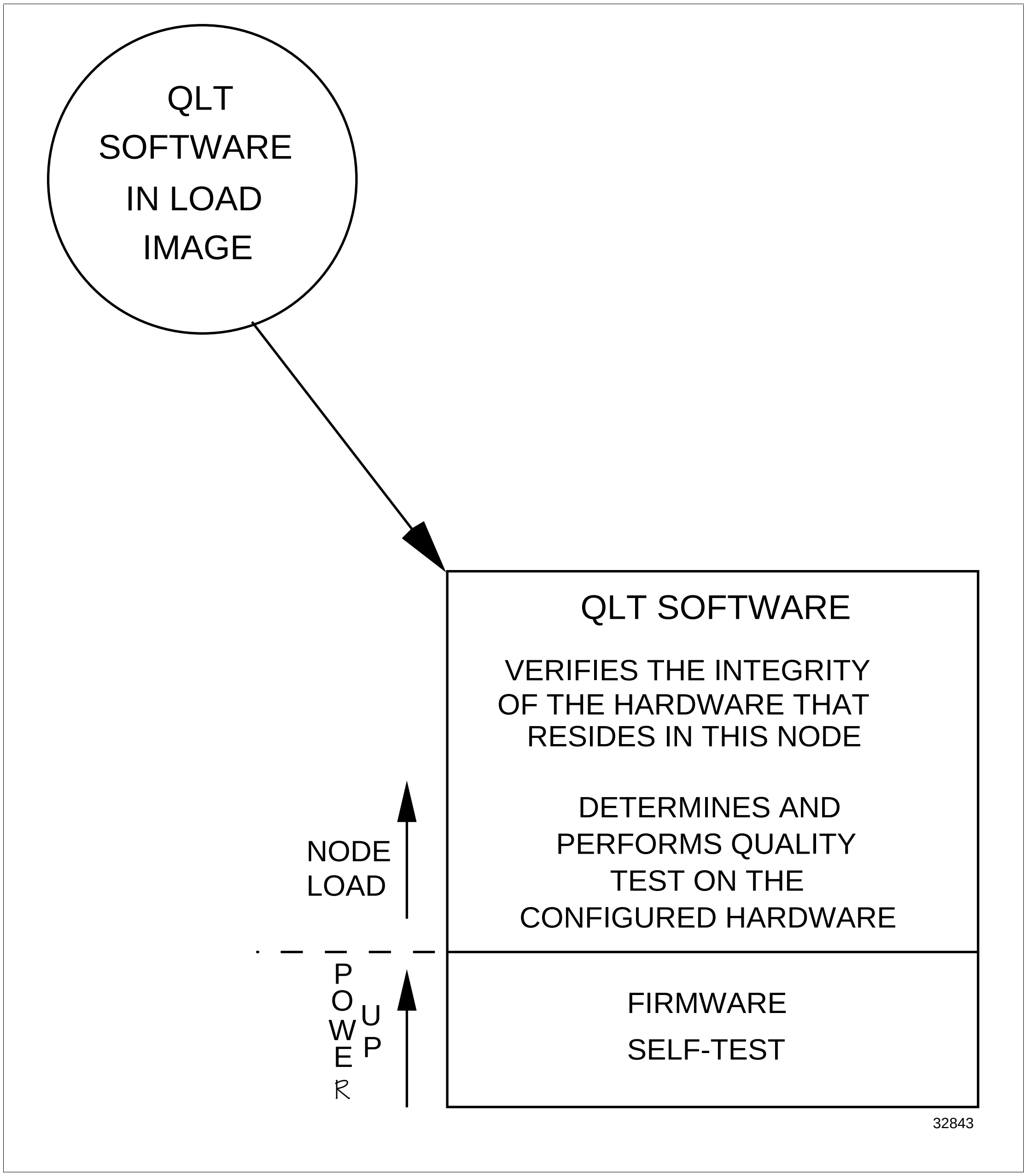
Queries


Request Node
Service Node
Application Request
Query Request Task
Query Service Task
Parameter Access Request Task
History Query
Parameter Query
Point Query
32830

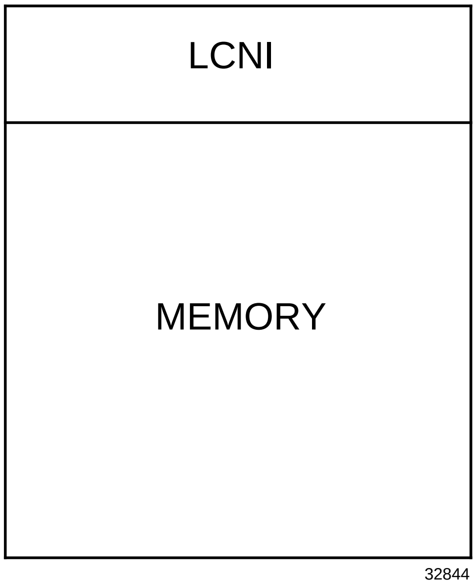
On-Process Analysis



Quality and Logic Test



On-Process Test



Background testing of the LCNI checks the ability to determine cyclic redundancy check errors.

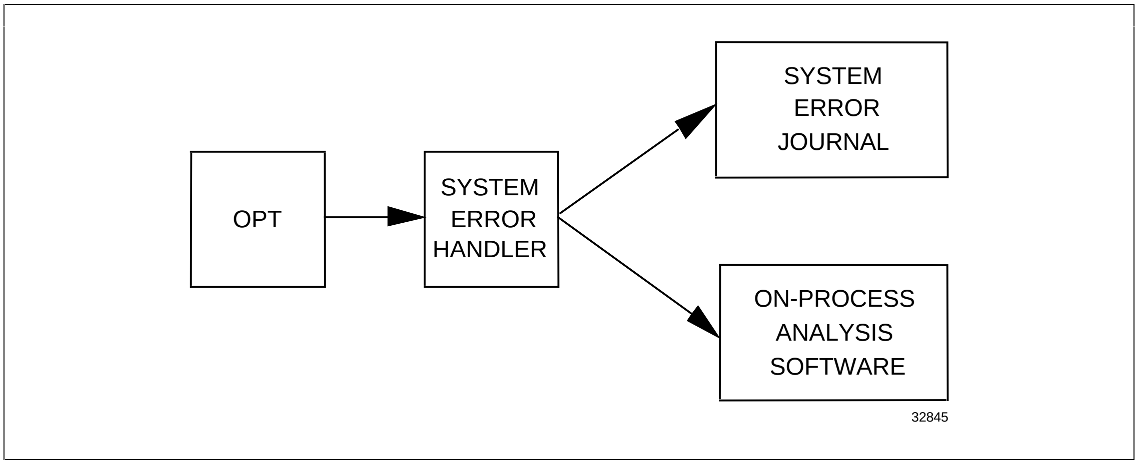
— — — — — — — — —

Memory Scrub uses EDAC to determine and correct single- bit errors.

EDAC test determines the ability to detect and correct single-bit errors.

Load image integrity check determines if there is a problem with memory caused by a hardware or software problem.

OPT reports all detected errors to the System Error Handler, which reports the errors to the System Error Journal and On-Process Analysis for subsequent analysis.



LAST PAGE