

AD-A138 481

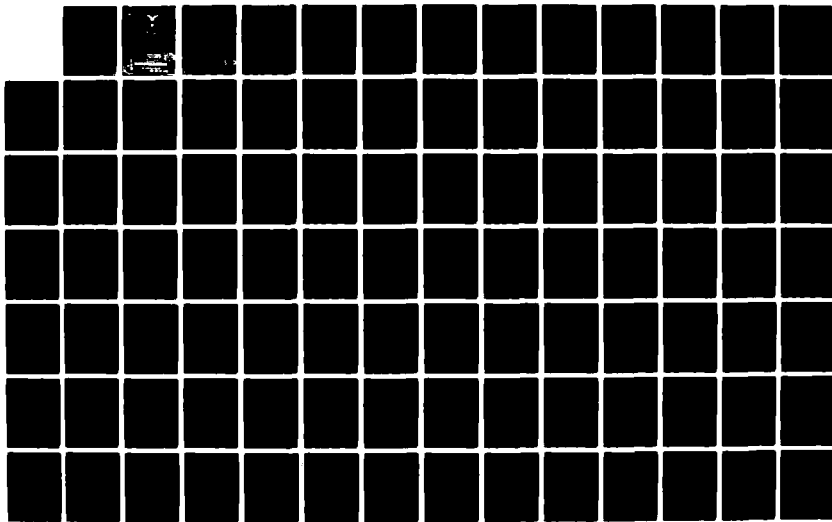
PROTOCOL STANDARDS AND IMPLEMENTATION WITHIN THE
DIGITAL ENGINEERING LABO..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P W PHISTER
DEC 83 AFIT/GE/EE/83D-58-PT-1

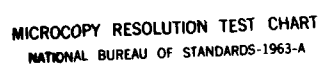
1/4

UNCLASSIFIED

F/G 17/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A138481



AFIT/GE/EE/83D-58

Protocol Standards and Implementation
Within the Digital Engineering Laboratory
Computer Network (DELNET)
Using the
Universal Network Interface Device
(UNID)

THESIS

(Part 1 of 2)

AFIT/GE/EE/83D-58

Paul W. Phister, Jr.
Capt USAF

DTIC
ELECTE
MAR 1 1984
S D B

Approved for public release; distribution unlimited.

AFIT/GE/EE/83D-58

Protocol Standards and Implementation
Within the Digital Engineering Laboratory
Computer Network (DELNET)
Using the
Universal Network Interface Device
(UNID)

THESIS

(Part 1 of 2)

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Paul W. Phister, Jr., BS.EE, MS.SM

Capt

USAF

Graduate Electrical Engineering

December 1983

Approved for public release; distribution unlimited.

Preface

This investigation represents a first draft of the required protocols to implement the Air Force Institute of Technology's Digital Engineering Laboratory Network (DELNET). Hopefully, the work accomplished during this investigation will lay down a firm foundation for continued research regarding the Universal Network Interface Device (UNID).

At this time I would like to thank my thesis advisor, Dr. Gary Lamont, for allowing me the freedom to accomplish this thesis as I perceived it should have been done. I would also like to thank my three thesis readers: LtCol Harold Carter, Lt Col Antone Kusmanoff, and Maj Walt Seward for their much regarded comments throughout this thesis effort. Their experience and patience has greatly improved the quality of this investigation.

I would like to especially thank two fellow students who have helped, and hopefully vice versa, throughout this thesis effort: Capt Bill Matheson and Lt Ken Cole. Their ideas and comments came in mighty handy at times.

But above all, I would especially like to thank my wife Jo Anne (TJ) for being my Tech Editor and keeping my verbs in agreement. Without her willingness to watch me disappear for weeks on end this thesis effort would not have been possible.

Table of Contents

	<u>Page</u>
Preface.....	ii
List of Figures.....	vii
List of Tables.....	xi
Abstract.....	xii
Glossary.....	xiii
I. Introduction.....	1-1
Background.....	1-1
UNID History.....	1-1
UNID Status at Start of this Investigation.....	1-15
UNID Design Goal.....	1-18
Hardware Design Goal.....	1-18
Software Design Goal.....	1-21
Overview of the Thesis.....	1-22
Problem.....	1-22
Scope.....	1-22
Assumptions.....	1-23
Standards.....	1-23
Approach.....	1-23
Equipment.....	1-24
II. DELNET Protocol Requirements.....	2-1
Introduction.....	2-1
Physical Layer Protocol.....	2-5
Overview of the Physical Layer.....	2-5
Physical Layer as Applied to the DELNET.....	2-5
Network Layer Protocol	2-6
Overview of the Network Layer.....	2-6
Network Layer as Applied to the DELNET.....	2-6
Transport Layer Protocol	2-10
Overview of the Transport Layer.....	2-10
Transport Layer as Applied to the DELNET.....	2-15
Session Layer Protocol	2-17
Overview of the Session Layer.....	2-17
Session Layer as Applied to the DELNET.....	2-21
Presentation Layer Protocol	2-21
Overview of the Presentation Layer.....	2-21
Presentation Layer as Applied to the DELNET.....	2-23
Application Layer Protocol	2-24
Overview of the Application Layer.....	2-24
Application Layer as Applied to the DELNET.....	2-26
Chapter Summary.....	2-26

Table of Contents (cont)

	<u>Page</u>
III. DELNET Software Test Plan.....	3-1
Introduction.....	3-1
Objectives.....	3-4
Success Criterion.....	3-4
Testing Procedures.....	3-5
Phase I: UNID Local Side Data Transfer.....	3-5
Phase II: UNID Network Side Data Transfer.....	3-6
Phase III: Local-to-Networkd and Network-to-Local Data Transfer.....	3-7
Phase IV: Local-to-Local Loopback Data Transfer	3-7
Phase V: Local-to-Local Data Transfer.....	3-7
Phase VI: Network-to-Network Loopback Data Transfer.....	3-8
Phase VII: Host-to-Host Data Transfer via 2-UNID Network.....	3-8
Phase VIII: Host-to-Host Data Transfer via 18-pair Cable.....	3-9
Phase IX: Host-to-Host Data Transfer via 3-UNID Network.....	3-9
Hardware and Software Test Configurations.....	3-9
Hardware Devices Required.....	3-19
Chapter Summary.....	3-19
IV. DELNET Software Implementation.....	4-1
Introduction.....	4-1
UNID Tables.....	4-1
UNID Local Operating System.....	4-3
UNID Network Operating System.....	4-9
UNID Memory Map and Linking Procedures.....	4-13
Zilog Software.....	4-16
Chapter Summary.....	4-21
V. Software Testing and Validation Results.....	5-1
Introduction.....	5-1
Phase I: UNID Local Side Data Transfer.....	5-1
Phase II: UNID Network Side Data Transfer.....	5-2
Phase III: Local-to-Networkd & Network-to-Local Data Transfer.....	5-3
Phase IV: Local-to-Local Loopback Data Transfer.	5-4
Phase V: Local-to-Local Data Transfer.....	5-5
Phase VI: Network-to-Network Loopback Data Transfer.....	5-7
Phase VII: Host-to-Host Data Transfer via 2-UNID Network.....	5-7
Phase VIII: Host-to-Host Data Transfer via 18-pair Cable.....	5-8

Table of Contents (cont)

	<u>Page</u>
Phase IX: Host-to-Host Data Transfer via 3-UNID Network.....	5-8
Chapter Summary.....	5-8
VI. Conclusions and Recommendations.....	6-1
Introduction.....	6-1
Conclusions.....	6-1
Recommendations.....	6-3
Bibliography.....	Bib-1
Appendix A: DELNET Monitor User's Guide.....	A-1
Introduction.....	A-1
Table of Contents.....	A-1
Zilog Modifications.....	A-1
UNID Cable Connections and Physical Layer Enhancements.....	A-3
UNID Start-up Procedures.....	A-8
Zilog Host Start-up Procedures.....	A-15
DELNET Cable Layout.....	A-17
Appendix Summary.....	A-19
Appendix B: N-Layer Concept Description.....	B-1
Introduction.....	B-1
Table of Contents.....	B-1
Principles of Layering.....	B-1
Blocking.....	B-6
Segmenting.....	B-7
Multiplexing/Splitting.....	B-8
Flow Control.....	B-9
Error Detection.....	B-9
Appendix Summary.....	B-10
Appendix C: DELNET Protocol Standard.....	C-1
Introduction.....	C-1
Table of Contents.....	C-1
Physical Layer Protocol Specification.....	C-2
Data Link Layer Protocol Specification.....	C-4
Network Layer Protocol Specification.....	C-5
Routing.....	C-5
Flow Control.....	C-10
DATAGRAM Service.....	C-10
Virtual Circuit Service.....	C-11
Transport Layer Protocol Specification.....	C-20
DATAGRAM Service.....	C-20
Virtual Circuit Service.....	C-31
Session Layer Protocol Specification.....	C-39

Table of Contents (cont)

	<u>Page</u>
Presentation Layer Protocol Specification.....	C-46
Application Layer Protocol Specification.....	C-46
Appendix D: Detailed Listing of Test Procedures.....	D-1
Introduction.....	D-1
Table of Contents.....	D-2
Testing Procedures.....	D-3
Appendix E: Data Dictionary.....	E-1
Introduction.....	E-1
Table of Contents.....	E-1
Appendix Summary.....	E-42
Appendix F: Software Listings.....	F-1
Introduction.....	F-1
Table of Contents.....	F-1
Section I - Setup Software for UNIDS.....	F-2
Section II - UNID Shared Modules.....	F-26
Section III - UNID Local Operating System Software.....	F-39
Section IV - UNID Network Operating System Software.....	F-109
Section V - Zilog Operating System Software.....	F-181
Vita.....	V-1



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
PER CALL JC	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

List of Figures

<u>Figure</u>	<u>Page</u>
1-1 Initial Concept of a Multi-Ring Base Network.....	1-3
1-2 Network Operating System Hierarchy.....	1-5
1-3 Universal Network Interface Device.....	1-6
1-4 UNID Internal Configuration.....	1-7
1-5 Initial DELNET Configuration.....	1-8
1-6 ISO Reference Model applied to DELNET.....	1-8
1-7 UNID Local Operating System Structure Chart.....	1-12
1-8 UNID Network Operating System Structure Chart.....	1-14
1-9 Link Access Procedure Frame Format.....	1-15
1-10 DELNET Frame Format.....	1-16
1-11 DELNET Data Packet Format.....	1-17
1-12 DELNET Hardware Topology.....	1-20
1-13 DELNET Software Topology.....	1-21
2-1 Overview Structure of a Node.....	2-3
2-2 General Message Format.....	2-4
2-3 CCITT X.121 Addressing Scheme.....	2-7
2-4 Sample DELNET Addressing Relationships.....	2-8
2-5 Sample DELNET Topology.....	2-10
2-6 Transport Layer Relationships.....	2-11
2-7 Session Layer Relationships.....	2-18
2-8 Presentation Layer Relationships.....	2-21
2-9 Application Layer Relationships.....	2-24
3-1 Incoming Local Table to LCLCTB Data Transfer.....	3-10
3-2 Incoming Local Table to LCNTTB Data Transfer.....	3-10
3-3 NTLCTB to Local Channel Ports Data Transfer.....	3-11

List of Figures (cont)

<u>Figure</u>	<u>Page</u>
3-4 LCLCTB to Local Channel Ports Data Transfer.....	3-11
3-5 LCNTTB to Outgoing Network Channel Data Transfer..	3-12
3-6 Incoming Network Channel to NTLCTB Data Transfer..	3-12
3-7 Internal Network-to-Network Data Transfer.....	3-12
3-8 Internal Local-to-Network Data Transfer.....	3-13
3-9 Internal Network-to-Local Data Transfer.....	3-14
3-10 Local-to-Local Test Message Transfer.....	3-14
3-11 Zilog-to-Zilog Local-to-Local Test Configuration..	3-15
3-12 Clockwise Portion of the Network Ring.....	3-16
3-13 Counter-Clockwise Portion of the Network Ring.....	3-16
3-14 Zilog-to-Zilog (2-UNID) Test Configuration.....	3-17
3-15 INTEL/432-to-Zilog (2-UNID) Test Configuration....	3-17
3-16 Zilog-to-Zilog (3-UNID) Test Configuration.....	3-18
3-17 Zilog-to-INTEL/432 (3-UNID) Test Configuration....	3-19
4-1 UNID Local and Network Operating Systems' Tables..	4-2
4-2 ROUTE_IN Procedure for UNID LOS.....	4-4
4-3 ROUTE_OUT Procedure for UNID LOS.....	4-5
4-4 DET_DEST Procedures for UNID LOS.....	4-7
4-5 ROUTE_IN Procedure for UNID NOS.....	4-10
4-6 ROUTE_OUT Procedure for UNID NOS.....	4-12
4-7 UNID Operating System Memory Map.....	4-14
4-8 Zilog Operating Systems' Tables.....	4-16
4-9 MAIN Procedure for Zilog Operating System.....	4-17
4-1 ROUTE_IN Procedure for Zilog Operating System.....	4-17
4-11 ROUTE_OUT Procedure for Zilog Operating System....	4-18

List of Figures (cont)

<u>Figure</u>	<u>Page</u>
4-12 Procedures BUILD_ and EXTRACT_TRANSPORT_HEADER....	4-19
5-1 ASCII Code Chart.....	5-6
A-1 UNID Network and Local Port Connections.....	A-3
A-2 UNID DCE Jumper Configuration.....	A-5
A-3 UNID-Zilog Cable Connections.....	A-8
A-4 Zilog System Start-up Message.....	A-9
A-5 N.INSIO_U2.MAP - UNID Network OS Memory Map.....	A-12
A-6 L.VINT_U2.MAP - UNID Local OS Memory Map.....	A-14
A-7 DELNET Cable Path.....	A-17
A-8 Cable Box Connections.....	A-18
B-1 Layering Concept.....	B-2
B-2 Entity Relationships.....	B-3
B-3 Data Unit Interrelationships.....	B-5
B-4 (N)-Layer Blocking.....	B-7
B-5 (N)-Layer Segmenting.....	B-8
C-1 Data Link Layer Header.....	C-4
C-2 Bldg 640 Basement Country Codes.....	C-7
C-3 Bldg 640 1st Floor Country Codes.....	C-8
C-4 Bldg 640 2nd Floor Country Codes.....	C-9
C-5 Network Layer Header.....	C-10
C-6 Call Request/Incoming Call Packet Format.....	C-11
C-7 Call Accepted/Connected Packet Format.....	C-12
C-8 DTE and DCE Clear Confirmation Packet Format.....	C-13
C-9 Clear Request/Indication Packet Format.....	C-13
C-10 DTE and DCE Data Packet Format.....	C-14

List of Figures (cont)

<u>Figure</u>	<u>Page</u>
C-11 DTE and DCE Interrupt Packet Format.....	C-15
C-12 DTE and DCE Interrupt Confirmation Packet Format..	C-15
C-13 DTE and DCE RR Packet Format.....	C-16
C-14 DTE and DCE RNR Packet Format.....	C-16
C-15 Reset Request/Indication Packet Format.....	C-17
C-16 DTE and DCE Reset Confirmation Packet Format.....	C-17
C-17 Restart Request/Indication Packet Format.....	C-18
C-18 DTE and DCE Restart Confirmation Packet Format....	C-18
C-19 DTE REJ Packet Format.....	C-19
C-20 Transport Layer DATAGRAM Header Format.....	C-20
C-21 TPDU Packet Format.....	C-32
C-22 Establish PDU Header Format.....	C-40
C-23 Accept PDU Header Format.....	C-41
C-24 Abort PDU Header Format.....	C-42
C-25 Finish PDU Header Format.....	C-43
C-26 Disconnect PDU Header Format.....	C-43
C-27 Not_Finished PDU Header Format.....	C-43
C-28 Please_Turn PDU Header Format.....	C-44
C-29 Cancel PDU Header Format.....	C-44
C-30 Expedited PDU Header Format.....	C-44
C-31 Data PDU Header Format.....	C-44
C-32 Transaction PDU Header Format.....	C-45
C-33 Presentation Layer Header Format.....	C-46
C-34 Application Layer Header Format.....	C-47
F-1 UNID Initialization and File Transfer Procedures..	F-3

List of Tables

<u>Table</u>	<u>Page</u>
A-1 UNID RS-232 Pin Assignments.....	A-4
A-2 UNID RS-449 Pin Assignments.....	A-6
A-3 UNID Network Card to RS-449 Pin Assignments.....	A-7
C-1 RS-232 Standard Pin Assignments.....	C-2
C-2 RS-449 Standard Pin Assignments.....	C-3
C-3 Address Conversion to Internet Header Format.....	C-26
C-4 Internet Options.....	C-26

Abstract

Development of the Air Force Institute of Technology's Digital Engineering Laboratory Network (DELNET) was continued with the development of an initial draft of a protocol standard for all seven layers as specified by the International Standards Organization's (ISO) Reference Model for Open Systems Interconnections. This effort centered on the restructuring of the Network Layer to perform Datagram routing and to conform to the developed protocol standards and actual software module development of the upper four protocol layers residing within the DELNET Monitor (Zilog MCZ 1/25 Computer System). Within the guidelines of the ISO Reference Model the Transport Layer was developed utilizing the Internet Header Format (IHF) combined with the Transport Control Protocol (TCP) to create a 128-byte Datagram. Also a limited Application Layer was created to pass the Gettysburg Address through the DELNET. This study formulated a first draft for the DELNET Protocol Standard and designed, implemented, and tested the Network, Transport, and Application Layers to conform to these protocol standards.

Glossary

NOTE: This Glossary is mostly taken from Reference 26 to aid the reader in the (N)-Layer terminology. For a general discussion on the (N)-Layer concept refer to Appendix B. Since each (N)-Layer identifier can take on the prefix (N-1), (N), or (N+1), only one definition is given with the others implied.

Acknowledgement: A function used between peer (N)-Entities to obtain a higher probability of detection of (N)-Protocol-Data-Unit loss than provided by the (N-1)-Layer.

Blocking: A function of an (N)-Entity to map multiple (N)-Service-Data-Units on one (N)-Protocol-Data-Unit.

Centralized Multi-Endpoint-Connection: A multi-endpoint-connection where data sent by the entity associated with the central connection-endpoint is received by all other entities, while data sent by one of the other entities is only received by the central entity.

Concatenation: A function of an (N)-Entity to map multiple (N)-Protocol-Data-Units on one (N-1)-Service-Data-Unit.

Control_Code: A code used to identify the addressing scheme used in the Datagram by the End Users.

Correspondent (N)-Entities: (N)-Entities at the ends of an (N-1)-Connection.

Country_Code: This is a cluster of UNIDs connected in a dual ring servicing up to 64 Local Area Networks.

Datagram: A finite length data packet, with destination host address and source address, that can be exchanged in its entirety between hosts.

Deblocking: A function of an (N)-Entity to identify multiple (N)-Service-Data-Units which are contained on one (N)-Protocol-Data-Unit.

Decentralized Multi-Endpoint-Connection: A multi-endpoint-connection where data sent by any entity associated with a connection-endpoint is received by all other entities.

DELNET: Abbreviation for Digital Engineering Laboratory Computer Network.

Demultiplexing: The function of an (N)-entity to identify multiple (N)-Connections within (N-1)-Service-Data-Units received on one single (N-1)-Connection supporting the multiple (N-1)-Connections.

Glossary (cont)

Expedited (N-1)-Service-Data-Unit: A small (N-1)-Service-Data-Unit whose transfer is expedited. The (N-1)-Layer ensures that an expedited-data-unit will not be delivered after any subsequent service-data-unit or expedited unit sent on that connection. An expedited (N-1)-Service-Data-Unit is intended to be processed by the receiving (N)-Entity with priority over normal (N-1)-Service-Data-Units. An Expedited (N-1)-Service-Data-Unit may also be referred to as an (N-1)-Expedited-Data-Unit.

Flow Control: A function for the control of the data flow within a layer or between adjacent layers.

Gateway: A node at which two or more networks are connected, with the fundamental role of serving as the boundary between the internal protocols of the connected networks.

Global-Title: A title which is unique within the OSI environment and comprises two parts, a Title-Domain-Name and a Local-Title.

Host_Code: A code having the values 0-255 to which a particular Host, located within a Local Area Network, is uniquely assigned within a particular Network.

LAN: Abbreviation for Local Area Network.

Local-Title: A title which is unique within a Title-Domain.

Multi-Connection-Endpoint-Identifier: A identifier required to specify which connection-endpoint of a multi-connection-endpoint should accept data that is being transferred.

Network_Code: A code having the values 0-15 to represent a particular UNID within a Country.

(N-1)-Layer: The next lower layer in the OSI Hierarchy.

(N-1)-Service-Data-Unit: The amount of (N-1)-Interface-Data whose identity is preserved from one end of an (N-1)-Connection to the other.

(N)-Address: An identifier which tells where an (N)-Service-Access-Point may be found.

(N)-Connection: An association established by the (N)-Layer between two or more (N+1)-Entities for the transfer of data.

(N)-Connection-Endpoint: A terminator at one end of an (N)-Connection within an (N)-Service-Access-Point.

Glossary (cont)

(N)-Connection-Endpoint-Identifier: An identifier of an (N)-Connection-Endpoint which can be used to identify the corresponding (N)-Connection at an (N)-Service-Access-Point.

(N)-Connection-Endpoint-Suffix: A part of an (N)-Connection-Endpoint-Identifier which is unique within the scope of an (N)-Service-Access-Point.

(N)-Data-Communication: The part of the (N)-Function corresponding to transferring of (N)-Protocol-Data-Units according to an (N)-Protocol over one or more (N-1)-Connection.

(N)-Data-Sink: That (N)-Entity that receives (N-1)-Service-Data-Units from an (N-1)-Connection.

(N)-Data-Source: That (N)-Entity that enters (N-1)-Service-Data-Units from an (N-1)-Connection.

(N)-Data-Transmission: The part of the (N)-Service that conveys (N)-Service-Data-Units from one (N+1)-Entity for reception at one or more (N+1)-Entities via (N)-Connections.

(N)-Directory: An (N)-Function by which the Global-Title of an (N)-Entity is translated into the (N-1)-Address of an (N-1)-Service-Access-Point to which it is attached.

(N)-Duplex-Transmission: (N)-Data Transmission of (N)-Service-Data-Units in both directions at the same time.

(N)-Entity: An active element within an (N)-Subsystem.

(N)-Facility: A part of an (N)-Service.

(N)-Function: A part of activity of (N)-Entities.

(N)-Half-Duplex-Transmission: (N-1)-Data-Transmission of (N)-Service-Data-Units in either direction one direction at a time; the choice of direction is controlled by an (N+1)-Entity.

(N)-Interface-Control-Information: Information transferred between an (N+1)-Entity and an (N)-Entity to coordinate their joint operation.

(N)-Interface-Data: Information transferred from an (N+1)-Entity to an (N)-Entity for transmission to a correspondent (N+1)-Entity over an (N)-Connection, or conversely, information transferred from an (N)-Entity to an (N+1)-Entity which has been received over an (N)-Connection from a correspondent (N+1)-Entity.

Glossary (cont)

(N)-Interface-Data-Unit: The unit of information transferred across the service-access-point between an (N+1)-Entity and an (N)-Entity in a single interaction and which contains (N)-Interface-Control-Information and/or possibly the whole or part of an (N)-Service-Data-Unit.

(N)-Layer: A well defined subdivision of the OSI architecture, which constitutes subsystems of the same rank (N).

(N)-One-Way-Communication: (N)-Data Communication such that (N)-Protocol-Data-Units are transferred in one pre-assigned direction.

(N)-Protocol: A set of rules and formats which determines the communication behavior of (N)-Entities in the performance of (N)-Functions.

(N)-Protocol-Control-Information: Information exchanged between (N)-Entities, using an (N-1)-Connection, to coordinate their joint operation.

(N)-Protocol-Data-Unit: A unit of data specific in an (N)-Protocol which consists of (N)-Protocol-Control-Information and possibly (N)-User-Data.

(N)-Relay: An (N)-Function through which an (N)-Entity forwards data received from a Correspondent (N)-Entity to another Correspondent (N)-Entity.

(N)-Service: A capability of the (N)-Layer and the layer beneath it, which is provided to (N+1)-Entities at the boundary between the (N)-Layer and the (N+1)-Layer.

(N)-Service-Access-Point: The access means by which (N)-Services are provided by an (N)-Entity to an (N+1)-Entity and formats.

(N)-Service-Access-Point-Address: An identifier which tells where an (N)-Service-Access-Point may be found.

(N)-Service-Connection-Endpoint-Identifier: An identifier which uniquely specifies an (N)-connection within the environment of the correspondent (N+1)-Entities.

(N)-Simplex-Transmission: (N)-Data Transmission of (N)-Service-Data-Units in one pre-assigned direction only.

(N)-Subsystem: An element in a hierarchical division of a system which interacts only with elements in the next higher division and the next lower division of that system.

Glossary (cont)

(N)-Two-Way-Alternate-Communication: (N)-Data-Communication such that (N)-Protocol-Data-Units are transferred in both directions, one direction at a time.

(N)-Two-Way-Simultaneous-Communication: (N)-Data-Communication such that (N)-Protocol-Data-Units are transferred in both directions at the same time.

(N)-User-Data: The data transferred between (N)-Entities on behalf of the (N+1)-Entities for whom the (N)-Entities are providing service.

(N+1)-Layer: The next higher layer in the OSI Hierarchy

OSI: Abbreviation for Open System Interconnection.

Peer-Entities: Entities within the same (N)-Layer.

Port_Code: A code that represents the destination port address for the incoming Datagram or message.

Recombining: The function of an (N)-Entity to identify one (N)-Connection within (N-1)-Service-Data-Units received on several (N-1)-Connections supporting the (N)-Connection.

Reset: A function which permits the correspondent (N)-Entities to come back to a predefined state with a possible loss or duplication of data.

Segmenting: A function of an (N)-Entity to map one (N)-Service-Data-Unit out of multiple (N)-Protocol-Data-Units.

Sequencing: A function of the (N)-Layer to provide the (N)-Service of delivering data in the same order it was sent.

Splitting: A function within the (N)-Layer by which more than one (N-1)-Connection is used to support one (N)-Connection.

Sub-Layer: A grouping of functions in a layer.

Title: A permanent identifier for an Entity.

Title-Domain: A subset of the title space within the OSI.

Title-Domain-Name: An identifier which uniquely identifies a Title-Domain within the OSI environment.

UNID: Abbreviation for Universal Network Interface Device.

Virtual Circuit: A circuit or channel that is established between source and destination packet switches and usually requires some form of setup prior to data transfer.

I. INTRODUCTION

Background

UNID History

The Electrical Engineering Digital Engineering Laboratory (DEL) located at the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, Ohio, functions primarily as a student research center in the area of computer systems.

This investigation is part of the Institute's continuing effort to link all the computers within the Engineering Laboratory into a network called the Digital Engineering Laboratory Network (DELNET), under the direction of the Electrical Engineering Department. The DELNET is the result of a number of influences, including the recent increase in computer networks, USAF operational interest in base-level telecommunication using computers, and AFIT's concern for advancing computer network research and optimizing internal computer network resources. Due to interest in these areas, AFIT has developed a sequence of thesis projects that have laid the necessary groundwork for a network configuration within the DEL (Refs 1,4,18,19,27,34, and 37).

There are many different topologies for computer networks. The more common ones are called Bus, Complete, Ring (Loop), Star, and Tree (Refs 13:77-85, 22:18-26, 39:24-32, 49:9). In a report by the 1842nd Electronics Engineering Group, subsequently called the Weaver Report (Ref 53), it was stated that a ring or loop network appeared to be applicable

to base-level computer communications. The other networks are commonly known configurations, but only the ring topology will be discussed. For more information on these other topologies, refer to the references quoted.

Simply stated, in the ring concept, all processors (or nodes) are connected to a common communications path which is configured into a closed ring. The message is put into a "frame" and deposited onto the ring network; it continues around the ring and is repeated by each node until the addressee recognizes it, whereupon the frame is accepted and withdrawn from the network. There is no need for a central controller, causing the network to be transparent to the user. The Weaver Report states that the processors and nodes within the ring network would require connectivity with processors and nodes in another computer network. Thus another computer network, which does not necessarily have to be of the same topology, could be connected to the first by a gateway node that transfers traffic between networks with the proper protocols and formats. Figure 1-1 illustrates the initial concept of a multi-ringed base network, when applied to an Automatic Digital Network (Autodin), as outlined in the Weaver Report. The following terms and abbreviations are used in the figure:

1. A ring network is called a User Community.
2. Gateway nodes are called Inter-Ring Interface Nodes.
3. DCO is Dial Central Office.
4. ESS is Electronic Switching Center.

5. CAP is Communications Access Processor.

6. LDMX is Local Distribution Message Exchange.

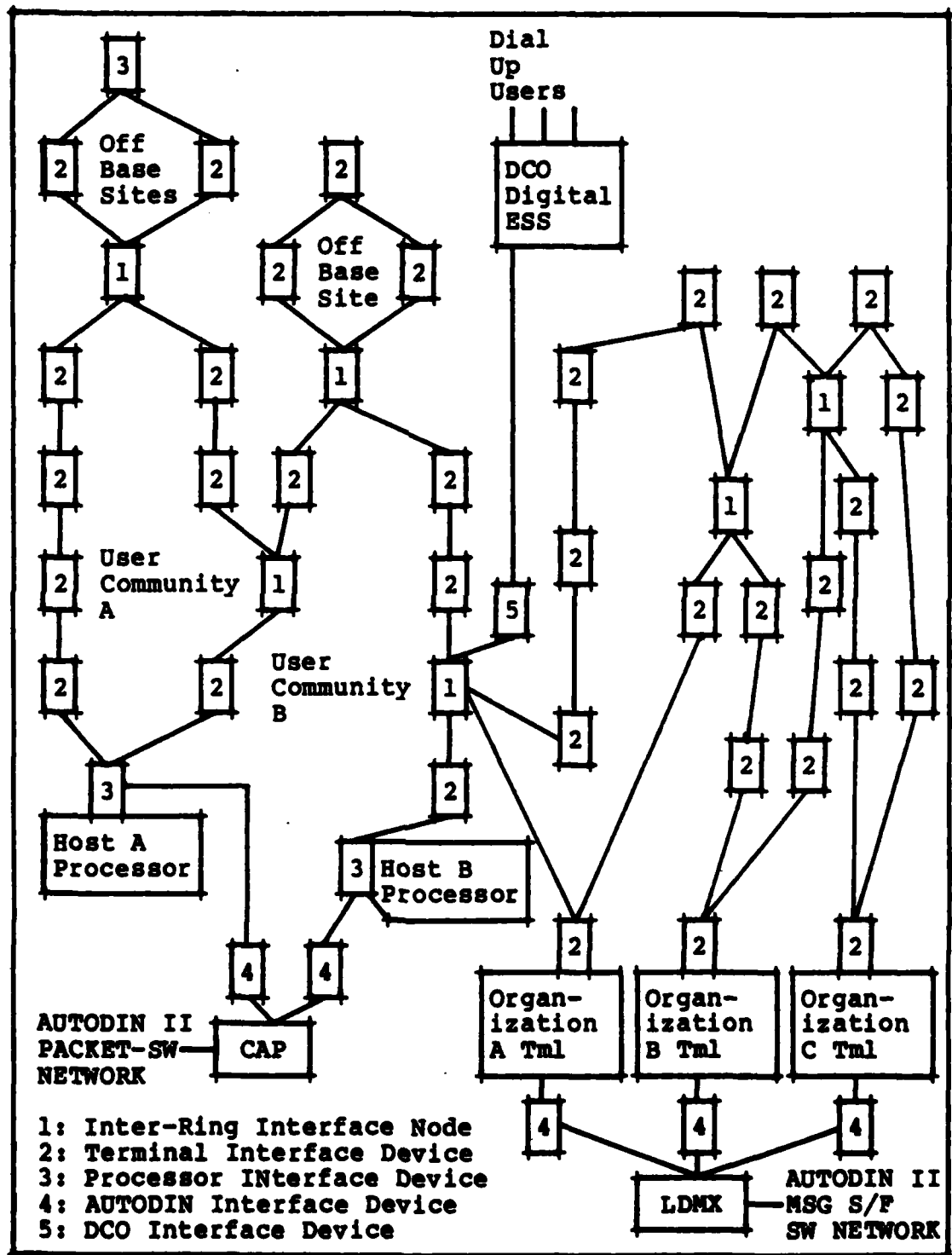


Figure 1-1. Initial Concept of a Multi-Ring Base Network

This multi-ring concept has many advantages for a base-level computer network. From the automatic data processing (ADP) side, communications control is simplified in that a communications front-end is no longer required for host processors. The host communicates to different nodes via a single high-speed multiplexed port, the same port that normally interfaces with the front-end processor of a Host. All that is required, according to the Weaver Report, is a ring interface device. As a processor or node is added to the ring, an interface device is added. The Weaver Report summarized that a multi-ring network concept would be a primary candidate for a "typical" base-level teleprocessing and narrative message network of the late 1980's. A key to the multi-ring concept was the development of five different devices to handle the requirements of interfacing the multiple rings. Figure 1-1 also illustrates these devices. This basic idea was then expanded and tasked to Rome Air Development Center (RADC) for incorporation into a postdoctoral study program.

It was recognized that the various interface devices proposed by the Weaver Report had similar features and were required to perform similar functions. Thus, it was decided that a single flexible and versatile interface device could be developed which could meet the separate requirements of each of the five proposed interface devices in a computer ring. Since the device must be flexible enough to provide interfacing for a number of processors and nodes with

various other different types of equipment, it was decided to call the ring interface device a Universal Network Interface Device (UNID) (Ref 4:2).

In 1978, Ravenscroft outlined the initial conceptual design. Figure 1-2 illustrates the Network Operating System Hierarchy that was developed by Ravenscroft (Ref 34:13).

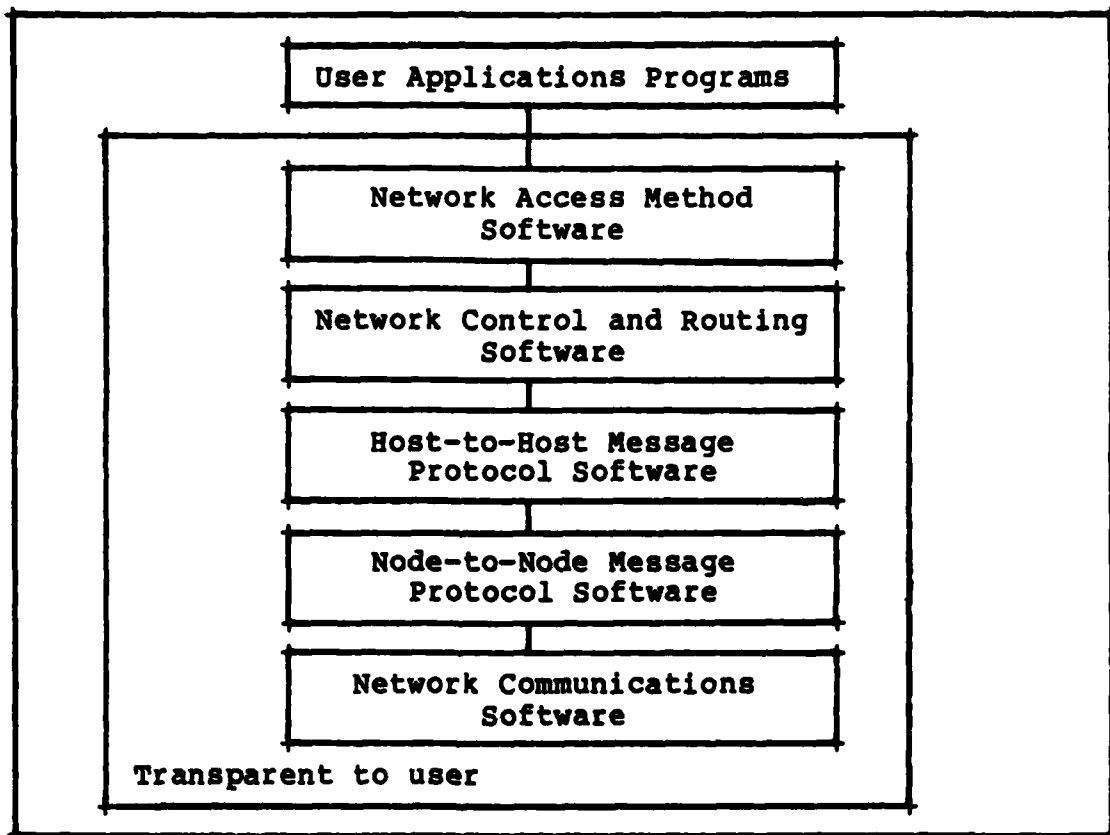


Figure 1-2. Network Operating System Hierarchy

Later in 1978, Sluzevich separated the design of the UNID into four tasks (Ref 37:7-8):

1. Define the functional requirements of the UNID.
2. Translate functional requirements into system design (Hardware and Software).
3. Design the UNID's Hardware.

4. Design the UNID's Software.

Figure 1-3 gives the layout of the UNID from the results of the thesis by Sluzevich.

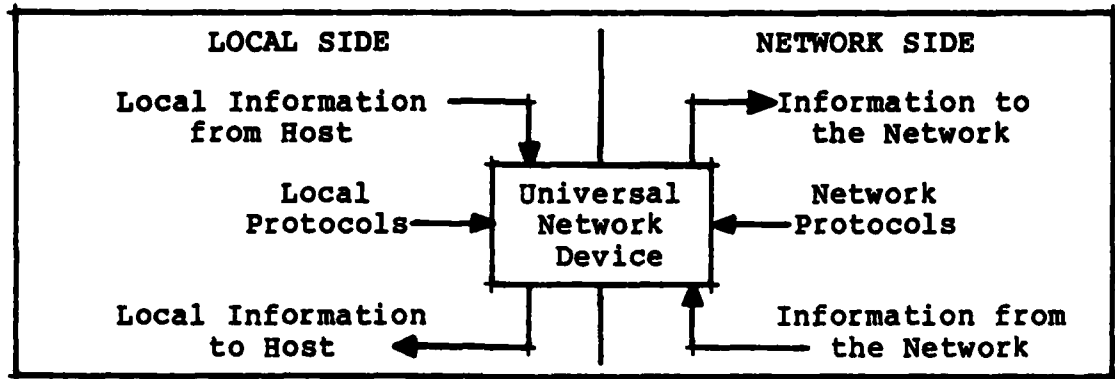


Figure 1-3. Universal Network Interface Device

Thus, the UNID evolved into a modular style. This approach was used to provide the required degree of universality to the UNID. To implement this modular approach, three cards were designed. The first was the "input card", which provided RS-232C interfacing with four individual full duplex ports, expandable with the addition of more input cards. The second card was the "network card" to connect the UNID into a communications network. The Network Card consisted of two full duplex RS-422/RS-423 (electrical) and RS-449 (mechanical/functional) ports with the number expandable through the addition of more network cards. The last card developed was the "processor card". Its function was to provide the control and processing capabilities for both the Local and Network sides.

In 1979, Brown (Ref 4: 7-8) upgraded the UNID's basic design to two Z-80 Microprocessor Boards which shared a common block of memory. Each of the Z-80 processor boards was

to perform specific tasks, one for the Network side and the other for the Local side. The Network Card, controlled by processor Y, was designed around the Z80-SIO (Serial Input/Output), since the SIO provided capability for interfacing with different protocols. If the message was addressed to the UNID device, then the Network Card would place the message into the shared memory. The Local Card, controlled by processor X, was designed to provide four RS-232C compatible input/output ports to interface with either Data Circuit Terminating Equipment or Data Terminal Equipment. The Local Card would be responsible for transferring the message to the correct end user.

The final UNID configuration consisted of two microprocessor boards which shared a 16K block of Random-Access-Memory(RAM). Access to the shared memory was provided by the Processor Cards. The X-Processor was used to interface with network traffic through the Network Card and the Y-Processor was used to interface with local traffic through the Local Card. Figure 1-4 illustrates the UNID at this time.

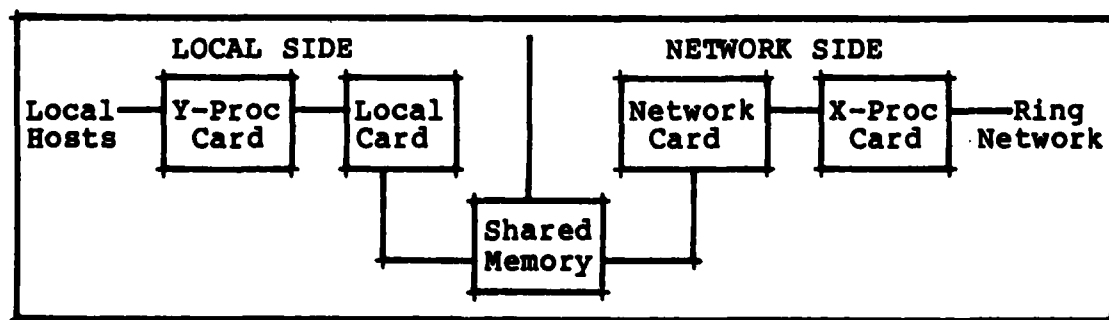


Figure 1-4. UNID Internal Configuration

In 1980, Baker (Ref 1:46-64) addressed the need for the complete testing of the prototype UNID along with the

development and testing of the software and hardware required to have an operational device. Baker also modified an existing monitor program and terminal to improve the testing of both the hardware and software for the UNID. At the end of the 1980 phase, two new memory boards were developed; the Shared and System Memory Boards (Ref 1:11-16).

In 1981, Geist indicated that the UNID protocol structure was ready for the DELNET. Its initial concept is shown in Figure 1-5 (Ref 18:5).

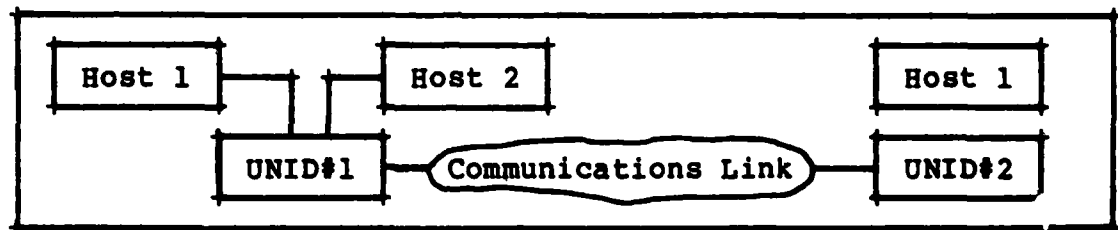


Figure 1-5. Initial DELNET Configuration

The International Standards Organization (ISO) has developed "The Reference Model of Open Systems Interconnection" in an effort to provide international standardization of network protocols (Refs 24:57-59, 26). Figure 1-6 gives the ISO model as applied to the DELNET.

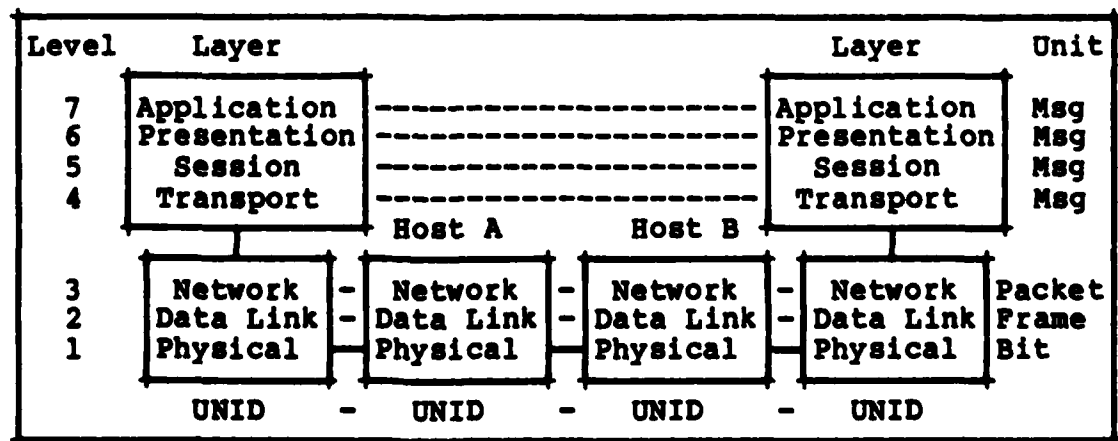


Figure 1-6. ISO Reference Model applied to DELNET

A brief description of each of the ISO protocol layers follows (Refs 19:2.6-2.9, 49:16-21, 26:55-100):

1. The Physical Layer. The Physical Layer provides mechanical, electrical, functional and procedural characteristics to activate, maintain and deactivate physical connections for bit transmission between data-link-entities, possibly through intermediate systems, each relaying bit transmission within the Physical Layer.

2. The Data Link Layer. The Data Link Layer provides functional and procedural means to establish and release data-link-connections among network-entities. This is accomplished by creating frames of data.

3. The Network Layer. The Network Layer provides the means to establish, maintain, and terminate network-connections between systems containing communication application-entities and provides the functional and procedural means to exchange network-service-data-units between transport-entities over network connections(Routing).

4. The Transport Layer. This is also known as the Host-to-Host layer, and it provides transparent transfer of data between session-entities. The Transport Layer relieves the transport users from any concern with the detailed way in which reliable and cost effective transfer of data is achieved.

5. The Session Layer. This layer is the user's interface into the network. Its purpose is to provide the means necessary for cooperating presentation-entities to organize

and synchronize their dialogue and manage their data exchange. To do this, the Session Layer provides services to establish a session-connection between two presentation-entities and to support their orderly data exchange interactions.

6. The Presentation Layer. This layer is tasked to perform functions that can be offered as library routines. The Presentation Layer is concerned only with the syntactic view of the data and not with its semantics. Some of the tasks performed by the Presentation Layer are format compatability conversion, text compression, encryption, and terminal standardization.

7. The Application Layer. As the highest layer in the Reference Model of Open Systems Interconnection (OSI), the Application Layer provides a means for the application-processes to access the OSI environment to exchange meaningful information.

Taking into account the preceding protocol methodology, operational software was developed for the UNID. The UNID's job is three phased: it must concentrate the input communication from four local channels; it must distribute local network communication to the identified destination; and it must store and forward network traffic not addressed to the connected host computers. Therefore, Geist (Ref 18) addressed the required software needed to support host-to-node and host-to-host connections. With four local channels and one network port, the UNID can be thought of as having

five sources and five destinations for data. The local side must be capable of accepting input from any local communication channel and routing that data either to another local channel or to the network via the shared memory area. Parallel to that operation, the network side must be capable of accepting information from the network port and routing it to either a local channel or back out onto the network. These data flow requirements lend themselves very nicely to data tables. These tables are important in this thesis effort; therefore, a more detailed description of each table along with associated procedures follows:

1. Init_L_Tab: Written in Assembly Language. Initializes the data tables used by the Local Operating System.

2. Init_U_Shtab: Written in Assembly Language. Initializes the data tables shared by both the Local and Network Operating Systems.

3. Invint: Written in Assembly Language. Initializes the Input/Output vector interrupt process.

4. Route_In: Written in PLZ Language. Routes data from the four local input tables on the Local side to their correct output tables.

5. Route_Out: Written in PLZ Language. Routes data from the Local-to-Local and Network-to-Local tables to the correct output channel on the Local side.

6. Mov_Seq: Written in Assembly Language. Moves a sequence of bytes from one location in memory to another.

7. Ld_Tab_Hskp: Written in PLZ Language. Initializes a specified table after servicing any receiving data.

8. Det_Dest: Written in PLZ Language. Determines the proper destination of the data being processed.

9. Srvc_Tab_Hskp: Written in PLZ Language. Initializes a specified table after servicing data being transmitted.

10. Trnpkt: Written in PLZ Language. Sets up the data and port addresses for packet transmission to the Local side and drives transmission until all the data has been sent.

The interrelationship among these Local Operating System modules is shown in Figure 1-7 (Ref 18:64).

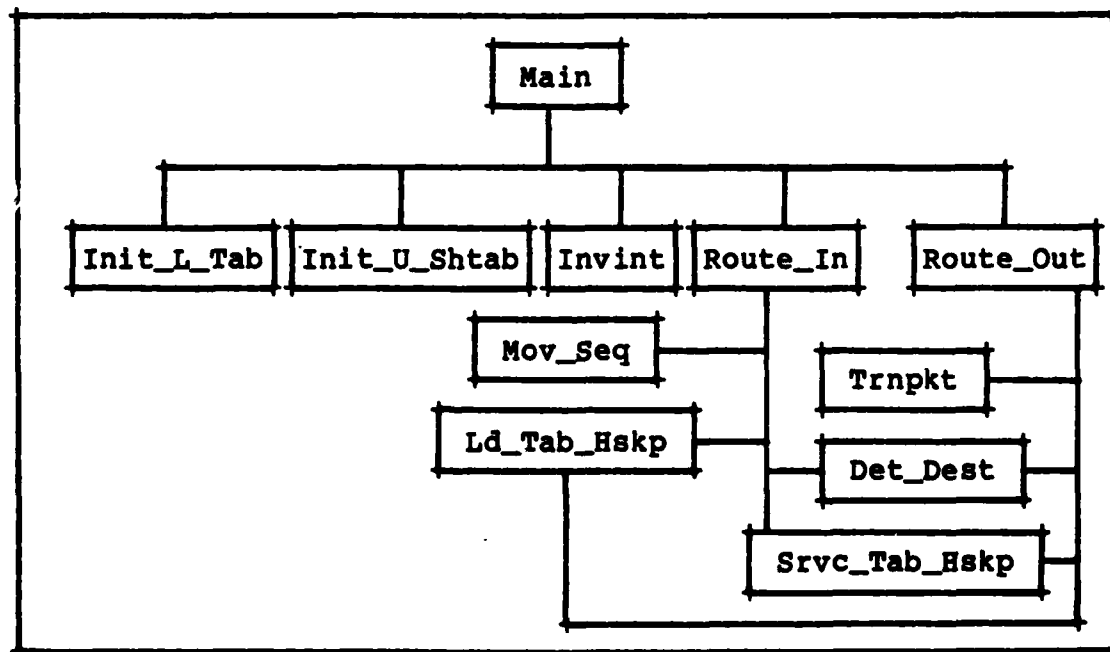


Figure 1-7. UNID Local Operating System Structure Chart

For the Local Operating System (L.OS) , the typical data block would flow from a local channel to an input table.

There it would be examined for destination and would be sent either to the network or back to the local channel. If sent to the network, the Network Operating System (N.OS) would accept the packet through the shared tables, located in the shared memory (Refer to Figure 1-4.), and process it. As with the L.OS, the Network Tables and procedures are described as follows:

1. Init_N_Tab: Written in Assembly Language. Initiates the data tables used by the NOS.

2. Insio: Written in PLZ Language. Initiates the serial port Input/Output process.

3. Route_In: Written in PLZ Language. Routes frames and packets from the network input tables to the correct output table.

4. Route_Out: Written in PLZ Language. Routes data from the Network-to-Network and Local-to-Network tables to the Network output channel.

5. Frame_Data: Written in PLZ Language. Establishes the correct frame around the data packets for network transmission.

6. Mov_Seq: Written in Assembly Language. Moves a sequence of bytes from one location in memory to another.

7. Ld_Tab_Hskp: Written in PLZ Language. Initializes a specified table after servicing a receiving packet.

8. Det_Dest: Written in PLZ Language. Determines the proper destination of a specified packet.

9. Srvc_Tab_Hskp: Written in PLZ Language. Initializes a specified table after servicing a transmitting packet.

10. Trnmit: Written in PLZ Language. Sets up the data and port addresses for frame transmission to the Network and drives transmission until an entire frame has been sent.

The interrelationship among these data tables is illustrated in Figure 1-8 (Ref 18:71).

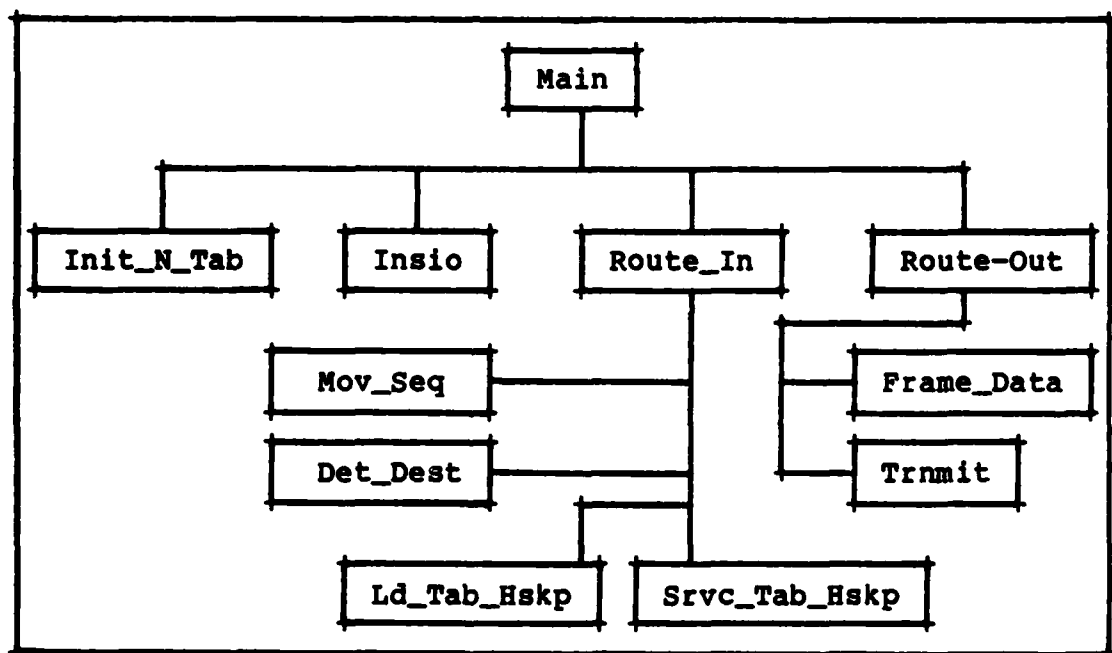


Figure 1-8. UNID Network Operating System Structure Chart

In the area of protocols, the ISO Reference Model has become the basic protocol model for the development of the UNIDs within the DELNET. Geist recommended that the next step in the development of the UNID be the incorporation of a dependable interrupt capability, including error recovery, at each potential host system. In late 1981, Papp developed extensive hardware that yielded two prototype processors

which were documented, tested, and operational (Refs 18:100-103 and 27).

UNID Status at Start of this Investigation

In 1982, Hazelton improved the first three protocol layers of the UNID as shown in Figure 1-6, which are (Ref 19):

1. Physical Layer. On the local side of the UNID, the data link uses the RS-232C standard. Only nine out of the available 25 pins are used. The data transfer from the host to the UNID is serial at 19.2 Kilobits per second. On the network side, although the ports are not currently installed, the data flow will be serial using a modified RS-449 standard and a fiber optic link at Two Megabits per second (Ref 27:93-94).

2. Data Link Layer. The data link layer is governed specifically by the CCITT standard for Link Access Procedures (LAPB). This standard specifies the frame structure as shown in Figure 1-9 (Ref 16:429-432).

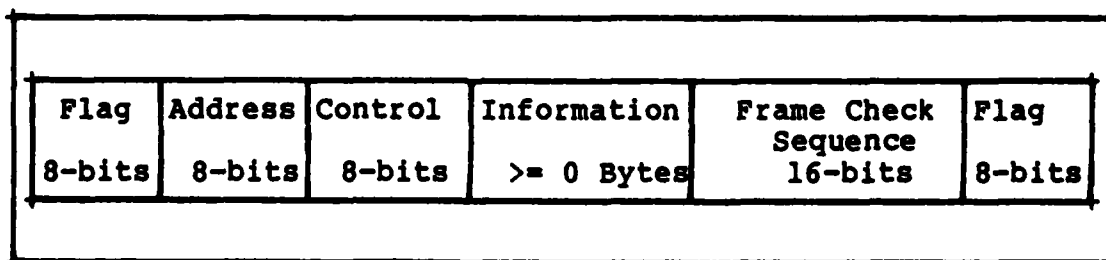


Figure 1-9. Link Access Procedure Frame Format

There are two types of frames used in the UNID. The first is an Information Frame (I-Frame). This contains the information that is to be routed. The second frame is the Supervisory Frame (S-Frame) which, currently, is for book-

keeping only and informs the sending UNID that the receiving UNID has received the frame that was sent. Figure 1-10 illustrates the current DELNET Frame Format (Ref 19:4-6).

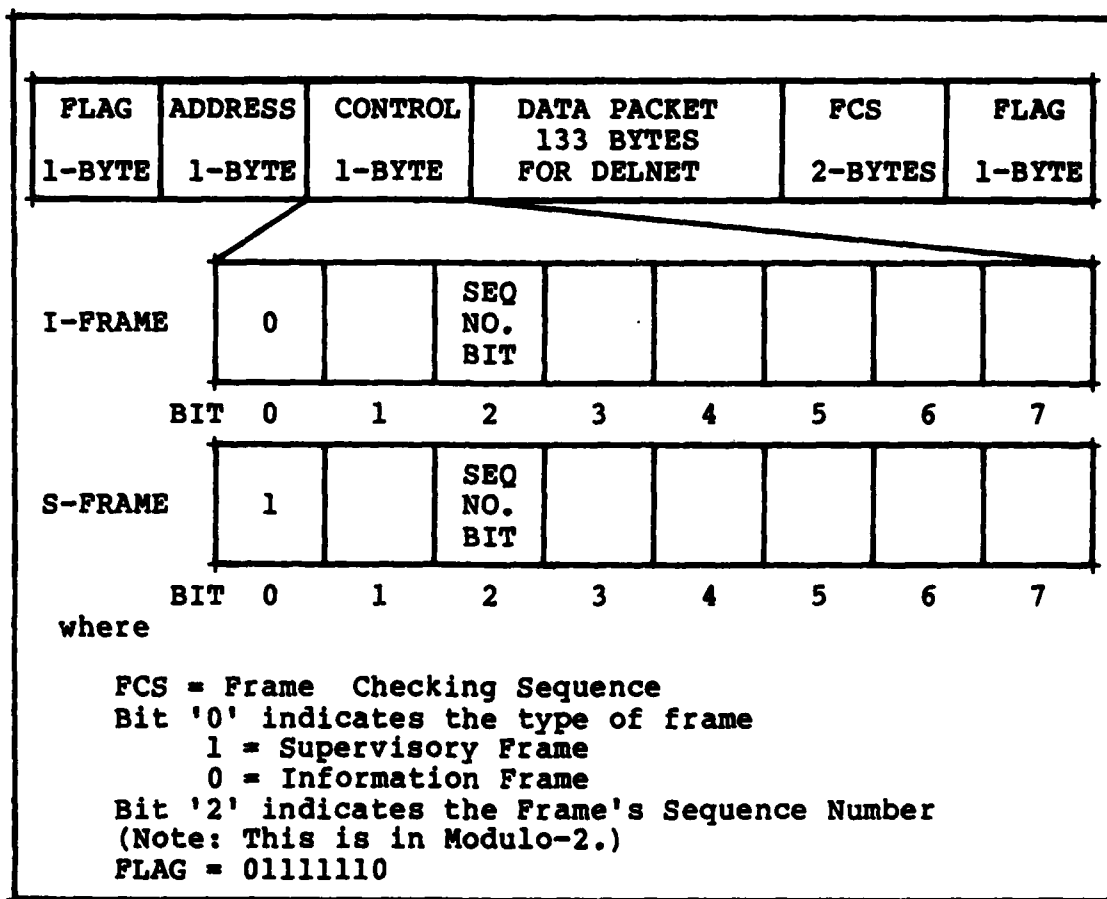


Figure 1-10. DELNET Frame Format

The software developed in 1981 by Geist gave an excellent foundation for the Network Layer software for the UNID. This software has tables and pointers which are required to maintain the various UNID internal routings of the packets, shown in Figures 1-7 and 1-8. In 1982, Hazelton implemented the previously designed software by augmenting the header information scheme and by processing the data flow accordingly.

3. Network Layer. Since the routing of the DELNET's packets is in a ring configuration, the routing algorithm in the network layer is simplified. Packets simply travel unidirectional within the ring and are withdrawn from the ring when they get to the destination UNID. Figure 1-11 gives the Network Layer's view of the Data Frame (Refs 18 and 19:5-5). Also illustrated in Figure 1-11 are the header locations for the upper four protocol layers as specified by the ISO Reference Model and will be discussed in Chapter 2 (Ref 26).

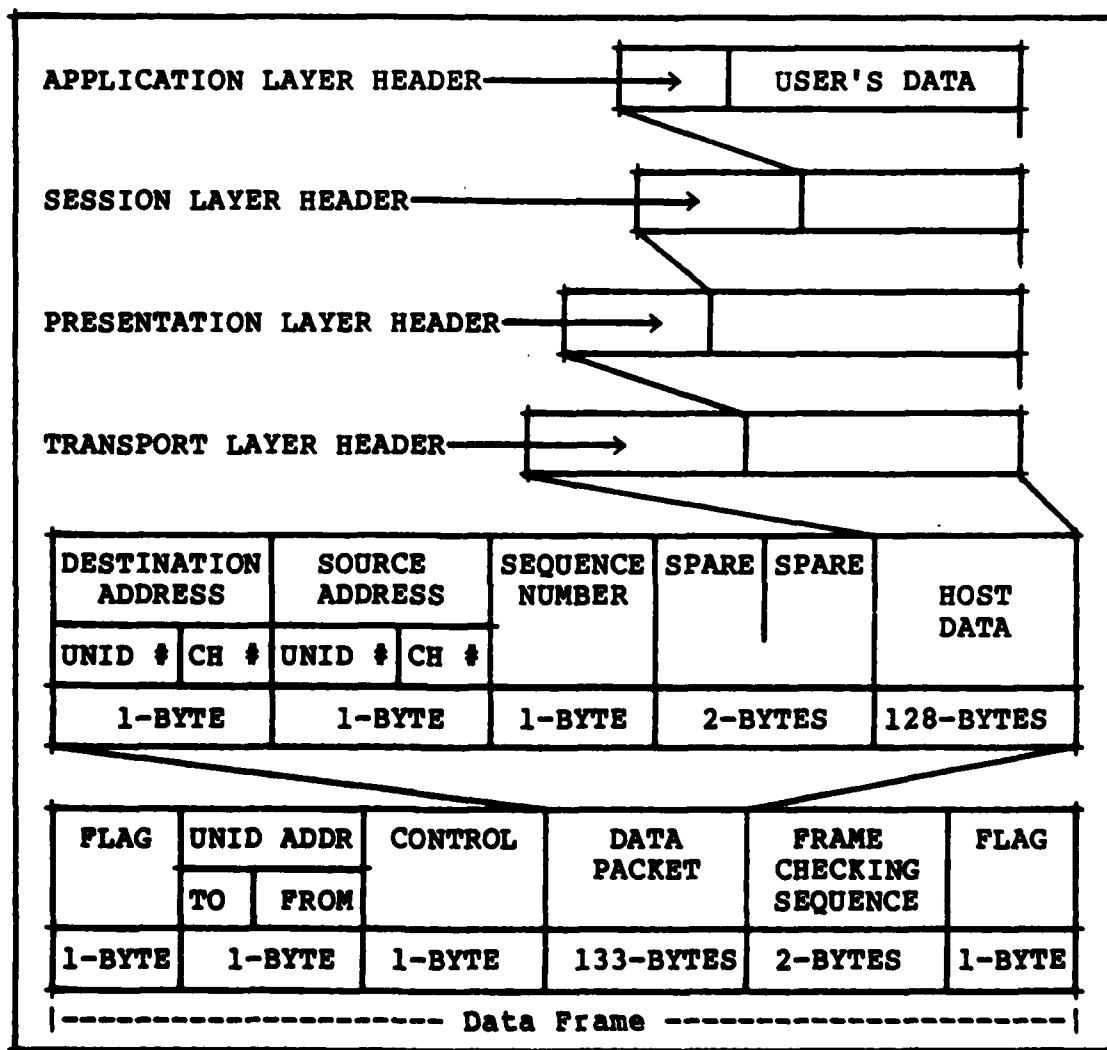


Figure 1-11. DELNET Data Packet Format

Although Hazelton did substantial work in the first three layers of protocol, X.25 implementation has a long way to go. In fact, only about 5% of X.25 at the start of this investigation has been incorporated into the UNID. This is, however, enough to pass a packet (in Datagram form) through the UNID so that the upper protocol layers can be implemented.

Hazelton's recommendations that directly relate to this thesis effort are (Ref 18:7.4-7.7):

1. Complete the initial three levels of protocol to fully implement X.25 Protocol;
2. Develop the successive higher levels of protocols;
3. Develop a real-time network monitoring system which can provide network analysis and evaluation.

A basic conclusion of Hazelton's study was that, while improvements to the present DELNET protocol levels are important, development of the higher layers of protocol is essential in order to obtain a working DELNET in the foreseeable future (Ref 18:7.4-7.7).

UNID Design Goal

The UNID has advanced significantly since its conception in 1977; however, considerable effort is required to achieve a "true" ring network. The design goal of the UNID can be considered in two aspects, hardware and software.

Hardware Design Goal

The hardware design goal is to have a full

duplex dual-ring network interfacing all the Local Area Networks operating within the Digital Engineering Laboratory, which includes the following major systems; Zilog MCZ 1/25, LSI-11, TT-10, Eclipse B-29, and VAX 11-780. The UNIDs will allow connection of up to sixty-four hosts within each Local Area Network under the normal addressing scheme and up to 4,096 using extended addressing. Figure 1-12 gives a conceptual view of the overall hardware topology.

Although the initial design was for a single ring configuration (1977), the UNIDs were upgraded to a dual ring configuration (1983). The only change was in Routing Control. With the dual ring, a simplified routing scheme was developed to determine the shortest path between two UNIDs (1983). Essentially this meant Route_Left or Route_Right. The dual ring is also illustrated in Figure 1-12.

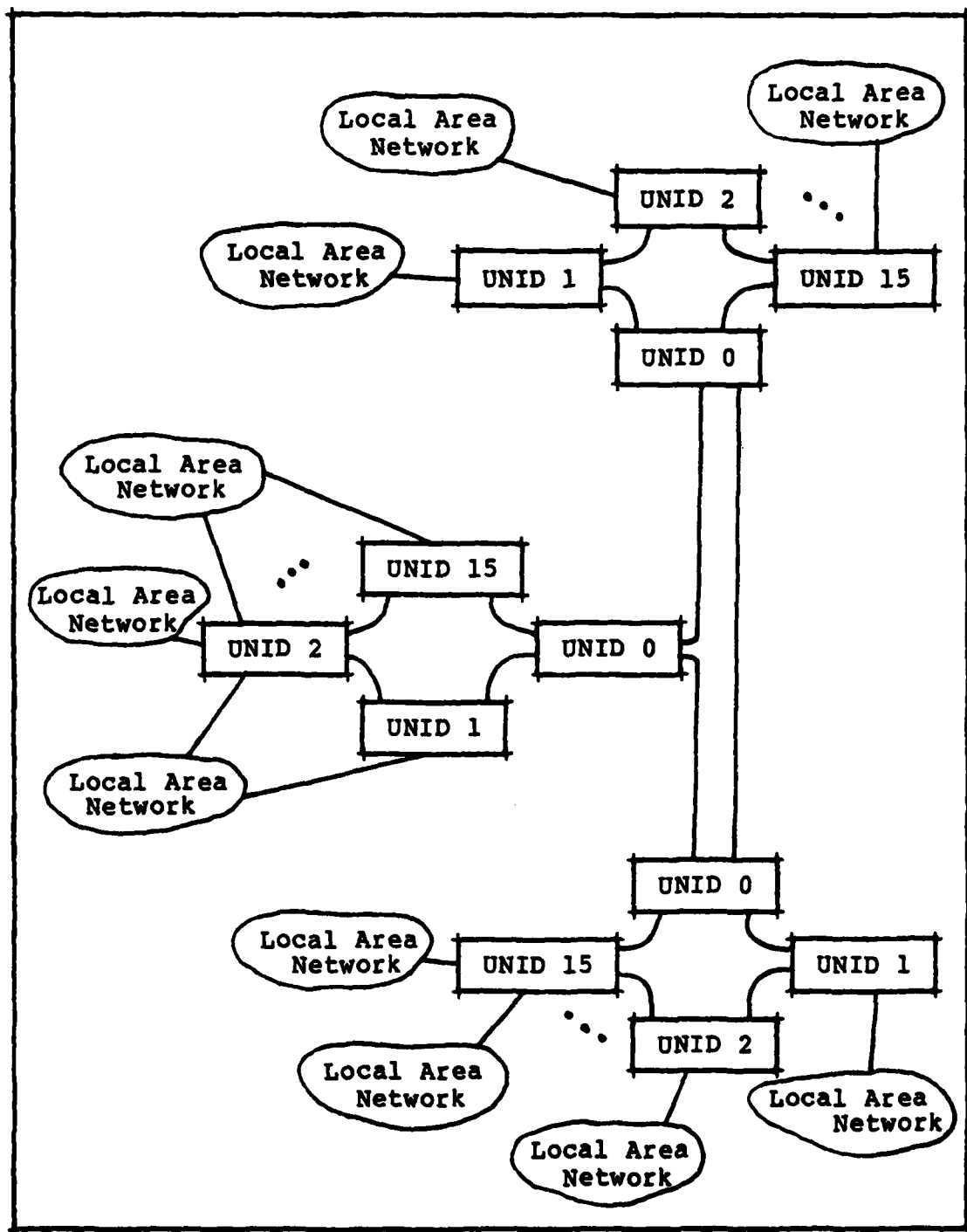


Figure 1-12. DELNET Hardware Topology

Software Design Goal

The software goal is to implement fully the Reference Model of the Open Systems Interconnection (Refs 10,26,28, and 48). This is illustrated in Figure 1-13.

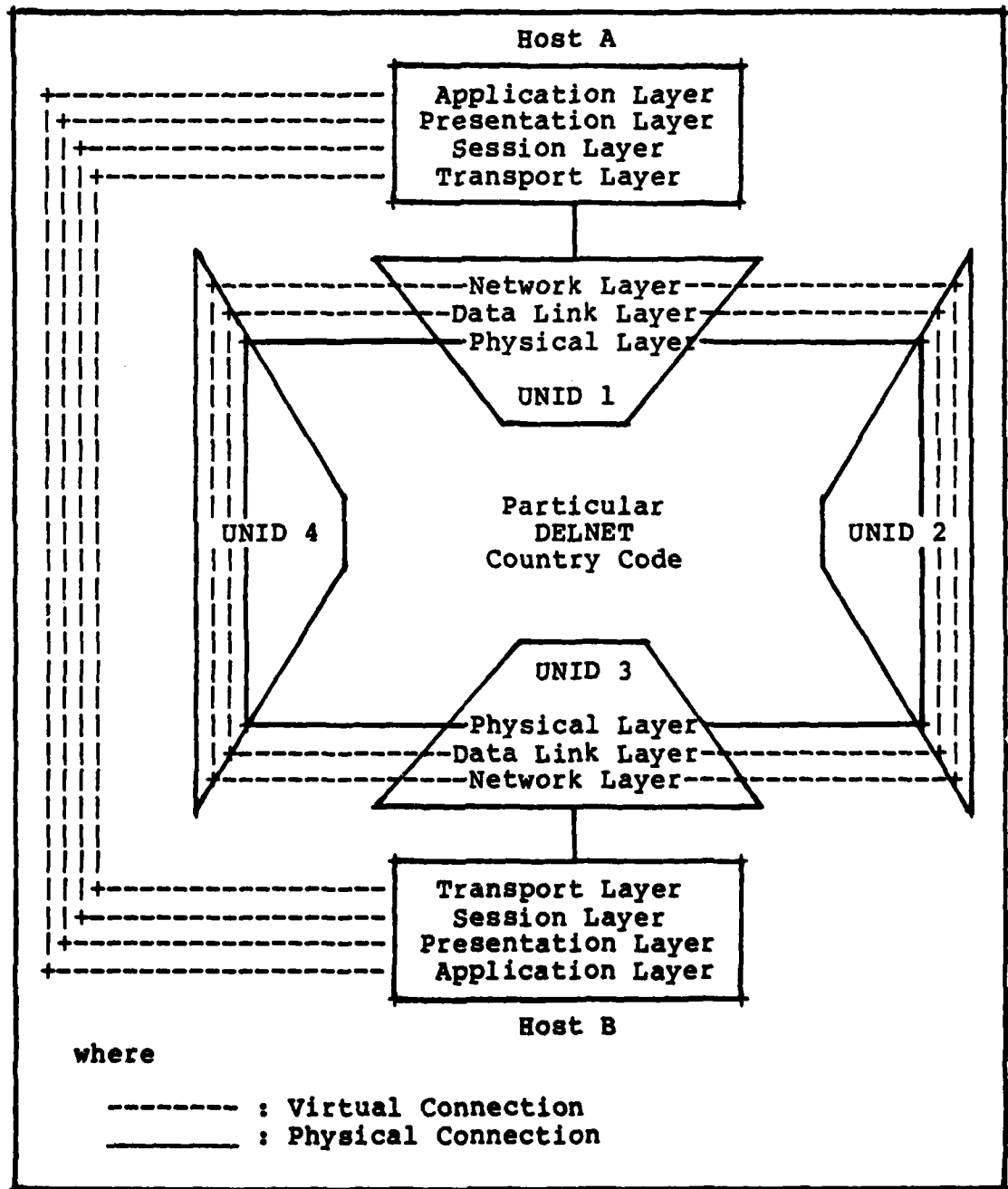


Figure 1-13. DELNET Software Topology

Overview of the Thesis

Problem

Although the UNID currently does not have all the capabilities of X.25 in its first three protocol layers (i.e. Physical, Data Link, and Network), it was decided to first make the UNID "minimally" operational in the AFIT DELNET, then later make additional software improvements to enhance the network. The term "minimally" referred to the ability to send a test message from the Zilog MCZ 1/25 H-19 Terminal (Port 1) through the UNID and back from the UNID to the Zilog MCZ 1/25 H-19 Terminal (Port 2) without an error.

The purpose of this thesis effort was threefold: first, to develop the protocols required for the AFIT DELNET implementation; and second, to implement a Network Layer within the UNID that would follow the determined protocol standards. This included End-to-End protocol design, implementation, and testing of the appropriate protocol layers. The last was to develop an initial test plan to be used to test the DELNET software.

Scope

This investigation designed, implemented, and tested software algorithms to incorporate into the AFIT DELNET, particularly the Network and Data Link Layer protocols as specified in the Open Systems Reference Model, DP-7498 dated 6 August 1981, using the Zilog MCZ 1/25 Computer system (Ref 26).

Assumptions

Two assumptions were made at the outset of this effort:

1. Algorithms currently operational within the first three protocol layers perform as specified (Ref 7).
2. UNID hardware works correctly.

Standards

The following standards were followed:

1. ISO Open Systems Reference Model DP-7498.
2. CCITT X.1, X.2 and X.95 for Class of Service.
2. CCITT X.121 for routing control.
3. CCITT X.25 for network control.
4. ISO3309-1976(E) for Data Link Control.
5. HDLC.
6. CCITT LAPB.
7. RS-232, RS-422/RS-423, RS-449 at Physical Layer.

Approach

This investigation was accomplished in three stages. The first stage was to determine the necessary protocol requirements as specified in the Open Systems Reference Model for the Network, Transport, Session, Presentation, and Application Layer protocols for incorporation into the AFIT DELNET. This also included a detailed literature search for methods of designing and testing required software algorithms. The second stage was to design a network standard that all AFIT computer systems would follow in order to be connected into the AFIT DELNET. The third and most important

stage was to modify the Network Layer protocol to handle routing of packets from one Local Area to another; write a UNID user's guide to include the cable connections and special operating instructions; and upgrade existing software for easier debugging and testing.

Equipment

The following equipment was required:

1. One Heathkit H-19 Terminal.
2. Two prototype UNIDs.
3. One ZILOG MCZ Dual Disc Computer System.
4. One Spinwriter Printer.
5. Four M-3A Terminals used as monitors.
6. Associated RS-232C interconnecting cables.
7. Associated RS-449 interconnecting cables.

II. DELNET PROTOCOL REQUIREMENTS

Introduction

Within the DELNET, the UNIDs will act as gateways to deliver error free messages to the Local Area Network (LAN) servicing the destination Host (Ref 49:354-358). This effort concentrated on the protocol specification (See Appendix C) and software implementation (Chapter 4) based on the ISO Reference Model (Ref 26). The three principle aspects of any protocol specification within a node are structure, interface, and function. They are defined as (Ref 28:114):

1. Structure - the identification and naming of the basic components of the architecture, the specification of the valid interconnections between components, and the identification and naming of major compound subsystems;

2. Interface - the specification of generic signals exchanged between interconnected components and, where appropriate, the specification of signal format, including electrical, mechanical, and timing aspects; and

3. Function - the closed form specification of behavior for the basic components; in computer networks these basic components are almost always combinational functions or discrete state machines.

The overall structure of a node using the above concepts is shown in Figure 2-1 (Ref 28:115). Each node is divided into layers, called N-Layers (See Appendix B), which consist of a set of layer Ports, one or more layer Routers, and a layer Manager. The general purpose of a layer Port is to

serve as an end-point for a connection between itself and another Peer Port with which it exchanges control and data messages via the services provided by the lower layers.

Each Port has an associated layer Port Address and supports end-to-end protocols of the following kinds: data transfer with send/receive acknowledgement; queueing; sequencing; segmenting; blocking; contention resolution; flow control; translation; expedited and normal flows; error detection; and error recovery. Layering of the network and the division of each Node Layer into a control Manager and data Ports induces a natural layering between nodes as shown in Figure 2-2 (Ref 28:116). For this effort the layering followed the ISO Model as discussed in Chapter 1.

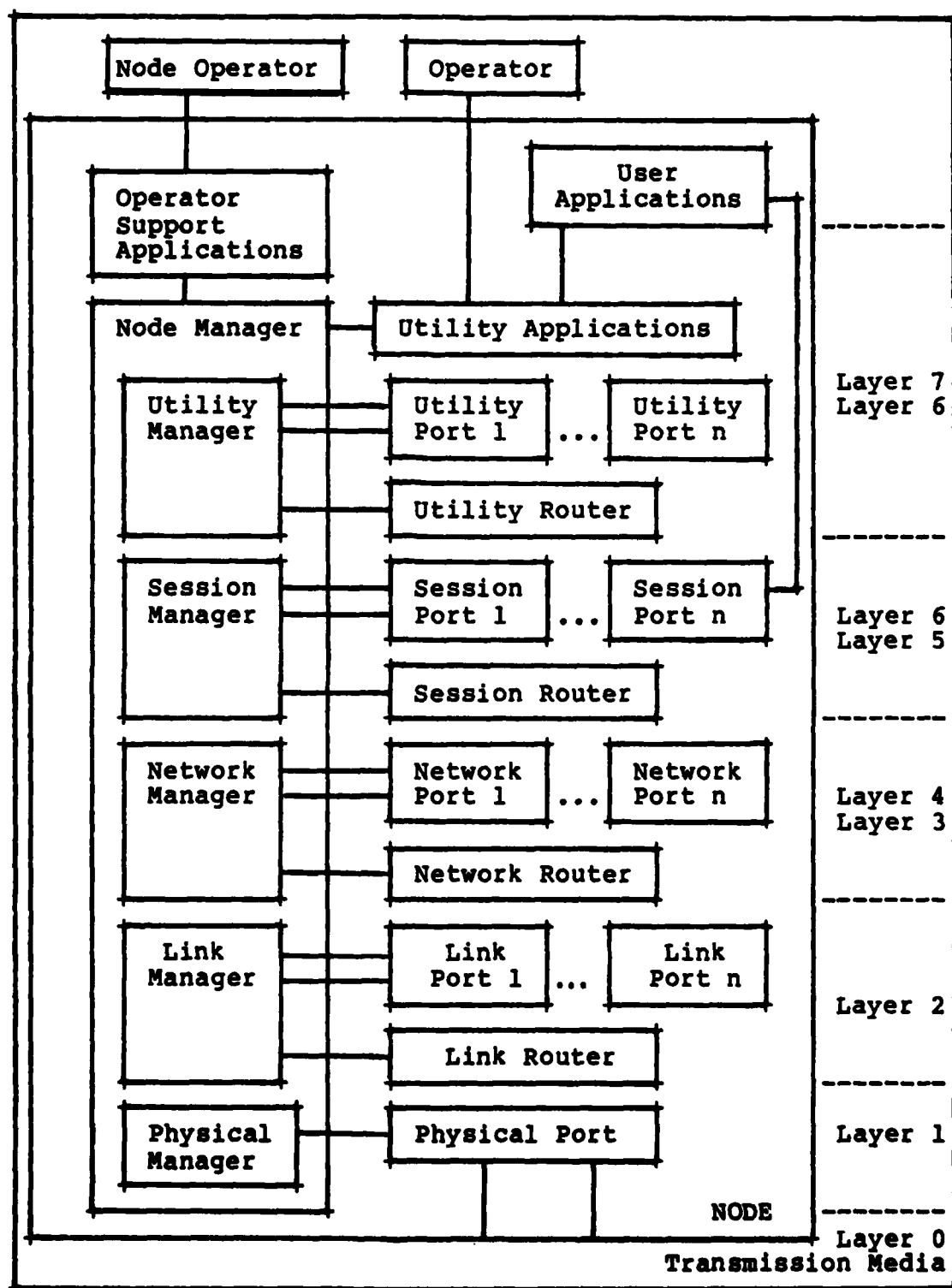


Figure 2-1. Overview Structure of a Node

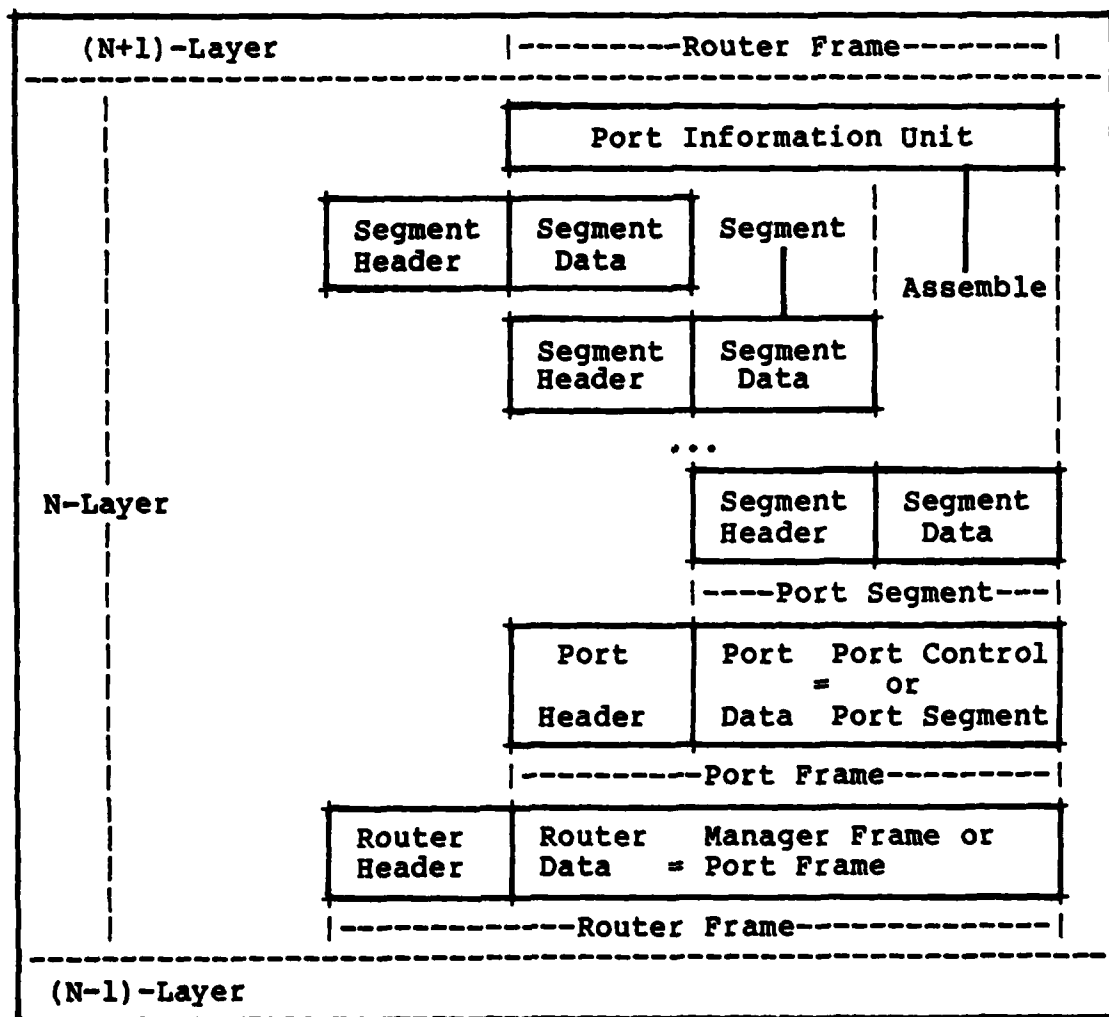


Figure 2-2. General Message Format

The rest of this chapter addresses each of the seven layers of the ISO Model in relation to the DELNET using Piatkowski's guidelines for protocol specification. All requirements will lean toward providing fast and error free service to each Local Area Network serviced by a particular UNID.

Physical Layer Protocol

Overview of the Physical Layer

The Physical Layer provides mechanical, electrical, functional and procedural capabilities to activate, maintain and deactivate physical connections for bit transmission between data-link-entities, possibly through intermediate systems, with each relaying bit transmission within the Physical Layer.

The Physical Layer provides the following services or elements of services: physical-connections; physical-service-data-units; physical-connection-endpoints; data-circuit identification; fault condition notification; and quality of service parameters (Ref 26).

Physical Layer as Applied to the DELNET

At this stage in the UNID's development, both the Network and Local Ports had to be installed. Previous efforts established the standard to use (Refs 27 and 37), i.e., RS-232 on the Local Side and RS-449 on the Network Side. (See Chapter 6 for recommendations concerning Physical Layer enhancements). However, the actual connections on the outside of the UNID were never installed.

Also, little design or attention was paid to the Physical Layer on the Network Side of the UNID in past efforts. Initially, only Channel A of the Z80-SIO was activated for both send and receive. To incorporate the dual-ring concept as shown in Figure 1-12 both Channel A and Channel B must be operational.

Network Layer Protocol

Overview of the Network Layer

The Network Layer provides the means to establish, maintain and terminate network-connections between systems containing communicating application-entities and also the functional and procedural means to exchange network-service-data-units between transport-entities over network connections.

The Network Layer controls routing and relay considerations associated with the establishment and operation of a given network-connection, including the case where several transmission resources are used in tandem or in parallel.

Network Layer functions must cover the wide variety of configurations supporting network-connections ranging from network-connections supported by point-to-point configurations, to network-connections supported by complex combinations of subnetworks with different characteristics. According to the ISO Model, the Network Layer performs the following functions: routing and relaying; network-connections; network-connection multiplexing; segmenting and blocking; error detection; error recovery; sequencing; flow control; expedited data transfer; reset; service selection; and Network Layer Management (Ref 26).

Network Layer as Applied to the DELNET

As mentioned in Chapter 1, the Network Layer was designed using the Datagram service option. Most of the

functions mentioned above are not operational within the Network Layer of the UNID. This effort was to get the DELNET operational on a "minimal" basis, i.e., have it transport packets to servicing Local Area Networks (LANs). Therefore, only the routing and relaying functions of the Network Layer were addressed.

To ensure flexibility and ease of future equipment additions, CCITT X.121 routing standard was used within the DELNET (Ref 16). According to CCITT X.121, there are fourteen decimal digits allotted to the address function as shown in Figure 2-3.

Country			Network	Host								Port		
1	2	3		4	5	6	7	8	9	10	11	12	13	14

Figure 2-3. CCITT X.121 Addressing Scheme

Instead of fourteen decimal digits representing the entire address of either source or corresponding destination, the DELNET will use the terms Country Code (CC), Network Code (NC) (where Network Code will equal a UNID number), Host Code (HC), and Port Code (PC) within a 32-bit address word (See Appendix C Network Layer for complete discussion). Figure 2-4 illustrates a particular part of the DELNET with its address relationships. Figure 2-4 also indicates extended addressing capability associated with CCITT X.121. (This is discussed further in the Transport Layer section of this chapter).

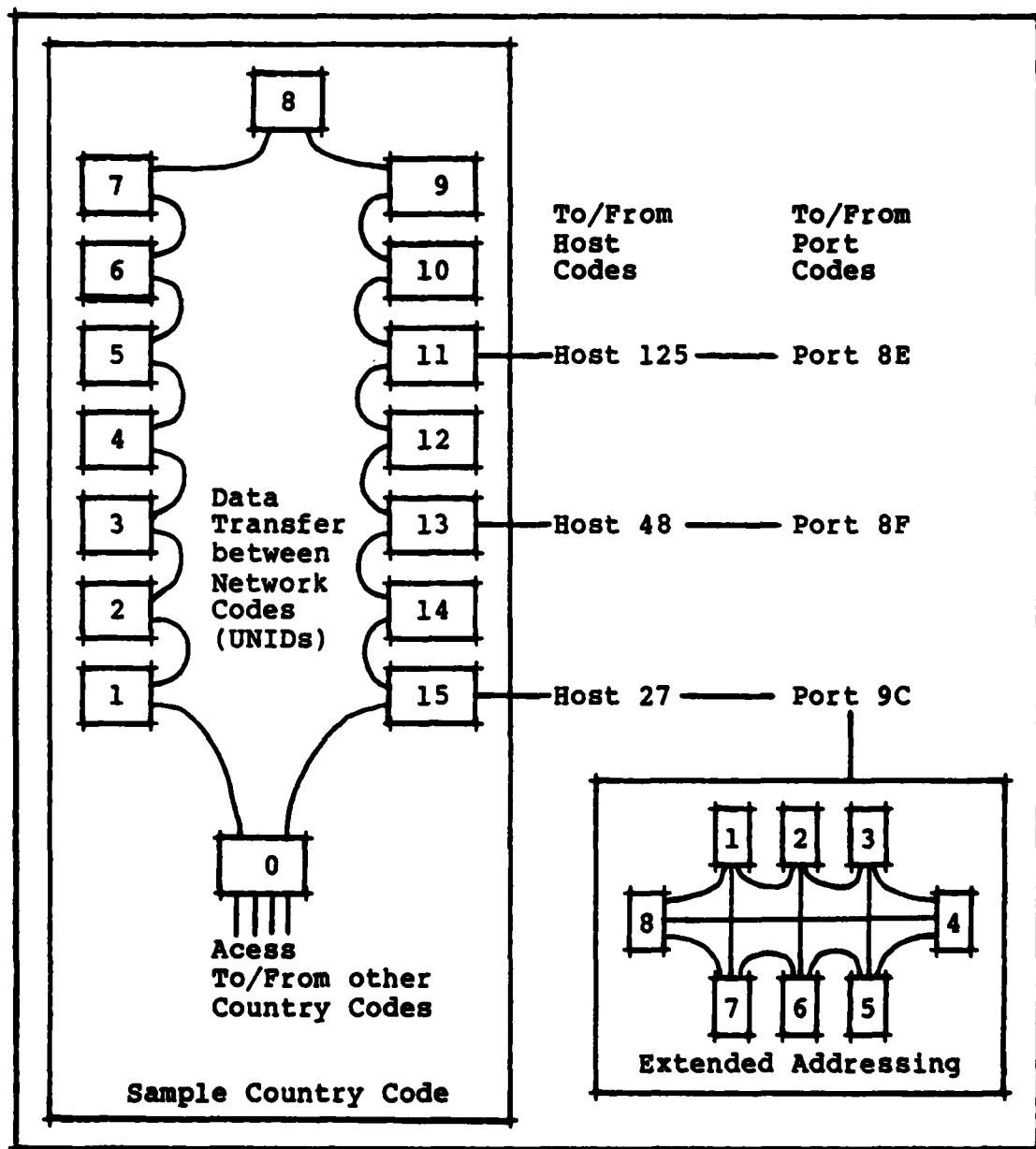


Figure 2-4. Sample DELNET Addressing Relationships

To minimize the response time to the user, the Network Layer will route the messages by first looking at the Country Code then at the Network Code to see if the outgoing message is destined for the Network. If not, then the Host Code will indicate which local port the message is destined to reach.

If the message is destined for the Network Side of the

UNID, then the Country Code is examined to see if the message is destined for another Country within the DELNET. If so, then the servicing UNID will route the message to UNID_0.

UNID_0 will play a special role within the DELNET. This UNID will have RS-449 Ports on both Local and Network Sides. On the Country Side it will be responsible for Flow Control, initial start-up, and system resets, among other things. Chapter 6 addresses UNID_0 concerning recommendations for further theses.

On the Inter-Country Side, UNID_0 will be responsible for routing the message to the correct Country within DELNET. This side of the UNID will use a distributed routing scheme (See Chapter 6) to ensure that the message is delivered in the minimum length of time.

Figure 2-4 above illustrates only one Country Code. Figure 2-5 below gives a possible DELNET topology using the CCITT X.121 routing standard for six Country Codes with only eight UNIDs connected per Country Code. This figure also demonstrates the unique role of UNID_0.

The illustration in Figure 2-5 is not representative of the total connectability of the DELNET. In its total configuration, there will be 16 Country Codes, 16 Networks (UNIDs) per Country Code, and 256 Hosts per UNID. This will give DELNET the capability of interconnecting a total of 65,536 Hosts. If each Host had the capability of using the extended addressing mode with one of its ports and had 16 sub-Hosts connected, then the total number of Hosts within

the DELNET would be 1,048,576.

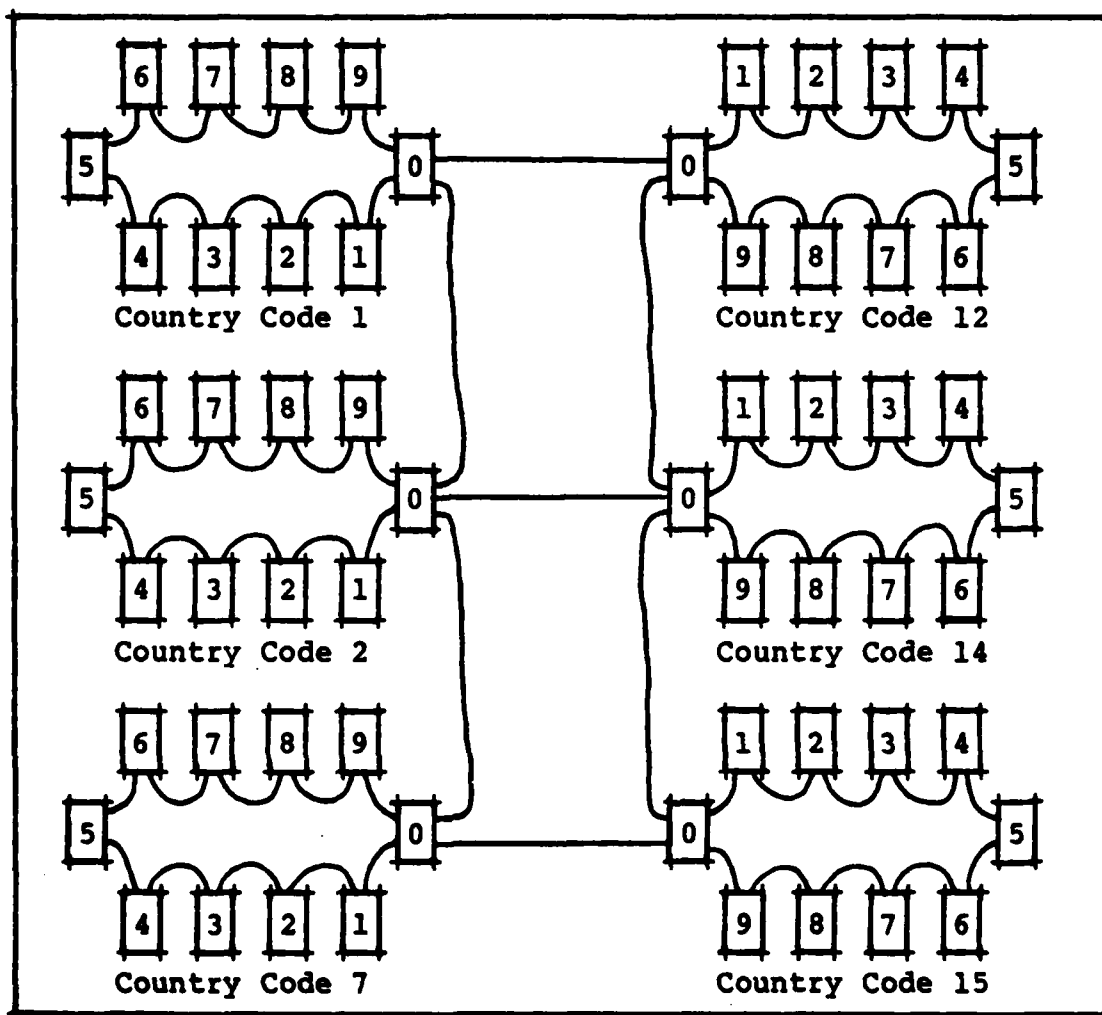


Figure 2-5. Sample DELNET Topology

Transport Layer Protocol

Overview of the Transport Layer

Figure 2-6 illustrates the Transport Layer and its relationship to the Session and Network Layers using N-Layer terminology (Refs 41,42,43,44,45, and 46). (See Appendix B for a general discussion concerning the N-Layer concept).

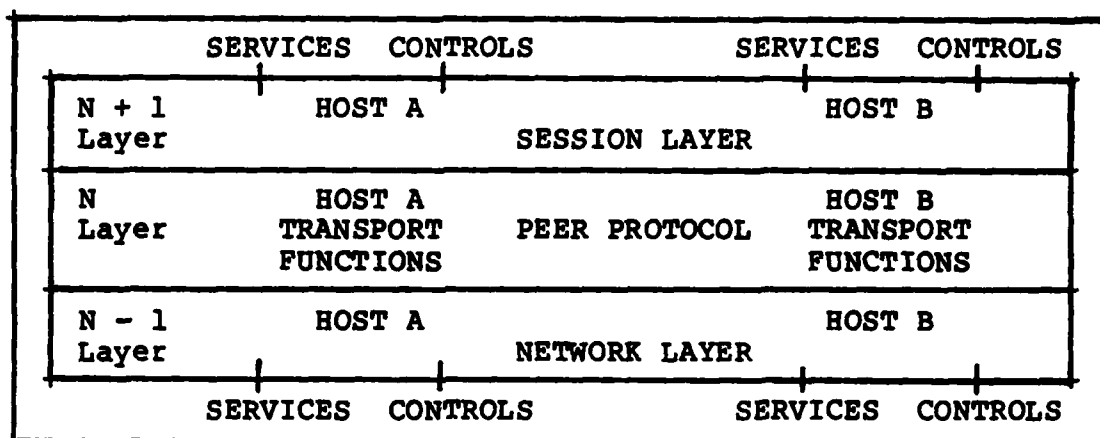


Figure 2-6. Transport Layer Relationships

The Transport Layer is the highest layer associated with the "providers" of communication service. The Transport Layer provides a transparent universal data transfer mechanism to the higher layers which, in general, represent the "users" of the communication service. The Transport Layer exists to provide the transport-service associated with the underlying services provided by the supporting layers. Briefly, the Transport Layer performs the following functions: provides transparent transfer of data between session-entities; relieves the transport users from any concern with the detailed way in which reliable and cost effective transfer of data is achieved; and optimizes the use of the available network-service to provide the performance required by each communicating transport user at minimum cost. This optimization will be achieved within the constraints imposed by considering the global demands of all concurrent transport users and the overall quality and capacity of the network-service available to the Transport Layer. Since the network-service provides network-connections from any transport-

entity to any other (including the use of tandem subnetworks), and relieves the Transport Layer of any concern with switching, routing, and relaying, all protocols defined in the Transport Layer will have end-to-end significance, where the ends are defined as the correspondent transport-entities. In other words, the Transport Layer is end-system oriented and transport protocols operate only between corresponding end-systems.

The Transport Layer also uniquely identifies each of its users by its transport-address. Users of the Transport Layer are provided with the means to establish, maintain, and release transport-connections which represent a two-way simultaneous data path between a pair of transport-addresses. More than one transport-connection can be established between the same pair of transport-addresses; the means by which the user can distinguish between the transport-connection-end-points will be provided by the Transport Layer, in terms of transport-connection-end-point-identifiers.

Lastly, the Transport Layer performs all those functions which are needed by the Session Layer (See Figure 2-6) that are not provided by the Network Layer. A range of end-to-end services are possible from the simplest, which provide only an addressing structure useful to support routing, but which provide no guarantee on information delivery or flow control, to the most complex, which offer guarantees about correct delivery, flow control and other services (Refs 26,52).

The ISO Model describes three phases of operation within

the Transport Layer, which are: establishment; data transfer; and release (Ref 26:77). The goal of the establishment phase is to establish a transport-connection between two transport users. Transport-connections are established to a peer transport address for users of the transport-service. The functions of the Transport Layer during this phase must match the requested class of services provided by the Network Layer. These classes of service represent globally predefined combinations of parameters and the grades of service to be provided. These service classes are intended to cover the transport-service requirements of the various types of traffic generated by the session-entities. These service classes are characterized by selected values of various parameter combinations such as throughput, transit delay and connection set-up delay and by various guaranteed values of parameters which are related to the residual error rate and service availability.

There are numerous other functions performed during the establishment phase. One of the first functions performed is the selection of Network-service which best matches the requirements of the connection. This selection takes into account the requested class of service.

The decision about whether to multiplex transport-connections onto a single network-connection is another function performed in the establishment phase (Ref 26:36). Another consideration within the establishment phase is the creation of an optimum transport-protocol-data-unit size.

An important function performed in the establishment phase is that of mapping transport-addresses onto network-addresses. This is performed by an (N)-address-mapping function (Ref 26:76). There are two different kinds of (N)-address-mapping functions that can exist within the transport layer; hierarchical (N)-address-mapping and (N)-address-mapping by tables. If an (N)-address is always mapped into only one (N-1)-address, then hierarchical construction of addresses can be used such that the (N)-address is made of an (N-1)-address and an (N)-suffix. In this case, the (N)-address-mapping function simply consists of recognition of the hierarchical structure of the (N)-address and extraction of the (N-1)-address it contains while (N)-address-mapping by tables is simply a table lookup for the desired address or addresses.

This hierarchical structure of addresses within a given layer simplifies (N)-address-mapping functions within that layer because of the permanent nature of the mapping it presupposes. It is imposed by the Basic Reference Model only in the N-Layer in order to allow more flexibility in (N)-address-mappings and to provide for the case where one (N)-entity attached to more than one (N-1)-service-access-point supports only one (N)-service-access-point identified by an (N)-address.

During the establishment phase, there must be means to distinguish different transport connections between the same pair of transport-service-access-points (connect ID). Lastly,

the establishment phase sets up the transport of users' data.

The second phase of operation within the Transport Layer is the data transfer phase (Ref 26:78). After the establishment phase is completed the data transfer phase begins. The purpose of the data transfer phase is to transfer transport-service-data-units between the two session-entities connected by the transport-connection. This purpose is achieved by means of the transmission of transport-protocol-data-units and by the following functions, each of which is used or not used in accordance with the requested class of service selected in the establishment phase; sequencing, blocking, concatenation, segmenting, multiplexing or splitting, flow control, error detection and/or correction, and error recovery. Within the data transfer phase, the network must also be able to handle the transfer of expedited data. Therefore the Transport Layer must be able to alert the system that this packet takes precedence over other packets.

The third and last phase performed within the Transport Layer is the release phase (Ref 26:79). The purpose of the release phase is to release the transport-connection and may include the following functions: notification of reason for release; identification of the transport-connection released; and possible additional information as required.

Transport Layer as Applied to the DELNET

The Transport Layer within the DELNET will follow two standards, one for Datagram service and the other for Virtual Circuit connection.

Considering Datagram service first, the Transport Layer will follow the Transport Control Protocol (TCP) which is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols (e.g. ISO Model) which supports multi-network applications (Refs 32,41,42,43, 44, 45, and 46). The TCP provides reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks (Refs 32:1-2 and 9:3-13). The TCP interfaces on one side to user or to application processes and on the other side to lower level protocols. Within the DELNET, the TCP will be the lowest level within the Host interfacing with the Network Layer in the UNID. Attached to the TCP header is the Internet Header Format (IHF) which makes up the Datagram Header. The use of TCP requires the addition of the IHF; however, the IHF could reside either in the Host or the UNID. Within the DELNET, the IHF will reside in the Host, for two reasons. First, it allows the UNIDs to operate in both Datagram and Virtual Circuit mode with little change on either end. Secondly, with the IHF residing in the Host, it can operate in the same manner as the Network Layer in the UNID, enabling the Host to forward the message to the appropriate destination Host with minimal transfer time.

The second protocol standard, which the Transport Layer will follow, when the DELNET is operating in the Virtual Circuit Mode, is the Federal Information Processing Standard (FIPS) (Refs 30,40,41,42,43,44,45, and 46). The reason for

this selection is twofold. First, it can be obtained from the National Bureau of Standards (NBS) and run on a VAX using UNIX. Secondly, if AFIT can create a fully operational UNID, using this protocol, then the NBS will certify it.

The DELNET will not operate in the Virtual Circuit Mode of operation at this time. This is because the X.25 standard has not been fully implemented in the UNIDs. Currently, as mentioned in Chapter 1, only Datagram service has been partially implemented; therefore, the first implementation of the Transport Layer within the DELNET will be the TCP plus IHF protocol.

One of the major thrusts of this effort was to begin to define the protocol standards to be used within the DELNET. Therefore, Appendix C contains both the TCP and FIPS protocol standard headers to provide a basis for a complete protocol standard to be developed for use within the DELNET.

Session Layer Protocol

Overview of the Session Layer

The services of any protocol layer are the capabilities which it offers to a user in the next higher protocol layer. In order to provide its service, a protocol layer builds its functions on the services which it requires from the next lower protocol layer. Figure 2-7 illustrates this notion of service hierarchy and shows the relationship of two correspondent N-Layer users and their associated N-Layer peer protocol entities. (See Appendix B for a general discussion concerning the N-Layer concept).

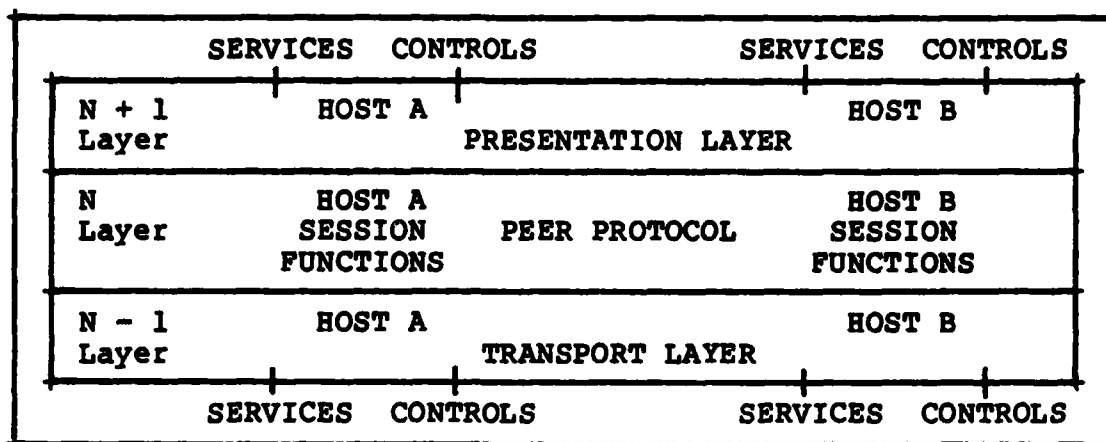


Figure 2-7. Session Layer Relationships

The purpose of the Session Layer is to manage data traffic between cooperating high-level protocol entities. Transfer of this data is known as a session. The Session Layer forwards requests for data transfer to the Transport Layer. To do this, the Session Layer establishes a session-connection between two presentation-entities and supports orderly data exchange interactions (Ref 26:63-64). The Session Layer controls the delivery of data to the high-level user (quarantining), controls the interactions between high-level users (dialogue management), and imposes a structure on the data it transfers (data delimiting) (Ref 6:37-39).

Quarantining occurs when a session-user desires to withhold some data from its correspondent for some length of time and this data is known as a quarantine-unit. The end of a quarantine-unit indicates that the quarantined data may be delivered to the receiving session-layer; thus, the quarantining facility may be used to force data delivery as well as to hold data back from delivery. The receiving session layer will not deliver any fragment of the

quarantine-unit to the session-user until all the data in that unit is available for delivery (Ref 6:40-41).

The Session Layer imposes a structure on the interaction between correspondents called a dialogue. There are three possible varieties of dialogue: two-way-simultaneous, two-way-alternate, and one-way (Ref 6:41). The information sent during a single turn is called an interaction-unit.

The Session Layer imposes a minimal level of structure on the flow of data in a session. Any imposition of structure upon a raw stream of data could be deferred by each protocol layer to the protocol layer above until ultimately the two highest protocol layers (Application and Presentation) inherit the responsibility. However, this is contrary to the principles of functional layering as illustrated in the ISO Model. The ISO Model attempts to find useful common functions and offer them as services at the lowest possible protocol layer. Within the Session Layer, this is handled by Session-Service-Data-Units (SSDU). The SSDU can be seen as a way for session-users to segment the data stream into sections, with the boundaries between them maintained across the network. It is the responsibility of the local user to identify the beginning and end of an SSDU for the Session Layer. The end result of the SSDU is that the local session-user has a certain amount of control over how data will be transferred across the remote session user interface (Ref 6:42-44).

A session is also a cooperative relationship between high-level protocol entities. The session passes through

three phases: establishment, data transfer, and termination. These phases are closely related to the phases within the Transport Layer discussed earlier. In the simplest case, in which the mapping between transport connections and sessions is one-to-one, the phases of the session would be equivalent to the phases of the transport connection. However, more complex relationships are possible; for example, a session might employ more than one transport connection or several sessions might employ the same transport connection.

Establishment of a session will include the establishment of at least one transport-connection with an appropriate type and grade of transport service and appropriate security authentication (Ref 6:38).

Data Transfer within a session is controlled by the Transport Layer.

Termination of a session will dissolve the cooperative relationship between session-users in an orderly way, so that no data is lost and the resources of the distributed system are released appropriately (Ref 6:39).

To summarize, the Session Layer provides the following services to the Presentation Layer (Ref 26:63-71): session-connection establishment; session-connection release; normal data exchange; quarantine service; expedited data exchange; interaction management; session-connection synchronization; and lastly, exception reporting.

Session Layer as Applied to the DELNET

As mentioned in the Transport Layer, the NBS protocol specifications (TCP and TPDU) were selected. Therefore, within the Session Layer, the FIPS Session Layer Protocol, called SPDU, will be used for the DELNET (Ref 7:108-118). This protocol is compatible with both TCP and TPDU and will provide flexibility in both design and future enhancements which will be incorporated within the DELNET.

As mentioned earlier in this thesis, the NBS has already implemented the upper four protocol layers on a VAX using the forementioned protocol structures. Incorporation of these standards will at least provide LAN connection to the UNIDS.

Presentation Layer Protocol

Overview of the Presentation Layer

Figure 2-8 gives the N-Layer relationship of the Presentation Layer to its neighboring protocol layers. (See Appendix B for a general discussion of the N-Layer concept).

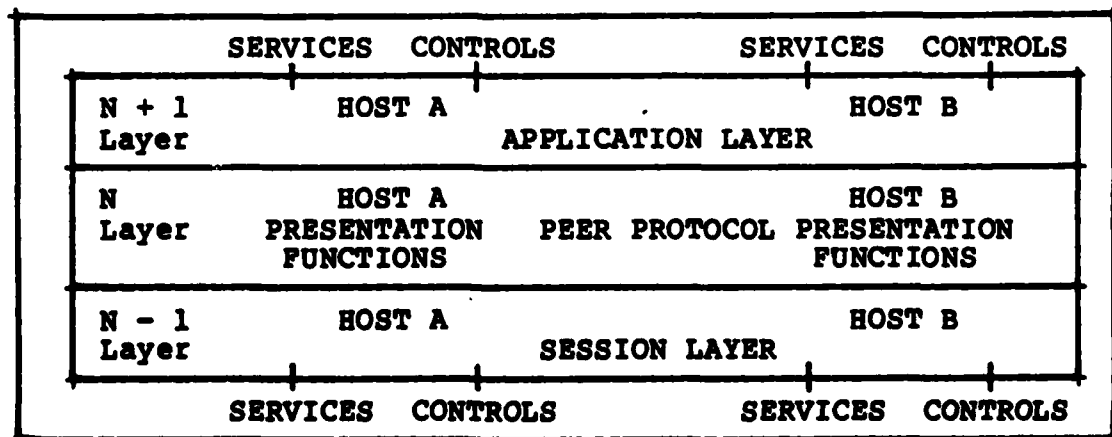


Figure 2-8. Presentation Layer Relationships

The Presentation Layer provides for the representation of information that application-entities either communicate

or refer to in their dialogue. The Presentation Layer covers two complementary aspects of this representation of information: the data syntax which is the representation of data to be transferred between application-entities; and the presentation-image-syntax which is the representation of the data structure which application-entities refer to in their dialogue, along with the representations of actions which may be performed on this data structure (Ref 26:58-62).

The Presentation Layer is concerned only with the syntactic view of the presentation image and of transferred data and not with its semantics. The Presentation Layer provides for a common representation to be used between application-entities. This relieves application-entities of any concern with the problem of "common" representation of information, i.e., it provides them with syntax independence. This syntax independence can be described in two ways: first, the Presentation Layer provides common syntactical elements which are used by application-entities; and second, it also provides the application-entities the transformation required between these syntaxes and the common syntax needed for correct communication between application-entities (Ref 26:59-60).

The Presentation Layer performs the following functions, according to the ISO Model (Ref 26:60): session establishment request; data transfer; negotiation and renegotiation of data syntax and presentation-image-syntax; transformation of data syntax and presentation-image-syntax, including data

transformation and formatting and special purpose transformations (e.g. compression); and lastly, session termination request.

To summarize, the Presentation Layer provides the following services to the Application Layer: transformation of data syntax (which is concerned primarily with code and character set conversions and with the modification of the layout of the data); transformation of presentation-image-syntax (which is concerned with the adaptation of presentation-data-syntax including adaptation of actions on the presentation-image); selection of data syntax (which provides the means of initially selecting a data syntax and subsequently modifying the selection); and selection of presentation-image-syntax (which provides the means for initially selecting/modifying the presentation-image-syntax).

Presentation Layer as Applied to the DELNET

Few protocol specifications are available for the Presentation Layer. This is because of the large number of variations possible within the Presentation Layer.

The traffic within the DELNET will use ASCII characters. The Presentation Layer will convert the data into the appropriate character set for the end-user's machine's operation. This will require that the end-user have only one conversion table.

The Presentation Layer will also be responsible for encryption/decryption of the data. This will not be specified but a 32-bit word will be reserved for future

expansion in this area. Another function that will be performed within the Presentation Layer is that of menu operation. The DELNET will be menu driven with the operator selecting desired options and values.

The last function of the Presentation Layer is conversion from a logical address to a 32-bit address to be used by the Transport Layer for source and destination addressing. This will allow the operator to select a logical address, which the DELNET will convert into its unique address. For example, if the operator wants to send data from the Zilog MCZ 1/25 Computer System located in the basement of Bldg 640 Rm 67 to an LSI-11 Computer System located on the second floor in Room 242 then the source logical address would be ZLG\$A\$CON (Console) and the destination logical address would be LSI\$A\$PRI (Printer).

Application Layer Protocol

Overview of the Application Layer

Figure 2-9 gives the N-Layer relationships of the Application Layer and its neighboring protocol layers.

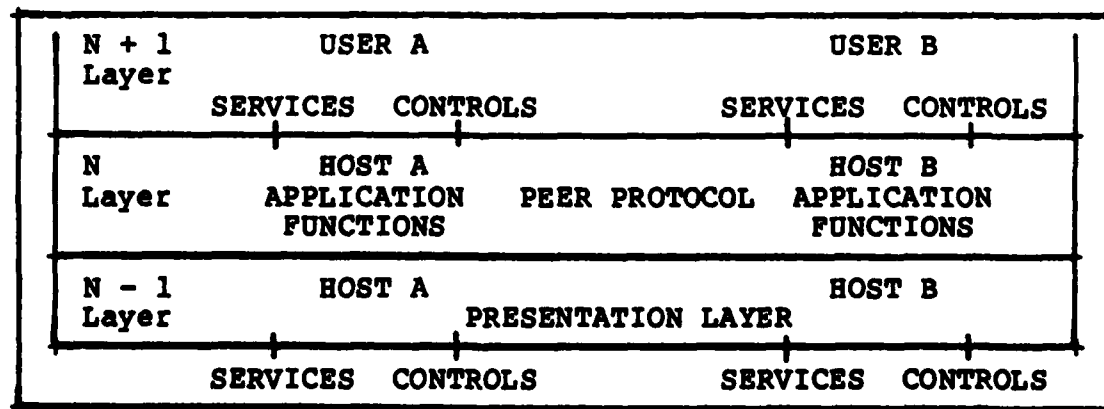


Figure 2-9. Application Layer Relationships

(See Appendix B for a general discussion of the N-Layer concept).

As the highest layer in the Reference Model of Open Systems Interconnection (OSI), the Application Layer provides a means for the application processes to access the prescribed OSI environment (Ref 26:55-57). Hence, the Application Layer does not interface with a higher layer. The purpose of the Application Layer is to serve as the window between correspondent application-processes which are using the OSI to exchange meaningful data. The Application Layer contains all functions which imply communication between open systems and which are not already performed by the lower protocol layers. These include functions performed by programs as well as by human operators.

Application-processes exchange information by means of application-entities, application-protocols, and presentation-services. As the only layer in the Reference Model that directly services the application-processes, the Application Layer necessarily provides all OSI services directly usable by application-processes.

In addition to information transfer, the Application Layer provides the following services: identification of intended communication partners; determination of the current availability of the intended communication partners; establishment of authority to communicate; agreement on privacy mechanisms; authentication of intended communication partners; determination of cost allocation methodology;

determination of adequacy of resources; determination of the acceptable quality of service (e.g. response time, tolerable error rate, and cost); synchronization of cooperating applications; selection of the dialogue discipline, including the initiation and release procedures; agreement on responsibility for error recovery; agreement on the procedures for control of data integrity; and finally, identification of constraints on data syntax (character sets, data structure).

Application Layer as Applied to the DELNET

Initially, within the DELNET, the Application Layer will perform both file and message transfers. This protocol layer, as with the Transport and Session Layers, will follow the protocols as specified by the National Bureau of Standards (NBS). These are: File Transfer Protocol (FTP) (Ref 15) and Message Transfer Protocol (MTP) (Ref 14). As this protocol layer is expanded, more and more application functions will be added to the DELNET, such as editing or executing on a remote system.

Chapter Summary

This chapter outlined the basic requirements of the protocol layers of the ISO Reference Model that must be accomplished in order to implement a Computer Network within AFIT.

The next stage in the software life cycle is to develop a comprehensive plan to test the implementation of the DELNET. The next chapter presents a detailed test plan that

is to be followed concerning the software and hardware portions of the UNIDs within the DELNET.

III. DELNET SOFTWARE TEST PLAN

Introduction

This chapter defines, in nine phases, a complete test plan to be applied to the software portions of UNID#1, UNID#2, UNID#3, and the Zilog MCZ 1/25 Computer System in relation to the DELNET.

There are many types of testing that could be performed on both the UNID's and the Zilog. However, software testing can be divided into three distinct categories: module testing, integration testing, and systems testing. Within these three categories there are many testing methods available, with the more common ones being path and boundary testing. This Test Plan is partitioned into three major sections.

The first section (Phases I, II, IV, and VI) consists of module testing. Path testing was selected as the method of module testing due to the "relatively" straightforward implementation of both UNID and Zilog software, that is, they both operate sequentially. This method will ensure that each of the UNID and Zilog modules operate properly for a given path through that module. Whenever appropriate, boundary testing will be utilized. This will be particularly useful when the UNID determines which local port a particular Host is connected.

The second section of this Test Plan (Phase III) consists of integration testing. Again, path testing will be employed as there are only two possible paths between local

and network sides of the UNID. This method will ensure that the interface between the local and network sides of the UNID functions correctly for data packet transfer.

The third, and last, section of this Test Plan (Phases V, VII, VIII, and IX) consists of system testing. This section of testing will also use path testing as there are only two possible paths between the Host and UNID. This method will ensure that the 128-byte Datagram is transmitted and/or received from Host-to-Host.

While some tests seem redundant to others, they were designed to test each line of software code for each particular aspect of the UNID and Zilog operation. Each stage will test a specific software module and will also test each possible data path through that module. In this manner, any software or hardware errors can be detected as quickly as possible.

All the testing outlined in this chapter will depend on two things: the first is the continued operation of the UNID's hardware (UNID#1 and UNID#2); and the second is the completion of the design of a third UNID (UNID#3).

As discussed in the Requirements Section of this investigation (Chapter 2), this test plan will test each protocol layer for correct operation. As a review, remember that the first three protocol layers (Physical, Data Link, and Network) will reside within the UNIDs, while the upper four protocol layers (Transport, Session, Presentation, and Application) will reside in the Host.

Within the UNIDs, the Physical Layer will be tested to ensure that data is put into the correct input table and that data is sent out the RS-232 port correctly. The Data Link Layer will be tested to ensure that a 135-byte Data Packet is correctly created and that the Data Packet Header is correctly removed before the 128-byte data block is sent out to a Local Host. The Network Layer will be tested to ensure that the incoming 128-byte data block is routed correctly and that a 135-byte Frame is correctly created and sent out onto the Network Ring.

Within the Host, the Transport Layer will be tested to ensure that the outgoing test message is correctly transformed into a 128-byte Datagram. The Session and Presentation Layers will not be tested at this time as there is only one Host on the DELNET. The Application Layer will be tested to ensure that TEST_MSG_1 and TEST_MSG_2 are correctly routed to and from the UNID.

Using the above, the nine testing phases are as follows: Phase I tests the internal operation of the local side; Phase II tests the internal operation of the network side; Phase III tests internal data transfer from the local side to the network side and from the network side to the local side; Phase IV tests local-to-local message transfer, using loopback, on the local side of the UNID; Phase V tests local-to-local message transfer using the Zilog MCZ 1/25 Computer System; Phase VI tests network-to-network message transfer, using loopback, on the network side of the UNID (this is a

test of the dual ring); Phase VII tests the message transfer, via the two UNID dual ring, between two remote Hosts; Phase VIII tests the installation of the 18-pair cable by placing UNID#2 on the second floor and sending a test message up to UNID#2 and back to UNID#1 and then to a local host, located in the basement. The last phase, Phase IX, tests the addition of a third UNID and the transfer of data between remote Hosts via a three UNID dual ring network.

Objectives

The objective of this test plan is to provide a comprehensive testing procedure, given a limited networking environment, for UNID#1, UNID#2, and the Zilog MCZ 1/25 Computer System (DELNET Monitor).

Success Criteria

This test plan will be deemed a success when a test message is sent via the UNIDs to a remote Host. This test message will be in two parts. The first part is a 68-character message (called TEST_MSG_1), "Now is the time for all good men to come to the aid of their country.", representing one Frame of data (One Frame is 128 bytes from the Host with first 60-bytes Transport Layer overhead). The second part (called TEST_MSG_2), is the Gettysburg Address by President Abraham Lincoln. This was selected because it consisted of 1584 characters providing a 23-Frame message (56-bytes for Datagram overhead and 72-bytes of user data per Datagram).

Testing Procedures

Phase I: UNID Local Side Data Transfer

This phase is divided into five stages so that both Procedures ROUTE_IN and ROUTE_OUT located within PLZ language Modules L.MAIN_U1 and L.MAIN_U2, on the local side of the UNID, can be thoroughly tested. This phase insures that the destination of an incoming 128-byte data block from the local side and a data packet from the network side is first correctly determined and routed to the correct output table and subsequently to the correct output data port.

Stage One tests data transfer from the four input tables to the Local-to-Local Table. This stage will ensure that the 128-byte incoming data block is correctly routed to the Local-to-Local Table. This stage tests each of the sixteen routing possibilities (four input Local Channels routed to four output Local Channels).

Stage Two tests data transfer from the four input tables to the Local-to-Network Table. This stage has four routing possibilities for the incoming 128-byte data block.

Stage Three tests data transfer from the Network-to-Local Table to each of the four local channels. This stage has four routing possibilities for the outgoing 128-byte data block.

Stage Four is from the Local-to-Local Table to each of the four local channels. This stage has only four routing possibilities for the outgoing data block.

The last stage, Stage Five, tests for each of the three

possible incoming channel errors that are presently detectable on the local side of the UNID. These are: invalid Control_Code, invalid Country_Code, and invalid Network_Code. This stage has a total of twelve configurations to test the three error conditions on each of the four local channels.

Phase II: UNID Network Side Data Transfer

This phase is divided into three stages so that both Procedures ROUTE_IN and ROUTE_OUT within PLZ language Modules N.MAIN_U1 and N.MAIN_U2 on the network side of the UNID can be thoroughly tested. Phase II testing ensures that the destinations of both a Data Packet and Frame are correctly determined and are routed either to the correct input table or out the correct network channel.

Stage One tests data transfer from the Local-to-Network Table to each of the outgoing network channels (Channel-A and Channel-B). There are two tests to perform.

The second stage tests data transfer from the two incoming Network Channel's to the Network-to-Local Table. There are two tests to perform. In each case the Network-to-Local Table must be examined to ensure that the Frame Header was correctly stripped from the incoming Data Frame. Also, the outgoing network channel must be checked to make sure an S-Frame was correctly sent back to the sending UNID indicating that the I-Frame was received.

The last stage is Network-to-Network message transfer. There are two tests to perform. In both cases the outgoing Frame should be examined to ensure that the FROM Destination

Address was changed to the current UNID number and a valid S-Frame was sent back to the sending UNID.

Phase III: Local-to-Network and Network-to-Local Data Transfer

This phase of testing ensures that the shared PLZ language Modules U.LIB_U1, U.LIB_U2, U.SHTAB_U1, and U.SHTAB_U2 function correctly for transfer of data from the local side to the network side and vice versa. This testing is divided into six stages. The first four testing stages ensure that an incoming 128-byte data block, from each of the four local channels, is correctly routed (via Channel-A or Channel-B) onto the dual network ring as a 135-byte Frame (eight tests). The last two stages ensure that a 135-byte Frame, from each of two network channels, is correctly routed to each of the local channels (eight tests).

Phase IV: Local-to-Local Loopback Data Transfer

This testing is divided into four stages. Each of the four stages will test data transfer from one local channel to each of the remaining three. This phase of testing ensures that the assembly language modules L.VINT_U1 and L.VINT_U2 function correctly.

Phase V: Local-to-Local Data Transfer

This testing phase is divided into four stages. Each testing stage ensures that the test message can be sent to the UNID from the Zilog Computer System and received from the UNID correctly. In each stage both test messages will be sent.

Stage One tests message transfer from the Zilog serviced

by Local Channel-1 back thru Local Channel-1 port of the UNID to the Zilog.

Stage Two tests message transfer from the Zilog serviced by Local Channel-2 back thru Local Channel-2 port of the UNID to the Zilog.

Stage Three tests message transfer from the Zilog serviced by Local Channel-3 back thru Local Channel-3 port of the UNID to the Zilog.

The last stage, Stage Four tests message transfer from the Zilog serviced by Local Channel-4 back thru Local Channel-4 port of the UNID to the Zilog.

Phase VI: Network-to-Network Loopback Data Transfer

This testing is divided into two stages. The first stage tests data transfer out Network Channel A and into Network Channel B. The second stage tests data transfer out Network Channel B and into Network Channel A. This phase of testing ensures that the assembly language modules N.INSIO_U1 and N.INSIO_U2 function correctly.

Phase VII: Host-to-Host Data Transfer via 2-UNID Network

This testing phase is divided into two stages. Each stage assures that a test message can be routed correctly to a remote Host. Both TEST_MSG_1 and TEST_MSG_2 will be sent for each stage of testing.

Stage One tests message transfer routing from the Zilog MCZ 1/25 Computer System to the INTEL/432 Computer System. The test message will then subsequently be displayed on the CRT screen or a printout will be made on the INTEL/432

Printer.

Stage Two is the opposite of Stage One. Here the test message arrives from the INTEL/432 Computer System and is either displayed on the H19 CRT screen or a printout is made on the spinwriter printer.

Phase VIII: Host-to-Host Data Transfer via 18-Pair Cable

This testing phase is divided into two stages. Each stage will ensure that the 18-pair cable connecting the basement with the second floor works correctly. UNID#2 will be on the second floor and UNID#1 will be in the basement. The first stage is routing the test message from UNID#1 to UNID#2 and the second phase is routing the test message from UNID#2 to UNID#1. Both TEST_MSG_1 and TEST_MSG_2 will be sent.

Phase IX: Host-to-Host Data Transfer via 3-UNID Network

This phase of testing is divided into two stages. Each stage is to test the addition of a third UNID on the network. The first stage is from the Zilog MCZ 1/25 Computer System, via the three UNID computer network, to the INTEL/432 Computer System and the second is the reverse.

Hardware and Software Test Configurations

The following text and figures illustrate what must be done to perform correctly the aforementioned test procedures:

Phase I Stage 1: Internal transfer of a 128-byte data block from Local Channel-1 (LC01TB), Local Channel-2 (LC02TB), Local Channel-3 (LC03TB), and Local Channel-4 (LC04TB) to the Local-to-Local Table (LCLCTB). This is shown

in Figure 3-1.

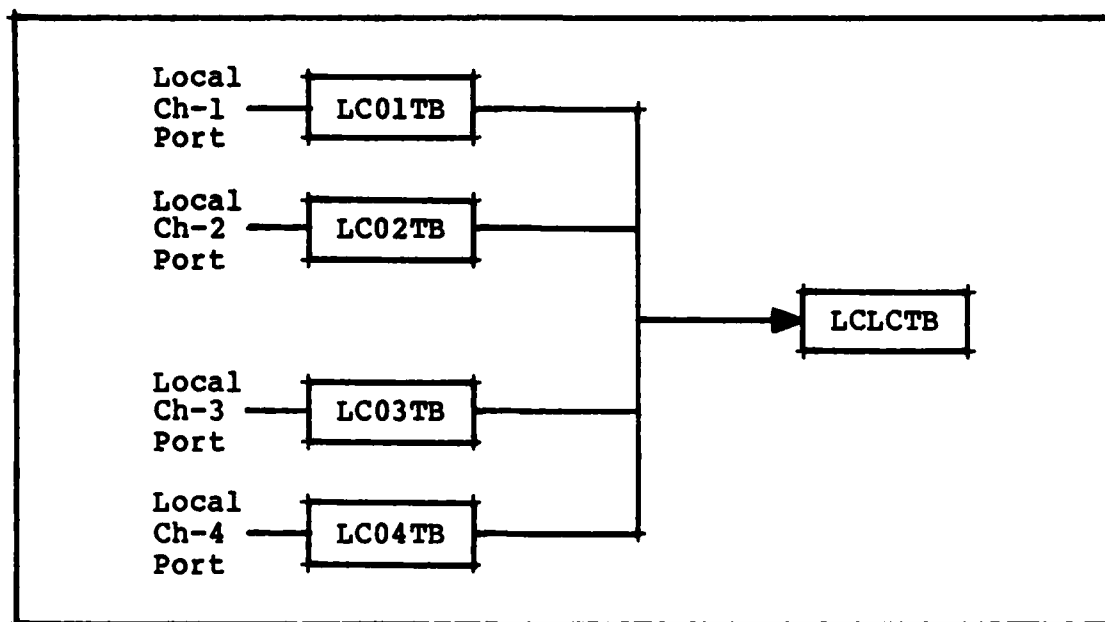


Figure 3-1. Incoming Local Table to LCLCTB Data Transfer

Phase I Stage 2: Internal transfer of 128-byte data block from LC01TB, LC02TB, LC03TB, and LC04TB to the Local-to-Network Table (LCNTTB). This is shown in Figure 3-2.

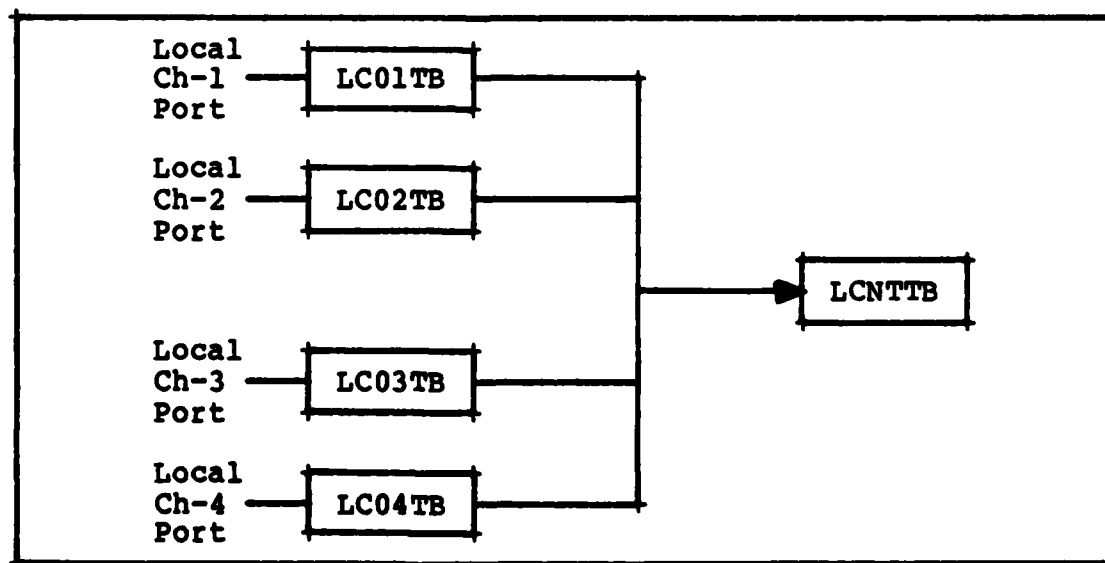


Figure 3-2. Incoming Local Table to LCNTTB Data Transfer

Phase I Stage 3: Internal transfer of 128-byte data

block from Network-to-Local Table (NTLCTB) to one of the four local output ports. This is illustrated in Figure 3-3.

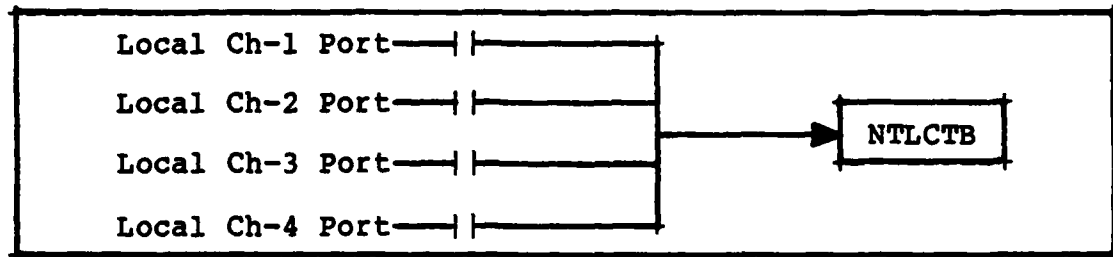


Figure 3-3. NTLCTB to Local Channel Ports Data Transfer

Phase I Stage 4: Internal transfer of 128-byte data block from Local-to-Local Table to one of the four local output ports. This is also shown in Figure 3-4.

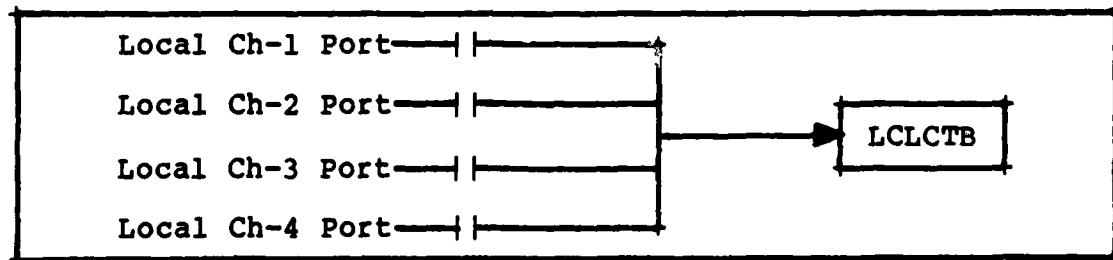


Figure 3-4. LCLCTB to Local Channel Ports Data Transfer

Phase I Stage 5: Internal detection of invalid CONTROL_CODE, invalid COUNTRY_CODE, and invalid NETWORK_CODE.

The configuration for this stage of testing is the same as Phase I Stage 1 (Figure 3-1).

Phase II Stage 1: Internal data transfer from Local-to-Network Table to each of the outgoing network channel tables (OUTFRAME_CHA_TB and OUTFRAME_CHB_TB). This is shown in Figure 3-5.

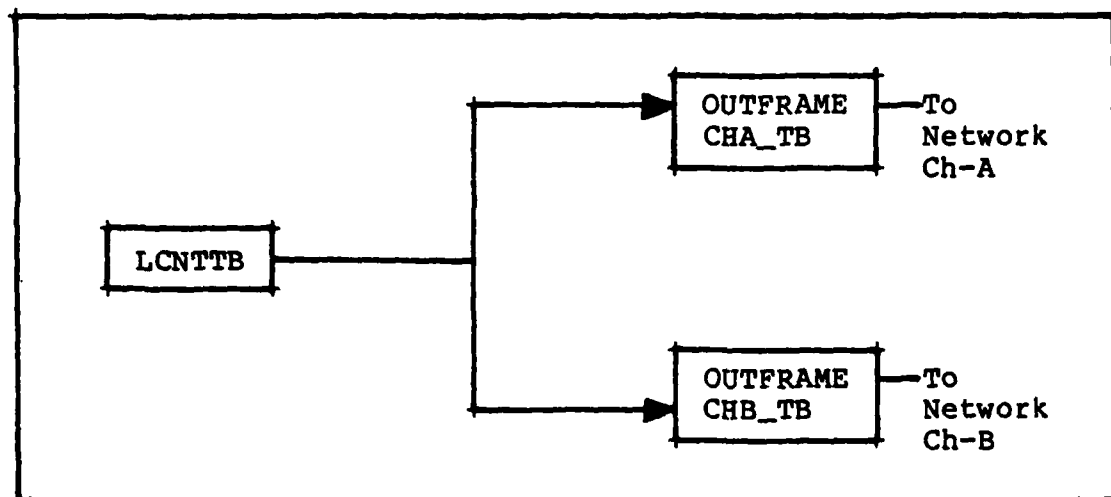


Figure 3-5. LCNTTB to Outgoing Network Channel Data Transfer

Phase II Stage 2: Internal 135-byte data Frame transfer from Network Channel Tables (NT01TB and NT02TB) to Network-to-Local Table. This is shown in Figure 3-6.

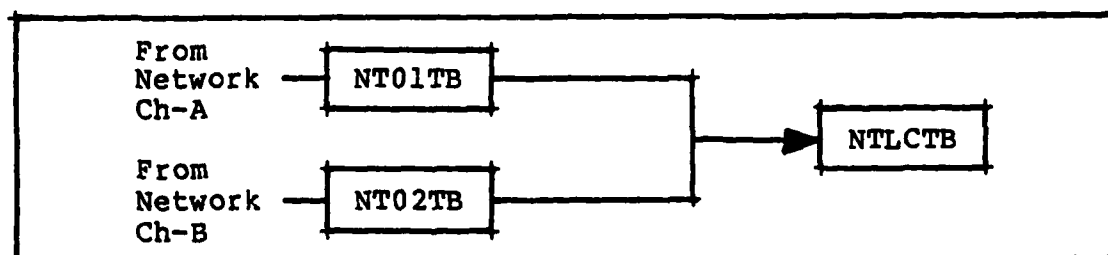


Figure 3-6. Incoming Network Channel to NTLCTB Data Transfer

Phase II Stage 3: Internal Network-to-Network 135-byte data Frame transfer. This is shown in Figure 3-7.

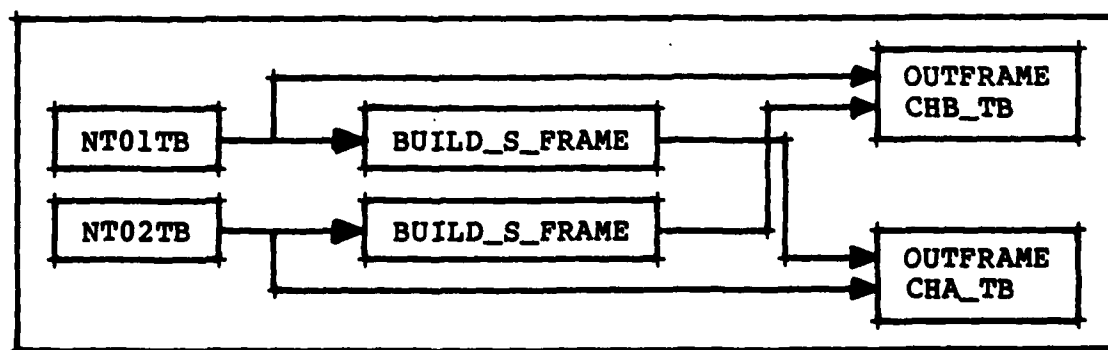


Figure 3-7. Internal Network-to-Network Data Transfer

Phase III Stage 1: Internal data transfer from Local Channel-1 to outgoing Network Channels A and B. This is shown in Figure 3-8.

Phase III Stage 2: Internal data transfer from Local Channel-2 to outgoing Network Channel's A and B. This is shown in Figure 3-8.

Phase III Stage 3: Internal data transfer from Local Channel-3 to outgoing Network Channel's A and B. This is shown in Figure 3-8.

Phase III Stage 4: Internal data transfer from Local Channel-4 to outgoing Network Channel's A and B. This is shown in Figure 3-8.

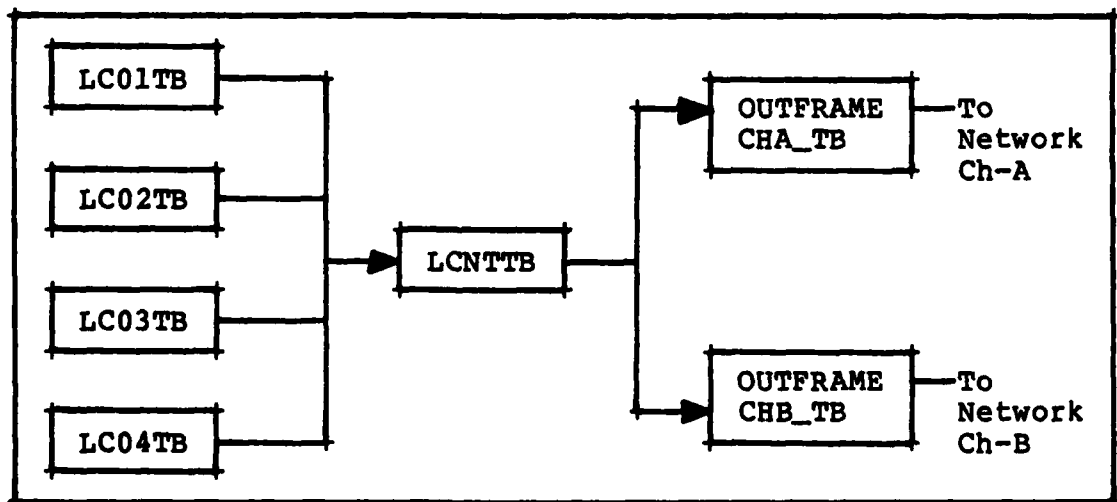


Figure 3-8. Internal Local-to-Network Data Transfer

Phase III Stage 5: Internal data transfer from incoming Network Channel-A to each of the four local channels. This is shown in Figure 3-9.

Phase III Stage 6: Internal data transfer from incoming Network Channel-B to each of the four local channels. This is illustrated in Figure 3-9.

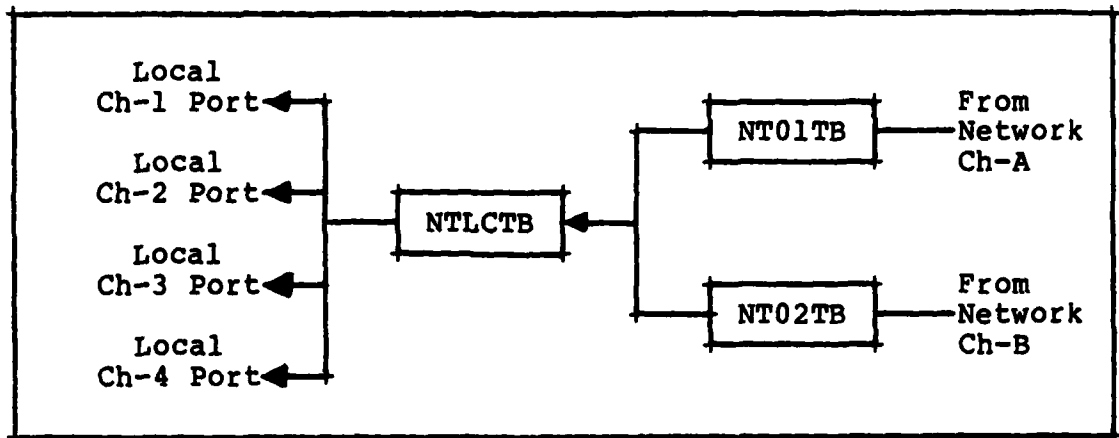


Figure 3-9. Internal Network-to-Local Data Transfer

The following stages refer to Figure 3-10.

Phase IV Stage 1: Local loopback test from Local Channel-1 to the other three local channels.

Phase IV Stage 2: Local loopback test from Local Channel-2 to the other three local channels.

Phase IV Stage 3: Local loopback test from Local Channel-3 to the other three local channels.

Phase IV Stage 4: Local loopback test from Local Channel-4 to the other three local channels.

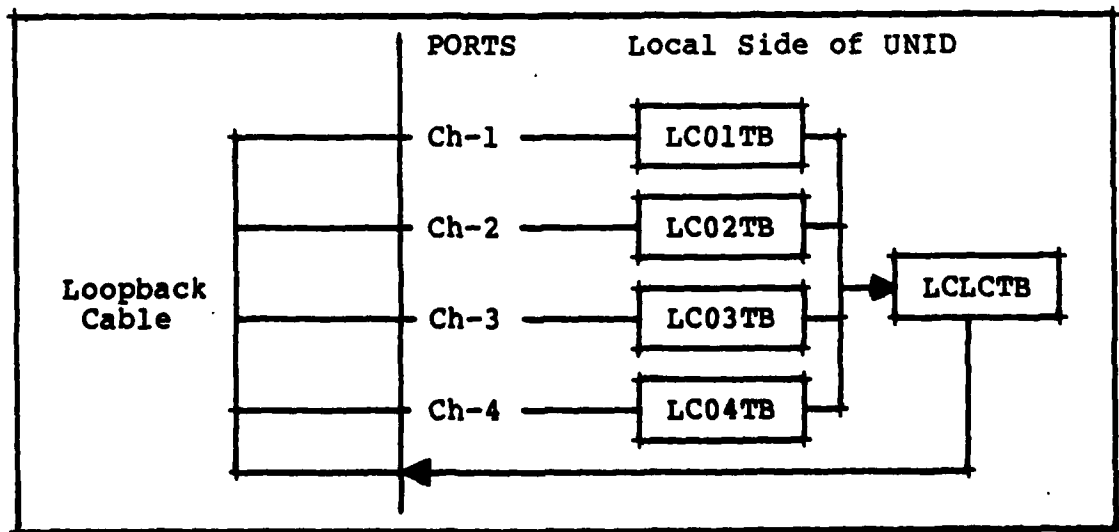


Figure 3-10. Local-to-Local Test Message Transfer

The following stages refer to Figure 3-11.

Phase V Stage 1: Transfer of a test message from the Zilog MCZ 1/25 Computer System being serviced by UNID Local Channel-1 back to the Zilog MCZ 1/25 Computer System through the four local channels of the UNID.

Phase V Stage 2: Transfer of a test message from the Zilog MCZ 1/25 Computer System being serviced by UNID Local Channel-2 back to the Zilog MCZ 1/25 Computer System through the four local channels of the UNID.

Phase V Stage 3: Transfer of a test message from the Zilog MCZ 1/25 Computer System being serviced by UNID Local Channel-3 back to the Zilog MCZ 1/25 Computer System through the four local channels of the UNID.

Phase V Stage 4: Transfer of a test message from the Zilog MCZ 1/25 Computer System being serviced by UNID Local Channel-4 back to the Zilog MCZ 1/25 Computer System through the four local channels of the UNID.

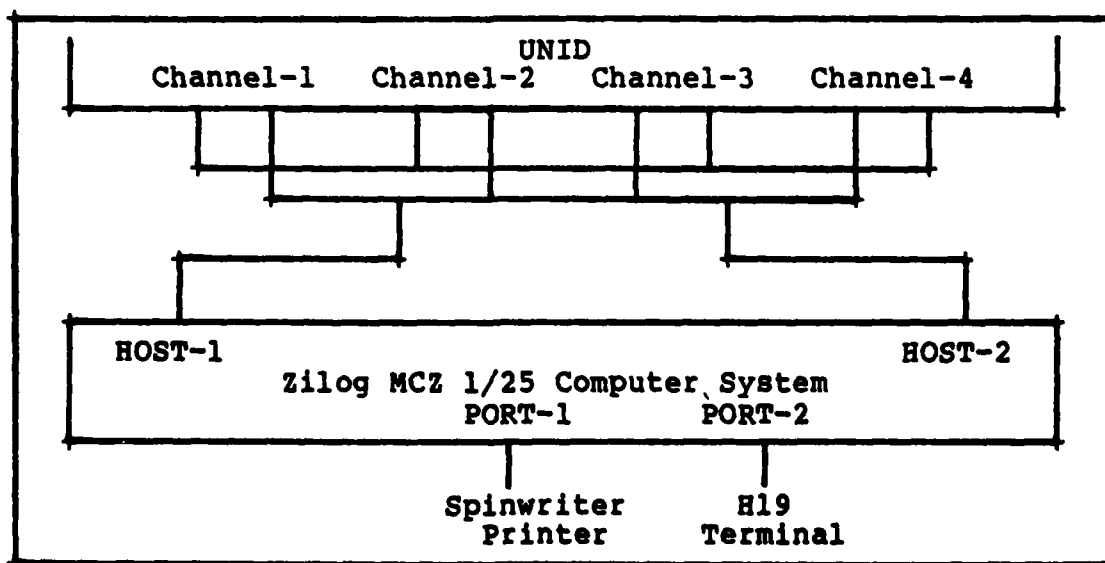


Figure 3-11. Zilog-to-Zilog Local-to-Local Test Configuration

Phase VI Stage 1: OUTFRAME_CHA_TB to NT01TB. Loopback test for clockwise portion of the network ring as illustrated in Figure 3-12.

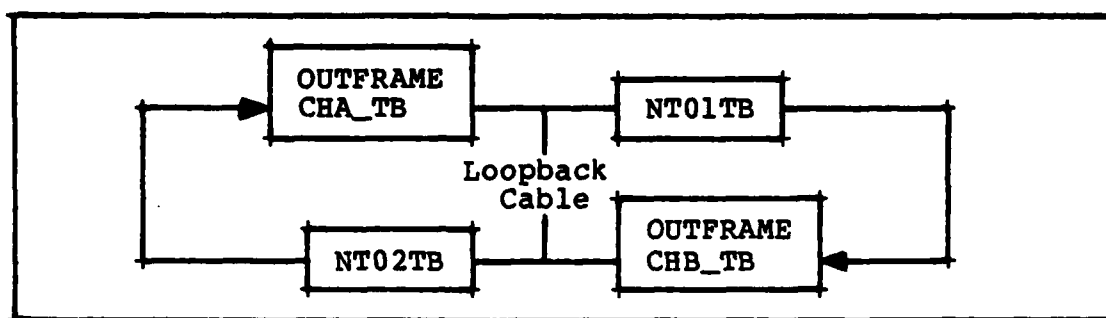


Figure 3-12. Clockwise Portion of the Network Ring

Phase VI Stage 2: OUTFRAME_CHB_TB to NT02TB. Loopback test for counter-clockwise portion of the network ring as shown in Figure 3-13.

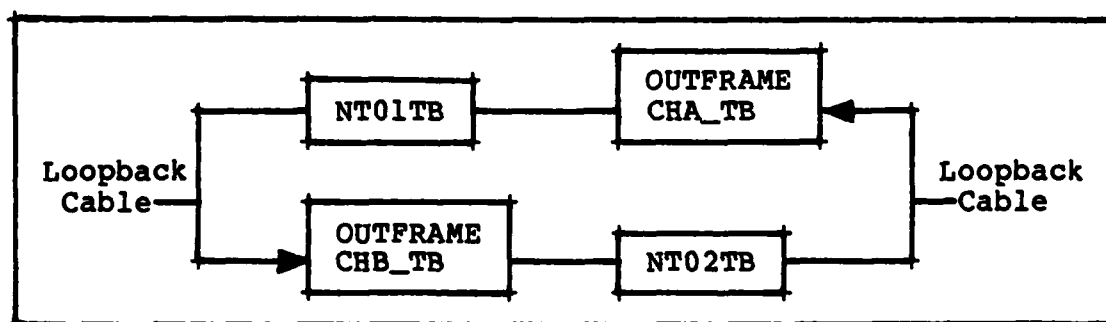


Figure 3-13. Counter-Clockwise Portion of the Network Ring

Phase VII Stage 1: Transfer of a test message from the Zilog MCZ 1/25 back to the Zilog MCZ 1/25 via a two UNID computer network. This is shown in Figure 3-14.

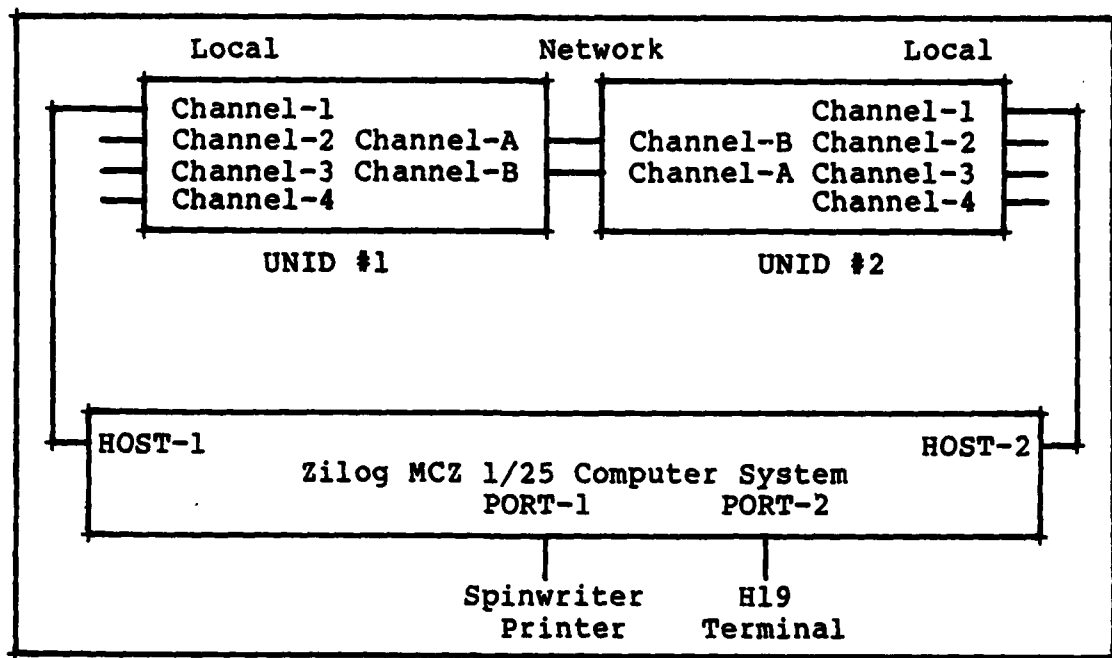


Figure 3-14. Zilog-to-Zilog (2-UNID) Test Configuration

Phase VII Stage 2: Transfer of a test message from the INTEL/432 Computer System to the Zilog MCZ 1/25 Computer System and vice versa via a two UNID computer network. This is shown in Figure 3-15.

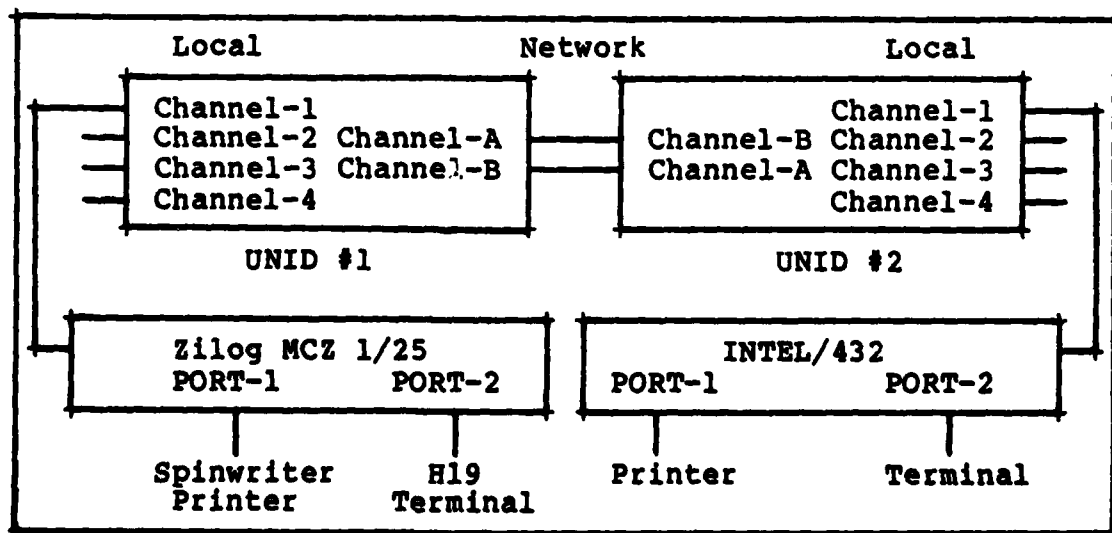


Figure 3-15. INTEL/432-to-Zilog (2-UNID) Test Configuration

Phase VIII Stage 1: Transfer of a test message via the

18-pair cable to a remote Host. This is shown in Figure 3-14, where the 18-pair cable is between UNID#1 and UNID#2.

Phase VIII Stage 2: Transfer of a test message via the 18-pair cable from a remote Host. This is also illustrated in Figure 3-15. The connection between UNID#1 and UNID#2 will be via the 18-pair cable.

Phase IX Stage 1: Transfer of a test message to/from the Zilog MCZ 1/25 Computer System, via a three-UNID dual ring network. This is Illustrated in Figure 3-16.

Phase IX Stage 2: Transfer of a test message from the Zilog MCZ 1/25 computer System to the INTEL/432 Computer System and vice versa via a three-UNID dual ring network. This is illustrated in Figure 3-17.

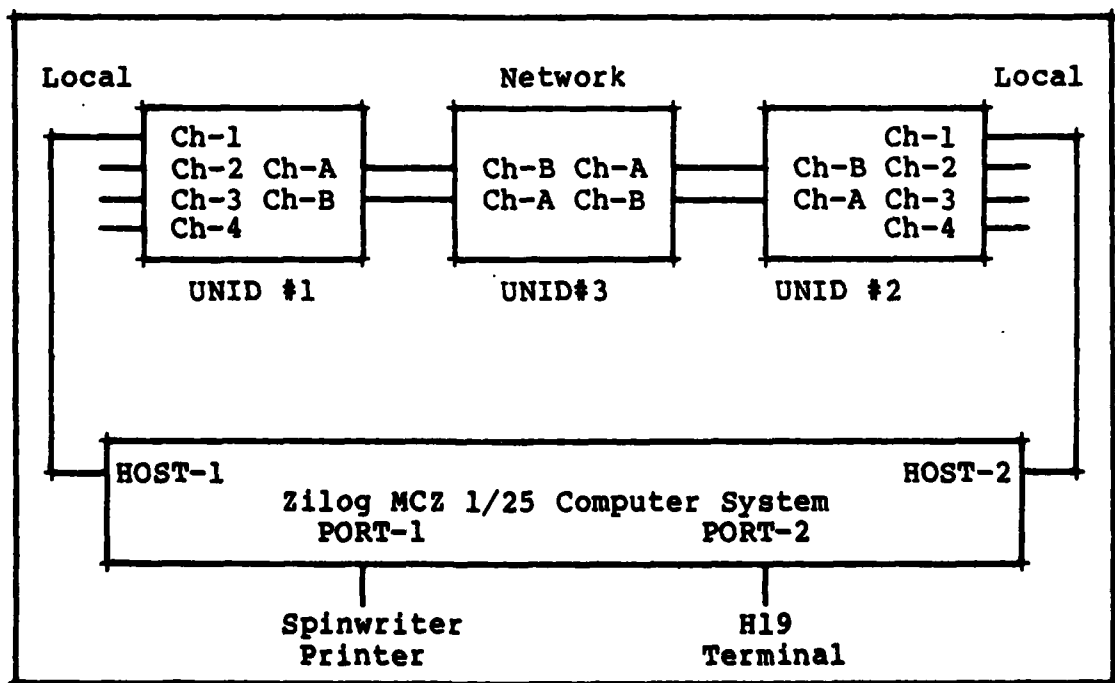


Figure 3-16. Zilog-to-Zilog (3-UNID) Test Configuration

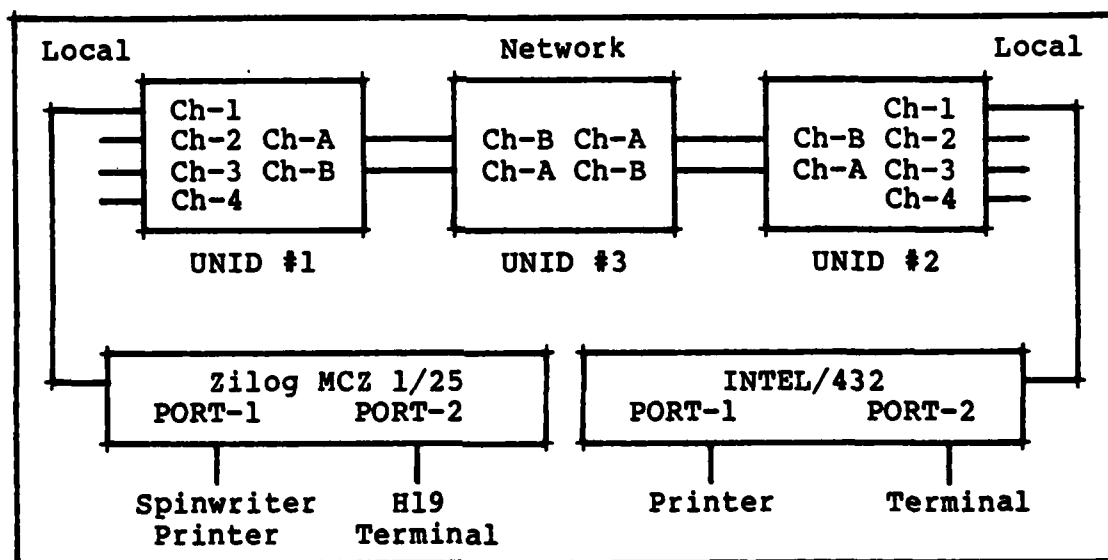


Figure 3-17. Zilog-to-INTEL/432 (3-UNID) Test Configuration

Hardware Devices Required

The following list of equipment was used in this test plan: Zilog MCZ 1/25 Dual Disk Computer System; INTEL-432 Dual Disk Computer System; Two H-19 CRT Terminals; Spinwriter Printer; H-47 Printer; Two Z80 UNIDS (UNID#1, UNID#2); and a Z86 UNID (UNID#3).

Chapter Summary

This chapter describes a complete test plan to be used in the initial stages of the implementation of the DELNET. However, before this test plan could be implemented the UNID's local and network side software had to be modified to reflect the protocol requirements and standards as outlined in Chapter 2 (see also Appendix C). The next chapter addresses the changes that had to be made on both the local and network operating systems of the UNID and the software that had to be developed for the Zilog MCZ 1/25 Computer System to implement the upper four protocol layers.

IV. DELNET SOFTWARE IMPLEMENTATION

Introduction

This chapter discusses the software and hardware changes that were made to both the local and network operating systems of the UNID (to allow the passage of 135-byte data frames in a dual ring networking environment) and the software developed for the Zilog MCZ 1/25 Computer system (to allow the passage of 128-byte data blocks to the UNID) (Refs 55,56,57,60, and 63).

The first section of this chapter deals with the changes required to the sizes of the UNID's table. The second section deals with the changes to the local operating system, while the third section deals with the network operating system software changes. The fourth section covers the updated memory map and the new linking procedures for both local and network operating systems.

The fifth, and last, section covers the software developed for the Zilog MCZ 1/25 Computer System to enable it to communicate with the UNID.

UNID Tables

In the previous investigation, it was assumed that a Data_Packet entered the UNID with the Source and Destination bytes already inserted (Ref 19:5-6). If the data was destined for the network, the local operating system of the UNID created a Frame and subsequently transferred the data to the network side of the UNID via the shared data tables. In a networking environment, this is incorrect. The correct

procedure is for the local side to create a Data_Packet and move it to the network side of the UNID via the shared data tables. The network side would then subsequently create a Frame and deposit the Frame onto the network.

With this in mind, both the local, network, and shared data tables had to be restructured. The table sizes had to reflect Data_Packets on the local side and Frames on the network side. Figure 4-1 illustrates the new restructured data tables.

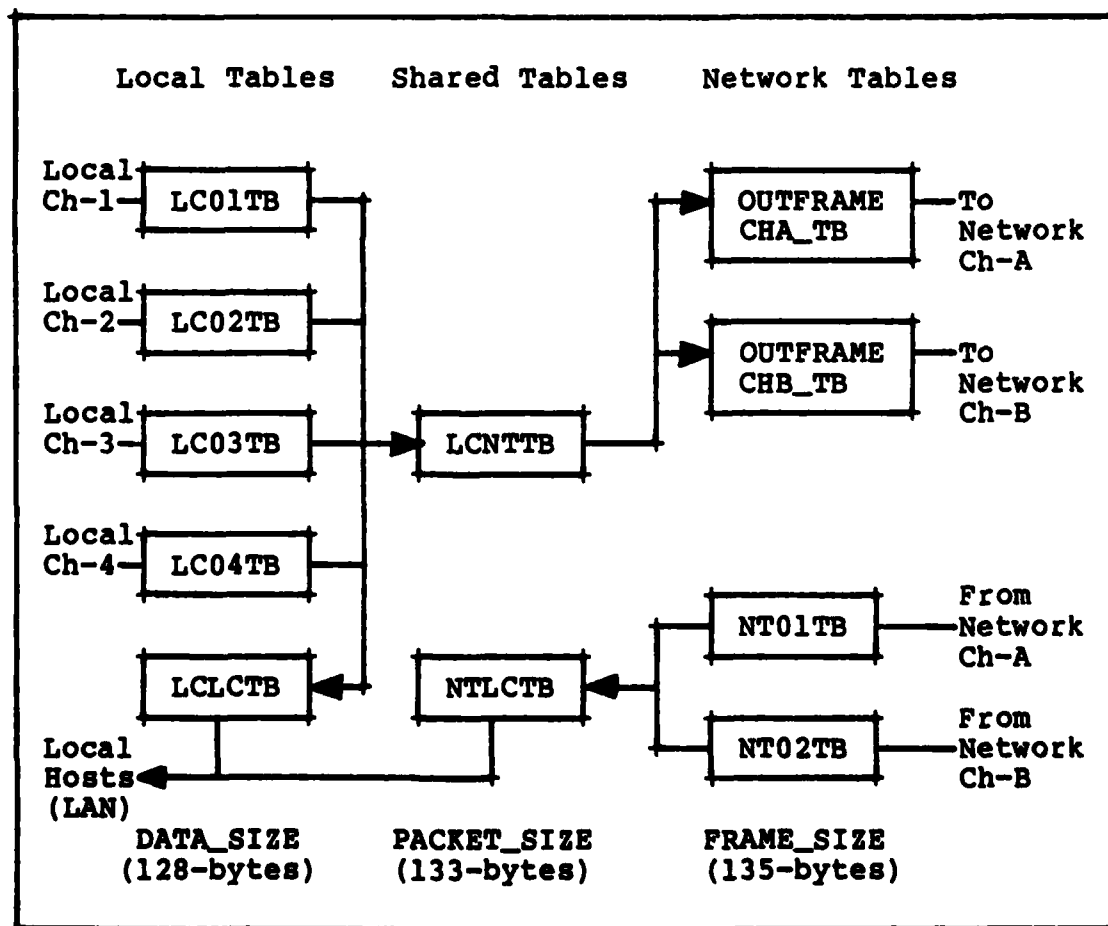


Figure 4-1. UNID Local and Network Operating Systems' Tables

UNID Local Operating System

On the local side of the UNID, the RS-232 ports were installed, enabling connection to local Hosts (See Appendix A for detailed discussion). The USARTs for Channels 2, 3, and 4 were re-wired to match the wiring of Channel-1 USART. This is also discussed in detail in Appendix A.

During the previous thesis effort, little attention was given to creating a Network Layer within the UNID using a standard protocol as discussed in Appendix C (Ref 19).

In a networking environment, using the Datagram Service option, the data flow is such that the UNID will accept a block of 128-bytes (current configuration and subject to change) from the local Host (see Figure 4-1). The format for this block of data, as discussed in Chapter 2, consists of IHF (Internet Header Format) plus TCP (Transport Control Protocol) plus DATA (User data) (See Appendix C Transport Layer for protocol standard). The IHF plus TCP headers consist of 56-bytes representing about 44% of Transport Layer overhead. The local operating system of the UNID determines the data's destination and if the information is destined for the network, the local operating system will first create a Data_Packet (See Appendix C Network Layer for protocol standard) then send it to the network side of the UNID via the shared data tables. The network operating system of the UNID will then create a Frame (See Appendix C Data Link Layer for protocol standard) and route it onto the network accordingly. Refer to Appendix E for the Data Dictionary and

Appendix F for the software listing of the UNIDs local side.

Since the local side of the UNID is responsible for routing and creating a Data_Packet (See Appendix C Network Layer for Routing Standard) instead of a Frame, ROUTE_IN, ROUTE_OUT, and DET_DEST procedures had to be modified. The ROUTE_IN procedure was modified so that it would build an I-PACKET rather than create an I-FRAME (Ref 19:5-9 Fig 14). ROUTE_OUT was changed to strip a Data_Packet Header instead of a FRAME Header. DET_DEST was significantly restructured to bring it into agreement with the protocol standard.

Taking the ROUTE_IN Procedure first, Figure 4-2 gives the updated structure chart.

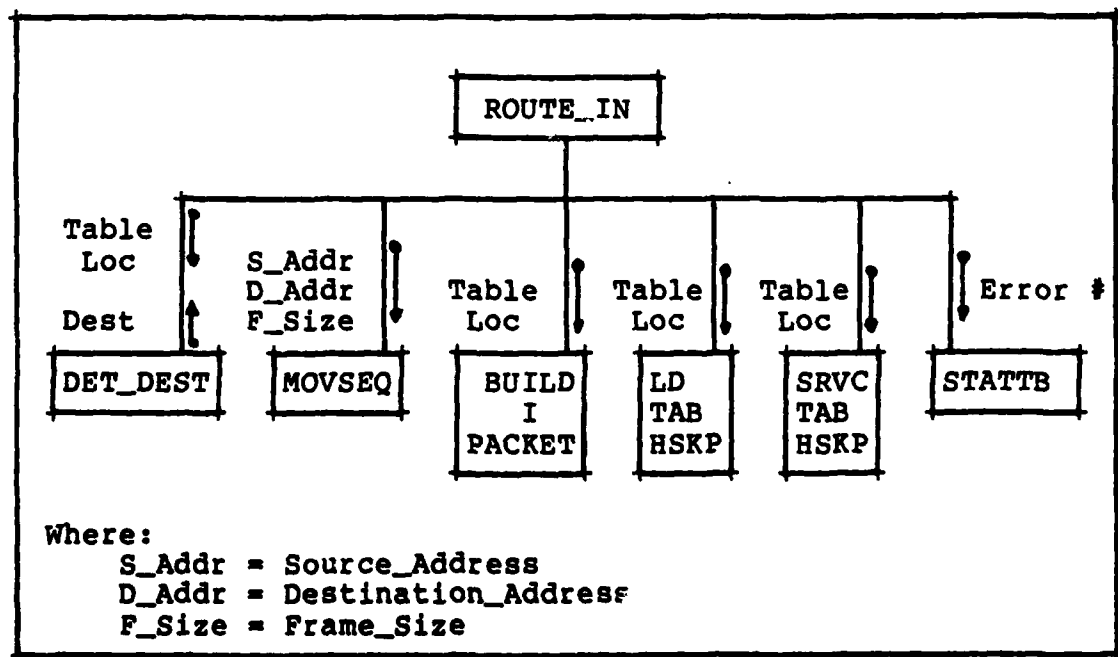


Figure 4-2. ROUTE_IN Procedure for UNID LOS

In Pseudo-English the processing of the ROUTE_IN Procedure was changed to the following:

```

Enter ROUTE_IN Procedure
If 128-byte data block present in LC01TB Then
  Determine block's DESTINATION
  If DESTINATION = 'LL' Then
    Move 128-byte data block to LCLCTB
    Update LCLCTB pointers
  Endif
  If DESTINATION = 'LN' Then
    Insert SOURCE_ADDRESS
    Call BUILD_I_PACKET
    Move 133-byte data packet to LCNTTB
    Update LCNTTB pointers
  Endif
  If DESTINATION = 'ER' Then
    Increment error status table
  Endif
Endif
Update LC01TB

```

Repeat above sequence for LC02TB, LC03TB, and LC04TB.
END ROUTE_IN Procedure

Very few changes were made to the ROUTE_OUT Procedure. These changes reflect the manipulation of a Data_Packet instead of a Frame. Figure 4-3 gives the updated structure chart for the ROUTE_OUT Procedure.

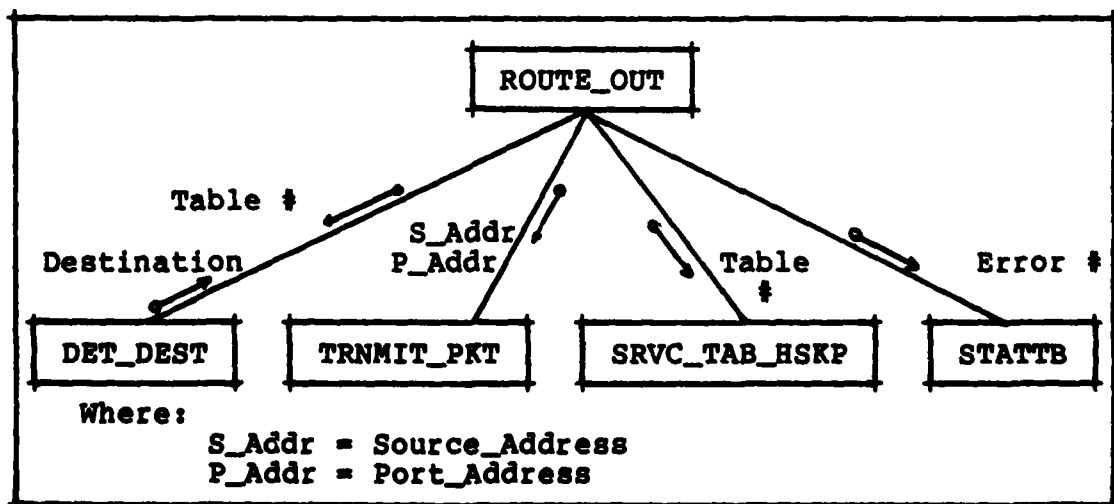


Figure 4-3. ROUTE_OUT Procedure for UNID LOS

In Pseudo-English the processing of the ROUTE_OUT

Procedure was changed to the following:

```
ENTER ROUTE_OUT Procedure
If 128-byte data block present in LCLCTB Then
  Determine data block's DESTINATION
  If DESTINATION = Channel-1 Then
    Transmit 128-bytes out Local Channel-1
  Endif
  If DESTINATION = Channel-2 Then
    Transmit 128-bytes out Local Channel-2
  Endif
  If DESTINATION = Channel-3 Then
    Transmit 128-bytes out Local Channel-3
  Endif
  If DESTINATION = Channel-4 Then
    Transmit 128-bytes out Local Channel-4
  Endif
  If DESTINATION = 'ER' Then
    Increment Error Status Table
  Endif
  Update LCLCTB pointers
Endif

If a 133-byte data packet present in NTLCTB Then
  Determine data packet's DESTINATION
  If DESTINATION = Channel-1 Then
    Strip Data Packet Header
    Transmit 128-bytes out Local Channel-1
  Endif
  If DESTINATION = Channel-2 Then
    Strip Data Packet Header
    Transmit 128-bytes out Local Channel-2
  Endif
  If DESTINATION = Channel-3 Then
    Strip Data Packet Header
    Transmit 128-bytes out Local Channel-3
  Endif
  If DESTINATION = Channel-4 Then
    Strip Data Packet Header
    Transmit 128-bytes out Local Channel-4
  Endif
  If DESTINATION = 'ER' Then
    Update Error Status Table
  Endif
  Update NTLCTB pointers
Endif
End ROUTE_OUT Procedure
```

The DET_DEST procedure became six separate procedures due to its original size and the implementation of data coupling considerations. The six procedures are: DET_DEST_ONE

(Determines destination of incoming data from Local Channel-1); DET_DEST_TWO (Determines destination of incoming data from Local Channel-2); DET_DEST_THREE (Determines destination of incoming data from Local Channel-3); DET_DEST_FOUR (Determines destination of incoming data from Local Channel-4); DET_DEST_LL (Determines destination of data located in Local-to-Local Table); and finally, DET_DEST_LN (Determines the destination of data located in Local-to-Network Table). Each procedure performed the same function but on a different data table. This change facilitated improved data coupling between the modules. Figure 4-4 illustrates the structure chart for the basic DET_DEST procedure. Each DET_DEST procedure extracts the CONTROL_CODE, COUNTRY_CODE, NETWORK_CODE, and HOST_CODE to determine the destination of the 128-byte data block.

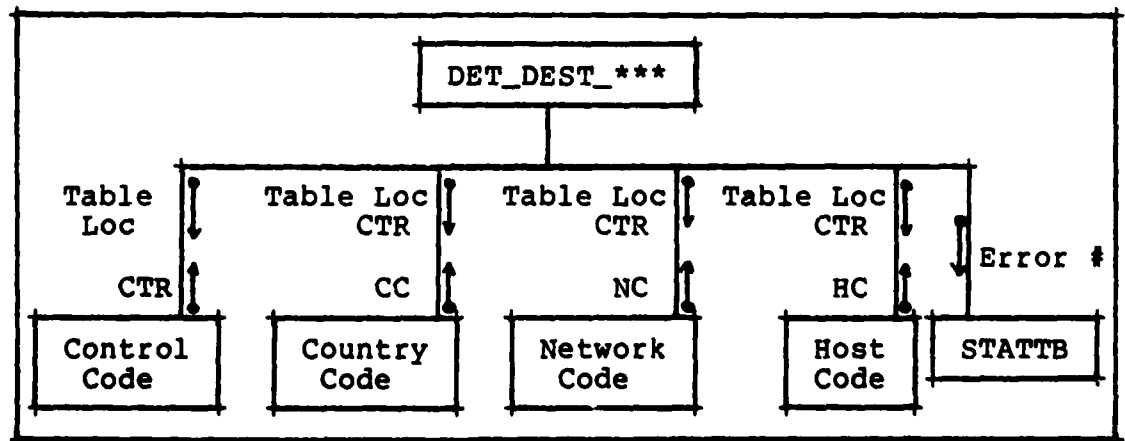


Figure 4-4. DET_DEST Procedures for UNID LOS

As illustrated, the DET_DEST procedure was changed to perform its routing function according to the prescribed protocol standard (See Appendix E and Appendix F for Data Dictionary and software listings). In Pseudo-English the

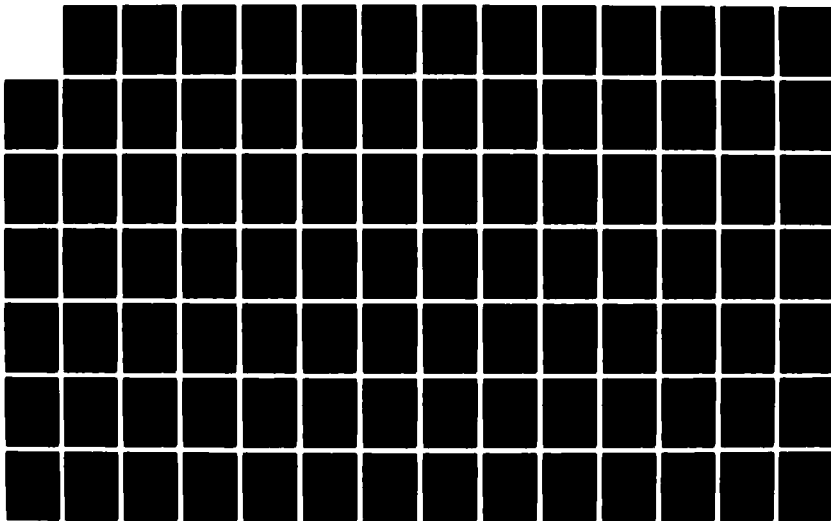
AD-A138 481

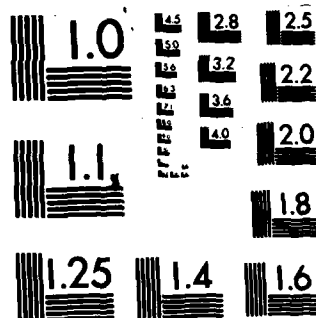
PROTOCOL STANDARDS AND IMPLEMENTATION WITHIN THE
DIGITAL ENGINEERING LABO..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P W PHISTER
DEC 83 AFIT/GE/EE/83D-58-PT-1 F/G 17/2

2/4

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DET_DEST Procedure is as follows:

a. If incoming data is in Local Channel Tables LC01TB, LC02TB, LC03TB, or LC04TB:

```
Determine CONTROL_CODE
For a CONTROL_CODE of '0000' do the following:
  Determine COUNTRY_CODE
  If COUNTRY_CODE is valid Then
    Determine NETWORK_CODE
    If NETWORK_CODE is valid Then
      Determine DESTINATION_ADDRESS
      If DESTINATION_ADDRESS is for another Network Host
        Then Determine HOST_CODE
          If 0 <= HOST_CODE <= 63
            Then DESTINATION_ADDRESS = NETWORK_CODE + Ch-1
          If 64 <= HOST_CODE <= 127
            Then DESTINATION_ADDRESS = NETWORK_CODE + Ch-2
          If 128 <= HOST_CODE <= 191
            Then DESTINATION_ADDRESS = NETWORK_CODE + Ch-3
          If 192 <= HOST_CODE <= 255
            Then DESTINATION_ADDRESS = NETWORK_CODE + Ch-4
          Else DESTINATION_ADDRESS = THIS_UNID_NBR
        Endif
      Else Set NETWORK_CODE Error
    Endif
  Else Set COUNTRY_CODE Error
Endif
Else Set CONTROL_CODE Error
END.
```

b. If data is from a local Host destined for a local Host or from the network destined for a local Host, then:

```
Determine CONTROL_CODE
If CONTROL_CODE is valid Then
  Determine HOST_CODE
  If 0 <= HOST_CODE <= 63
    Then DESTINATION_ADDRESS = Channel-1
  Endif
  If 64 <= HOST_CODE <= 127
    Then DESTINATION_ADDRESS = Channel-2
  Endif
  If 128 <= HOST_CODE <= 191
    Then DESTINATION_ADDRESS = Channel-3
  Endif
  If 192 <= HOST_CODE <= 255
    Then DESTINATION_ADDRESS = Channel-4
  Endif
Else Set CONTROL_CODE Error
Endif
```

UNID Network Operating System

As with the local side, the network ports were installed to enable access between UNIDs. The prescribed standard for the network side is RS-449 (mechanical) and the installation of the network ports is discussed in detail in Appendix A.

When data is destined for a local host from the Network, the UNID first accepts a Frame, strips off the Data Link Layer, then transfers the Data_Packet to the UNID's local operating system via the shared data tables. It then sends back an S-Frame to the sending UNID acknowledging the receipt of a valid I-Frame. The local operating system examines the Data_Packet to determine the destination then sends the 128-byte data block (minus Data_Packet Header) to the appropriate local Host (Refer to Appendix E for the Data Dictionary and Appendix F for the software listings of the UNIDs network side.)

When data is destined for another UNID, the network operating system first sends an S-Frame to the sending UNID, then transfers the I-Frame to the appropriate outgoing network port for transmission back onto the network. The UNID will change the Data Link FROM address to its UNID_NUMBER (See Appendix C Data Link Layer for Header Format).

As a design solution, the S-Frame is zeroed out in all cases except in the Data Link Frame Header. This is done so that the receiving UNID will not treat it as a valid I-Frame if an error occurs in the CONTROL_WORD of the S-Frame Header.

With the above items in mind, both procedures ROUTE_IN

and ROUTE_OUT on the network side of the UNID had to be modified. Figure 4-5 illustrates the new structure chart for the Procedure ROUTE_IN.

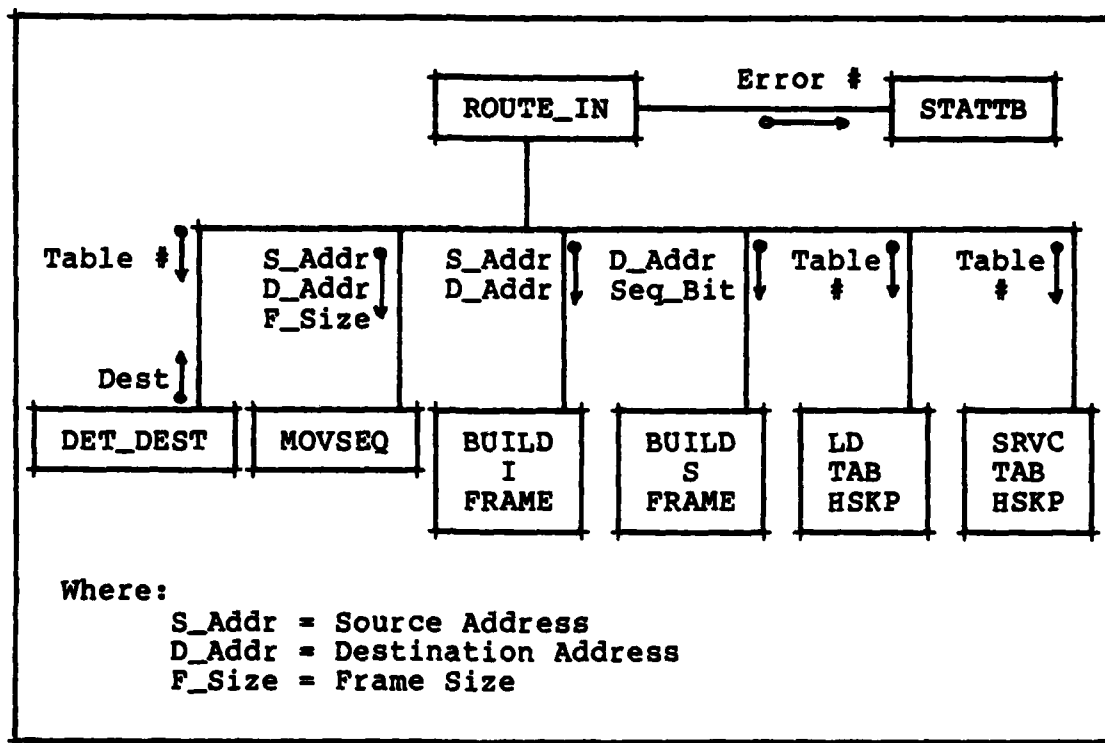


Figure 4-5. ROUTE_IN Procedure for UNID NOS

The following describes, in Pseudo-English, the ROUTE_IN Procedure illustrated in Figure 4-5.

```

ENTER ROUTE_IN Procedure
If a FRAME is present in NT01TB Then
  Determine frame DESTINATION
  If DESTINATION = 'NN' Then
    Determine INPUT_SEQ_BIT
    Call Build_S_FRAME to send ACKNOWLEDGE_A
    Move S-Frame to OUTFRAME_CHA_TB
    Update OUTFRAME_CHA_TB pointers
    Move I-Frame to OUTFRAME_CHB_TB
    Update OUTFRAME_CHB_TB pointers
  Endif
  If DESTINATION = 'NL' Then
    If frame = S-FRAME Then Determine if S-FRAME is
      positive ACKNOWLEDGE_A of last I-FRAME Sent
    Endif
    If frame = I-FRAME Then
      Determine INPUT_SEQ_BIT
    Endif
  Endif
Endif

```

```

        Call BUILD_S_FRAME to send ACKNOWLEDGE_A
        Move S-FRAME to OUTFRAME_CHA_TB
        Update OUTFRAME_CHA_TB pointers
        Strip off Frame Header and Move packet to NTLCTB
        Update NTLCTB pointers
    Endif
Endif
Update NT01TB pointers
Endif
If a FRAME is present in NT02TB Then
    Determine frame DESTINATION
    If DESTINATION = 'NN' Then
        Determine INPUT_SEQ_BIT
        Call BUILD_S_FRAME to send ACKNOWLEDGEMENT_B
        Move S-Frame to OUTFRAME_CHB_TB
        Update OUTFRAME_CHB_TB pointers
        Move I-Frame to OUTFRAME_CHA_TB
        Update OUTFRAME_CHA_TB pointers
    Endif
    If DESTINATION = 'NL' Then
        If frame = S-FRAME Then Determine if S-FRAME is
            positive ACKNOWLEDGE_B of last I-FRAME Sent
        Endif
        If frame = I-FRAME Then
            Determine INPUT_SEQ_BIT
            Call BUILD_S_FRAME to send ACKNOWLEDGE_B
            Move S-FRAME to OUTFRAME_CHB_TB
            Update OUTFRAME_CHB_TB pointers
            Strip off Frame Header and Move packet to NTLCTB
            Update NTLCTB pointers
        Endif
    Endif
    Update NT02TB pointers
Endif
If 133-byte data packet present in LCNTTB Then
    Determine data packet DESTINATION
    If DESTINATION = Network Channel-A Then
        BUILD_I_FRAME for Network Channel-A
    Endif
    If DESTINATION = Network Channel-B Then
        BUILD_I_FRAME for Network Channel-B
    Endif
Endif
END ROUTE_IN Procedure

```

The ROUTE_OUT Procedure was changed to reflect a dual ring network. Since each direction on the ring would have different operating parameters, ROUTE_OUT was modified accordingly. Each network channel has associated with it a

TIME_DELAY, an ACKNOWLEDGEMENT, and a SEQ_BIT value.

The next change to ROUTE_OUT was to divide the transmit procedure (TRNMIT) into two separate transmitting procedures (XMITCA and XMITCB). This ensured that processing for network Channel-A would be independent of the processing of network Channel-B.

Figure 4-6 gives the structure chart for the updated ROUTE_OUT Procedure.

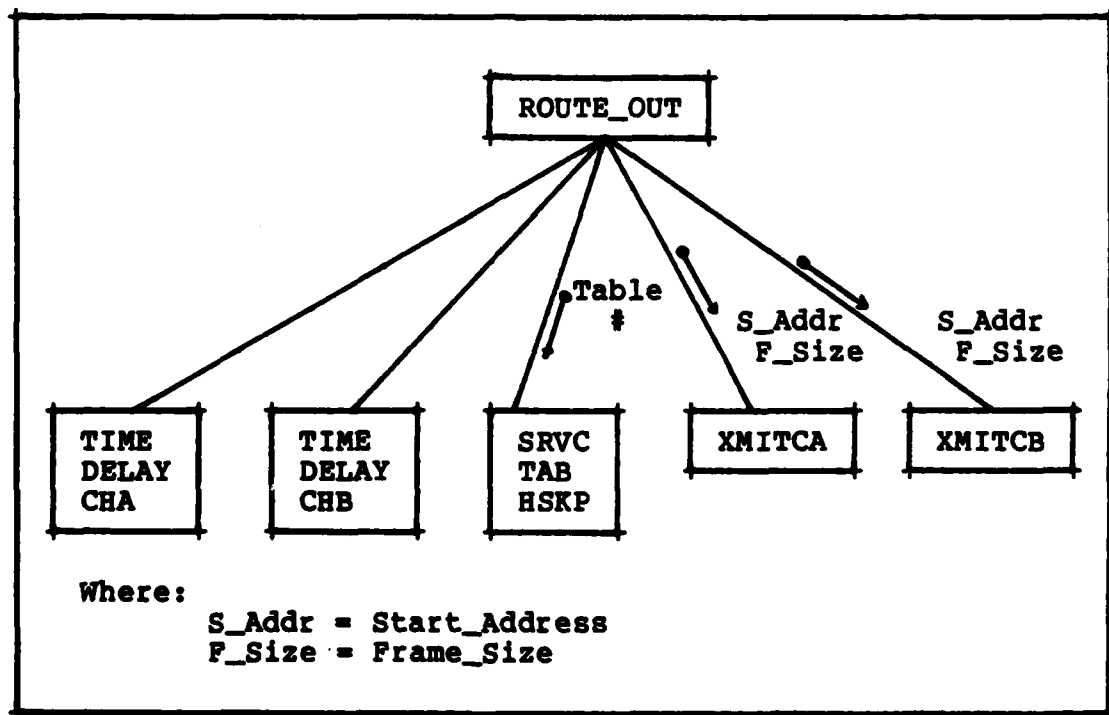


Figure 4-6. ROUTE_OUT Procedure for UNID NOS

The following describes, in Pseudo-English, the Procedure ROUTE_OUT as shown in Figure 4-6:

```

ENTER ROUTE_OUT Procedure
If a FRAME is present in OUTFRAME_CHA_TB Then
  If DESTINATION is valid
    Then
      If TIMCHA = FALSE Then
        TIME_DELAY_CHA
      Endif
      If ACKNOWLEDGE_A = FALSE and TIMCHA = TRUE Then

```

```

        Transmit FRAME out network Channel-A
        Set ACKNOWLEDGE_A = FALSE
        Start TIME_DELAY_CHA
    Endif
    If ACKNOWLEDGE_A = TRUE Then
        Update OUTFRAME_CHA_TB pointers
        Set SEQ_BIT_A = NOT SEQ_BIT_A
    Endif
Else
    Update Error Status Table
    Update OUTFRAME_CHA_TB pointers
Endif
Endif

If a FRAME is present in OUTFRAME_CHB_TB Then
    If DESTINATION is valid
    Then
        If TIMCHB = FALSE Then
            TIME_DELAY_CHB
        Endif
        If ACKNOWLEDGE_B = FALSE and TIMCHB = TRUE Then
            Transmit FRAME out network Channel-B
            Set ACKNOWLEDGE_B = FALSE
            Start TIME_DELAY_CHB
        Endif
        If ACKNOWLEDGE_B = TRUE Then
            Update OUTFRAME_CHB_TB pointers
            Set SEQ_BIT_B = NOT SEQ_BIT_B
        Endif
    Else
        Update Error Status Table
        Update OUTFRAME_CHB_TB pointers
    Endif
Endif

```

END ROUTE_OUT Procedure

UNID Memory Map and Linking Procedures

With all of the changes described, the memory map and linking procedures on the Zilog MCZ 1/25 Computer System of both the UNID's Local and Network Operating Systems were subsequently changed. Figure 4-7 gives the updated memory map for both the Local and Network Operating Systems of UNID#1 (UNID#2 is identical except for the module names). For a more detailed version of the memory map, refer to the DELNET Monitor User's Guide in Appendix A. As the UNIDs are modified

and expanded in the future, the memory map as shown in Figure 4-7 will change. It is advised that an updated version of the UNID's memory map be kept at all times.

Memory (Hex)	UNID Local Operating System (Z-80 Processor)	UNID Network Operating System (Z-80 Processor)	Memory (Hex)
0000 0FFF	LOCAL SYSTEM ROM	NETWORK SYSTEM ROM	0000 0FFF
1000 1FFF	LOCAL SYSTEM RAM	NETWORK SYSTEM RAM	1000 1FFF
2000 219C	L.VINT_U1	N.INSIO_U1	2000 21ED
219D 3B19	L.TAB_U1	N.TAB_U1	21EE 3770
3B20 5B41	L.MAIN_U1	N.MAIN_U1	3771 4D24
5B42 6FFF	NOT USED	NOT USED	4D25 6FFF
7000 7233	PLZ.MATH		7000 7233
7234 7FFF	NOT USED		7234 7FFF
8000 8081	U.LIB_U1		8000 8081
8082 8B67	U.SHTAB_U1		8082 8B67
8B68 FFFF	NOT USED		8B68 FFFF

Figure 4-7. UNID Operating System Memory Map

In order to create the memory map shown in Figure 4-7,
the following PLZ Linking commands were used:

UNID#1:

Local Operating System:

```
PLINK $=2000 L.VINT_U1 L.TAB_U1 L.MAIN_U1
      $=7000 PLZ.MATH
      $=8000 U.LIB_U1 U.SHTAB_U1
```

Network Operating System:

```
PLINK $=2000 N.INSIO_U1 N.TAB_U1 N.MAIN_U1
      $=7000 PLZ.MATH
      $=8000 U.LIB_U1 U.SHTAB_U1
```

UNID#2:

Local Operating System:

```
PLINK $=2000 L.VINT_U2 L.TAB_U2 L.MAIN_U2
      $=7000 PLZ.MATH
      $=8000 U.LIB_U2 U.SHTAB_U2
```

Network Operating System:

```
PLINK $=2000 N.INSIO_U2 N.TAB_U2 N.MAIN_U2
      $=7000 PLZ.MATH
      $=8000 U.LIB_U2 U.SHTAB_U2
```

Note: The PLZ.MATH module must be linked to L.MAIN_U1,
L.MAIN_U2, N.MAIN_U1, and N.MAIN_U2 due to the constraints of
the Zilog MCZ 1/25 when using multiplication and division of
integers and numbers.

Zilog Software

As specified by the requirements outlined in Chapter 2, the Host will contain the Transport, Session, Presentation, and Application Layer protocols. The Zilog MCZ 1/25 Computer System was chosen as the first computer to be programmed to act as a DELNET Host because of its compatibility with the UNID software.

The basic structure of the software within the Zilog follows very closely the software structure used within the UNIDs. Figure 4-8 gives the internal data tables used to transfer the 128-byte data blocks to and from the UNID (See Appendix E for the Data Dictionary and Appendix F for the listings of the Zilog_Host software).

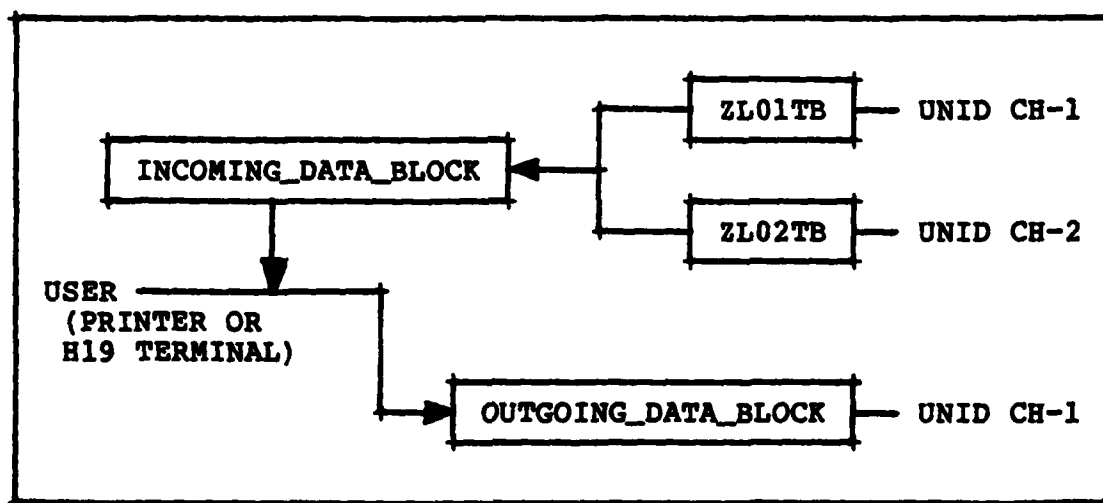


Figure 4-8. Zilog Operating Systems' Tables

The main driver for the Zilog_Host operating system was tailored similar to that of the UNIDs. Figure 4-9 illustrates the MAIN module of the Zilog_Host.

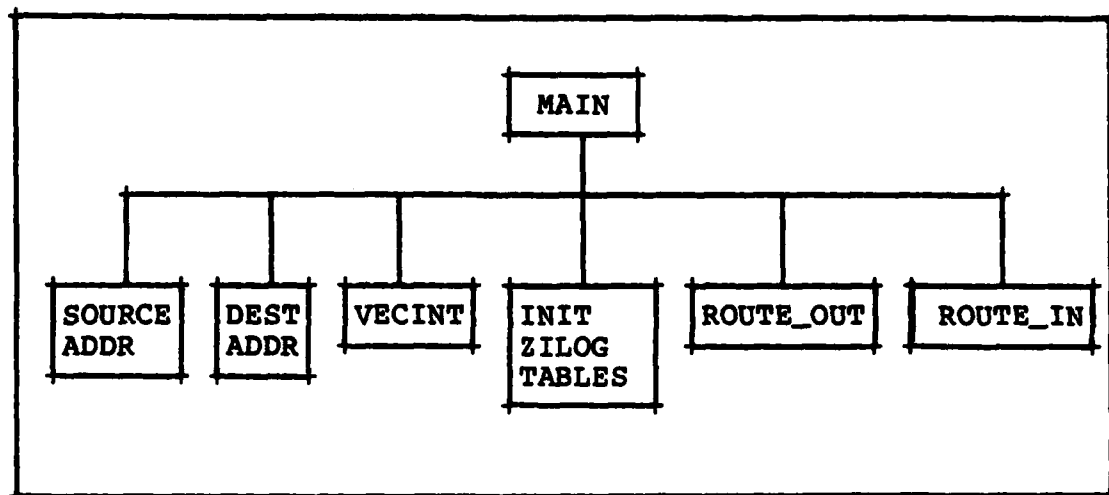


Figure 4-9. MAIN Procedure for Zilog Operating System

As with the UNIDs, the Procedures ROUTE_IN and ROUTE_OUT control the data transfer within the Operating System. The Zilog ROUTE_IN structure chart is shown in Figure 4-10 while the ROUTE_OUT structure chart is shown in Figure 4-11.

Due to the byte manipulations possible within the Zilog Operating System, the Source and Destination addresses were subdivided into byte segments for software implementation.

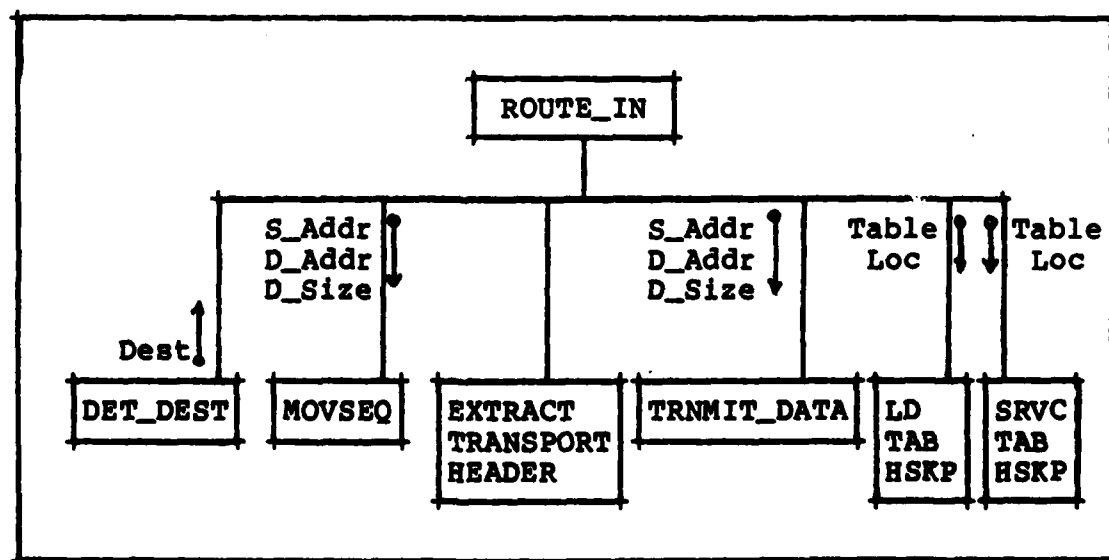


Figure 4-10. ROUTE_IN Procedure for Zilog Operating System

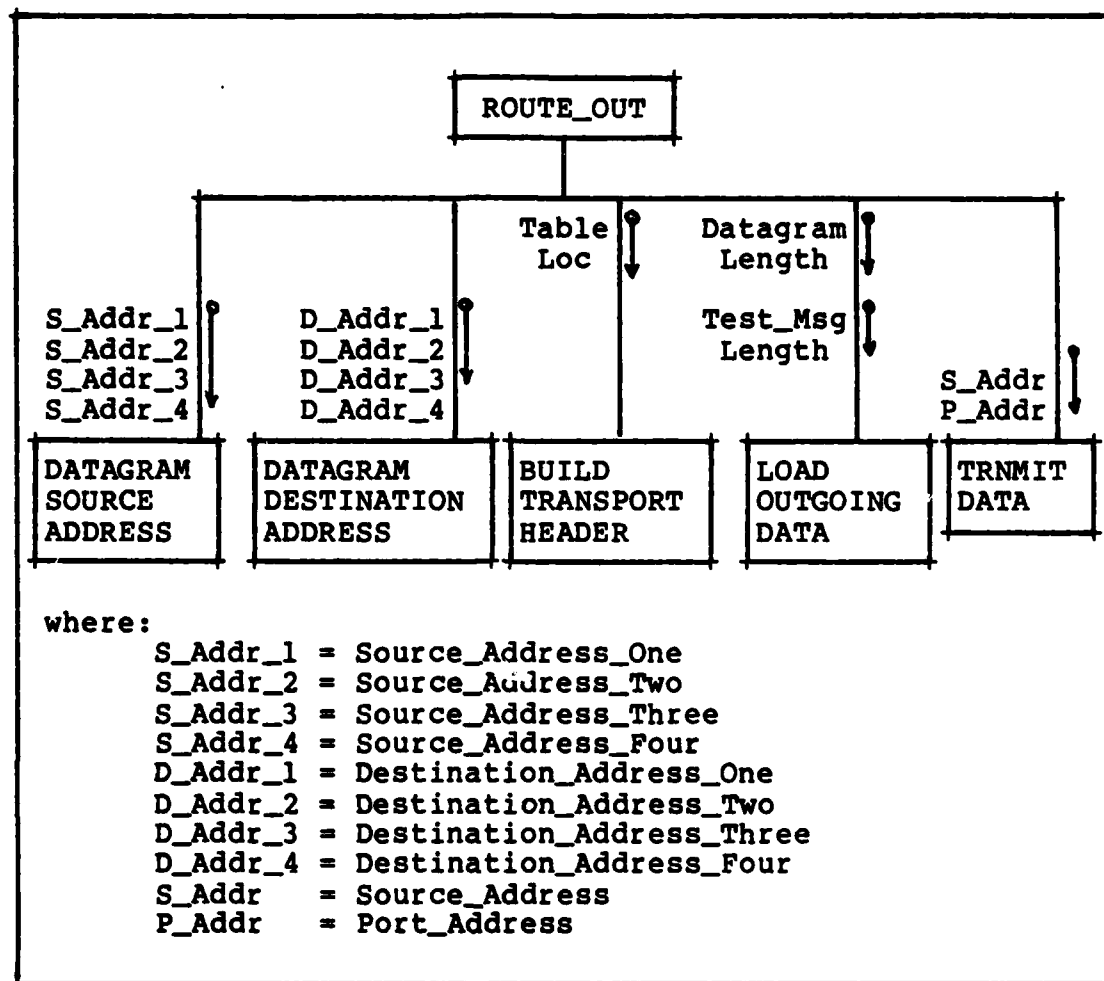


Figure 4-11. ROUTE_OUT Procedure for Zilog Operating System

As discussed in Chapter 2, the Zilog MCZ 1/25 Computer System will be responsible for implementing the upper four protocol layers (See Appendix C for protocol standards) with the first implementation of the Transport Layer being the Datagram Option. Figure 4-12 illustrates the structure chart for the procedures EXTRACT_TRANSPORT_HEADER and BUILD_TRANSPORT_HEADER. All the sub-procedures that are listed are currently just software stubs except for the Procedure DATAGRAM_DESTINATION_ADDRESS. This is to allow a "minimal" Destination Addressing scheme within the Zilog Hostenabling

some routing to be accomplished within the DELNET.

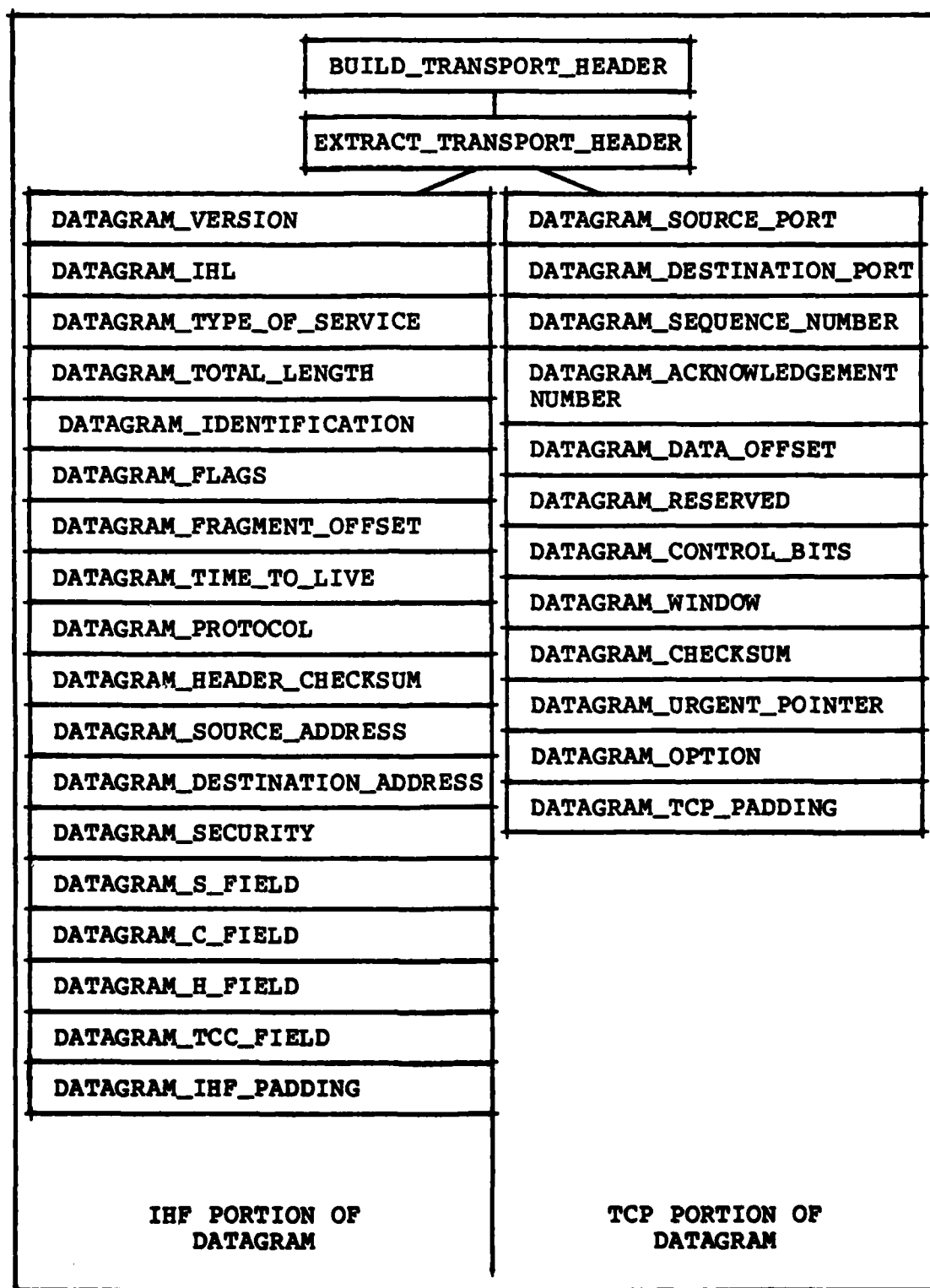


Figure 4-12. Procedures BUILD_ and EXTRACT_TRANSPORT_HEADER

In Pseudo-English the processing of the ROUTE_IN
Procedure is as follows:

```
Enter ROUTE_IN Procedure
Set DATAGRAM_LENGTH to 56
If a 128-byte data block is present in ZL01TB Then
  Move the 128-byte data block to INCOMING_DATA_BLOCK
  EXTRACT_TRANSPORT_HEADER
  Determine data block's DESTINATION
  If DESTINATION = PRINTER Then
    TRANSMIT 128-byte data block to PRINTER
  Endif
  If DESTINATION = H19 TERMINAL Then
    TRANSMIT 128-byte data block to H19 TERMINAL
  Endif
  Updata ZL01TB pointers
Endif

If a 128-byte data block is present in ZL02TB Then
  Move the 128-byte data block to INCOMING_DATA_BLOCK
  EXTRACT_TRANSPORT_HEADER
  Determine data block's DESTINATION
  If DESTINATION = PRINTER Then
    TRANSMIT 128-byte data block to PRINTER
  Endif
  If DESTINATION = H19 TERMINAL Then
    TRANSMIT 128-byte data block to H19 TERMINAL
  Endif
  Updata ZL02TB pointers
Endif

End ROUTE_IN Procedure
```

In Pseudo-English the processing of the ROUTE_OUT
Procedure is as follows:

```
Enter ROUTE_OUT Procedure
Zero out OUTGOING_DATA_BLOCK Table
Determine 128-byte data block's SOURCE_ADDRESS
Determine 128-byte data block's DESTINATION_ADDRESS
Build a 128-byte DATAGRAM by calling:
  BUILD_TRANSPORT_HEADER
  LOAD_OUTGOING_DATA at end of DATAGRAM Header
TRANSMIT DATAGRAM to the UNID
End ROUTE_OUT Procedure
```

Chapter Summary

This chapter first outlined all the changes that had to be made to the hardware and software of both the local and network operating systems of the UNID. Extensive changes were required in order to align the system with the protocol standard of Appendix C.

The last part of this chapter outlined the basic structure of the Zilog MCZ software which will allow a 128-byte Datagram to be sent to the UNID. The current software contains software stubs with the exception of the Destination_Address procedure.

The next chapter discusses the results of the software testing as outlined in Chapter 3 and included in Appendix D.

V. DELNET SOFTWARE TESTING AND VALIDATION RESULTS

Introduction

This chapter discusses the results of the software testing performed on both UNIDs and the Zilog MCZ 1/25 Computer System as prescribed by the Software Test Plan of Chapter 3. A detailed listing of the software testing procedures performed can be found in Appendix D.

To aid in the software testing, over 100 software test points were included in the operating software of both UNIDs and the Zilog computer system. These test points can be found in the software listings located in Appendix F. As the software is expanded and improved, these test points can be modified and/or removed as required. When the UNIDs become fully operational, it is recommended that most of these test points be eliminated to ensure maximum operating speed of the UNIDs.

Phase I: UNID Local Side Data Transfer

This testing phase validated the local side modules, L.MAIN, L.LIB, L.TAB, U.LIB, and U.SHTAB, for both UNID#1 and UNID#2. The testing was divided into five stages.

The first stage tested the internal data transfer from the four input tables (LC01TB, LC02TB, LC03TB, and LC04TB) to the Local-to-Local Table (LCLCTB) destined for the four local channels. In each case the data was correctly routed to the LCLCTB table with the correct local destination.

The second stage tested the internal data transfer from the four input tables to the Local-to-Network Table (LCNTTB)

destined for the network side of the UNID. In each case the data was first converted to a Data_Packet correctly and subsequently routed to the network side of the UNID.

The third stage tested the internal data transfer from the Network-to-Local Table (NTLCTB) to each of the four local channels. In each case the Data_Packet header was correctly removed and the resultant 128-byte data block was sent out the correct local channel.

The fourth stage tested the data transfer from the Local-to-Local Table (LCLCTB) to each of the four local channel ports. In each case the data block was correctly routed out the designated local channel port.

The fifth and last stage tested the internal detection of invalid Control_Code, Country_Code, and Network_Code. There were twelve tests performed with each test correctly identifying the error and the Error_Table subsequently incremented.

Phase II: UNID Network Side Data Transfer

This testing phase validated the network side modules; N.MAIN, N.TAB, N.LIB, U.LIB, and U.SHTAB. This testing phase was divided into three stages.

The first stage tested the internal data transfer from the Local-to-Network Table (LCNTTB) to each of the two network ports. In each case the Data_Packet was converted to an I-Frame and routed out the correct network port.

The second stage tested the internal data transfer from the two network channels (NT01TB, and NT02TB) to the Network-

to-Local Table (NTLCTB). In each case the I-Frame was sent to the NTLCTB as a correct Data_Packet, with an S-Frame indicating receipt of a correct I-Frame sent back to the sending UNID.

The third and last stage tested the internal Network-to-Network data transfer. There were two tests performed, one in each direction on the network ring. In each test the created I-Frame was correctly routed to the opposite network table. Also an S-Frame was subsequently created and inserted in the network table from which the I-Frame originated indicating correct reception (as if it originated from another UNID) of the I-Frame.

Phase III: Local-to-Network & Network-to-Local Data Transfer

This phase tested the interface between the local and network sides of the UNID. This testing phase was divided into six stages.

The first stage tested the data transfer from Local Channel-1 to each of the two outgoing network channels. In both cases the 128-byte data block was correctly converted into a 135-byte I-Frame and sent out the network port.

The second stage tested the data transfer from Local Channel-2 to each of the two outgoing network channels. In both cases the 128-byte data block was correctly converted into a 135-byte I-Frame and sent out the network port.

The third stage tested the data transfer from Local Channel-3 to each of the two outgoing network channels. In both cases the 128-byte data block was correctly converted

into a 135-byte I-Frame and sent out the network port.

The fourth stage tested the data transfer from Local Channel-4 to each of the two outgoing network channels. In both cases the 128-byte data block was correctly converted into a 135-byte I-Frame and sent out the network port.

The fifth stage tested the data transfer from Network Channel-A (NT01TB) to each of the four local channels. In each test the incoming 135-byte I-Frame was correctly sent out the designated local channel as a 128-byte data block.

The sixth and last stage tested the data transfer from Network Channel-B (NT02TB) to each of the four local channels. In each of the tests the incoming 135-byte I-Frame was correctly sent out the designated local channel as a 128-byte data block.

Phase IV: Local-to-Local Loopback Data Transfer

This phase of testing validated the local side module L.VINT. This phase was divided into four stages.

The first stage tested the data transfer from Local-Channel-1 to each of the other three local channels via a loopback RS-232 cable. In each case the 128-byte data block was correctly routed out and subsequently received into the receiving local channel table.

The second stage tested the data transfer from Local-Channel-2 to each of the other three local channels via a loopback RS-232 cable. In each case the 128-byte data block was correctly routed out and subsequently received into the receiving local channel table.

The third stage tested the data transfer from Local-Channel-3 to each of the other three local channels via a loopback RS-232 cable. In each case the 128-byte data block was correctly routed out and subsequently received into the receiving local channel table.

The fourth and last stage tested the data transfer from Local-Channel-4 to each of the other three local channels via a loopback RS-232 cable. In each case the 128-byte data block was correctly routed and subsequently received into the receiving local channel table.

Phase V: Local-to-Local Data Transfer

This phase of testing validated the Zilog_Host software modules; ZILOG_APPLICATION, ZILOG_TRANSPORT, ZILOG_PHYSICAL, ZILOG_LIB, and ZILOG_TABLES. Since the data are put on line as ASCII characters, the Spinwriter was unable to print out the ASCII equivalent of certain hexadecimal values; therefore, the Datagram header was composed of printable characters (2F = /) with only the Source and Destination values in proper hexadecimal values. Figure 5-1 is provided as a conversion between ASCII and hexadecimal for the Spinwriter.

Bits					Comm. Codes		Printable Characters						
B7 _____					0	0	0	0	1	1	1	1	
B6 _____					0	0	1	1	0	0	1	1	
B5 _____					0	1	0	1	0	1	0	1	
B4	B3	B2	B1	R/C	0	1	2	3	4	5	6	7	
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p	
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q	
0	0	1	0	2	STX	DC2	"	2	B	R	b	r	
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t	
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w	
1	0	0	0	8	BS	CAN	(	8	H	X	h	x	
1	0	0	1	9	HT	EM	)	9	I	Y	i	y	
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	11	VT	ESC	+	;	K	[	k	{	
1	1	0	0	12	FF	FS	,	<	L	\	l		
1	1	0	1	13	CR	GS	-	=	M	]	m	}	
1	1	1	0	14	SO	RS	.	>	N	^	n	~	
1	1	1	1	15	SI	US	/	?	O	_	o	DEL	

Figure 5-1. ASCII Code Chart

This phase of testing was divided into four stages as follows:

The first stage tested the data transfer from the zilog_Host to the UNID via Local Channel-1 and back. There were two test messages sent and each was correctly displayed

on the H19 terminal.

The second stage tested the data transfer from the Zilog_Host to the UNID via Local Channel-2 and back. There were two test messages sent and each was correctly displayed on the H19 terminal.

The third stage tested the data transfer from the Zilog_Host to the UNID via Local Channel-3 and back. There were two test messages sent and each was correctly displayed on the H19 terminal.

The fourth and last stage tested the data transfer from the Zilog_Host to the UNID via Local Channel-4 and back. There were two test messages sent and each was correctly displayed on the H19 terminal.

Phase VI: Network-to-Network Loopback Data Transfer

This phase of testing validated the network side module N.INSIO for both UNID#1 and UNID#2. This phase of testing was divided into two stages.

The first stage tested the data transfer from Network Channel-A to Network Channel-B. The I-Frame was sent out with a corresponding S-Frame received correctly.

The second stage tested the data transfer from Network Channel-B to Network Channel-A. The I-Frame was sent out with a corresponding S-Frame received correctly.

Phase VII: Host-to-Host Data Transfer via Two-UNID Network

This testing phase was not carried out as prescribed by the Test Plan due to hardware problems on UNID#1.

Phase VIII: Host-to-Host Data Transfer via 18-pair Cable

This testing phase was not carried out as prescribed by the Test Plan due to hardware problems on UNID#1; however, the 18-pair cable was checked by an ohmmeter and found to be installed correctly. A test was conducted with the RS-232 cable by connecting the Zilog to the UNID's Local Channel-1 via 1100 feet of RS-232 cable. At 19.2 Kbps there were too many bit errors for the UNID to correctly route the incoming 128-byte data blocks. This is to be expected as the RS-232 standard specifies a maximum distance of 50 feet.

Phase IX: Host-to-Host Data Transfer via Three-UNID Network

This phase of testing was not carried out due to UNID#3 not being fully operational.

Chapter Summary

This chapter summarized the detailed software testing that was performed on both UNIDs and the Zilog MCZ 1/25 Computer System. Six of the nine phases of testing prescribed in the Test Plan were successfully carried out with Phase VIII partially completed.

VI. CONCLUSIONS AND RECOMMENDATIONS

Introduction

The purpose of this investigation was to lay a firm foundation for future software design and implementation of the DELNET. The first complete draft of the DELNET Protocol Standard was developed and the software within the UNIDs modified to follow the prescribed standard. Most of the DELNET Test Plan was accomplished; however, the problems with UNID#1 hardware precluded its completion.

This investigation has laid a broad foundation upon which the UNIDs can be developed and expanded within the prescribed protocol standards. It has brought the UNID software within the protocol standards as set forth by the ISO Reference Model (Ref 26).

Conclusions

The design and testing techniques performed on the Data Link and Network Layers of the UNID in the previous thesis effort were found inadequate (Ref 19). Therefore, great time and effort was devoted to the reconstruction of the Local and Network Sides of the UNIDs to align them with the prescribed protocol standards.

At the start of this investigation only about 5% of the X.25 Protocol Standard was implemented on the UNIDs. Currently it is estimated that about 25% of X.25 has been implemented. The only portion of X.25 that is currently implemented is the Datagram Option, with the more difficult part, the Virtual Circuit, yet to be designed and

implemented. This path was taken to allow the UNIDs to become operational on a limited scale.

Although much design and implementation was accomplished during this investigation effort, no attempt was made to:

1. Design, implement or test the upper protocol layers on any other computer system other than the Zilog MCZ 1/25.

2. Design, implement or test any error detection or correcting algorithms.

3. Totally implement X.25 protocol within the first three protocol layers within the UNIDs.

4. Perform any permanent hardware modifications within the UNIDs.

5. Perform any network analysis on the DELNET, such as throughput, network utilization, or bandwidth optimization.

Lastly, from the results of this investigation, the Zilog Operating System needs to be upgraded in order for it to handle the large software programs anticipated for the Application, Presentation, Session, and Transport Layers. For example, the size of the ZILOG_TRANSPORT.S module is currently 634 blocks (1 Block=80 Bytes). When the module is compiled the created ZILOG_TRANSPORT.L module is 1022 blocks. The ZILOG_TRNAsPORT.Z module is 97 blocks and the resultant ZILOG_TRANSPORT.OBJ module is also 97 blocks. Before the intermediate modules are deleted there is only 67 free blocks left on the Zilog disk. As the ZILOG_TRANSPORT.S module is expanded during the next thesis effort this will subsequently cause problems.

Recommendations

Because this investigation effort is a continuation of previous research, the overall recommendation is to continue with the DELNET development project as this project has provided a learning experience (Refs 1, 4, 18, 19, 27, 34, and 37) about networking that is unavailable in other investigation areas. The following specific recommendations are provided:

1. Incorporation of the total CCITT X.25 Protocol Standard. Since the UNIDs will correctly pass the basic structure of Datagrams, the next step is to do a top-down design of the Virtual Connection option available under X.25. This design should be carefully done using the Structure Analysis Design Technique (SADT).

2. Incorporation of total RS-232 and RS-449 Standards. Currently, the UNIDs use very little of the total capabilities offered by both RS-232 and RS-449 Standards. It is recommended that their capabilities be incorporated into the hardware of the UNIDs so that future software development can utilize their capabilities to the fullest.

3. UNID Start-up. Currently, the Zilog MCZ 1/25 Computer System is used initially to start up the UNIDs. As the DELNET is expanded to incorporate more and more Local Hosts, each host (in its particular language which may not be compatible with the PLZ Language of the UNIDs) must keep this start-up software and the software of the UNIDs. This will pose a significant problem, which is that the PLZ and Zilog

Assembly Language Modules of the UNID's software will have to reside in foreign machines such as the INTEL 432, and the VAX 11/780. Therefore, it is recommended that research and a top-down analysis be performed enabling either the Zilog MCZ 1/25 Computer System to start-up the UNIDs through UNID#1 or another computer system to perform the "Wake-up function". Whichever computer system is used, this design will involve Supervisory Frames sent on the Network, first to "Wake-up" the UNIDs and secondly to pass the necessary Operating System Software. This will alleviate the Local Hosts from having to store the UNIDs' software.

4. Role of UNID_0 in the DELNET. As designed, UNID_0 will play a special role within the DELNET. This UNID will control the traffic within its particular Country_Code and alert all of its UNIDs of the status of the Country on a periodic basis. This UNID will also monitor the status of other connecting Countries thru their UNID_0. It is therefore recommended that research be performed to develop the design methodology and software required enabling UNID_0 to perform its required supervisory functions. Since the Z86 UNID is still in its early stages of software development, it is further recommended that this UNID be designated UNID_0.

5. Implementation of NBS Transport Layer. AFIT received from the National Bureau of Standards (NBS) a copy of the Transport Layer used on a VAX 11/780 under the UNIX Operating System. Since the VAX 11/780 will be connected into the DELNET, it is recommended that this software be used. The

primary advantage to this is that it is currently running at the NBS so if problems arise they can and have shown a willingness to aid us whenever possible.

6. Parallel development for the DELNET. As prescribed in the Protocol Standard of Appendix C, the upper four protocol layers will reside within the Host. At this point in the DELNET implementation, development of each Host's software can be developed in parallel to that of the Zilog MCZ 1/25 Computer System. Currently, there is only a minimum Datagram option available using the Zilog system. At this point it is recommended that the Zilog_Host software be incorporated within the Intel/432 and LSI-11 systems so that Phases VII, VIII, and IX of the Test Plan can be completed.

7. Handshaking protocol for the UNIDs and the Zilog. As currently implemented, there is no software handshaking within the UNIDs or Zilog to: a) stop processing, b) resume processing, and c) recover lost Datagrams. With more Hosts attached to the UNIDs, the ten shared tables and the ten local tables per local channel, plus the ten network tables, may not be enough to handle large numbers of data transfers. It is therefore recommended that the UNIDs' software be expanded to include handshaking between: Zilog-UNID, UNID Local-UNID Network, and UNID-UNID. This will be a large undertaking but can be accomplished in parallel with the other avenues of software development.

8. Completion of Software Test Plan. As noted in the testing results section of this investigation (Chapter 5),

the software test plan was not finished due to hardware problems with UNID#1. Therefore, it is recommended that once UNID#1 is repaired the software test plan be completed, especially Phase's VII and IX which were designed to test the dual-network ring.

In conclusion, a large amount of time and resources have gone into the design, implementation, and testing of the UNIDs within the DELNET. There is, however, more development that has to be accomplished before the UNIDs can truly pass data to and from a local area network. It is felt however, that this project is truly worthwhile and should be continued in the years to come as a learning process in the fields of computer networking and protocol design.

Bibliography

1. Baker, L., Capt. USAF Prototype and Software Development for Universal Network Interface Device, MS thesis. AFIT-GCS-EE-80D-4. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1980. (AD A100787).
2. Ball, A., et al. "Videotex Networks," IEEE Computer, 13 (12): 8-14 (December 1980).
3. Blumer, T.; Tenney, R., "A Formal Specification Technique and Implementation Method for Protocols," Computer Networks, 6: 201-217 (June 1982).
4. Brown, E., Capt. USAF Prototype Universal Network Interface Device, MS thesis. AFIT-GE-EE-79-8. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1979. (AD A080173).
5. Bunch, S. "Control Structure Overhead in TCP," IEEE Trends and Applications: Computer Network Protocols, 121-127 (May 1980).
6. Burruss, J. Features of the Transport and Session Protocols. Draft Report ICST/HLNP-80-1, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 3-66, March 1980.
7. Burruss, J. et al. Specification of the Session Protocol. Draft Report ICST/HLNP-81-2, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 108-118, March 1981.
8. Ceferin, J., "Modular Protocols Improve Industrial Network Control," Computer Design, 59-62 (January 1983).
9. Callon, R. Features of Internetwork Protocol. Draft Report ICST/HLNP-80-8, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 1-64, July 1980.
10. "Data Processing-Open Systems Interconnection-Basic Reference Model," Computer Networks, 5: 81-118 (May 1981).
11. desJardins, R.; White, G. "ISO/ANSI Reference Model of Open Systems Interconnection," IEEE Trends and Applications: Computer Network Protocols, 47-53 (May 1980).
12. de Maine, P. "A Programming Language for Networks," Data Communications, 10(7): 70-76 (July 1981).
13. Dineson, M. "Broadband Local Networks Enhance Communication Design," Electronic Design News, 26(5): 77-85 (March 4, 1981).

14. Features of a Message Transfer Protocol. Draft Report ICST/CBOS-82-1, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 11-42, November 1981.

15. Features of the File Transfer Protocol (FTP) and the Data Presentation Protocol (DPP). Draft Report ICST/HLNP-80-6, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 24-61, September 1980.

16. Folts, H.; Karp, H. "Data Communication - High-level Data Link Control Procedures-Frame Structure," ISO3309-1976(E). Data Communications Standards. New York, New York: McGraw-Hill Publishing Co., 1978.

17. Folts H., Karp, H., "Provisional Recommendation X.25", Data Communications Standards, 243-283, New York: McGraw-Hill Publications Co., 1978.

18. Geist, J., Capt. Development of the Digital Engineering Laboratory Computer Network: Host-to-Node/Host-to-Host Protocols. MS thesis, AFIT-GCS-EE/81D-8. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1981.

19. Hazelton, C., 1st Lt. Continued Development and Implementation of the Protocols for the Digital Engineering Laboratory Network. MS thesis, AFIT-GE-EE-82D-37. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1982.

20. Information Processing Systems - Open Systems Interconnection - Transport Protocol Specification. International Standards Organization, Washington, D.C., Draft Proposal ISO/DP-8073, 64-81, June 1982.

21. Local Network Control and Interface Facility Volume 3: X.25 UNIX Host Interface Design and Implementation. Draft Report ICST/LANP-003, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 1-35, March 1981.

22. McQuillan, J. "Local Network Architectures," Computer Design, 18(5): 18-26 (May 1979).

23. Mockapetris, P. Communication Environments for Local Networks. ISI-RR-82-103. Marina del Rey, California: Information Sciences Institute, University of California, December 1982.

24. Mokhoff, N. "Local Data Nets: Untying the Office Knot," IEEE Spectrum, 18(4): 57-59 (April 1981).

25. Naming and Addressing in Computer Based Message Systems. Draft Report ICST/CBOS-82-4, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for computer Sciences and Technology, 9-33, August 1982.

26. Open Systems Interconnection-Basic Reference Model-ISO Reference Model. DP-7498, New York: International Standards Organization, August 1981.

27. Papp, C., Capt. Prototype Delnet Using the Universal Network Interface Device. MS thesis, AFIT-GE-EE-81D-46. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1981.

28. Piatkowski, T., "The ISO-ANSI Open Systems Reference Model-A Proposal for a Systems Approach," Computer Networks, 4: 111-124 (April 1980).

29. Postel, J., Address Mappings. Report No. RFC 796. Network Working Gp: USC/Information Sciences Institute, September 1981.

30. Postel, J., Assigned Numbers. Report No. RFC 790. Network Working Gp: USC/Information Sciences Institute, September 1981.

31. Postel, J.(ed), DOD Standard Internet Protocol. Report No. RFC 791, 11-23, Arlington, Virginia: Defense Advanced Research Projects Agency (DARPA) Information Processing Techniques Office, September 1981.

32. Postel, J.(ed), DOD Standard Transmission Control Protocol. Report No. RFC 793. Arlington, Virginia: Defense Advanced Research Projects Agency (DARPA) Information Processing Techniques Office, 15-19, September 1981.

33. Postel, J., Service Mappings. Report No. RFC 795. Network Working Gp: USC/Information Sciences Institute, September 1981.

34. Ravenscroft, D., Capt. Electrical Engineering Digital Design Laboratory Communications Network. MS thesis. AFIT-GCS-EE-78-16, Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1978. (AD AO64729).

35. Requirements Analysis of Local Area Computer Networks. Draft Report ICST/LANP-81-3, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 6-20, July 1981.

36. Service Specification of a Message Transfer Protocol. Draft Report ICST/CBOS-82-3, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 12-39, February 1982.

37. Sluzevich, S., Capt. Preliminary Design of a Universal Network Interface Device. MS thesis. AFIT-GE-EE-78-41. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1978. (AD AO64059).

38. "Software for Ethernet LANs Reaches Transport Level," Computer Design, (January 1983).

39. Soi, I., "A Review of Computer-Communication Network Classification Schemes," IEEE Communications, 19(2): 24-32 (March 1981).

40. Specification for Message Format for Computer Based Message Systems. Proposed Federal Information Processing Standard, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 15-75, April 1982.

41. Specification of a Transport Protocol for Computer Communications Volume 1: Overview and Services. Draft Report ICST/HLNP-83-1, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 5-58, January 1983.

42. Specification of a Transport Protocol for Computer Communications Volume 2: Class 2 Protocol. Draft Report ICST/HLNP-83-2, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 152-174, February 1983.

43. Specification of a Transport Protocol for Computer Communications Volume 3: Class 4 Protocol. Draft Report ICST/HLNP-83-3, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 217-237, February 1983.

44. Specification of a Transport Protocol for Computer Communications Volume 4: Service Specifications. Draft Report ICST/HLNP-83-4, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 1-70, January 1983.

45. Specification of a Transport Protocol for Computer Communications Volume 5: Guidance for the Implementor. Draft Report ICST/HLNP-83-5, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 1-28, January 1983.

46. Specification of a Transport Protocol for Computer Communications Volume 6: Guidance for Implementation Selection. Draft Report ICST/HLNP-83-6, Washington, D.C.: Dept of Commerce National Bureau of Standards, Institute for Computer Sciences and Technology, March 1983.

47. Stack, T.; Dillencourt, K., "Protocols for Local Area Networks," IEEE Trends and Applications: Computer Network Protocols, 83-93 (May 1980).

48. Standards for Local Computer Networks. Draft Report ICST/LANP-80-2, Washington, D.C.:Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 8-20, March 1980.

49. Tanenbaum, A., Computer Networks. Prentice-Hall Software Series, Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1981.

50. Voydock, V.; Kent, S. Security in Higher Level Protocols: Approaches, Alternatives and Recommendations. Draft Report ICST/HLNP-81-19, Washington, D.C.:Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 159-171, September 1981.

51. Voydock, V. Service Specification of a Network Interprocess Communication Protocol. Draft Report ICST/HLNP-80-15, Washington, D.C.: Dept of Commerce National Bureau of Standards Institute for Computer Sciences and Technology, 1-43, October 1980.

52. Watson, R., "TPC Interface and End-to-End(Transport) Protocol Design Issues," Distributed Systems-Architecture and Implementation-An Advanced Course, Berlin: Springer-Verlag, 1981.

53. Weaver, T., An Engineering Assessment Toward Economic, Feasible and Responsive Base-Level Communications Through the 1980's. 1842-EEG/EEIC-TR-78-5(FOUO). Scott AFB, Illinois:1842 Electronics Engineering Group, 31 October 1977.

54. Zilog, Inc. MCZ-1/20.25 Hardware User's Manual, Manufacturer's data. Cupertino, Ca.: Zilog, Inc., 1977.

55. Zilog, Inc. PLZ User Guide (03-3096-01), Manufacturer's data. Cupertino, Ca.: Zilog, Inc., July 1979.

56. Zilog, Inc. Z80-MCB Hardware User's Manual, Manufacturer's data. Cupertino, Ca.: Zilog, Inc., 1977.

57. Zilog, Inc. Z80-MCB Software User's Manual (03-3004-01), Manufacturer's data. Cupertino, Ca.: Zilog, Inc., May 1978.

58. Zilog, Inc. Z80-RIO Operating Sys. User's Manual(03-0072-01), Manufacturer's data. Cupertino, Ca.: Zilog, Inc., September 1978.

59. Zilog, Inc. Z80-RIO Relocating Assembler and Linker User's Manual, Manufacturer's data. Cupertino, Ca.: Zilog, Inc., 1978.

60. Zilog, Inc. Z80-SIB User's Manual (03-0051-00),
Manufacturer's data. Cupertino, Ca.: Zilog, Inc., July 1978.

61. Zilog, Inc. Z80-SIO Technical Manual (03-3033-01),
Manufacturer's data. Cupertino, Ca.: Zilog, Inc., August
1978.

62. Zilog, Inc. Z80-RIO Text Editor User's Manual (Rev B),
Manufacturer's data. Cupertino, Ca: Zilog, Inc., 1978.

63. Zilog, Inc, Z80-CTC/Z80A-CTC Technical Manual,
Manufacturer's data. Cupertino, Ca: Zilog, Inc., 1977.

APPENDIX A

DELNET Monitor User's Guide

Introduction

Within the DELNET, the Zilog MCZ 1/25 Computer System will function as the DELNET Monitor and will reside within Country Code 0000. Initialization of the UNIDs using the Zilog MCZ 1/25 Computer System has become a rather involved process. Therefore, this System Monitor User's Guide is provided so that UNID connections to the Zilog MCZ 1/25 Computer System can be performed with a minimum of effort (Refs 54, 56, 58, 60, 61, and 63).

Table of Contents

This appendix is subdivided into six sections, which are:

<u>Section</u>	<u>Page</u>
Zilog Modifications.....	A-1
UNID Cable Connections and Physical Layer Enhancements.....	A-3
UNID Start-up Procedures.....	A-8
Zilog_Host Start-up Procedures.....	A-15
DELNET Cable Layout.....	A-17
Appendix Summary.....	A-19

Zilog Modifications

The Zilog MCZ 1/25 Computer System was modified to act as two Hosts with the SIB Board wired as follows (Ref 60:5-15):

1. J1: Sets addresses for CTCs and USARTs. Current connections are pins 1-12, 2-11, 3-10, 4-14, and 5-13 which indicate the following CTC and USART addressing:

1-12	CTC0	80H-83H
2-11	CTC1	84H-87H

3-10	CTC2	88H-8BH
4-14	USART 0 and 1	8CH-8FH
5-13	USART 2 and 3	90H-93H

2. J2: Specifies the baud rate clocks. Current connections are pins 1-16, 2-5, 2-6, 4-12, 4-15, 6-13, and 13-14.

3. J3: Specifies which external clocks are to be used. Current connections are pins 4 (MCB PHI/2)-11(CK/T0), 5 (On Board PHI/32)-13(CK/T2), and 6 (On Board PHI/2)-12(CK/T1).

4. J4: Sets address range for CTCs and USARTs. Current connections are pins 5-16, 1-7, and 4-6. This sets the following address ranges:

CTC		USART	
80H	SIB-CTC0-Channel-0	8C	SIB-USART0-Data
81H	SIB-CTC0-Channel-1	8D	SIB-USART0-Command
82H	SIB-CTC0-Channel-2	8E	SIB-USART1-Data
83H	SIB-CTC0-Channel-3	8F	SIB-USART1-Command
84H	SIB-CTC1-Channel-0	90	SIB-USART2-Data
85H	SIB-CTC1-Channel-1	91	SIB-USART2-Command
86H	SIB-CTC1-Channel-2	92	SIB-USART2-Data
87H	SIB-CTC1-Channel-3	93	SIB-USART3-Command
88H	SIB-CTC2-Channel-0		
89H	SIB-CTC2-Channel-1		
8AH	SIB-CTC2-Channel-2		
8BH	SIB-CTC2-Channel-3		

The following signals are present on J5, J6, J7, and J8:

1-TxDB 2-RxDB 3-RTSB 4-CTSB 5-DSRB 6-DTRB 7- +12v
8-Spare 9-TDSR 10-TDTR 11-TCTS 12-TRTS 13-TRxD 14-TTxD

5. J5: USART0 This is connected to J-112. Current pin connections are 1-14, 2-13, 3-12, 4-11, 5-11, and 6-10.

6. J6: USART1 This is connected to J-113. Current pin connections are 1-14, 2-13, 3-12, 4-11, 5-11, and 6-10.

7. J7: USART2 This is connected to J-104. Current pin connections are: 1-13, 2-14, 4-7, 5-10, 6-9, 7-8, and 8-11.

8. J8: USART3 This is connected to J-114. Current pin

connections are 1-14, 2-13, 3-12, 4-11, 5-11, and 6-10.

UNID Cable Connections and Physical Layer Enhancements

Although both the network and local ports were wired on the motherboard of the UNID (Ref 27), they were not readily accessible for use. Therefore, the first enhancement of the UNIDs within the Physical Layer was the installation of the RS-232 Local Ports and the RS-449 Network Ports on the outside of the UNIDs. Figure A-1 gives the resultant layout.

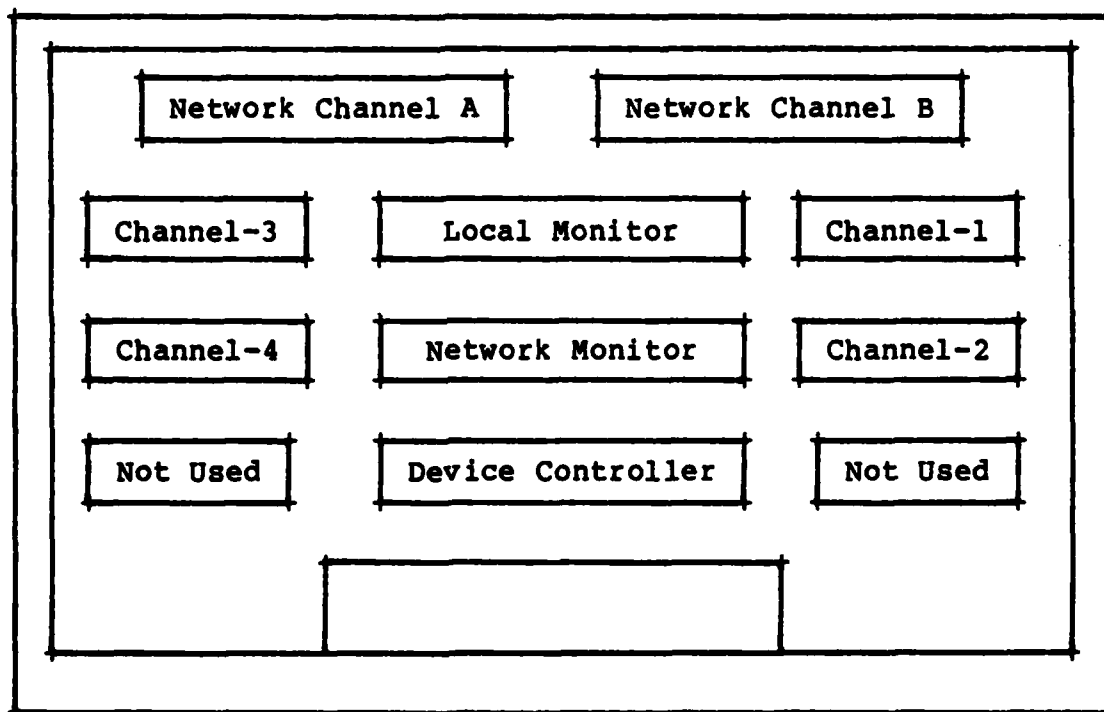


Figure A-1. UNID Network and Local Port Connections

The adopted standard on the local side (i.e. connecting the UNID with Host, Local Monitor, Network Monitor, and the Device Controller) was the RS-232 standard. However, not all the pins for an RS-232 standard are used by the above devices. Table A-1 gives the RS-232 Standard pin numbers and clarification as to what pins are and are not connected to

the listed devices.

Table A-1. UNID RS-232 Pin Assignments

RS-232 Pin Number	Function	PIN CONNECTED						
		MONITOR L	DEV N	CTR	CHANNEL 1	2	3	4
1	Protective Ground	X	X	X	X	X	X	X
2	Transmitted Data	X	X		X	X	X	X
3	Received Data	X	X		X	X	X	X
4	Request to Send	X	X		X	X	X	X
5	Clear to Send	X	X	X	X	X	X	X
6	Data Set Ready	X	X		X	X	X	X
7	Signal Ground	X	X		X	X	X	X
8	Received Line Signal Detector	X	X		X	X	X	X
9	Reserved for Testing							
10	Reserved for Testing							
11	Unassigned							
12	Secondary Receive Signal Detect							
13	Secondary Clear to Send							
14	Secondary Transmit Data			X				
15	Transmit Signal Element Timing							
16	Secondary Receive Data							
17	Receive Signal Element Timing							
18	Unassigned			X				
19	Secondary Request to Send							
20	Data Terminal Ready	X	X		X	X	X	X
21	Signal Quality Detector							
22	Ring Detector							
23	Data Signal Rate Selector(DTE)							
24	Data Signal Rate Selector(DCE)							
25	Unassigned							

The USARTs for Local Channels-2, -3, and -4 were not wired to allow handshaking between local Hosts and the UNID. However, Local Channel-1 was wired correctly. This was because Channel-1 was used to load the system software from the Zilog MCZ 1/25 Computer System into the UNID and handshaking was required to perform this task. Therefore,

Local Channels-2, -3, and -4 were rewired to allow handshaking as shown in Figure A-2.

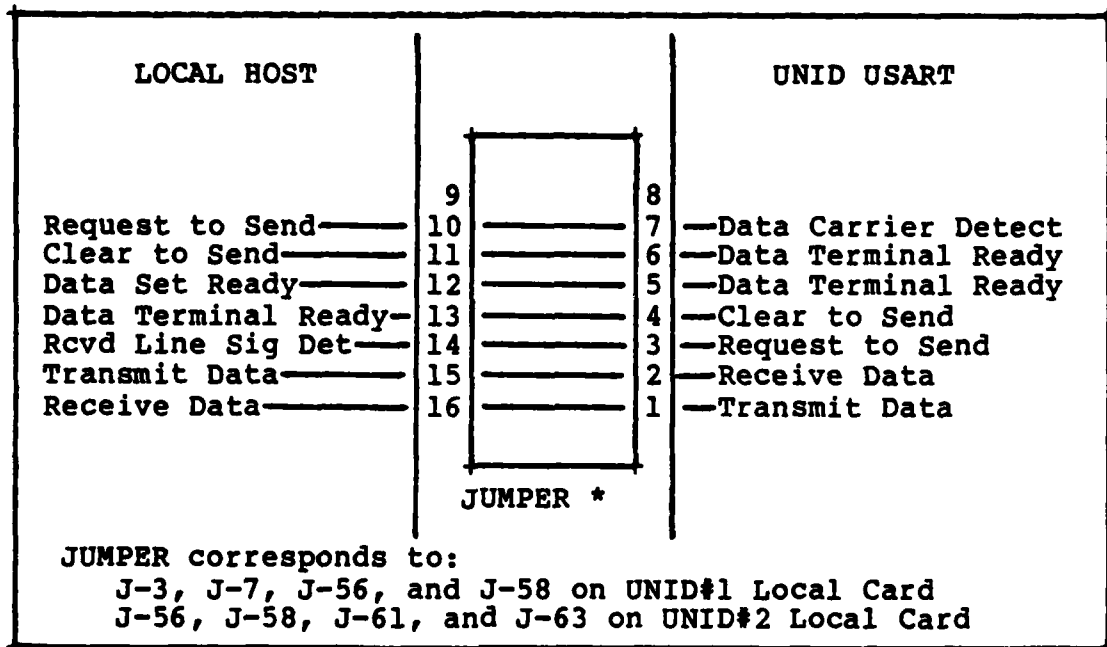


Figure A-2. UNID DCE Jumper Configuration

As mentioned, the second enhancement to the Physical Layer was the connection of the output jacks according to the RS-449 standard on the Network Side. The Network Side had temporary connections on J-53 and J-54 located on the Network Card. Therefore, a connection was made from J-53 and J-54 to the RS-449 jacks located on the back of the UNID. Table A-2 shows which pins have been connected for both Network Channels.

Table A-2. UNID RS-449 Pin Assignments

RS-449 Pin		PIN CONNECTED	
Number	Function	Network Channel-A	Network Channel-B
1	Shield	X	X
2	SI-Signaling Rate Detector		
3	Spare		
4	SD-Send Data	X	X
5	ST-Send Timing		
6	RD-Receive Data	X	X
7	RS-Request to Send	X	X
8	RT-Receive Timing		
9	CS-Clear to Send	X	X
10	LL-Local Loopback		
11	DM-Data Mode		
12	TR-Terminal Ready	X	X
13	RR-Receiver Ready	X	X
14	RL-Remote Loopback		
15	IC-Incoming Call		
16	SF-Select Frequency		
	SR-Signaling Rate Indicator		
17	TT-Terminal Timing		
18	TM-Test Mode		
19	SG-Signal Ground	X	X
20	RC-Receive Common		
21	Spare		
22	SD-Send Data	X	X
23	ST-Send Timing		
24	RD-Receive Data	X	X
25	RS-Request to Send	X	X
26	RT-Receive Timing		
27	CS-Clear to Send	X	X
28	IS-Terminal in Service		
29	DM-Data Mode		
30	TR-Terminal Ready	X	X
31	RR-Receiver Ready	X	X
32	SS-Select Standby		
33	SQ-Signal Quality		
34	NS-New Signal		
35	TT-Terminal Timing		
36	SB-Standby Indicator		
37	SC-Send Common		

Table A-3 gives the connections from J-53 and J-54 to the RS-449 pin connections.

Table A-3. UNID Network Card to RS-449 Pin Assignments

Channel-A RS-449 Pin J-53 Number		Channel-B RS-449 Pin J-54 Number	
1	-	1	-
2	22	2	22
3	25	3	25
4	30	4	30
5	24	5	24
6	27	6	27
7	31	7	31
8	1	8	1
9	19	9	19
10	13	10	13
11	9	11	9
12	6	12	6
13	12	13	12
14	7	14	7
15	4	15	4
16	+5v	16	+5v

Figure A-3 illustrates the cable connections required to connect the UNIDs with the Zilog MCZ 1/25 Computer System. Using Figure A-3 as a reference, listed below are the appropriate pin connections.

1. UNID Internal connections:

- a. J-21 is Channel One
- b. J-19 is Channel Two
- c. J-17 is Channel Three
- d. J-15 is Channel Four
- e. J-18 is Local Monitor
- f. J-16 is Network Monitor
- g. J-20 is UNID Device Controller

2. Zilog MCZ 1/25 External Connections:

- a. J-112 is Channel One of UNID#1
- b. J-113 is Channel One of UNID#2
- b. J-104 is Spinwriter Printer
- c. J-106 is H19 Terminal

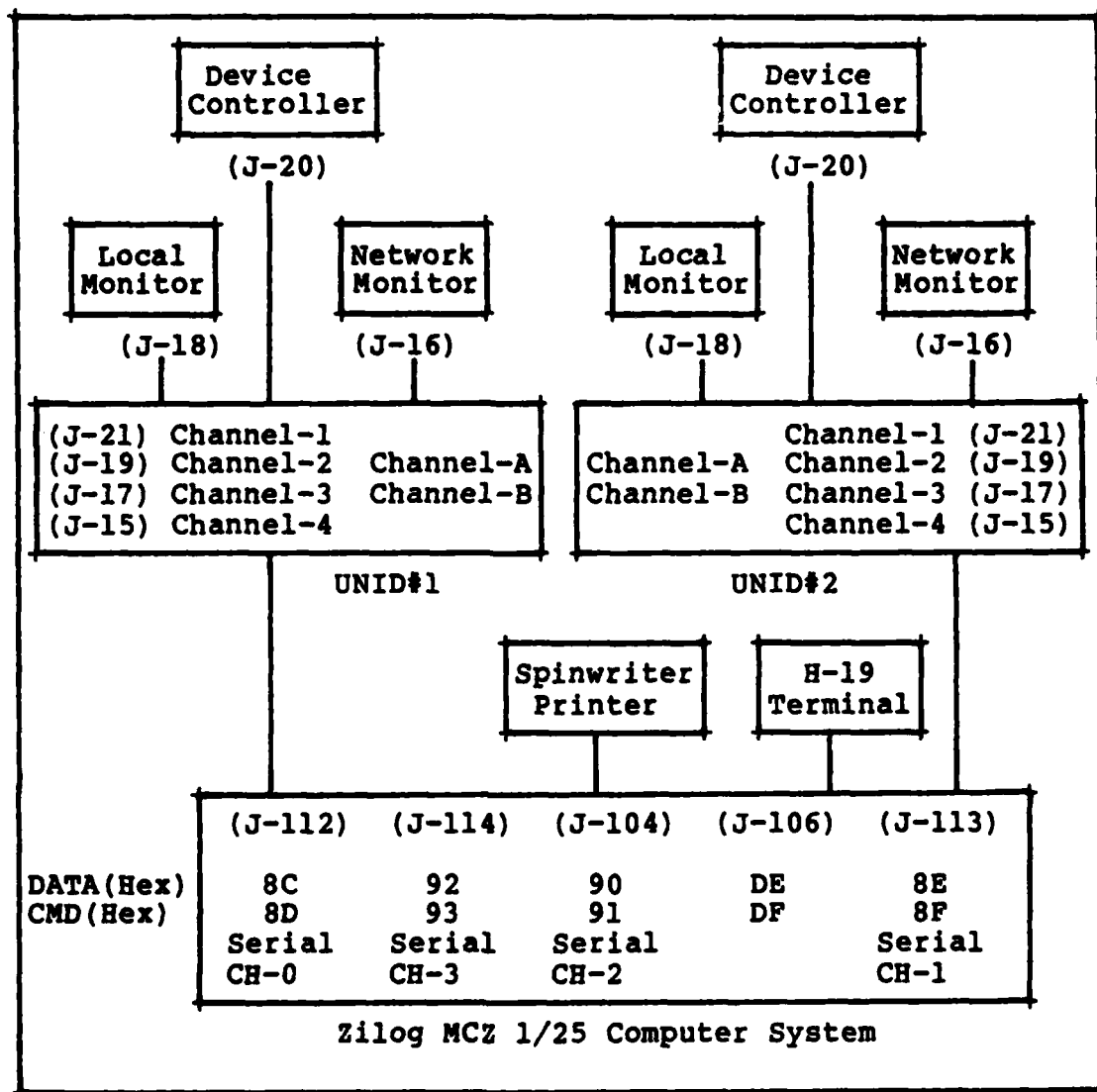


Figure A-3. UNID-Zilog Cable Connections

UNID Start-up Procedures

The following procedures will successfully bring up both UNIDs to their full operating condition:

1. Turn power on for all equipment. (Make sure Spinwriter is on and out of Local Mode.)
2. Boot up Zilog MCZ 1/25 Computer System by putting the boot floppy disk in the lower drive upside down. The disks have to be put in upside down because the drive

heads are located on the top and not the bottom of the disk head. Once the boot disk is in place, close the door and push the RETURN button on the H19 Terminal. This should boot up the system. Figure A-4 illustrates what should appear on the H19 Terminal.

\$SPRINTER active as SYSLST to Spinwriter

26 June 81: OS BOOT EDITOR AND PLZ.

This disk was built to facilitate software development with PLZ. It does not contain all the software available for the MCZ1/25 system. Refer to disk 13-3001-01A, MCZ/PDS RIO Floppy-790612, for additional software.

Please set the date, use: DATE yymmdd
RIO REL 2.06

%

Figure A-4. Zilog System Start-up Message

3. If this is the initial boot of the day, then the DATE must be set. If the DATE is not set then all files created or modified will have a ***** listed in the directory. This could cause compile problems.

4. For each UNID that is being brought on line do the following:

a. Put in the local operating system disk of the selected UNID upside down in the top drive (Drive 2).

b. Type in the command:

"DO SETUP.UNID_unid number"

For example to bring up UNID#2 type "DO SETUP.UNID_TWO". The UNID's software is interrupt driven, therefore this command

will enable UNID#1 to get programs from the Zilog MCZ 1/25 Computer System. At this time, there are only two UNIDs operational; therefore, to bring up the first UNID type "DO SETUP.UNID_ONE".

c. Next, remove the local system disk and insert the selected UNID's network operating system disk in the upper disk drive (upside down).

d. Be sure both the local and network monitors have been reset by the UNID's device controller. Using the local monitor of the UNID that is coming on line, type in the command "L N.INSIO_U2". This command will load the N.INSIO_U2 program from the Zilog MCZ 1/25 Computer System into the selected UNID.

e. At this time, the program N.INSIO_U2 must be moved over to the Network Side of the UNID. The initial design of the UNID allowed only the local monitor to interface with the Zilog MCZ 1/25 Computer System; therefore, the Network Operating System must first be loaded into the Local Side of the UNID then transferred over to the Network Side. This is accomplished with by the following commands:

M A000 2000 4FFF - Means move 4FFF bytes of memory from starting location 2000 to starting location A000. (Note: The Network Operating System is loaded into the UNID starting at memory location 2000.) Using the network monitor of the UNID coming on line (make sure it has been reset by the UNID Device Controller), type in the following: "M 2000 A000 4FFF". This moves 4FFF bytes of memory from starting location

A000 to starting location 2000.

f. At this point, the Network Operating System of the UNID is ready. What is still needed, however, is to reset the Program Counter to the start of the program. The start of the Network program is determined by the starting location of the module MAIN. This starting address is found in the Network Map. To get a Network Map, type "PRINT N.INSIO_U2.MAP" on the H19 Terminal. Figure A-5 gives an example of a Network Map. Once you have a Network map find the starting location for N.MAIN_U2. This is shown in Figure A-5 by the * and is 4E35 for this example. Also note that as the software for the Network Operating System is modified, this entry location will change. Once the entry location for MAIN is found type in the following command: "R PC (space) 0000 (space) 4E35". Now the Program Counter is reset.

g. To execute the network program, type either "G" or "GO" on the network monitor.

h. At this point you should see the following on the network monitor in rapid succession:

```
UNID#2 NETWORK OS
VERSION 27 SEP 83
EXECUTING
TP_1: ENTERING INIT_N_TAB PROCEDURE
TP_2: ENTERING INSIO PROCEDURE
TP_3: STARTING ROUTE_IN-ROUTE_OUT LOOP
TP_4: ENTERING ROUTE_IN PROCEDURE
TP_42: ENTERING ROUTE_OUT PROCEDURE
TP_43: END OF ROUTE_IN-ROUTE_OUT LOOP
TP_4: ENTERING ROUTE_IN PROCEDURE
TP_42: ENTERING ROUTE_OUT PROCEDURE
TP_43: END OF ROUTE_IN-ROUTE_OUT LOOP
```

The above indicates that the network program is running correctly on the Network Side of the UNID.

MODULE	ORIGIN	LENGTH	GLOBAL	ADDRESS	MODULE
N.INSIO_U2	2000	01FD	OUTFRAME_CHB_NS	3727	N.TAB_U2
N.TAB_U2	21FD	1583	OUTFRAME_CHB_SZ	372B	N.TAB_U2
N.MAIN_U2	3780	1753	OUTFRAME_CHB_TB	31E1	N.TAB_U2
PLZ.MATH	7000	0234	RECSEQ	8023	U.LIB_U2
U.LIB_U2	8000	0082	SNDSEQ	8053	U.LIB_U2
U.SHTAB_U2	8082	0AE6	STATTB	8AF2	U.SHTAB_U2
LINKAGE INFO	8B68	0000	STCTC2	21CD	N.INSIO_U2
GLOBAL	ADDRESS	MODULE	STCTC3	21E5	N.INSIO_U2
CTCNOA	3780	N.MAIN_U2	TIMCHA	3784	N.MAIN_U2
CTCNOB	3781	N.MAIN_U2	TIMCHB	3785	N.MAIN_U2
db?	7134	PLZ.MATH	XMITCA	2107	N.INSIO_U2
di?	701C	PLZ.MATH	XMITCB	2144	N.INSIO_U2
dsi?	7094	PLZ.MATH			
dw?	711E	PLZ.MATH	PROG:N.INSIO_U2--6B68	BYTES	
INIT_N_TAB	372D	N.TAB_U2	ENTRY: 2000		
INIT_U_SHTAB	8B06	U.SHTAB_U2			
INSIO	2080	N.INSIO_U2			
LCNTNE	85B6	U.SHTAB_U2			
LCNTNS	85B4	U.SHTAB_U2			
LCNTSZ	85B8	U.SHTAB_U2			
LCNTTB	8082	U.SHTAB_U2			
MAIN	4E35	N.MAIN_U2	*		
MAXNOA	3782	N.MAIN_U2			
MAXNOB	3783	N.MAIN_U2			
mb?	71E2	PLZ.MATH			
mdb?	712B	PLZ.MATH			
mdi?	7006	PLZ.MATH			
mdsi?	70D8	PLZ.MATH			
mdw?	7113	PLZ.MATH			
mi?	713C	PLZ.MATH			
MOVSEQ	8000	U.LIB_U2			
msi?	7194	PLZ.MATH			
mw?	71D6	PLZ.MATH			
NT01NE	2745	N.TAB_U2			
NT01NS	2743	N.TAB_U2			
NT01SZ	2747	N.TAB_U2			
NT01TB	21FD	N.TAB_U2			
NT02NE	2C91	N.TAB_U2			
NT02NS	2C8F	N.TAB_U2			
NT02SZ	2C93	N.TAB_U2			
NT02TB	2749	N.TAB_U2			
N TLCNE	8AEE	U.SHTAB_U2			
N TLCNS	8AEC	U.SHTAB_U2			
N TLC SZ	8AF0	U.SHTAB_U2			
N TLC TB	85BA	U.SHTAB_U2			
OUTFRAME_CHA_NE	31DD	N.TAB_U2			
OUTFRAME_CHA_NS	31DB	N.TAB_U2			
OUTFRAME_CHA_SZ	31DF	N.TAB_U2			
OUTFRAME_CHA_TB	2C95	N.TAB_U2			
OUTFRAME_CHB_NE	3729	N.TAB_U2			

Figure A-5. N.INSIO_U2.MAP - UNID Network OS Memory Map

i. Next, the Local Operating System of the selected UNID is loaded by first putting its Local Operating System floppy disk in the upper disk drive (upside down).

j. Using the local monitor of the UNID coming on line, type in the command "L L.VINT_U2". This should load the selected UNID's (in this case UNID#2) Local Operating System from the Zilog MCZ 1/25 Computer System to the selected UNID.

k. Again a map giving all the memory locations of the Local Side is required as with the Network Side of the UNID. Figure A-6 gives the Local Memory Map. Once a Local Memory Map is available, locate the module MAIN (this is indicated by the * in Figure A-6. Again note that as software is modified the memory locations will change) and find the starting location of MAIN (588D).

l. The Program Counter is reset to this starting location with the command: "R PC (space) 0000 (space) 588D".

m. At this point type in the command "GO" and the following should appear on the local monitor:

```
UNID#2 LOCAL OS
VERSION 27 SEP 83
EXECUTING
TP_1: ENTERING INIT_L_TAB PROCEDURE
TP_2: ENTERING INIT_U_SHTAB PROCEDURE
TP_3: ENTERING INVINT PROCEDURE
TP_4: STARTING ROUTE_IN-ROUTE_OUT LOOP
TP_5: ENTERING ROUTE_IN PROCEDURE
TP_22: ENTERING ROUTE_OUT PROCEDURE
TP_35: END OF ROUTE_IN-ROUTE_OUT LOOP
TP_5: ENTERING ROUTE_IN PROCEDURE
TP_22: ENTERING ROUTE_OUT PROCEDURE
TP_35: END OF ROUTE_IN-ROUTE_OUT LOOP
. . .
```

The above indicates that the UNID's Local Operating System is working correctly.

MODULE	ORIGIN	LENGTH	GLOBAL	ADDRESS	MODULE
L.VINT_U2	2000	019D	NTLCNS	8AEC	U.SHTAB_U2
L.TAB_U2	219D	1983	NTLCSZ	8AF0	U.SHTAB_U2
L.MAIN_U2	3B20	1E21	NTLCTB	85BA	U.SHTAB_U2
PLZ.MATH	7000	0234	RECSEQ	8023	U.LIB_U2
U.LIB_U2	8000	0082	SNDSEQ	8053	U.LIB_U2
U.SHTAB_U2	8082	0AE6	STATTB	8AF2	U.SHTAB_U2
LINKAGE INFO	8B68	0000	TDAADD	3B20	L.MAIN_U2
GLOBAL	ADDRESS	MODULE	TPRADD	3B22	L.MAIN_U2
db?	7134	PLZ.MATH	TRNMIT	208A	L.VINT_U2
di?	701C	PLZ.MATH			
dsi?	7094	PLZ.MATH			
dw?	711E	PLZ.MATH			
INIT_L_TAB	3ABB	L.TAB_U2			
INIT_U_SHTAB	8B06	U.SHTAB_U2			
INVINT	2020	L.VINT_U2			
LC01NE	269F	L.TAB_U2			
LC01NS	269D	L.TAB_U2			
LC01SZ	26A1	L.TAB_U2			
LC01TB	219D	L.TAB_U2			
LC02NE	2BA5	L.TAB_U2			
LC02NS	2BA3	L.TAB_U2			
LC02SZ	2BA7	L.TAB_U2			
LC02TB	26A3	L.TAB_U2			
LC03NE	30AB	L.TAB_U2			
LC03NS	30A9	L.TAB_U2			
LC03SZ	30AD	L.TAB_U2			
LC03TB	2BA9	L.TAB_U2			
LC04NE	35B1	L.TAB_U2			
LC04NS	35AF	L.TAB_U2			
LC04SZ	35B3	L.TAB_U2			
LC04TB	30AF	L.TAB_U2			
LCLCNE	3AB7	L.TAB_U2			
LCLCNS	3AB5	L.TAB_U2			
LCLCSZ	3AB9	L.TAB_U2			
LCLCTB	35B5	L.TAB_U2			
LCNTNE	85B6	U.SHTAB_U2			
LCNTNS	85B4	U.SHTAB_U2			
LCNTSZ	85B8	U.SHTAB_U2			
LCNTTB	8082	U.SHTAB_U2			
MAIN	588D	L.MAIN_U2			*
mb?	71E2	PLZ.MATH			
mdb?	712B	PLZ.MATH			
mdi?	7006	PLZ.MATH			
mdsi?	70D8	PLZ.MATH			
mdw?	7113	PLZ.MATH			
mi?	713C	PLZ.MATH			
MOVSEQ	8000	U.LIB_U2			
msi?	7194	PLZ.MATH			
mw?	71D6	PLZ.MATH			
NTLCNE	8AEE	U.SHTAB_U2			

PROG: L.VINT_U2 -6B68 BYTES
ENTRY: 2000

Figure A-6. L.VINT_U2.MAP - UNID Local OS Memory Map

o. Repeat the above process starting with Step 4 for each UNID that is to be brought on line.

p. To summarize the necessary steps: here is an example if UNID#2 is to be brought on line:

1. All equipment is on.
2. Insert UNID#2 Local Operating System disk.
3. Type, on the H19 Terminal, the command:
"DO SETUP.UNID_TWO".
4. Remove Local Operating System disk and insert Network Operating System disk.
5. Type, on UNID#2's Local Monitor, "L N.INSIO_U2".
6. Type, on UNID#2's Local Monitor, the command:
"M A000 2000 4FFF".
7. Type, on UNID#2's Network Monitor, the command:
"M 2000 A000 4FFF".
8. Type, on UNID#2's Network Monitor, the command:
"R PC 0000 4E35".
9. Type, on UNID#2's Network Monitor, the command:
"GO".
10. Remove Network Operating System disk and insert the Local Operating System disk.
11. Type, on UNID#2's Local Monitor, the command:
"L L.VINT_U2".
12. Type, on UNID#2's Local Monitor, the command:
"R PC 0000 588D".
13. Type, on UNID#2's Local Monitor, the command:
"GO".

Zilog_Host Start-up Procedures

The Zilog_Host will contain the upper four protocol levels as outlined by the ISO Reference Model (Application, Presentation, Session, and Transport). See Appendix C for the prescribed protocol standards and Chapter 4 for the

implementation within the DELNET (Ref 26). Currently, each protocol layer is designated a software module, i.e., Application Layer is designated Zilog_Application.

The first step in starting the Zilog_Host is to link all the modules together similar to that performed in the UNIDS software. The command is as follows:

```
PLINK $=5000 ZILOG_APPLICATION ZILOG_TRANSPORT ZILOG_PHYSICAL  
          ZILOG_TABLES ZILOG_LIB PLZ.MATH (E=MAIN)
```

The Zilog_Host software must start at 5000 due to the constraints imposed by the Zilog operating system. The PLZ.MATH module must be included as algebraic operations are used. The E=MAIN option will automatically set the programs entry point to Module Main.

The second step is to rename the linked modules to the following:

```
ZILOG_APPLICATION      to ZILOG_HOST_2  
ZILOG_APPLICATION.MAP  to ZILOG_HOST_2.MAP
```

The third step in the Zilog start-up procedures is to ensure that the Spinwriter is set for XON/XOFF protocol. This is accomplished by setting the fourth bit from the left of Switch One (located in the left front part of the Spinwriter) to the up position (sets it to a one). If the Spinwriter is currently on when this is accomplished, then the Spinwriter must be turned off to reset the protocol. Note: If this is performed and a Zilog Warmboot is desired then the warmboot will stop when the Spinwriter is queried. To continue the Zilog Warmboot, press "CTRL F" on the Spinwriter.

The fourth, and last, step is to type the following

command on the H19 Terminal: "ZILOG_HOST_2". The Zilog Start-up Header and software test points should appear on the Spinwriter indicating that the Gettysburg Address is being transmitted to the UNID via Channel-1. Note: Most of the Zilog_Host software test points have been converted to comments to speed up the transfer time. As the software is expanded these software test points can be used as needed.

DELNET Cable Layout

The DELNET cable was laid from the basement to the second floor providing the first dual-ring network at AFIT. Figure A-7 illustrates the resultant cable path.

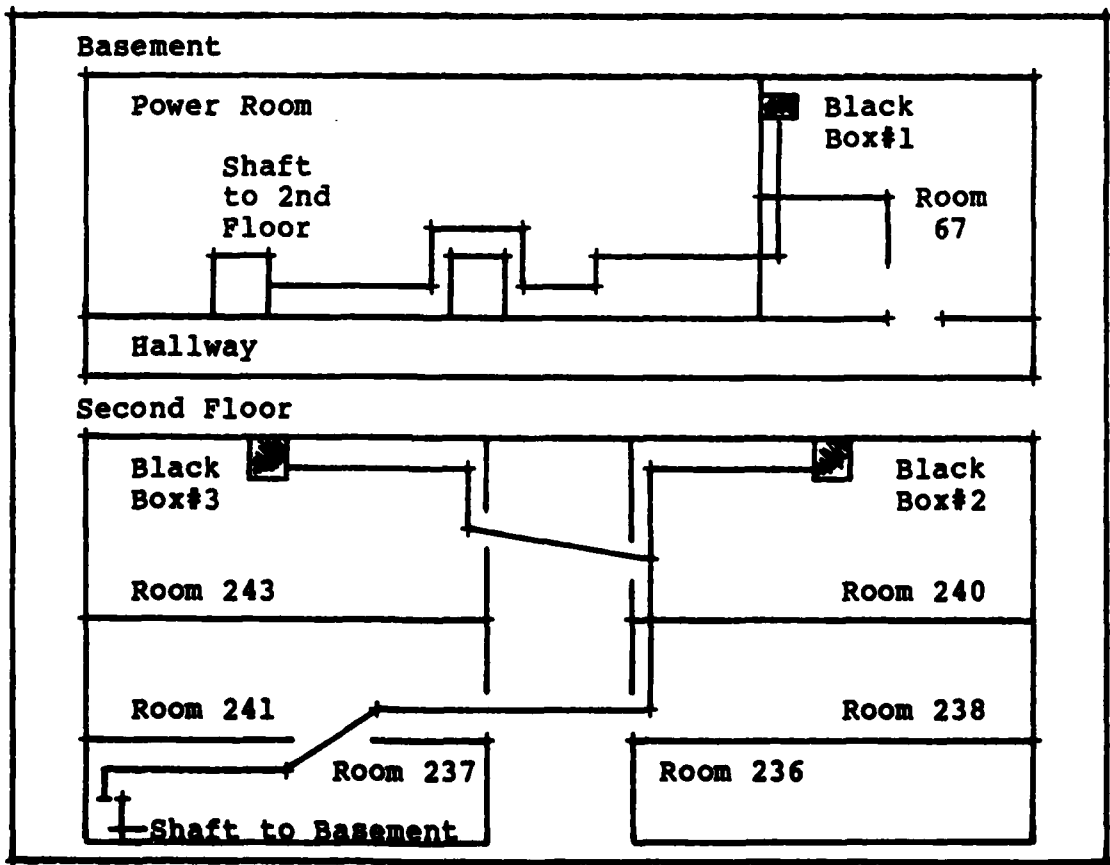


Figure A-7. DELNET Cable Path

As mentioned, the cable provides a dual-ring for both an

RS-232 and RS-449 link. Figure A-8 illustrates the cable box connections.

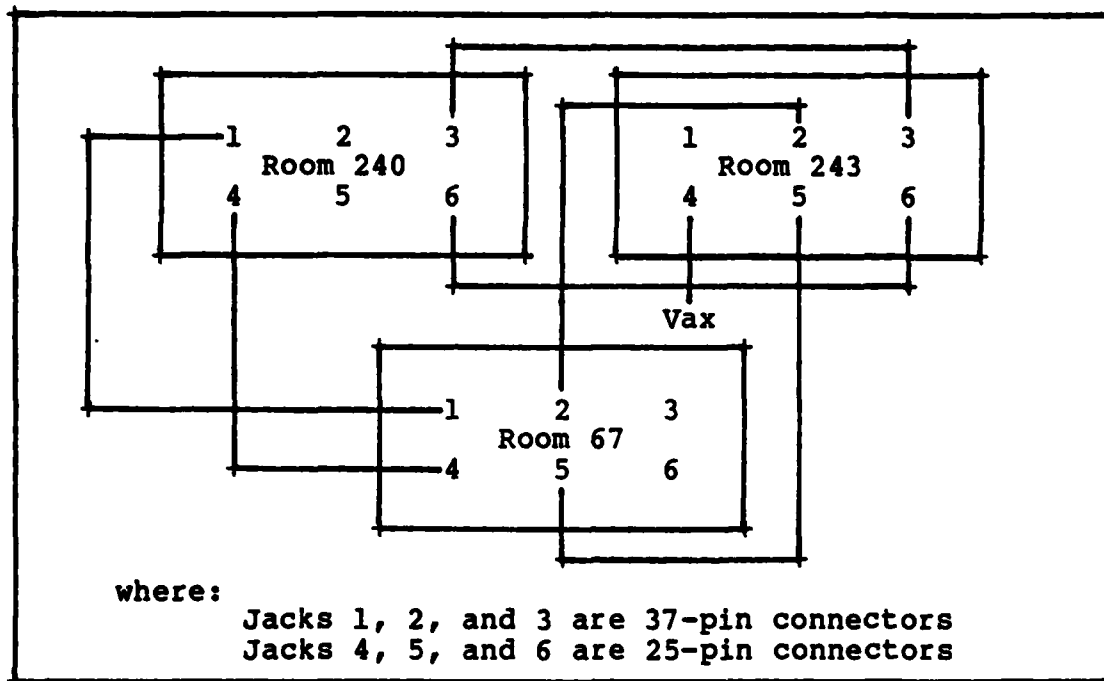


Figure A-8. Cable Box Connections

The cables were 18-pair and 6-pair twisted cable, connected as follows:

18-pair		6-Pair	
1-Ground	16-Brown	2-Red	4-Black
2-Black	21-Black	3-White	5-Black
3-White	22-White	6-Green	8-Black
4-Green	23-Green	7-Ground	
5-Blue	24-Yellow	20-Yellow	1-Black
6-Green	25-Orange	A-Brown	B-Black
7-Black	26-Red	C-Blue	D-Black
8-Yellow	27-Green		
9-Brown	28-Red		
10-White	29-Orange		
11-Red	30-Blue		
12-Black	31-Red		
13-Brown	34-Red		
14-Green	36-Yellow		
15-Black	35-Green		
17-Orange	33-Green		
18-Black	32-Black		
19-Red	37-Red		
20-Blue			

Note: A, B, C, and D are extra wires for future expansion.

The 18-pair cable has the following specifications:
Shielded 100 ohm low voltage computer cable, 24 AWG, .015 PE
Insulation, .050 PVC Jacket, Beldfoilshield 100%, Braid 90.5%
Belden Stock Number 9837.

Appendix Summary

This appendix discussed, in more detail than in Chapter 4, the hardware changes that were required to both the local and network side of the UNIDs. A detailed wiring connection diagram was included to facilitate proper connections for the Zilog MCZ 1/25 Computer System and the UNIDs. A step-by-step procedure was provided to correctly start-up both UNIDs and the Zilog_Host software. Lastly, details were provided as to the 6-pair and 18-pair cable that was installed for the DELNET.

APPENDIX B

N-Layer Concept Description

Introduction

The architecture of the ISO Reference Model was approached as three basic elements (Ref 27:7). These are: the application-processes which exist within the Open Systems Interconnection environment; the connections which join the application-processes and permit them to exchange information; and lastly, systems.

The functions and concepts described in this appendix are those known to be useful to achieve Open Systems Interconnection. However, all functions and concepts described will not necessarily be employed by each layer of the ISO Reference Model (Ref 26:84-90, and 41:9).

Table of Contents

This appendix is subdivided into seven sections, which are:

<u>Section</u>	<u>Page</u>
Principles of Layering.....	B-1
Blocking.....	B-6
Segmenting.....	B-7
Multiplexing/Splitting.....	B-8
Flow Control.....	B-9
Error Detection.....	B-9
Appendix Summary.....	B-10

Principles of Layering

Each system is viewed as being logically composed of an ordered set of subsystems, represented in Figure B-1.

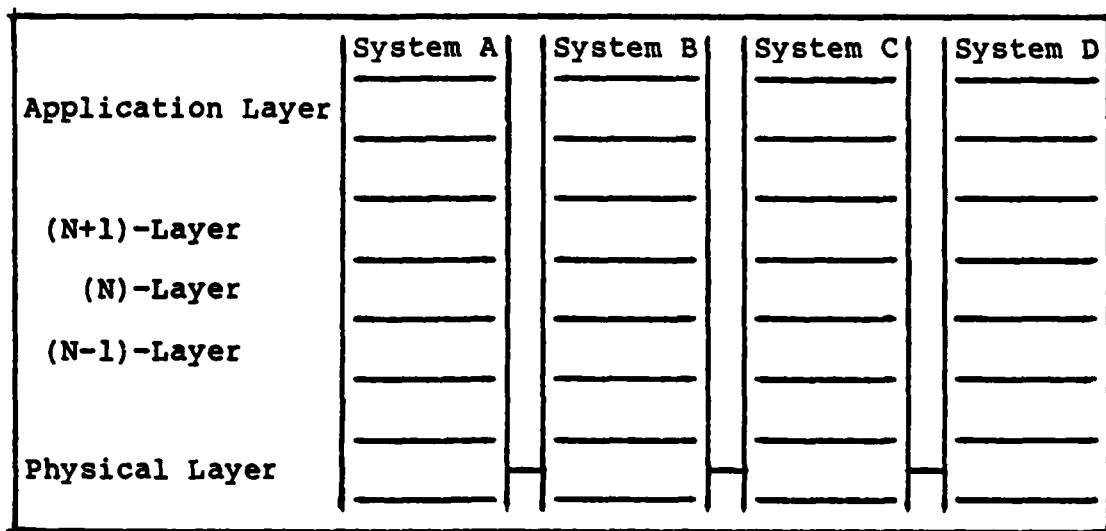


Figure B-1. Layering Concept

Adjacent subsystems communicate through their common interface. Subsystems of the same rank collectively form the (N)-Layer. A subsystem is made of one or several entities with an entity, called (N)-Entity, existing within each layer. Entities in the same layer are termed Peer-Entities. An entity in the next higher layer is an (N+1)-Entity and an entity in the next lower layer is an (N-1)-Entity. Except for the Application Layer, each (N)-Layer provides entities in the (N+1)-Layer with (N)-Services. In order for information to be exchanged between two or more (N+1)-Entities, an association must be established in the (N)-Layer using an (N)-Protocol. This association is called an (N)-Connection. (N)-Connections are provided by the (N)-Layer between two or more (N)-Service-Access-Points. The end of an (N)-Connection at an (N)-Service-Access-Point is called an (N)-Connection-Endpoint.

Each service provided by an (N)-Layer may be tailored by

choosing one or more (N)-Facility which determines the attributes of the service. The services of an (N)-Layer are provided to the next higher layer, using the (N)-Functions performed within the (N)-Layer and the services available from the next lower layer. An entity may provide services to one or more entities in the next higher layer and use the services of one or more entities in the next lower layer. A Service-Access-Point is the access means by which a pair of entities in adjacent layers use or provide services. These relationships are shown in Figure B-2.

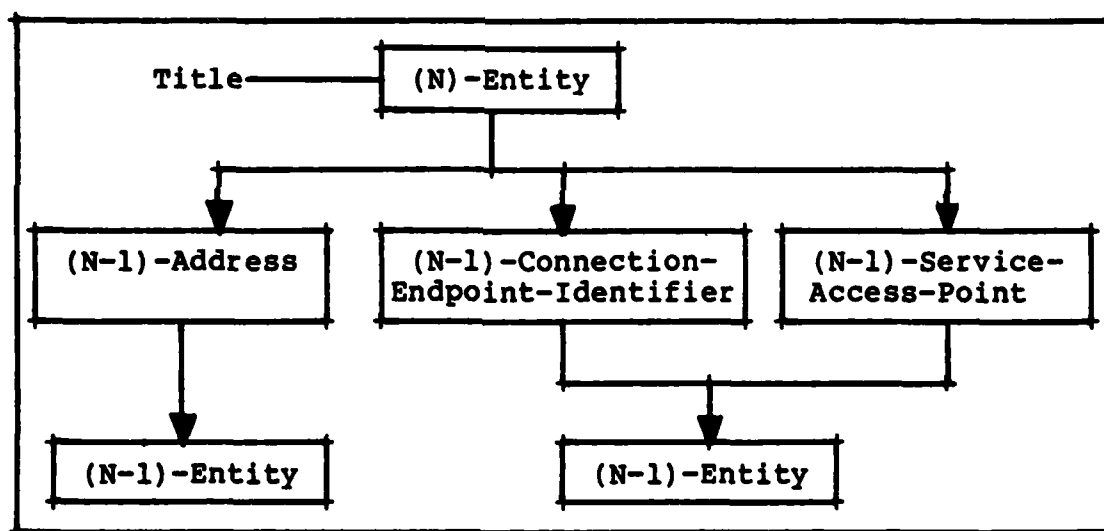


Figure B-2. Entity Relationships

An (N)-Directory concept is defined within the architecture. It serves as an (N)-Function within an (N)-Layer to translate the global-title (A global-title identifies the (N)-Layer) of an (N)-Entity into the (N)-Address to which it is attached. Determination of the correspondence between (N)-Addresses served by an (N)-Entity and the (N-1)-Addresses used for this purpose is performed by

an (N)-Mapping function.

Two different mapping functions may exist within a layer, hierarchical and mapping by tables (Ref 26). If an (N)-Address is always mapped into only one (N-1)-Address then hierarchical construction can be used where the (N)-Address is made of an (N-1)-Address and an (N)-Suffix. In the hierarchical case, the (N)-Mapping-Function simply consists of recognizing the structure of the (N)-Address and extracting the (N-1)-Address it contains. The (N)-Address consists of two parts. The first is an (N-1)-Address of the (N)-Entity which is supporting the current (N)-Service-Access-Point of the (N+1)-Entity. The second part is an (N)-Suffix which makes the (N)-Service-Access-Point uniquely identifiable within the (N-1)-Addresses. If (N)-Addresses can be mapped into several (N-1)-Addresses, or an (N)-Address is not permanently mapped into the same (N-1)-Address, then the mapping function must use tables to translate (N)-Addresses into (N-1)-Addresses.

An (N+1)-Entity requests (N)-Services via an (N)-Service-Access-Point which permits the (N+1)-Entity to interact with an (N)-Entity in the (N)-Layer. Both the (N)- and (N+1)-Entities attached to an (N)-Service-Access-Point are in the same system. An (N)-Entity may concurrently be attached to one or more (N+1)-Entities through (N)-Service-Access-Points. An (N)-Service-Access-Point is only attached to one (N+1)-Entity at a time. An (N)-Service-Access-Point may be reattached to the same or another (N+1)-Entity. A

Service-Access-Point is located by means of its address. An (N)-Address is also used by an (N+1)-Entity to request an (N)-Connection.

Information is transferred in various types of data units between peer entities and between entities attached to a specific service access point. The data units are shown in Figure B-3.

	Control	Data	Combined
(N)-(N) Peer Entities	(N) protocol control information	(N) user data	(N) protocol data units
(N)-(N-1) Adjacent Layers	(N-1) interface control information	(N-1) interface data	(N-1) interface data unit

Figure B-3. Data Unit Interrelationships

Using Figure B-3 as a guide, the data units are defined as follows (Ref 26): (N)-Protocol-Control-Information is information exchanged between two (N)-Entities, using an (N-1)-Connection, to co-ordinate their joint operation; (N)-User-Data is the data transferred between two (N)-Entities on behalf of the (N+1)-Entities for whom the (N)-Entities are providing services; (N)-Protocol-Control-Information is information exchanged between two (N)-Entities, using an (N-1)-Connection, to co-ordinate their joint operation; (N)-User-Data is the data transferred between two (N)-Entities on behalf of the (N+1)-Entities for whom the (N)-Entities are

providing services; (N)-Protocol-Data-Unit is a unit of data which consists of (N)-Protocol-Control-Information and possibly (N)-User-Data; and (N)-Interface-Control-Information is information exchanged between an (N+1)-Entity and an (N)-Entity to co-ordinate their joint operation.

An (N)-Protocol is a set of rules and formats by which (N)-Protocol-Information and possibly (N)-User-Data are exchanged between (N)-Entities in performance of (N)-Functions. (N)-Protocol-Identifiers name the specific protocols defined. An (N)-Protocol is used within the (N)-Layer. One or more (N)-Protocols may be defined for the (N)-Layer. An (N)-Entity may employ one or more (N)-Protocols. Meaningful communication between (N)-Entities over an (N-1)-Connection requires the agreed selection of one (N)-Protocol. Control information and user data are exchanged between (N)-Entities in (N)-Protocol-Data-Units, which is a unit of data specified in an (N)-Protocol; the protocol-data-unit contains (N)-Protocol-Control-Information and possibly (N)-User-Data.

Blocking

Blocking is a function used to collect several (N)-Service-Data-Units into a single (N)-Protocol-Data-Unit; the destination (N)-Entity separates the blocked (N)-Protocol-Data-Units (Ref 26:78). Blocking occurs when several (N)-Service-Data-Units with added (N)-Protocol-Control-Information form an (N)-Protocol-Data-Unit which is illustrated in Figure B-4 (Ref 26:42).

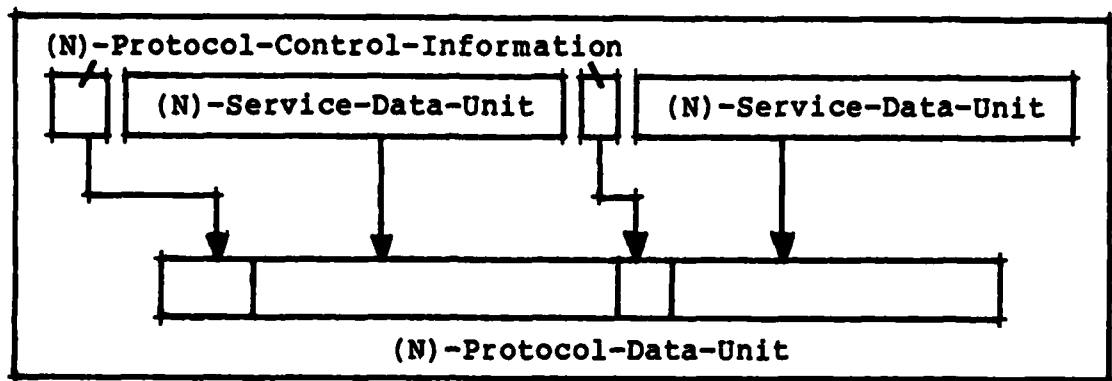


Figure B-4. (N)-Layer Blocking

Segmenting

Segmenting is a function used to split a single (N)-Service-Data-Unit into multiple (N)-Protocol-Data-Units where the destination (N)-Entity reassembles the segmented (N)-Protocol-Data-Units (Ref 26). Segmenting is the mapping of (N)-service-data-units into more than one (N)-Protocol-Data-Unit. Segmenting may occur when (N)-Protocol-Data-Units are mapped into (N-1)-Interface-Data-Units. It is necessary to preserve the identity of (N)-Service-Data-Units on an (N)-Connection. Thus, mechanisms must be available for identifying the segments of an (N)-Service-Data-Unit, such that the correspondent (N)-Entity can determine when the (N)-Service-Data-Unit is complete. When segmenting is performed, an (N)-Service-Data-Unit is segmented into several (N)-Protocol-Data-Units with added (N)-Protocol-Control-Information as shown in Figure B-5 (Ref 26:42).

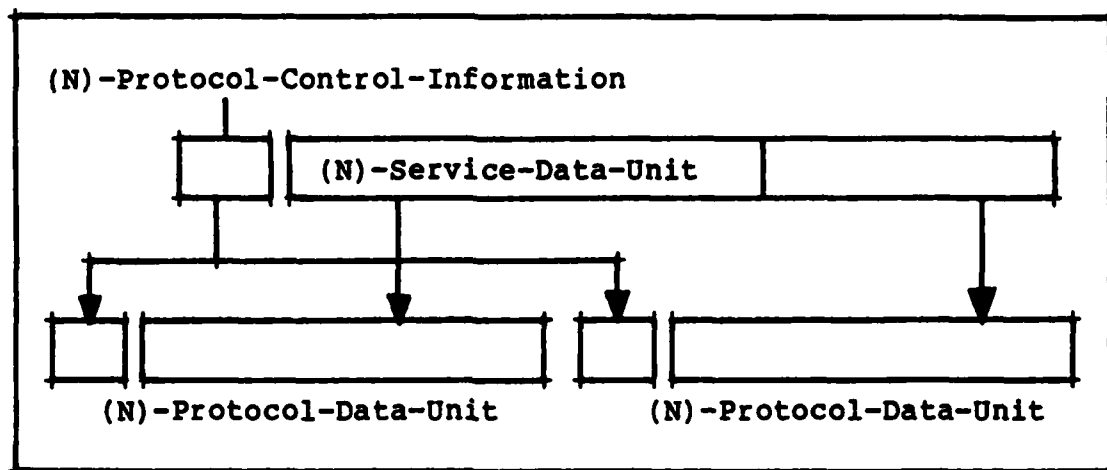


Figure B-5. (N)-Layer Segmenting

Multiplexing/Splitting

Multiplexing or splitting is a function used to enable a single network-connection to be shared between two or more (N)-Connections, or to split a single (N)-Connection into multiple network-connections (Ref 26). This is a function within the (N)-layer by which one (N-1)-Connection is used to support more than one (N)-Connection. Multiplexing may be needed in order to make more efficient or more economic use of the (N-1)-Service or to provide several (N)-Connections in an environment where only a single (N-1)-Connection exists. In order to ensure that (N)-User-Data from the various multiplexed (N)-Connections is not mixed, it is necessary that an identification of each of the individual (N)-Connections be provided and associated with the (N)-User-Data transferred over the multiplexed (N-1)-Connection. This identification is distinct from that of the (N)-Connection-Endpoint-Identifiers and is called an (N)-Protocol-Connection-Identifier. When the capacity of the (N-1)-

Connection is shared by the introduction of a multiplexing function, it is necessary that flow control functions be performed on each individual flow. When more than one (N)-Connection is prepared to send data, it is necessary that scheduling functions be established to select the next (N)-Connection to be serviced.

Flow Control

If flow control mechanisms are provided, they can only operate on (N)-Protocol-Data-Units and (N)-Interface-Data-Units. There are two types of flow control. The first is peer-to-peer flow control which requires protocol definitions and is based on Protocol-Data-Unit size. Peer-to-peer flow control mechanisms require that flow control information be included with the (N)-Protocol-Control-Information of an (N)-Protocol-Data-Unit. The second is (N)-Interface flow control which permits an (N+1)-Entity and an (N)-Entity servicing it to regulate the rate at which (N)-Interface-Data is sent onto the (N)-Connection or received from the (N)-Connection at the (N)-Service-Access-Point. (N)-Interface flow control is based on the (N)-Interface-Data-Unit size (Ref 26:40).

Error Detection

Error detection is the detection of the loss, corruption, duplication, misordering or misdelivery of (N)-Protocol-Data-Units. Error control is concerned with the detection of and recovery from errors and failures of different types. No single error control mechanism meets all the needs or is appropriate at all layers and for all

services within a layer. In general, perfect error control cannot be achieved at any given layer. All that is possible is to decrease unrecoverable errors by increasing the probability of error detection and recovery. Error control mechanisms must be developed around the knowledge of the types of errors that can occur, their frequency, and the structure of the service being protected (Ref 52). Determination of whether to apply error control and choice of an error control mechanism at a given layer will depend on a comparison of the costs involved in the mechanism and the cost of undetected or unrecovered errors. If error control is applied at lower levels, then the probability of undetected or uncorrected errors may be low enough that higher levels may choose not to provide additional error control (Ref 52).

Appendix Summary

This appendix described briefly the (N)-Layer concept that is used throughout the description of the ISO Basic Reference Model. For a more detailed description of the (N)-Layer concept refer to references: 16,17,19,23,24, and 31.

APPENDIX C

DELNET PROTOCOL STANDARD

Introduction

Although the first three layer protocol standards have already been developed , they are included in this appendix for completeness. This appendix represents the initial draft of the DELNET Protocol Standard of all seven protocol layers as outlined by the ISO Basic Reference Model (Ref 26).

Each protocol layer header contains an integral number of octets (one octet = 8-bits) with the numbering starting from zero, increasing in the order of transmission. The bits in an octet are numbered from zero to seven, where the zero bit is the low order bit and is transmitted first. All quantities described below are in binary representation. Where a field covering more than one octet is to be interpreted as the binary representation of an integer, the highest-numbered bit of the highest numbered octet of the field is the most significant bit of the integer and the lowest numbered bit of the lowest numbered octet is the least significant bit of the integer.

Table of Contents

This appendix is subdivided into seven sections, which are:

<u>Section</u>	<u>Page</u>
Physical Layer Protocol Specification.....	C-2
Data Link Protocol Specification.....	C-4
Network Layer Protocol Specification.....	C-5
Transport Layer Protocol Specification.....	C-20
Session Layer Protocol Specification.....	C-39
Presentation Layer Protocol Specification.....	C-46
Application Layer Protocol Specification.....	C-46

Physical Layer Protocol Specification

On the side interfacing the UNID with the various Hosts the RS-232 Standard will be followed. Table C-1 gives the pin connections for the RS-232 Standard which have been implemented thus far in the DELNET. One should note that the table indicates only those pins that are currently being used with no regard as to how they should be utilized.

Table C-1. RS-232 Standard Pin Assignments

RS-232 Pin Number	Function	LOCAL CHANNEL			
		1	2	3	4
1	Protective Ground	X	X	X	X
2	Transmitted Data	X	X	X	X
3	Received Data	X	X	X	X
4	Request to Send	X	X	X	X
5	Clear to Send	X	X	X	X
6	Data Set Ready	X	X	X	X
7	Signal Ground	X	X	X	X
8	Received Line Signal Detector	X	X	X	X
9	Reserved for Testing				
10	Reserved for Testing				
11	Unassigned				
12	Secondary Receive Signal Detect				
13	Secondary Clear to Send				
14	Secondary Transmit Data				
15	Transmit Signal Element Timing				
16	Secondary Receive Data				
17	Receive Signal Element Timing				
18	Unassigned				
19	Secondary Request to Send				
20	Data Terminal Ready	X	X	X	X
21	Signal Quality Detector				
22	Ring Detector				
23	Data Signal Rate Selector (DTE)				
24	Data Signal Rate Selector (DCE)				
25	Unassigned				

On the Network Side, which will connect the UNIDs in a double ring, the RS-449 Standard will be used. Table C-2 gives the RS-449 Standard pin connections, their corresponding functions, and shows which are currently

connected. Again, what is indicated is the pins that are currently in use and not how they should be utilized.

Table C-2. RS-449 Standard Pin Assignments

Pin Number	Function	NETWORK	
		CHANNEL-A	CHANNEL-B
1	Shield	X	X
2	Signaling Rate Indicator		
3	Spare		
4	Send Data	X	X
5	Send Timing		
6	Received Data	X	X
7	Request to Send	X	X
8	Receive Timing		
9	Clear to Send	X	X
10	Local Loopback		
11	Data mode		
12	Terminal Ready	X	X
13	Receiver Ready	X	X
14	Remote Loopback		
15	Incoming Call		
16	Select Frequency		
17	Signaling Rate Indicator		
18	Terminal Timing		
19	Test Mode		
20	Signal Ground	X	X
21	Receive Common		
22	Spare		
23	Send Data	X	X
24	Send Timing		
25	Received Data	X	X
26	Request to Send	X	X
27	Receive Timing		
28	Clear to Send	X	X
29	Terminal in Service		
30	Data Mode		
31	Terminal Ready	X	X
32	Receiver Ready	X	X
33	Select Standby		
34	Signal Quality		
35	New Signal		
36	Terminal Timing		
37	Standby Indicator		
	Send Common		

Data Link Layer Protocol Specification

The Data Link Layer was initially implemented in 1982 and is included here for completeness (Ref 19:4.1-4.8). Figure C-1 gives the Data Link Layer Header.

123																																									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1										
0 1 1 1 1 1 1 0										UNID ADDRESS										CONTROL										DATA											
DATA >= 0 BYTES <= 133 BYTES																																									
DATA										FRAME CHECKING SEQUENCE																				0 1 1 1 1 1 1 0											

Figure C-1. Data Link Layer Header

Octet 1 (Bits 0-7): Opening Flag. This will consist of 1-Byte (01111110). Indicates to the receiving UNID the start of a Frame.

Octet 2 (Bits 8-15): UNID Address. This will be the destination UNID within a given Country Code (See Network Layer for definition of a Country Code within the DELNET).

Octet 3 (Bits 16-23): Control. Not used at this time. When X.25 is implemented this section will be involved.

Octets 4-136 (Bits 24-1087(MAX)): Data bits. This consists of the upper five protocol layers that are transparent to the Data Link Layer. This represents 133-bytes of information (1064-bits).

Octet 137 (Bits 1088-1103): Frame Checking Sequence. Used for Error control within the Data Link Layer.

Octet 138 (Bits 1104-1111): Closing Flag. Will consist of 01111110. Indicates to the receiving UNID the end of a Frame.

Network Layer Protocol Specification

Routing

Within DELNET, CCITT X.121 and IHF (Refs 31:11-23, 49:373-377) format will be used with CC-Country Code, NC-Network Code, HC-Host Code, and PC-Port Code. The format is:

					1							2							3	
0	3	4	7	8	1	2	5	6	9	0	3	4	7	8	1					
0000				0000				0000				0000				0000				
CTR				CC		NC		HC				PC								

Bits 0-3: Control - This will inform the UNID which addressing scheme is in use. The breakdown is as follows:

- 0000 uses 8-bits for Network Addresses (4-CC, 4-NC) and 20-bits for Host Addresses (8-HC, 12-PC)
- 1000 uses 14-bits for Network Addresses (7-CC, 7-NC) and 14-bits for Host Addresses (6-HC, 8-PC)
- 1100 uses 20-bits for Network Addresses (8-CC, 12-NC) 8-bits for Host Addresses (4-HC, 4-PC)
- 1110 indicates Extended Addressing Mode

Bits 4-7: Country Code (CC) - This will give 16 possible codes as follows (See Figures C-2 thru C-4 for layout). The Country Code will be based on the building number and floor where the particular computer is located as follows:

- 0000 DELNET Monitor
- 0001 Bldg 640, Basement, Rooms 60-64, 66, 67
- 0010 Bldg 640, 1st Floor, Rooms 141 thru 151
- 0011 Bldg 640, 1st Floor, Rooms 100-104, 120-123
- 0100 Bldg 640, 1st Floor, Rooms 124 thru 134
- 0101 Bldg 640, 1st Floor, Rooms 105 thru 114
- 0110 Bldg 640, 1st Floor, Rooms 160-165, 115-119
- 0111 Bldg 640, 2nd Floor, Rooms 201 thru 208
- 1000 Bldg 640, 2nd Floor, Rooms 200, 209-220, 225, 230
- 1001 Bldg 640, 2nd Floor, Rms 221-224, 237, 239, 241, 243, 245
- 1010 Bldg 640, 2nd Floor, Rms 226-229, 232-236, 238, 240, 242
- 1011 Bldg 640, 2nd Floor, Rooms 231, 244, 246-265
- 1100 Bldg 640, 2nd Floor, Reserved For Expansion
- 1101 Bldg 641, 1st Floor
- 1110 Bldg 641, 2nd Floor
- 1111 Bldg 641, 3rd Floor

Bits 8-11: Network Code (NC)- This gives 16 possible values as follows:

- 0000 UNID_0	- 1000 UNID_8
- 0001 UNID_1	- 1001 UNID_9
- 0010 UNID_2	- 1010 UNID_10
- 0011 UNID_3	- 1011 UNID_11
- 0100 UNID_4	- 1100 UNID_12
- 0101 UNID_5	- 1101 UNID_13
- 0110 UNID_6	- 1110 UNID_14
- 0111 UNID_7	- 1111 UNID_15

Bits 12-19: Host Code (HC) - This gives 256 possible values with the following format:

- a. If Host is connected to UNID Port_1, then Local Area Network Hosts will be numbered from 0 to 63.
- b. If Host is connected to UNID Port_2, then Local Area Network Hosts will be numbered from 64 to 127.
- c. If Host is connected to UNID Port_3, then Local Area Network Hosts will be numbered from 128 to 191.
- d. If Host is connected to UNID Port_4, then Local Area Network Hosts will be numbered from 192 to 255.

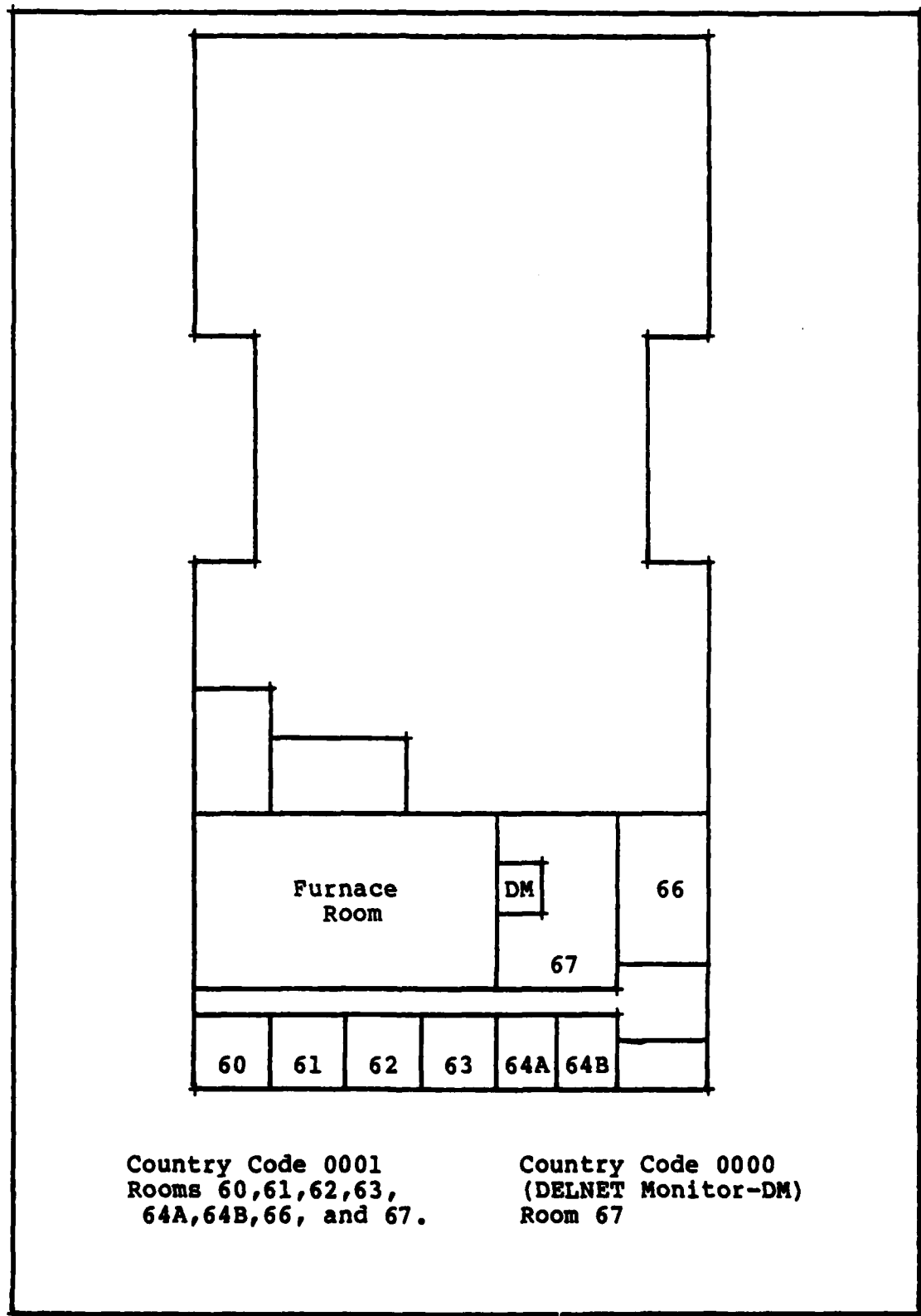
Bits 20-31: Port Code (PC). This section is subdivided into two parts depending on the bit values in the Control Section.

a. If Control = 0000 then the Port Code is seen as normal mode with the first 8-bits representing the Hosts Hexidecimal Port Address. The last four bits are set to zero.

b. If Control = 1110 then the Port Code gives a pointer to a subnetwork's Host and Port as follows:

Bits 20-23: Host Code. This gives 16 possible values.

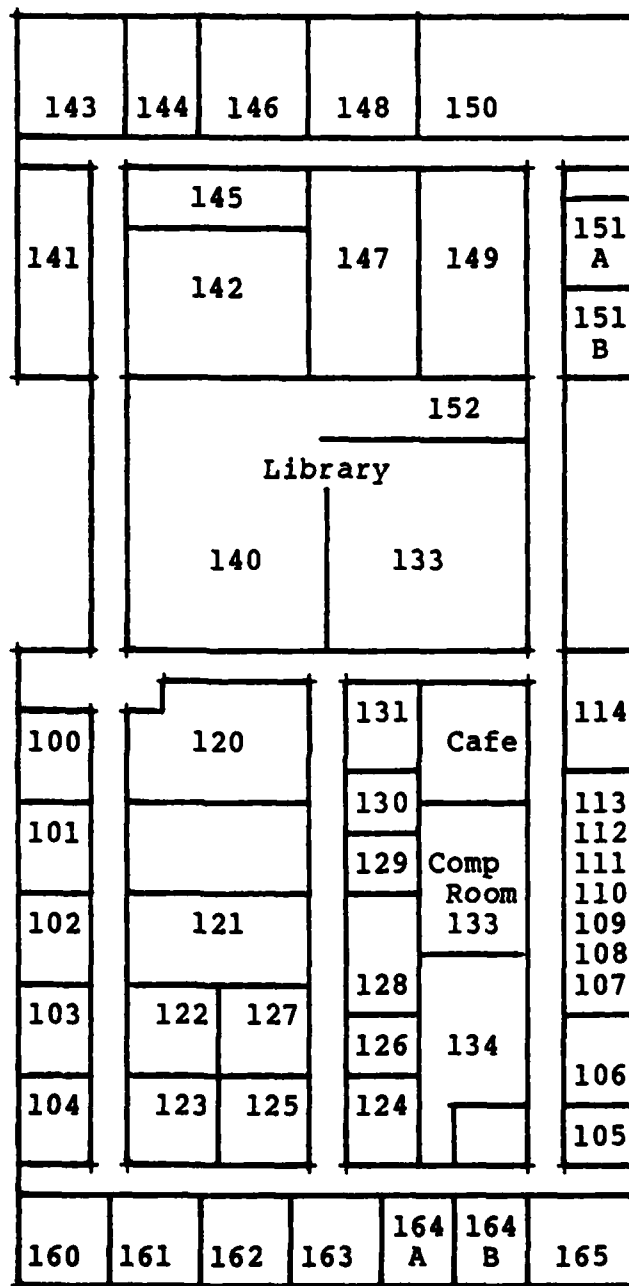
Bits 24-31: Port Code. The same as above.



Country Code 0001
Rooms 60,61,62,63,
64A,64B,66, and 67.

Country Code 0000
(DELNET Monitor-DM)
Room 67

Figure C-2. Bldg 640 Basement Country Codes



CC 0010: Rooms: 141,142,143,144,145,146,147,148,
149,150,151A,151B
CC 0011: Rooms: 100,101,102,103,104,120,121,122,123
CC 0100: Rooms: 124,125,126,127,128,129,130,131,133,134
CC 0101: Rooms: 105,106,107,108,109,110,111,112,113,114
CC 0110: Rooms: 160,161,162,163,164A,164B,165

Figure C-3. Bldg 640 1st Floor Country Codes

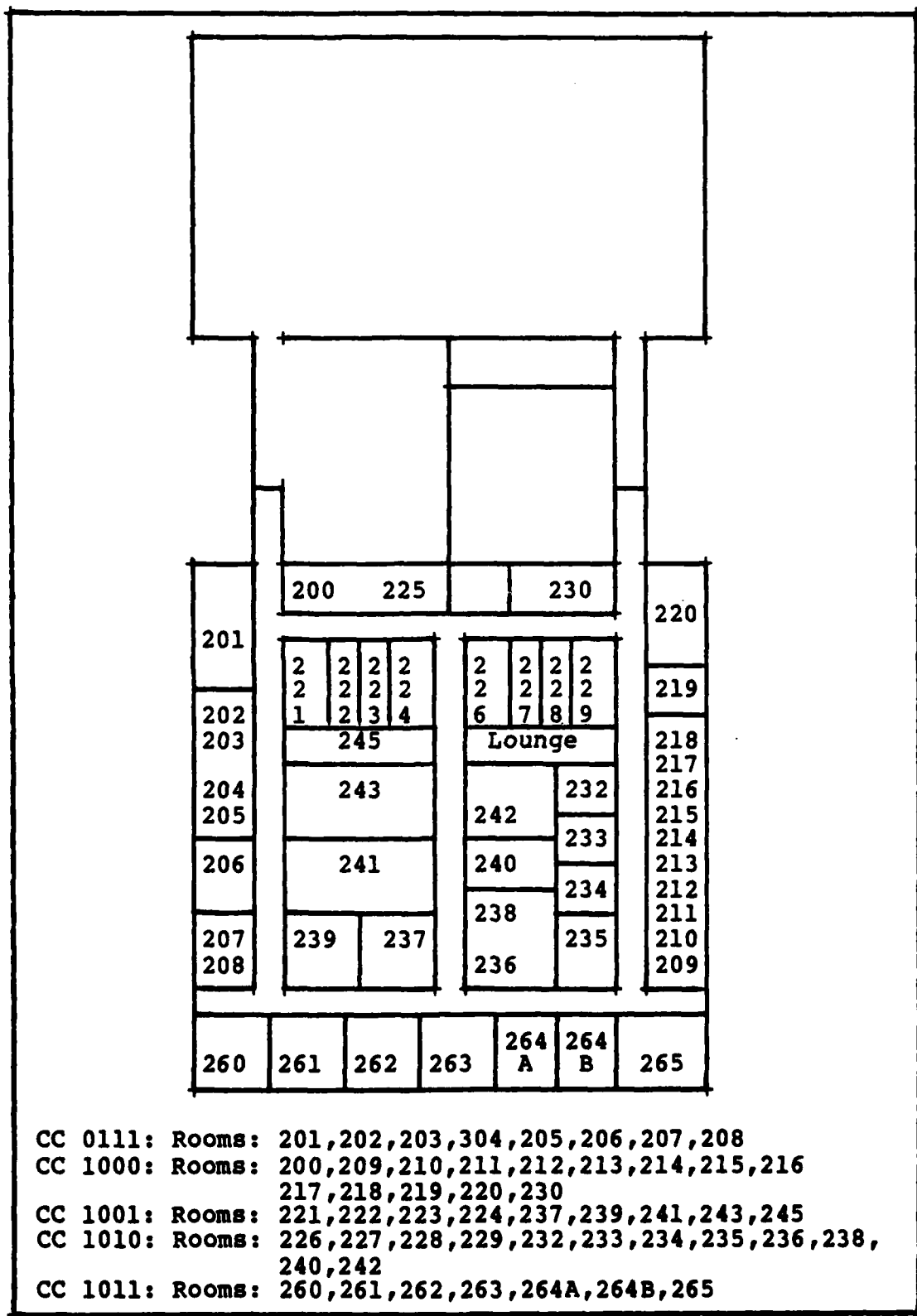


Figure C-4. Bldg 640 2nd Floor Country Codes

Flow Control

UNID_0 will be responsible for the flow control within a particular Country Code and will also control the data flow between different Country Codes. To perform the flow control function, UNID_0 will periodically send out an S-Frame with the 133-bytes used to control and monitor each of the UNIDs connected within the Country Code. On the Inter-Country Code side of UNID_0, all UNID_0s will periodically send S-Frames informing them of the status of the individual Countries within the DELNET.

DATAGRAM Service

The Network Layer has been partially implemented to accomodate Datagrams. Figure C-5 gives the Network Layer Header for DATAGRAM service (Ref 19:5-5).

1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1															
DEST ADDRESS UNID# CH#				SOURCE ADDRESS UNID# CH#				SEQUENCE NUMBER				SPARE			
SPARE				0 <= DATA <= 128 BYTES											

Figure C-5. Network Layer Header

Octet 1 (Bits 0-7): Destination Address. This octet contains the UNID and Channel Number of the destination UNID that services the Host called upon by the User.

Octet 2 (Bits 8-15): Source Address. This octet contains the UNID Number and Channel Number of the source UNID that represents the calling UNID.

Octet 3 (Bits 16-23): Sequence Number. This octet contains the Sequence Number of the Frame.

Octets 6-133 (Bits 40-1063): This consists of the upper protocol layers that are transparent to the Network r.

The Virtual Service header will follow the X.25 standard. Currently, X.25 Virtual Service is not implemented on the UNIDs. When implemented, Figures C-6 thru C-19 illustrate the protocol headers that will be used (Ref 17:243-283):

1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																							
GFI				LCGN				LCN				Packet Type Identifier				Calling DTE AL				Called DTE AL			
DTE Address																0 0 0 0							
0 0		Facility Length				Facilities																	
Call User Data																							

Octet 1 (Bits 0-7): Subdivided as follows:

```

0001: Call set-up and clearing, flow control
      interrupt, reset and restart packets using
      MODULO-8 sequencing numbering scheme.
1001: Data Packets using MODULO-8 sequencing.
1010: Data Packets using MODULO-128 sequencing.
1011: Call set-up and clearing, flow control
      interrupt, reset and restart packets using
      MODULO-128 sequencing numbering scheme.

```

The logical channel group number appears in every packet except in RESTART packets.

Octet 2 (Bits 8-15): Logical Channel Number.
The logical channel number appears in every packet except in RESTART packets.

Octet 3 (Bits 16-23): Packet Type Identifier.
0000 1011: Incoming Call or Call Request Packet

Octet 4 (Bits 24-31): Subdivided as follows:

Bits 24-27: Calling DTE Address Length. This gives the address length of the calling DTE.

Bits 28-31: Called DTE Address Length. This gives the address length of the called DTE.

Octets 5-20 (Bits 32-159): DTE Address.
This contains the calling and called DTE address. Each address can be up to eight octets in length.

Octet 21 (Bits 160-167): Facility Length.
This is a 12-bit length that gives the length of the Facility Field in octets. The first two bits are preset to 00.

Octet 22 (Bits 168-175): Facility Field.
The Facility Field is present only when the DTE is using an optional user facility requiring some indicator in the Call Request and Incoming Call Packets. The Facility Field is formatted as follows:

0000 0000: Reverse Charging not Requested
0000 0001: Reverse Charging Requested
0000 0010: Flow Control Parameters Selected

Octets 23-38 (Bits 176-303): Call User Data.
The call user data field has a maximum length of 16-octets. The contents of the field are passed unchanged to the called DTE.

2. Call Accepted and Call Connected Packet

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
0	0	0	1	LCGN				LCN				0 0 0 0 1 1 1 1																			

Figure C-7. Call Accepted/Connected Packet Format

Bits 0-3: General Format Identifier.
0001 = Call set-up and clearing.
Bits 4-7: Logical Channel Group Number.

Octet 3 (Bits 16-23): Packet Type Identifier.
0000 1111 = Call Accepted/Connected Packet

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0 0 0 1				LCGN								LCN								0 0 0 1 0 1 1 1																			

Bits 0-3: General Format Identifier.
0001 = Call set-up and clearing.
Bits 4-7: Logical Channel Group Number.

Octet 3 (Bits 16-23): Packet Type Identifier.
0001 0111 = DTE and DCE Clear Confirmation Packet

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0 0 0 1				LCGN								LCN								0 0 0 1 0 0 1 1				Clearing Clause															

Bits 0-3: General Format Identifier.
0001 = Call set-up and clearing.

Octet 3 (Bits 16-23): Packet Type Identifier.
0001 0011 = Clear Request/Indication Packet

Octet 4 (Bits 24-31): Clearing Clause. Values are:

0000 0000: DTE Clearing
 0000 0001: Number Busy
 0000 1001: Out of Order
 0001 0001: Remote Procedure Error
 0001 1001: Number Refuses Reverse Charging
 0000 0011: Invalid Call
 0000 1011: Access Barred
 0001 0011: Local Procedure Error
 0000 0101: Network Congestion
 0000 1101: Not Obtainable

5. DTE and DCE Data Packet

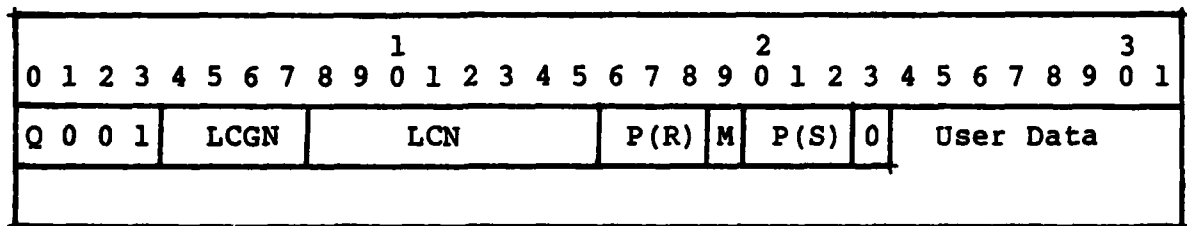


Figure C-10. DTE and DCE Data Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bit 0: Q = Data Qualifier
 Bits 1-3: General Format Identifier.
 001 = Call set-up and clearing.
 Bits 4-7: Logical Channel Group Number.

Octet 2 (Bits 8-15): Logical Channel Number.

Octet 3 (Bits 16-23): Subdivided as follows:

Bits 16-18: P(R) = Receive Sequence Count
 Bit 19: M = More Data Indication
 0: No more data
 1: More data
 Bits 20-22: P(S) = Send Sequence Count
 NOTE: In the event that the sequence numbering
 scheme is MODULO-128, this format is
 effected.
 Bit 23: Set to Zero.

Octets 4-20 (Bits 24-79): User Data.

6. DTE and DCE Interrupt Packet

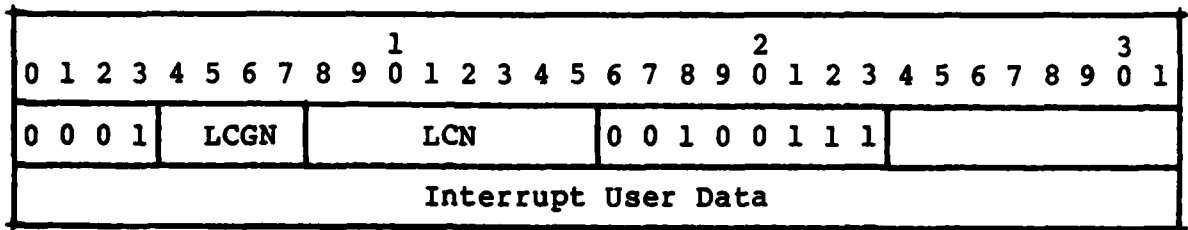


Figure C-11. DTE and DCE Interrupt Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier

0001 = Call set-up and clearing.

Bits 4-7: Logical Channel Group Number.

Octet 2 (Bits 8-15): Logical Channel Number.

Octet 3 (Bits 16-23): Packet Type Identifier.

0010 0111 = DTE and DCE Interrupt Packet

Octets 4-20 (Bits 24-79): Interrupt User Data.

7. DTE and DCE Interrupt Confirmation Packet

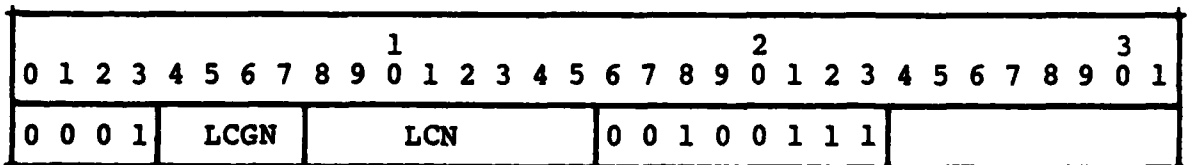


Figure C-12. DTE and DCE Interrupt Confirmation Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier

0001 = Call set-up and clearing.

Bits 4-7: Logical Channel Group Number.

Octet 2 (Bits 8-15): Logical Channel Number.

Octet 3 (Bits 16-23): Packet Type Identifier.

0010 0111 = DTE and DCE Interrupt Confirmation Packet

8. DTE and DCE RR Packet

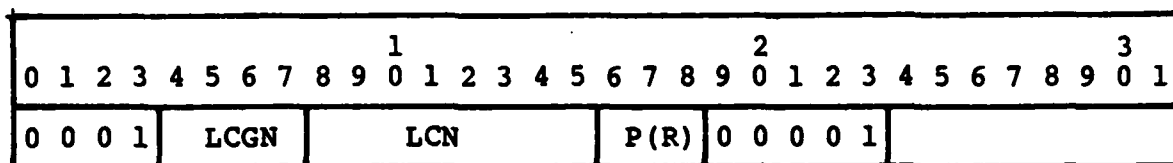


Figure C-13. DTE and DCE RR Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier
0001 = Call set-up and clearing.
Bits 4-7: Logical Channel Group Number.

Octet 2 (Bits 8-15): Logical Channel Number.

Octet 3 (Bits 16-23): Subdivided as follows:

Bits 16-18: P(R)
Bits 17-23: Set to 0001

NOTE: In the event that the sequence numbering scheme is MODULO-128, this format is affected.

9. DTE and DCE RNR Packet

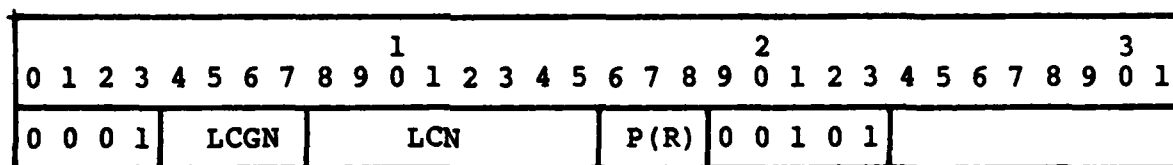


Figure C-14. DTE and DCE RNR Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier
0001 = Call set-up and clearing.
Bits 4-7: Logical Channel Group Number.

Octet 2 (Bits 8-15): Logical Channel Number.

Octet 3 (Bits 16-23): Subdivided as follows:

Bits 16-18: P(R)
Bits 19-23: Set to 00101

NOTE: In the event that the sequence numbering scheme is MODULO-128, this format is affected.

10. Reset Request and Reset Indication Packet

1																														2																														3																													
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																																																										
0 0 0 1				LCGN				LCN								0 0 0 1 1 0 1 1								Resetting Cause																																																																	
Diagnostic Code																																																																																									

Figure C-15. Reset Request/Indication Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier

0001 = Call set-up and clearing.

Bits 4-7: Logical Channel Group Number.

Octet 2 (Bits 8-15): Logical Channel Number.

Octet 3 (Bits 16-23): Packet Type Identifier.

00011011 = Reset Request/Indication Packet.

Octet 4 (Bits 24-31): Resetting Cause.

Possible values are:

```
0000 0000: DTE Reset
```

0000 0001: Out of Order

```
0000 0011: Remote Procedure Error
```

```
0000 0101: Local Procedure Error
```

0000 0111: Network Congestion

Octet 5 (Bits 32-39): Diagnostic Code.

This Field is set to 00000000 for Reset Request and Reset Indication Packets.

11. DTE and DCE Reset Confirmation Packet

										¹										²										³									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
0	0	0	1	LCGN					LCN					0 0 0 1 1 1 1 1																									

Figure C-16. DTE and DCE Reset Confirmation Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier

0001 = Call set-up and clearing.

Bits 4-7: Logical Channel Group Number.

Octet 2 (Bits 8-15): Logical Channel Number.

Octet 3 (Bits 16-23): Packet Type Identifier.
 0001 1111 = DTE and DCE Reset Confirmation Packet.

12. Restart Request and Restart Indication Packet

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
0 0 0 1				0 0 0 0				0 0 0 0 0 0 0 0								1 1 1 1 1 0 1 1								Restarting Cause															

Figure C-17. Restart Request/Indication Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier
 0001 = Call set-up and clearing.
 Bits 4-7: Logical Channel Group Number.
 Set to all zeros.

Octet 2 (Bits 8-15): Logical Channel Number.
 Set to zeros.

Octet 3 (Bits 16-23): Packet Type Identifier.
 1111 1011 = Restart request/Indication Packet.

Octet 4 (Bits 24-31): Restarting Cause.
 This contains the reason for the restart. For the Restart Request Packet the Restarting Cause is 00000000. For the Restart Indication Packet the Restarting values are:

0000 0001: Local Procedure Error
 0000 0011: Network Congestion

13. DTE and DCE Restart Confirmation Packet

123																																	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1								

Figure C-18. DTE and DCE Restart Confirmation Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier
 0001 = Call set-up and clearing.
 Bits 4-7: Logical Channel Group Number.
 Set to all zeros.

Octet 2 (Bits 8-15): Logical Channel Number.
Set to all zeros.

Octet 3 (Bits 16-23): Packet Type Identifier.
1111 1111 = DTE and DCE Restart Confirmation Packet.

14. DTE REJ Packet

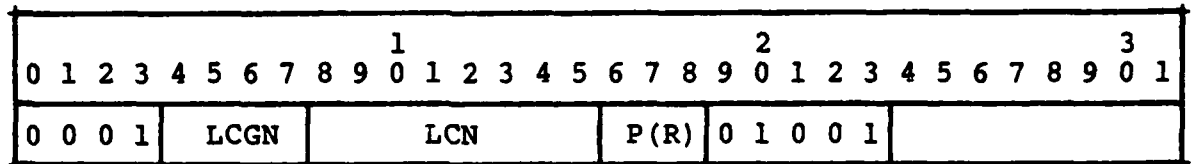


Figure C-19. DTE REJ Packet Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: General Format Identifier
0001 = Call set-up and clearing.
Bits 4-7: Logical Channel Group Number.

Octet 2 (Bits 8-15): Logical Channel Number.

Octet 3 (Bits 16-23): Subdivided as follows:

Bits 16-18: P(R)
Bits 19-23: Set to 01001

DATAGRAM Service

0 1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																															
VERSION		IHL		TYPE OF SERVICE								TOTAL LENGTH																			
IDENTIFICATION										FLAGS		FRAGMENT OFFSET																			
TIME TO LIVE				PROTOCOL				HEADER CHECKSUM																							
SOURCE ADDRESS																															
DESTINATION ADDRESS																															
OPTIONS																						PADDING									
SOURCE PORT										DESTINATION PORT																					
SEQUENCE NUMBER																															
ACKNOWLEDGEMENT NUMBER																															
DATA OFFSET		RESERVED				U R G	A C K	P S H	R S T	S Y N	F I N	WINDOW																			
CHECKSUM										URGENT POINTER																					
OPTIONS																						PADDING									
DATA																															

C-20

Following is the specification for the DATAGRAM Transport Header:

Internet Protocol Header:

1. Version is 4 bits in length. The Version Field indicates the format of the internet header. The possible values are (Ref 30):

- 0000: Reserved
- 0001 - 0011: Unassigned
- 0100: Internet Protocol (DELNET version)
- 0101: ST Datagram Mode
- 0110 - 1110: Unassigned
- 1111: Reserved

2. Internet Header Length (IHL) is 4 bits in length. This represents the length of the internet header in 32-bit words, and thus points to the beginning of the data. The minimum value for a correct header is five.

3. Type of Service is 8 bits in length. The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. The 8-bits are subdivided as follows:

a. Bits 0-2: Precedence

- 000 - Routine
- 001 - Priority
- 010 - Immediate
- 011 - Flash
- 100 - Flash Override
- 101 - CRITIC/ECP
- 110 - Internetwork Control
(Used within DELNET)
- 111 - Network Control
(Used within LAN)

b. Bit 3: (D) 0=Normal Delay, 1=Low Delay

c. Bit 4: (T) 0=Normal Throughput, 1=High Throughput

d. Bit 5: (R) 0=Normal Reliability, 1=High Reliability

e. Bits 6-7: Not Used

The Type of Service values depend on the network being

used. Whenever the DELNET interfaces with other networks, a knowledge of their assignments is required. Following are some assignments for typical computer networks (Ref 33):

a. AUTODIN II: The service choices are in two parts: Traffic Acceptance Catagories and Application Type. The Traffic Acceptance Catagories can be mapped into and out of the Type of Service precedence reasonably directly. The Application types can be mapped into the remaining Type of Service fields as follows:

TA	Delay	Throughput	Reliability	DTR	TA
I/A	1	0	0	000	Q/R
Q/R	0	0	0	001	Q/R
B1	0	1	0	010	B1
B2	0	1	1	011	B2
				100	I/A
				101	I/A
				110	I/A
				111	Error

b. ARPANET: The service choices to ARPANET are limited. There is one priority bit that can be mapped to the high order bit of the Type of Service precedence. The other choices are to use the regular "Type 0" messages instead of the uncontrolled "Type 3" messages, or to use single packet instead of multipacket messages. The mapping of ARPANET parameters into the Type of Service parameters are as follows:

Type	Size	Delay	Throughput	Reliability	DTR	Type	Size
0	S	1	0	0	000	0	M
0	M	0	0	0	001	0	M
3	S	1	0	0	010	0	M
3	M	Error	Error	Error	011	0	M
					100	3	S
					101	0	S
					110	3	S
					111	Error	

c. PRNET: There is no priority indication in PRNET. The two choices are to use the station routing as opposed to point-to-point routing, or to require acknowledgements rather than have any acknowledgements. The mapping of PRNET parameters into Type of Service parameters are:

Routing	Acks	Delay	Throughput	Reliability	DTR	Routing	Acks
PTP	no	1	0	0	000	STATION	no
PTP	yes	1	0	1	001	STATION	yes
STATION	no	0	0	0	010	STATION	no
STATION	yes	0	0	1	011	STATION	yes
					100	PTP	no
					101	PTP	yes
					110	PTP	no
					111	PTP	yes

d. SATNET: There is no priority indication with SATNET. The four choices are to use the block versus stream type, to select one of four delay categories, to select one of two holding time strategies, or to request one of three reliability levels. The mapping of SATNET parameters into Type of Service parameters is quite complex with $2*4*2*3=48$ possibilities.

4. Total Length is 16 bits in length. This is the length of the datagram, measured in octets, including internet header and data. This field allows a datagram to be up to 65,535 octets. All hosts must be prepared to accept up to 576 octets (whether they arrive whole or in fragments).

5. Identification is 16 bits in length. This is an identifying value assigned by the sender to aid in assembling the fragments of a datagram.

6. Flags is 3 bits in length. This is subdivided as:

- a. Bit 0: Must be zero
- b. Bit 1: 0=Net may fragment, 1=Do not fragment
- c. Bit 2: 0=Last Fragment, 1=More to come

7. Fragment Offset is 13 bits in length. This field indicates where in the datagram this fragment belongs. The first fragment has offset=0.

8. Time to Live is 8 bits in length. This field indicates the maximum time the datagram is allowed to remain in the internet system. If the value is zero then the datagram is discarded. The time is measured in units of seconds.

9. Protocol is 8 bits in length. This field indicates the next level protocol used in the data portion of the internet datagram. The following are currently assigned protocol numbers (Ref 30):

Number	Protocol
0000 0000	Reserved
0000 0001	ICMP
0000 0010	Unassigned
0000 0011	Gateway-to-Gateway
0000 0100	CMCC Gateway Monitoring Message
0000 0101	ST
0000 0110	TCP
0000 0111	UCL
0000 1000	Unassigned
0000 1001	Secure
0000 1010	BBN RCC Monitoring
0000 1011	NVP
0000 1100	PUP
0000 1101	Pluribus
0000 1110	Telenet
0000 1111	XNET
0001 0000	Chaos
0001 0001	User Datagram
0001 0011	DCN
0001 0100	TAC Monitoring
0001 0101 - 0011 1110	Unassigned

	0011 1111	Any Local Area Network
	0100 0000	SATNET and Backroom EXPAK
	0100 0001	MIT Subnet Support
0100 0010 -	0100 0100	Unassigned
	0100 0101	SATNET Monitoring
	0100 0110	Unassigned
	0100 0111	Internet Packet Core Utility
0100 1000 -	0100 1011	Unassigned
	0100 1100	Backroom SATNET Monitoring
	0100 1101	Unassigned
	0100 1110	WIDEBAND Monitoring
	0100 1111	WIDEBAND EXPAK
0101 0000 -	1111 1110	Unassigned
	1111 1111	Reserved

10. Header Checksum is 16 bits in length. This is a checksum on the header only. This is recomputed at each point when the internet header is processed.

11. Source Address is 32 bits in length. See Routing Section of the Network Layer for further breakdown.

12. Destination Address is 32 bits in length. See routing Section of the Network Layer for further breakdown.

Within the Source and Destination Addresses, a conversion must be performed to convert internet addresses to local net addresses and vice versa. Table C-3 gives this conversion for various common networks.

13. Options Field is of variable length. The options may or not appear in datagrams. All UNIDs must be able to decipher all options available within the DELNET. This field is broken down into subfields as shown in Table C-4.

Table C-3. Address Conversion to Internet Header Format

Network	Number	IP Header			
AUTODIN II	26	00011010 00000000 00000000 00000000			
			Host/Terminal		
ARPANET	10	00001010 00000000 00000000 00000000			
			Host	Logical	IMP
				Host	
DCNs	29-30	00010010 00000000 00000000 00000000			
Distributed Computing Networks			Host	Process	
			ID	ID	
EDN	21	00010101 00000000 00000000 00000000			
Experimental Data Network			Host	Logical	IMP
				Host	
LCSNET	18	00010010 00000000 00000000 00000000			
			Subnet	Reserved	Host
PRNET	1 BBN-PR	00000000 00000000 00000000 00000000			
Packet	2 SF-PR-1	PR Net		Host	
Radio	5 SILL-PR				
Net	6 SF-PR-2				
	9 BRAGG-PR				
	20 DC-PR				
SATNET	4	00000100 00000000 00000000 00000000			
Sat PN			Host		
WBCNET	28	00011100 00000000 00000000 00000000			
Wideband			Host	Local	
Comm Sat			Access	Address	
Packet Net			Number		

Table C-4. Internet Options

Class	No.	Length	Description
0	0	-	End of options list. This option occupies only one octet; it has no length octet.
0	1	-	No operation. This option occupies only one octet; it has no length octet.
0	2	11	Security. Used to carry Security, Compartmentation, User Group(TCC), and Handling Restriction Codes.
0	3	var	Loose Source Routing. Used to route the internet datagram based on information supplied by the source.
0	7	var	Record Route. Used to trace the route an internet datagram takes.
0	8	4	Stream Identification. Used to carry the stream identifier.
0	9	var	Strict Source Routing. Used to route the internet datagram based on information supplied by the source.
2	4	var	Internet Timestamp.

Some of the fields shown in Table C-4 and their corresponding values are:

End of Options	- 00000000	Type 0	
No Operation	- 00000001	Type 1	
Security	- 10000010	00001011	- Type 130 - 11 octets
S-Field	- 00000000	00000000	- Unclassified
		11110001	00110101 - Confidential
		01111000	10011010 - EFTO
		10111100	01001101 - MMMM
		01011110	00100110 - PROG
		10101111	00010011 - Restricted
		11010111	10001000 - Secret
		01101011	11000101 - Top Secret
		00110101	11100010 - Future Use
		10011010	11110001 - Future Use
		01001101	01111000 - Future Use
		00100100	10111101 - Future Use
		00010011	01011110 - Future Use
		10001001	10101111 - Future Use
		11000100	11010110 - Future Use
		11100010	01101011 - Future Use
C-Field	- 00000000	00000000	- Not Compartmented
H-Field	- 00000000	00000000	- No Restrictions
TCC-Field	- 00000000	00000000	00000000
Loose Source			
Record Route	- 10000011		- Type 131

14. IHF Padding is of variable length. The IHF Header must end on an even 4-byte boundary.

TCP Protocol Header:

1. Source Port is 16 bits in length.
2. Destination Port is 16 bits in length.

The Source and Destination Port are used to name the ends of logical connections which carry long term conversations. The following assigned ports use a small part of the possible ports available. Following are assigned values (Ref 30):

0000000000000000 - 0000000011111111 Network Wide Standard
Function

Within the Network Wide Standard Function, the following
are assigned:

General Assignment Description

0000	0000	0000	0001	Old Telnet
0000	0000	0000	0011	Old File Transfer
0000	0000	0000	0101	Remote Job Entry
0000	0000	0000	0111	Echo
0000	0000	0000	1001	Discard
0000	0000	0000	1011	Who is on? or SYSTAT
0000	0000	0000	1101	Date and Time
0000	0000	0000	1111	Who is up? or NETSTAT
0000	0000	0001	0001	Short Text Message
0000	0000	0001	0011	Char generator or TTYTST
0000	0000	0001	0101	New File Transfer
0000	0000	0001	0111	New Telnet
0000	0000	0001	1001	SMTP
0000	0000	0001	1011	NSW User System w/COMPASS
0000	0000	0001	1101	MSG-3 Internet Control Protocol
0000	0000	0001	1111	MSG-3 Authentication
0000	0000	0010	0001	Unassigned
0000	0000	0010	0011	IO Station Spooler
0000	0000	0010	0101	Time Server
0000	0000	0010	0111	Unassigned
0000	0000	0010	1001	Graphics
0000	0000	0010	1010	Name Server
0000	0000	0010	1011	Who is?
0000	0000	0010	1101	Message Processing Module
0000	0000	0010	1111	NI File Transfer Protocol
0000	0000	0011	0001	RAND Network Graphics Conference
0000	0000	0011	0011	Message Generator Control
0000	0000	0011	0101	AUTODIN II File Transfer Protocol
0000	0000	0011	0111	ISI Graphics Language
0000	0000	0011	1001	Message Transfer Protocol
0000	0000	0011	1011	New MIT Host Status
0000	0000	0011	1100	Unassigned
0000	0000	0011	1101	Unassigned
0000	0000	0011	1111	Unassigned

0000000010000000 - 000000010000011 Host Specific Functions

Within the Host Specific Functions the following are currently assigned:

General Assignment Description			
0000	0000	0100	0001 Unassigned
0000	0000	0100	0011 Datacomputer at CCA
0000	0000	0100	0111 Trivial File Transfer
0000	0000	0100	0111 NETRJS (EBCDIC) at UCLA-CCN
0000	0000	0100	1001 NETRJS (ASCII-68) at UCLA-CCN
0000	0000	0100	1011 NETRJS (ASCII-63) at UCLA-CCN
0000	0000	0100	1101 Any Private Remote Job Entry Server
0000	0000	0100	1111 Name or Finger
0000	0000	0101	0001 Unassigned
0000	0000	0101	0011 MIT ML Device
0000	0000	0101	0101 MIT ML Device
0000	0000	0101	0111 Any Terminal Link
0000	0000	0101	1001 SU/MIT Telnet Gateway
0000	0000	0101	1011 MIT Dover Spooler
0000	0000	0101	1101 BBN RCC Accounting
0000	0000	0101	1111 SUPDUP
0000	0000	0110	0001 Datacomputer Status
0000	0000	0110	0011 CAD/C - NIFTP via UCL
0000	0000	0110	0101 NPL - NIFTP via UCL
0000	0000	0110	0111 BNPL - NIFTP via UCL
0000	0000	0110	1001 CAMBRIDGE - NIFTP via UCL
0000	0000	0110	1011 HARWELL - NIFTP via UCL
0000	0000	0110	1101 SWURCC - NIFTP via UCL
0000	0000	0110	1111 ESSEX - NIFTP via UCL
0000	0000	0111	0001 RUTHERFORD - NIFTP via UCL
0000	0000	0111	0011 - 0000 0000 1000 0001 Unassigned
0000	0000	1000	0011 Datacomputer

0000 0000 1000 0100 - 0000 0000 1101 1111 Reserved

0000 0000 1000 0100 - 0000 0000 1101 1111 Reserved

0000 0000 1110 0000 - 1111 1111 1111 1111 Any Experimental Function

Within the Experimental Function the following values are assigned:

0000	0000	1110	0000 - 0000 0000 1110 1111 Unassigned
0000	0000	1111	0001 NCP Measurement
0000	0000	1111	0011 Survey Measurements
0000	0000	1111	0101 LINK
0000	0000	1111	0111 TIPSRV
0000	0000	1111	1001 - 0000 0000 1111 1111 RSEXEC

3. Sequence Number is 32 bits in length. The sequence number of the first data octet (one octet = 8 bits)

in the segment(except when SYN-bit is present). If SYN-bit is present the sequence number is the initial sequence number (ISN) and the first octet is ISN+1.

4. Acknowledgement (ACK) Number is 32 bits in length. If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

5. Data Offset is 4 bits in length. This contains the number of 32 bit words in the TCP Header thus indicating the start of the data. The TCP header will always be an intergral number of 32 bits.

6. Reserved is 6 bits in length. Not used at this time and must be set to zero.

7. Control Bits is 6 bits in length. This field contains the following sub- fields:

- a. URG: Indicates use of URGENT POINTER field
- b. ACK: Indicates use of ACKNOWLEDGEMENT field
- c. PSH: Push Function
- d. RST: Used to reset the connection
- e. SYN: Used to synchronize sequence numbers
- f. FIN: Indicates no more data from sender

8. Window is 16 bits in length. Indicates the number of data octets beginning with the one indicated in the acknowledgement field which the sender of this segment is willing to accept back from the receiver.

9. Checksum is 16 bits in length. The checksum field is provided to enable error detection/correction within the end-to-end environment. While computing the checksum, the checksum field itself is replaced with zeros.

10. Urgent Pointer is 16 bits in length. This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field is only interpreted in segments with the URG control bit set to a one.

11. Options is variable in length. Options may occupy space at the end of the TCP header and is a multiple of 8-bits in length. All options are included in the checksum. An option may begin on any octet boundary. A TCP must be able to implement all options occurring within the DELNET.

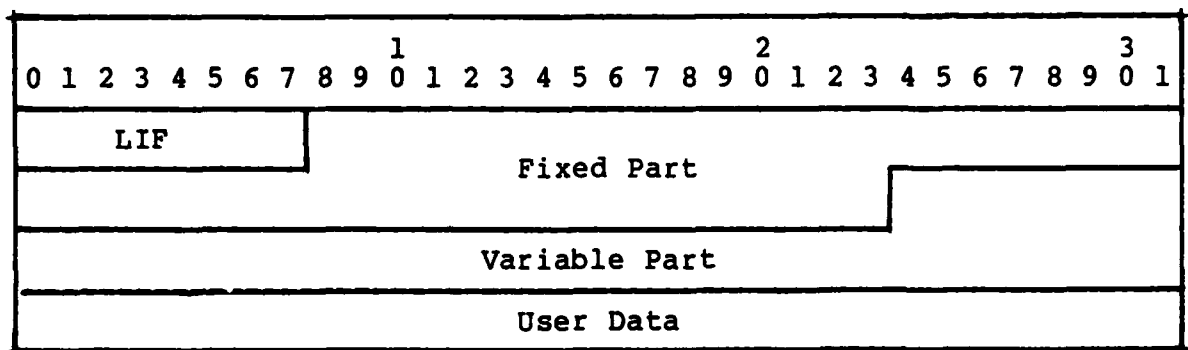
12. Padding Field is of variable length. The TCP Header must end on an even 4-byte boundary.

Virtual Circuit Service

As defined in the previous section, all the Transport Protocol Data Units (TPDU) will contain an integral number of octets. The octets in a TPDU are numbered starting from one and increasing in the order of transmission. The bits in an octet are numbered from zero to seven, where bit zero is the low-ordered bit (Refs 20:64-81, 43:217-241, 42:152-174, and 50:159-171).

There are two types of TPDUs: Data TPDUs used to transfer Transport-Service-Data-Units; and Control TPDUs used to control the transport protocol functions including any optional functions. A TPDU is divided into four parts: Length Indicator Field (LI); Fixed Part; Variable Part (optional);

For this definition, the maximum header format is displayed with corresponding octet locations for specified commands. The reader is reminded that not all commands must be given.



Octet 1 (Bits 0-7): Length Indicator Field. The length is a binary number with a maximum value of 254 (11111110). The length indicated is the header length, including parameters, but excluding the length indicator field and user data (Session, Presentation and Application Layer Headers plus end-users' message/data). The value 255 (11111111) is reserved for future extensions.

Octet 2: TPDU Code. The TPDU code is used to define the structure of the remaining header and contains the following:

Note: The xxxx portion is used to signal the CDT.

0000 0000 - Not Used
 This code is in use in compatible
 protocols not defined by ISO.
 1110 xxxx - CR Connection Request
 xxxx=0000 in Class 0 and 1
 xxxx=0001 in Preferred Class
 1101 xxxx - CC Connection Confirm
 1000 0000 - DR Disconnect Request
 1100 0000 - DC Disconnect Confirm
 1111 0000 - DT Data
 0001 0000 - ED Expedited Data
 0110 xxxx - AK Data Acknowledgement
 0010 0000 - EA Expedited Data Acknowledgement
 0101 xxxx - RJ Reject - In extendedheaderformat
 AK=0110 0000 and RJ=0101 0000
 0111 0000 - ERR TPDU Error
 0011 0000 - Not Used
 This code is in use in compatible
 protocols not defined by ISO.
 1001 xxxx - Not Used
 This code is in use in compatible
 protocols not defined by ISO.
 1010 xxxx - Not Used
 This code is in use in compatible
 protocols not defined by ISO.

Octets 3-4: DST-Reference. Reference selected by the transport entity for identification of the requested transport connection.

0000 0000 0000 0000: Connection Request

Octets 5-6: Source-Reference. Reference selected by the transport machine which identifies uniquely the requested connection in the local system environment.

Exceptions:

1. The source-reference is not used in a Graceful Close Request in Class 4.

2. In an ERR TPDU Octet-5 becomes the Reject Cause with the following four values:

0000 0000 - Reason Not Specified
 0000 0001 - Unimplemented Function
 0000 0010 - Invalid TPDU
 0000 0011 - Invalid Parameter

AD-A138 481

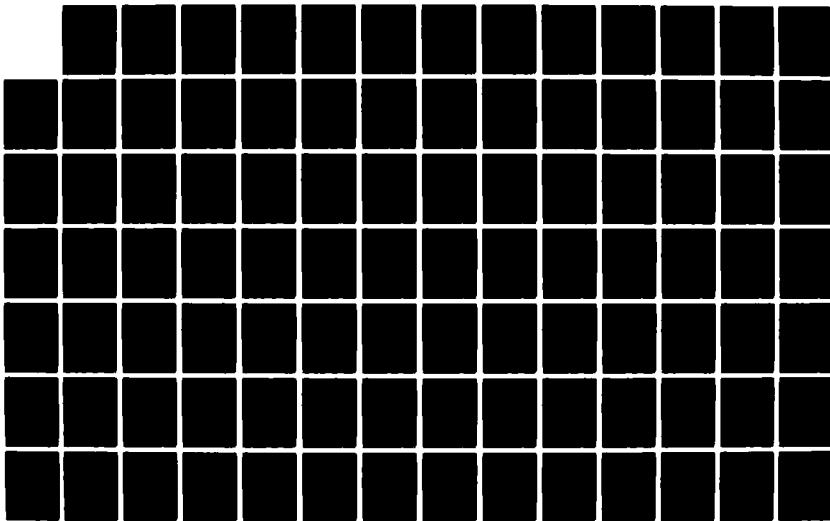
PROTOCOL STANDARDS AND IMPLEMENTATION WITHIN THE
DIGITAL ENGINEERING LABO..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P W PHISTER
DEC 83 AFIT/GE/EE/83D-58-PT-1

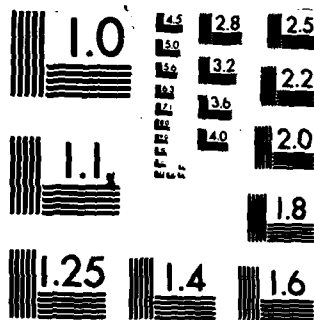
3/4

UNCLASSIFIED

F/G 17/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

3. In a Data TPDU, octets 5 and 6 become the send sequence number of the TPDU. The modulus of the sequence number is either 2^{*7} or 2^{*31} , depending on the size of the sequence space decided during connection establishment. The initial send sequence number is zero. In normal format, this field occupies bits 1-7 of Octet 5; in extended format, it occupies octets 5 through 7 and bits 1-7 of octet 8.

4. In a Expedited Data TPDU, octets 5 and 6 become the expedited sequence number. Its format is the same as in a Data TPDU.

5. In an Acknowledgement TPDU, octets 5 and 6 become the sequence number of the next expected Data TPDU number. Its format is the same as in a Data TPDU.

6. In a Expedited Acknowledgement TPDU, octets 5 and 6 become the sequence number indicating the acknowledged expedited data TPDU. Its format is the same as a Data TPDU. Octet 7 (Bits 56-63): Class Options.

Identifies the class of protocol requested for the desired connection. Acceptable values are:

Bits 56-59: Options to be used on the requested transport connection as follows:

- 0000 - Normal with 7-bit sequence number and 4-bit credit.
- 0010 - Extended with 31-bit sequence number and 16-bit credit.

Bits 60-63: Types of Classes available. Options are:

- 0000 - Class 0
- 0001 - Class 1
- 0010 - Class 2
- 0011 - Class 3
- 0100 - Class 4

Exceptions:

1. In a Disconnect Request TPDU, octet 7 becomes the reason code for disconnecting a transport connection.

Appropriate values are:

0000	0000	- Reason Not Given
0000	0001	- Addressed Transport User Busy
0000	0010	- Addressed Transport User Not Available
0000	0011	- Transport Address Unknown
1000	0000	- Normal User-Initiated Disconnect
1000	0001	- Insufficient Host Resources to Support Connection
1000	0010	- Parameter Negotiation Failed
1000	0011	- Duplicate Connection Detected
1000	0100	- Mismatched References
1000	0101	- Protocol Error
1000	0110	- Unused
1000	0111	- Reference Overflow
1000	1000	- Connection Request Refused
1000	1001	- Unused
1000	1010	- Header or Parameter Length Invalid
1100	1000	- Connection Rejected by Transport User
1100	1001	- Unacceptable Protocol Class
1100	1010	- Unacceptable Security Parameters
1100	1011	- Unacceptable Quality of Service Req.
1100	1100	- Unacceptable Version
1100	1101	- Unacceptable Sequence Space Width
1100	1110	- Unacceptable Use of Checksum Option
1100	1111	- Unacceptable TPDU Size
1111	1111	- Unknown Reason

Octets 8-257 (Bits 64-2055): Variable Part of TPDU Header. The variable part is used to define parameters relating to optional functions. If the variable part is present, it will contain one or more parameters. The number of parameters that may be contained in the variable part is indicated by the length of the variable part (which is a Length Indicator) minus the length of the fixed part. Since the currently defined minimum fixed part for headers which allow parameters is four octets, and since the length indication field is limited to a maximum of 254, the maximum length of the

variable part is 250 octets (2000 bits).

Octet 8 (Bits 64-71): Parameter Code Field (PCF).
Provides for a maximum of 255 different parameters as follows:

- 1100 0011: Checksum Parameter
(Result of Checksum Algorithm)
Must be present for Class 4.
- 1100 0000: TPDU Size
- 1100 0001: Calling Transport Service Access Point ID
Class 4 = Sender Transport Suffix
- 1100 0010: Called Transport Service Access Point ID
Class 4 = Receiver Transport Suffix
- 1100 0100: Version Number (Not used in Class 0)
- 1100 0101: Security Parameters (Not used in Class 0)
- 1100 0110: Additional Option Selection
- 1100 0111: Alt Protocol Class (Not used in Class 0)
- 1100 1000: Unit Data Options
- 1000 0101: Acknowledgement Time
- 1000 1001: Throughput
- 1000 0110: Residual Error Rate
- 1000 0111: Priority
- 1000 1000: Transit Delay

Octet 9 (Bits 72-79): Parameter Length Indication (PLI).
This indicates the length, in octets, of the parameter value field.

Octet 10 (Bits 80-87): Parameter Value Field. Contains the value of the parameter identified in the parameter code field.

Octet 11 (Bits 88-95): TPDU Size. A binary number indicating the proposed maximum TPDU size in octets, including the TPDU headers, to be used on the requested transport connection. Possible values:

- 0000 1101: 8192 octets (not allowed in Class 0 or 1)
- 0000 1100: 4096 octets (not allowed in Class 0 or 1)
- 0000 1011: 2048 octets
- 0000 1010: 1024 octets
- 0000 1001: 512 octets
- 0000 1000: 256 octets
- 0000 0111: 128 octets (Default value if not specified)

Octet 12 (Bits 96-103): Version Number. Value: 0000 0001

Octet 13 (Bits 104-111): Receiver Transport Suffix.

Parameter Code: 1100 0010. The transport user suffix supplied in the Connection Request TPDU which identifies a transport user to a transport machine.

Octet 14 (Bits 112-119): Sender Transport Suffix.

Parameter Code: 1100 0001. The transport user suffix supplied in the Connection Request TPDU which identifies a transport user to a transport machine.

Octets 15-25 (Bits 120-207): Security Field.

Parameter Code: 1100 0101. Allows for 11 octets of user defined information.

Octets 26-27 (Bits 208-223): Checksum.

Parameter Code: 1100 0011. The Fletcher checksum of all the octets in the TPDU, including both header and data portions.

Octet 28 (Bits 224-231): Additional Option Selection. Indicates whether the TPDUs for the remainder of the connection should contain a checksum parameter. Valid values are:

0000 0001 - Checksums should be used
0000 0011 - Checksums should not be used

Octets 29-30 (Bits 232-247): Alternative Protocol Class.

Parameter Code: 1100 0111. This parameter is used to specify alternative protocol classes for the connection. Valid values are:

0000 0000 - Class 0
0000 0001 - Class 1
0000 0010 - Class 2
0000 0011 - Class 3
0000 0100 - Class 4

Octets 31-32 (Bits 248-263): Acknowledgement Time.

Parameter Code: 1000 0101. This parameter conveys the maximum acknowledgement time to the remote transport entity. It is an indication only and not subject to negotiation. The time is expressed in milliseconds.

Octets 33-44 (Bits 264-359): Throughput. This is further broken down as:

- Octets 33-35: Target Value,
calling-called user direction
- Octets 36-38: Minimum Acceptable
calling-called user direction
- Octets 39-41: Target Value
called-calling user direction
- Octets 42-44: Minimum Acceptable
called-calling user direction

Octets 45-47 (Bits 360-383): Residual Error Rate.

Parameter Code: 1000 0110. This is broken down as:

- Octet 45: Target Value, power of 10.
- Octet 46: Minimum Acceptable, power of 10.
- Octet 47: TSDU size of interest, power of 2.

Octets 48-49 (Bits 384-399): Priority.

Parameter Code: 1000 0111. Integer value specified by the user.

Octets 50-57 (Bits 400-463): Transit Delay.

Parameter Code: 1000 1000. Further broken down as:

- Octets 50-51: Target Value
calling-called user direction
- Octets 52-53: Maximum Acceptable
calling-called user direction
- Octets 54-55: Target Value
called-calling user direction
- Octets 56-57: Maximum Acceptable
called-calling user direction

Octet 58 (Bits 464-471): Size of Sequence Space.

This parameter defines the size of the sequence spaces and

credit fields for the remainder of the connection. Three values are defined:

0000 0000 - 4-bit credit, 7-bit sequence
0000 0001 - 8-bit credit, 15-bit sequence
0000 0010 - 16-bit credit, 31-bit sequence

Octet 59 (Bits 472-479): Unit Data Options.

Parameter Code: 1100 1000. This parameter, if present, indicates an attempt to convey unit data, rather than an attempt to open a connection. Acceptable values are:

0000 0010 - 2-way
0000 0011 - 3-way

Octets 60-257 (Bits 480-2055): Remainder of variable section of transport Header.

Octets 258-289 (Bits 2056-2311): User Data. No user data are permitted in Class 0, and are optional in the other classes. Where permitted the users data may not exceed 32 octets.

Session Layer Protocol Specification

All Session Protocol Data Units (SPDUs) contain an integral number of octets. The bits in an octet are numbered from zero to seven, where bit zero is the low order bit and is transmitted first. All SPDUs are divided into three parts similar to the TPDUs. They are: the Fixed Part of the header; the Variable Part of the header; and the Data Field (Ref 7:108-118).

The fixed part of an SPDU header contains the frequently occurring parameters, including the functional type of the SPDU. The functional type of the SPDU determines the length and structure of the remaining fixed part of the header.

Since the protocol headers for the Session Layer differ a great deal, each type of PDU will be shown with the corresponding values. It is at this protocol layer where Datagram and Virtual Circuit Service merge into one protocol specification.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																																	
0 0 0 0 0 0 0 1																Version				DLG		TRN		PDUSZ									
QIN																QOUT																	
Receiver Session Address																Sender Session Address																	
Data Field																																	

```
01 - local
10 - foreign
00 - either
```

Octets 3-4 (Bits 16-31): This parameter is the proposed value for the size of the largest acceptable SPDU to be sent on this session connection. The size is expressed in octets and includes the SPDU headers.

Octets 5-6 (Bits 32-47): This parameter represents the size of the largest quarantined data the session machine can accept, expressed in octets and not including any SPDU headers.

Octets 7-8 (Bits 48-63): This parameter represents the size of the largest amount of quarantined data the session machine desires to send, expressed in octets and not including any SPDU headers.

Octets 9-10 (Bits 64-79): This parameter represents a 16-bit address of the receiver.

Octets 11-12 (Bits 80-95): This parameter represents a 16-bit address of the sender.

Octets 13-49 (Bits 96-391): The Establish PDU can have a Data Field of maximum length of 37 octets.

2. Accept PDU

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
0 0 0 0 0 0 0 1								Version				DLG		TRN		PDUSZ																							
QIN															QOUT																								
Data Field																																							

Figure C-23. Accept PDU Header Format

Octet 1 (Bits 0-7): This is the Accept Code.

Octet 2 (Bits 8-15): Subdivided as follows:

Bits 8-11: Identifies the version of the protocol to be followed during the session connection.

Bits 12-13: Identifies the type of dialogue in force for the session connection. Possible values are:

- 01 - monologue
- 10 - Two-way alternate
- 11 - Two-way simultaneous
- 00 - Any type of dialogue

Bits 14-15: This parameter represents the assigned value for the first turn of the dialogue. Possible values are:

- 01 - local
- 10 - foreign
- 00 - either

Octets 3-4 (Bits 16-31): This parameter is the assigned value for the size of the largest acceptable SPDU to be sent during this session.

Octets 5-6 (Bits 32-47): This parameter is the size of the largest amount of quarantined data the session will accept.

Octets 7-8 (Bits 48-63): This parameter represents the size of the largest amount of quarantined data the current session will send.

Octets 9-20 (Bits 64-159): Optional Data Field with a maximum length of 12 octets.

3. Abort PDU

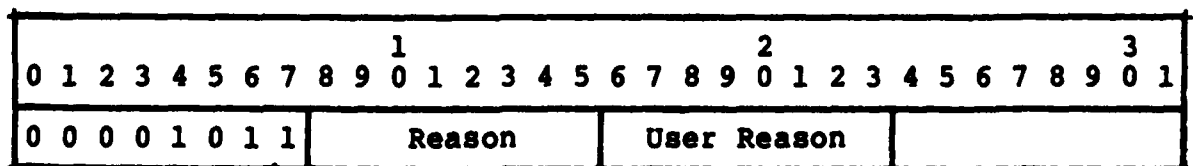


Figure C-24. Abort PDU Header Format

Octet 1 (Bits 0-7): This is the Abort code.

Octet 2 (Bits 8-15): This parameter gives the reason for disconnecting the session connection. Possible values are:

- 0000 0000 - Normal User-Initiated Disconnect
- 0000 0001 - Insufficient Host Resources to Support Connection
- 0000 0010 - Parameter Negotiation Failed
- 0000 0011 - Protocol Error
- 0000 0100 - Data in Establish PDU Too Long

Octet 3 (Bits 16-23): This parameter is supplied by the disconnecting user and delivered to the peer user. This octet is transparently transmitted and is not subject to interpretation by the session layer.

Figures C-25 thru C-29 represent only one octet commands.

4. Finish PDU

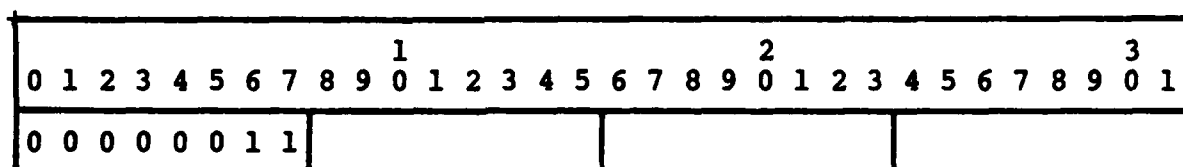


Figure C-25. Finish PDU Header Format

5. Disconnect PDU

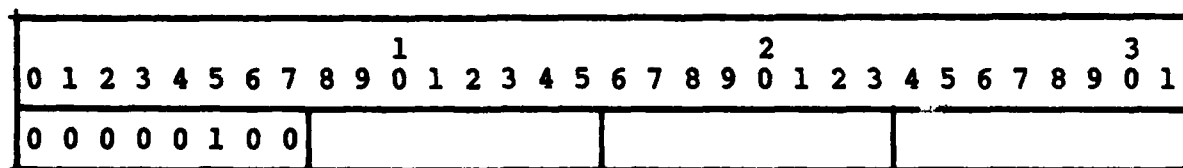


Figure C-26. Disconnect PDU Header Format

6. Not_Finished PDU

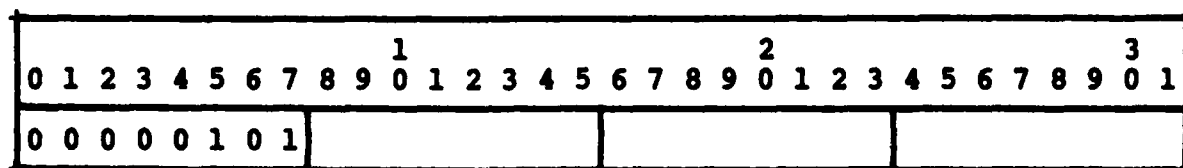


Figure C-27. Not_Finished PDU Header Format

7. Please_Turn PDU

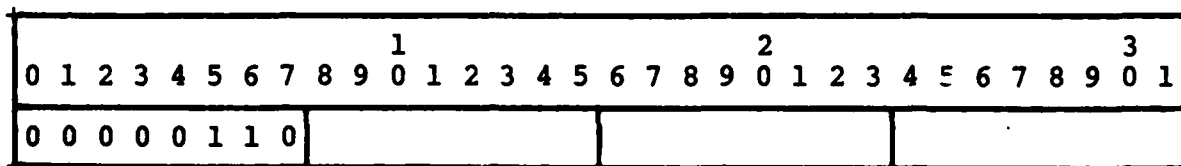


Figure C-28. Please_Turn PDU Header Format

8. Cancel PDU

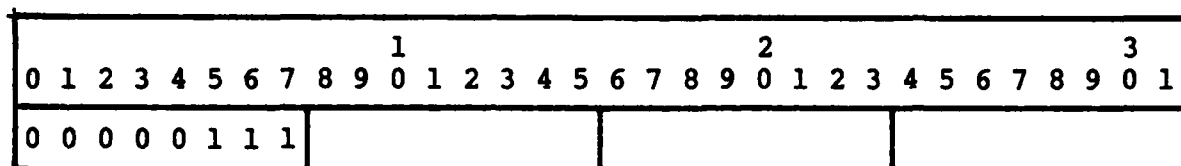


Figure C-29. Cancel PDU Header Format

9. Expedited PDU

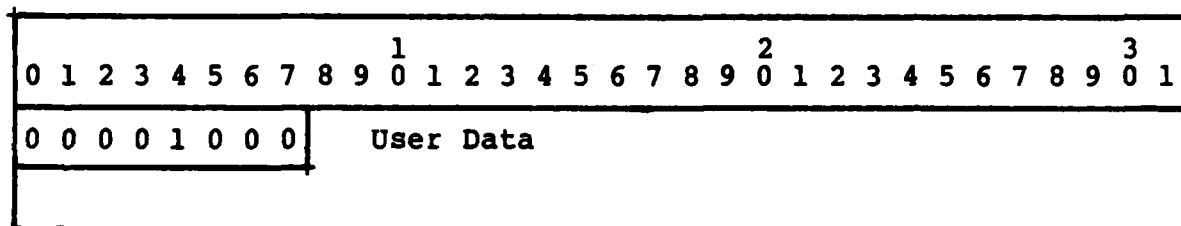


Figure C-30. Expedited PDU Header Format

Octets 2-16 (Bits 8-127): An expedited PDU can contain up to 15 octets of data.

10. Data PDU

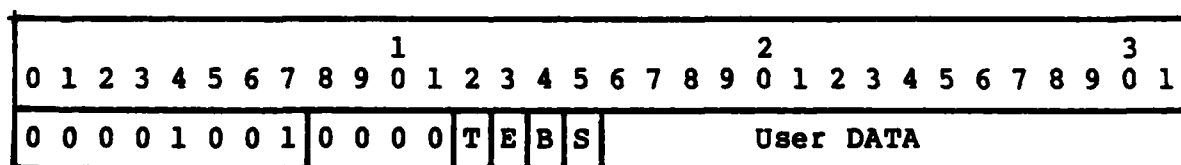


Figure C-31. Data PDU Header Format

Octet 2 (Bits 8-15): This is subdivided as follows:

Bits 8-11: Set to zero

Bit 12 : T (Turn)- On if the Data PDU marks the surrender of the dialogue turn

Bit 13 : E (EOQ) - On if the Data PDU marks the release of quarantined data

- Bit 14 : B (BOQ) - On if the Data PDU marks the beginning of quarantining data
- Bit 15 : S (SSDU) - On if the Data PDU is the last of a series of Data PDUs constituting a Session-Service-Data-Unit

Octet 3-? (Bits 16-?): This data field contains the SPDU data. The length of the data field is limited to the negotiated SPDU size for the session connection minus the length of the Data Header (2-octets). A Data PDU may be shorter than the negotiated length only if it is marked with the SSDU bit on.

11. Transaction PDU

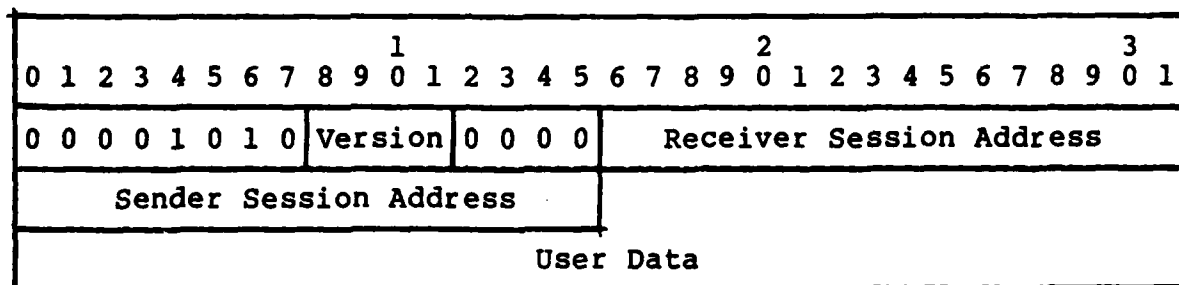


Figure C-32. Transaction PDU Header Format

Octet 1 (Bits 0-7): Transaction Code.

Octet 2 (Bits 8-15): This is broken down as:

Bits 8-11: Identifies the version of the protocol to be followed during the requested session transaction.

Bits 12-15: Set to zeros.

Octets 3-4 (Bits 16-31): This parameter represents the 16-bit address of the receiver.

Octets 5-6 (Bits 32-47): This parameter represents the 16-bit address of the sender.

Octets 7-? (Bits 48-?): The data field contains the session transaction data.

Presentation Layer Protocol Specification

Currently, the Presentation Layer protocol format will consist of only 2-octets as shown in Figure C-33. Ten octets are provided for future expansion and end-user use.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																															
1 2 3																															
Version										Data Format																					
0 0 0 0																															

Figure C-33. Presentation Layer Header Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: Version. This parameter identifies the version of the protocol to be followed during a requested connection.

Bits 4-7: Set to zero.

Octet 2 (Bits 8-15): Data Format. This parameter represents what character format the incoming data represents.

Octets 3-12 (Bits 16-104): Reserved for end-user's use.

Application Layer Protocol Specification

The Application Layer protocol is determined by the individual Hosts and users. Within the DELNET the Application Layer header, as shown in Figure C-34, will be structured the same as the Session Layer using FIPS. The first part will be the fixed part and the second will be the variable part with the user's commands.

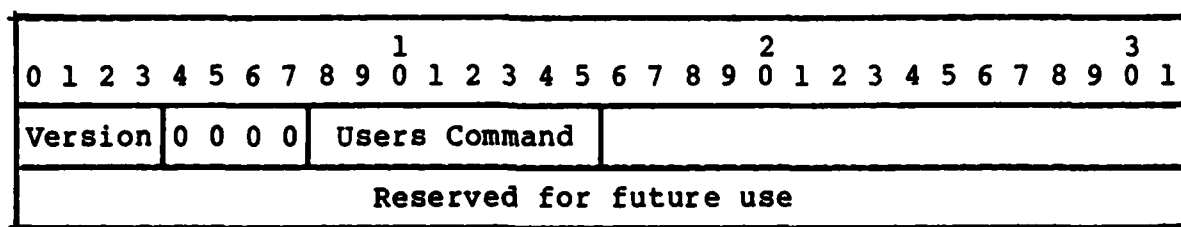


Figure C-34. Application Layer Header Format

Octet 1 (Bits 0-7): Subdivided as follows:

Bits 0-3: Contains the version of the protocol being used.

Bits 4-7: Set to zero.

Octet 2 (Bits 8-15): Contains the desired command to be performed. The structure is left up to the end user.

Octets 3-20 (Bits 16-79): Reserved for future use.

APPENDIX D

Detailed Listing of Test Procedures

Introduction

This appendix contains the detail step-by-step testing procedures for all nine phases of testing as outlined in the Test Plan (Chapter3) and is correspondingly subdivided into nine sections, one for each testing phase. The testing procedure used was path-testing. Since each test was designed to test a specific path some tests may seem redundant.

All memory location values as well as the UNID start-up procedures can be found in Appendix A. The following notes are provided for the reader:

1. Each time the software is modified its corresponding memory locations will subsequently change values. Therefore, the memory locations used in these testing procedures are valid only with the memory map found in Appendix A.

2. When testing with UNID#1 the value of NETWORK_CODE equals: one(1) when local-to-local and two(2) when local-to-network. When testing with UNID#2 the value of NETWORK_CODE equals: two(2) when local-to-local and one(1) when local-to-network.

3. The UNID software should be executed before the testing to set all the required constants. This limits the amount of data that has to be entered for each of the software tests.

4. A list of the Monitor commands can be found in Ref 27 pages 104-107.

Table of Contents

<u>Testing Procedures</u>	<u>Page</u>
Phase I.....	D-3
Stage 1.....	D-3
Stage 2.....	D-19
Stage 3.....	D-23
Stage 4.....	D-27
Stage 5.....	D-31
Phase II.....	D-43
Stage 1.....	D-43
Stage 2.....	D-45
Stage 3.....	D-47
Phase III.....	D-49
Stage 1.....	D-49
Stage 2.....	D-51
Stage 3.....	D-53
Stage 4.....	D-55
Stage 5.....	D-57
Stage 6.....	D-61
Phase IV.....	D-65
Stage 1.....	D-65
Stage 2.....	D-68
Stage 3.....	D-71
Stage 4.....	D-74
Phase V.....	D-77
Stage 1.....	D-77
Stage 2.....	D-78
Stage 3.....	D-79
Stage 4.....	D-80
Phase VI.....	D-81
Stage 1.....	D-81
Stage 2.....	D-82
Phase VII.....	D-83
Stage 1.....	D-83
Stage 2.....	D-84
Phase VIII.....	D-85
Stage 1.....	D-85
Stage 2.....	D-86
Phase IX.....	D-87
Stage 1.....	D-87
Stage 2.....	D-88

Testing Procedures

Phase I Stage 1: Internal transfer of the 128-byte data block from LC01TB, LC02TB, LC03TB, and LC04TB to LCLCTB destined for each of the four local ports. There are sixteen tests in this stage of testing.

1. LC01TB to LCLCTB destined for Local Channel-1:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 21 : Command - D 21AE <cr> 21 <cr> <cr>
LC01TB[18] = 3F : Command - D 21AF <cr> 3F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 13 (Local Host 19)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB reset pointers:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

2. LC01TB to LCLCTB destined for Local Channel-2:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 24 : Command - D 21AE <cr> 24 <cr> <cr>
LC01TB[18] = 3F : Command - D 21AF <cr> 3F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 43 (Local Host 67)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

3. LC01TB to LCLCTB destined for Local Channel-3:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 28 : Command - D 21AE <cr> 28 <cr> <cr>
LC01TB[18] = 8F : Command - D 21AF <cr> 8F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 88 (Local Host 136)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

4. LC01TB to LCLCTB destined for Local Channel-4:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 2C : Command - D 21AE <cr> 2C <cr> <cr>
LC01TB[18] = AF : Command - D 21AF <cr> AF <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = CA (Local Host 202)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

5. LC02TB to LCLCTB destined for Local Channel-1:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 22 : Command - D 26B4 <cr> 22 <cr> <cr>
LC02TB[18] = 0F : Command - D 26B5 <cr> 0F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 20 (Local Host 32)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

6. LC02TB to LCLCTB destined for Local Channel-2:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 26 : Command - D 26B4 <cr> 26 <cr> <cr>
LC02TB[18] = 4F : Command - D 26B5 <cr> 4F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 64 (Local Host 100)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

7. LC02TB to LCLCTB destined for Local Channel-3:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 2A : Command - D 26B4 <cr> 2A <cr> <cr>
LC02TB[18] = 5F : Command - D 26B5 <cr> 5F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = A5 (Local Host 165)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

8. LC02TB to LCLCTB destined for Local Channel-4:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 2D : Command - D 26B4 <cr> 2D <cr> <cr>
LC02TB[18] = DF : Command - D 26B5 <cr> DF <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = DD (Local Host 221)
Port_Code = FF0 (Not Used in this test.)
- c. To indicate reception of 128-byte data block set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

9. LC03TB to LCLCTB destined for Local Channel-1:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 21 : Command - D 2BBA <cr> 21 <cr> <cr>
LC03TB[18] = 9F : Command - D 2BBB <cr> 9F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 19 (Local Host 25)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

10. LC03TB to LCLCTB destined for Local Channel-2:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 24 : Command - D 2BBA <cr> 24 <cr> <cr>
LC03TB[18] = 1F : Command - D 2BBB <cr> 1F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 41 (Local Host 65)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

11. LC03TB to LCLCTB destined for Local Channel-3:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 2A : Command - D 2BBA <cr> 2A <cr> <cr>
LC03TB[18] = AF : Command - D 2BBB <cr> AF <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = AA (Local Host 170)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

12. LC03TB to LCLCTB destined for Local Channel-4:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 2C : Command - D 2BBA <cr> 2C <cr> <cr>
LC03TB[18] = 3F : Command - D 2BBB <cr> 3F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = C3 (Local Host 195)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

13. LC04TB to LCLCTB destined for Local Channel-1:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 20 : Command - D 30C0 <cr> 20 <cr> <cr>
LC04TB[18] = 3F : Command - D 30C1 <cr> 3F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 03 (Local Host 03)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

14. LC04TB to LCLCTB destined for Local Channel-2:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 24 : Command - D 30C0 <cr> 24 <cr> <cr>
LC04TB[18] = 1F : Command - D 30C1 <cr> 1F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 41 (Local Host 65)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

15. LC04TB to LCLCTB destined for Local Channel-3:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 28 : Command - D 30C0 <cr> 28 <cr> <cr>
LC04TB[18] = CF : Command - D 30C1 <cr> CF <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 8C (Local Host 140)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

16. LC04TB to LCLCTB destined for Local Channel-4:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 2F : Command - D 30C0 <cr> 2F <cr> <cr>
LC04TB[18] = EF : Command - D 30C1 <cr> EF <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = FE (Local Host 254)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 3AAF <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCLCTB : Command - D 35B5 80 <cr>
- b. To keep data in first 128-bytes of LCLCTB:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>

Phase I Stage 2: Internal transfer of the 128-byte data block from LC01TB, LC02TB, LC03TB, and LC04TB to LCNTTB destined for the network side of the UNID. There are four tests in this stage of testing.

1. LC01TB to LCNTTB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 10 : Command - D 21AE <cr> 10 <cr> <cr>
LC01TB[18] = 8F : Command - D 21AF <cr> 8F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCNTTB : Command - D 8082 85 <cr>
LCNTTB should contain a valid DATA_PACKET with the following header : 11 21 00 00 00.
- b. To keep data in first 133-bytes of LCNTTB reset pointers:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>

2. LC02TB to LCNTTB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 14 : Command - D 26B4 <cr> 14 <cr> <cr>
LC02TB[18] = 0F : Command - D 26B5 <cr> 0F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26AB 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = 40 (Local Host 64)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCNTTB : Command - D 8082 85 <cr>
LCNTTB should contain a valid DATA_PACKET with the following header: 12 22 00 00 00.
- b. To keep data in first 133-bytes of LCNTTB:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>

3. LC03TB to LCNTTB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 1B : Command - D 2BBA <cr> 1B <cr> <cr>
LC03TB[18] = FF : Command - D 2BBB <cr> FF <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = BF (Local Host 191)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:.

- a. The data should be in LCNTTB : Command - D 8082 85 <cr>
LCNTTB should contain a valid DATA_PACKET with the following header: 13 23 00 00 00.
- b. To keep data in first 133-bytes of LCNTTB:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>

4. LC04TB to LCNTTB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 1C : Command - D 30C0 <cr> 1C <cr> <cr>
LC04TB[18] = 0F : Command - D 30C1 <cr> 0F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = C0 (Local Host 192)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. The data should be in LCNTTB : Command - D 8082 85 <cr>
LCNTTB should contain a valid DATA_PACKET with the following header: 14 24 00 00 00.
- b. To keep data in first 133-bytes of LCNTTB:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>

Phase I Stage 3: Internal transfer of the 128-byte data block from table NTLCTB to the four local channel ports. There are four tests to perform in this stage of testing.

1. NTLCTB to Local Channel-1 Port:

Set-up using the Local Monitor:

- a. Place 133-bytes of data (HEX=85) into NTLCTB, set:
All bytes = CC : Command - F 85BA 863F CC <cr>
- b. Fill Destination Address bytes in NTLCTB as follows:
NTLCTB[00] = 11 : Command - D 85BA <cr> 11 <cr> <cr>
To examine : Command - D 85BA 85 <cr>
This will set the destination address as:
Destination UNID Number = 1
Destination Channel Number = 1
- c. To indicate reception of 133-byte data block set:
NTLCNE = 85 : Command - D 8AEE <cr> 85 <cr> <cr>
NTLCNS = 00 : Command - D 8AEC <cr> 00 <cr> <cr>
- d. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

The 128-byte block of data should be transmitted out
Local Channel-1 port.
Command - Viewed on the Local Monitor.

2. NTLCTB to Local Channel-2 Port:

Set-up using Local Monitor:

- a. Place 133-bytes of data (HEX=85) into NTLCTB, set:
All bytes = CC : Command - F 85BA 863F CC <cr>
- b. Fill Destination Address bytes in NTLCTB as follows:
NTLCTB[00] = 12 : Command - D 85BA <cr> 12 <cr> <cr>
To examine : Command - D 85BA 85 <cr>
This will set the destination address as:
Destination UNID Number = 1
Destination Channel Number = 2
- c. To indicate reception of 133-byte data block set:
NTLCNE = 85 : Command - D 8AEE <cr> 85 <cr> <cr>
NTLCNS = 00 : Command - D 8AEC <cr> 00 <cr> <cr>
- d. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

The 128-byte block of data should be transmitted out
Local Channel-2 port.
Command - Viewed on Local Monitor

3. NTLCTB to Local Channel-3 Port:

Set-up using Local Monitor:

- a. Place 133-bytes of data (HEX=85) into NTLCTB, set:
All bytes = CC : Command - F 85BA 863F CC <cr>
- b. Fill Destination Address bytes in NTLCTB as follows:
NTLCTB[00] = 13 : Command - D 85BA <cr> 13 <cr> <cr>
To examine : Command - D 85BA 85 <cr>
This will set the destination address as:
Destination UNID Number = 1
Destination Channel Number = 3
- c. To indicate reception of 133-byte data block set:
NTLCNE = 85 : Command - D 8AEE <cr> 85 <cr> <cr>
NTLCNS = 00 : Command - D 8AEC <cr> 00 <cr> <cr>
- d. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

The 128-byte block of data should be transmitted out
Local Channel-3 port.
Command - Viewed on Local Monitor

4. NTLCTB to Local Channel-4 Port:

Set-up using Local Monitor:

- a. Place 133-bytes of data (HEX=85) into NTLCTB, set:
All bytes = CC : Command - F 85BA 863F CC <cr>
- b. Fill Destination Address bytes in NTLCTB as follows:
NTLCTB[00] = 14 : Command - D 85BA <cr> 14 <cr> <cr>
To examine : Command - D 85BA 85 <cr>
This will set the destination address as:
Destination UNID Number = 1
Destination Channel Number = 4
- c. To indicate reception of 133-byte data block set:
NTLCNE = 85 : Command - D 8AEE <cr> 85 <cr> <cr>
NTLCNS = 00 : Command - D 8AEC <cr> 00 <cr> <cr>
- d. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

The 128-byte data block should be transmitted out
Local Channel-4 port.
Command - Viewed on Local Monitor

Phase I Stage 4: Internal routing of 128-byte data block from LCLCTB out the four local channels. There are four tests in this stage of testing.

1. LCLCTB to Local Channel-1 Port:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LCLCTB, set:
All bytes = CC : Command - F 35B5 3635 CC <cr>
- b. Fill Destination Address bytes in LCLCTB as follows:
LCLCTB[16] = 01 : Command - D 35C5 <cr> 01 <cr> <cr>
LCLCTB[17] = 11 : Command - D 35C6 <cr> 11 <cr> <cr>
LCLCTB[18] = 0F : Command - D 35C7 <cr> 0F <cr> <cr>
LCLCTB[19] = F0 : Command - D 35C8 <cr> F0 <cr> <cr>
To examine : Command - D 35B5 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = 10 (Local Host 16)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LCLCNE = 80 : Command - D 3AB7 <cr> 80 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- d. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- e. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- f. Start UNID : Command - GO <cr>

Display Results:

The 128-byte data block should go out Local Channel-1 port.
Command - Viewed on Local Monitor

2. LCLCTB to Local Channel-2 Port:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LCLCTB, set:
All bytes = CC : Command - F 35B5 3635 CC <cr>
- b. Fill Destination Address bytes in LCLCTB as follows:
LCLCTB[16] = 01 : Command - D 35C5 <cr> 01 <cr> <cr>
LCLCTB[17] = 15 : Command - D 35C6 <cr> 15 <cr> <cr>
LCLCTB[18] = 0F : Command - D 35C7 <cr> 0F <cr> <cr>
LCLCTB[19] = F0 : Command - D 35C8 <cr> F0 <cr> <cr>
To examine : Command - D 35B5 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = 50 (Local Host 80)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LCLCNE = 80 : Command - D 3AB7 <cr> 80 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- d. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- e. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- f. Start UNID : Command - GO <cr>

Display Results:

The 128-byte data block should go out Local Channel-2 port.
Command - Viewed on Local Monitor

3. LCLCTB to Local Channel-3 Port:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LCLCTB, set:
All bytes = CC : Command - F 35B5 3635 CC <cr>
- b. Fill Destination Address bytes in LCLCTB as follows:
LCLCTB[16] = 01 : Command - D 35C5 <cr> 01 <cr> <cr>
LCLCTB[17] = 1B : Command - D 35C6 <cr> 1B <cr> <cr>
LCLCTB[18] = 4F : Command - D 35C7 <cr> 4F <cr> <cr>
LCLCTB[19] = F0 : Command - D 35C8 <cr> F0 <cr> <cr>
To examine : Command - D 35B5 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = B4 (Local Host 180)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LCLCNE = 80 : Command - D 3AB7 <cr> 80 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- d. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- e. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- f. Start UNID : Command - GO <cr>

Display Results:

The 128-byte data block should go out Local Channel-3 port.
Command - Viewed on Local Monitor

4. LCLCTB to Local Channel-4 Port:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LCLCTB, set:
All bytes = CC : Command - F 35B5 3635 CC <cr>
- b. Fill Destination Address bytes in LCLCTB as follows:
LCLCTB[16] = 01 : Command - D 35C5 <cr> 01 <cr> <cr>
LCLCTB[17] = 1E : Command - D 35C6 <cr> 1E <cr> <cr>
LCLCTB[18] = 1F : Command - D 35C7 <cr> 1F <cr> <cr>
LCLCTB[19] = F0 : Command - D 35C8 <cr> F0 <cr> <cr>
To examine : Command - D 35B5 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = E1 (Local Host 225)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LCLCNE = 80 : Command - D 3AB7 <cr> 80 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- d. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- e. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- f. Start UNID : Command - GO <cr>

Display Results:

The 128-byte data block should go out Local Channel-4 port.
Command - Viewed on Local Monitor

Phase I Stage 5: Internal detection of invalid CONTROL_CODE, COUNTRY_CODE, and NETWORK_CODE within the incoming 128-byte data block. There are 12 tests in this stage of testing.

1. Invalid CONTROL_CODE in LC01TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 11 : Command - D 21AD <cr> 11 <cr> <cr>
LC01TB[17] = 21 : Command - D 21AE <cr> 21 <cr> <cr>
LC01TB[18] = 3F : Command - D 21AF <cr> 3F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 1 (INVALID)
Country_Code = 1
Network_Code = 2
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-1 in-processing.
- b. STATTB should reflect a CONTROL_CODE error.
Command - D 8AF2 23 <cr>

2. Invalid COUNTRY_CODE in LC01TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 04 : Command - D 21AD <cr> 04 <cr> <cr>
LC01TB[17] = 21 : Command - D 21AE <cr> 21 <cr> <cr>
LC01TB[18] = 3F : Command - D 21AF <cr> 3F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 4 (INVALID)
Network_Code = 2
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-1 in-processing.
- b. STATTB should reflect a COUNTRY_CODE error.
Command - D 8AF2 23 <cr>

3. Invalid NETWORK_CODE in LC01TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 91 : Command - D 21AE <cr> 91 <cr> <cr>
LC01TB[18] = 3F : Command - D 21AF <cr> 3F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 9 (INVALID)
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-1 in-processing
- b. STATTB should reflect a NETWORK_CODE error.
Command - D 8AF2 23 <cr>

4. Invalid CONTROL_CODE in LC02TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 11 : Command - D 26B3 <cr> 11 <cr> <cr>
LC02TB[17] = 21 : Command - D 26B4 <cr> 21 <cr> <cr>
LC02TB[18] = 3F : Command - D 26B5 <cr> 3F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 1 (INVALID)
Country_Code = 1
Network_Code = 2
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-2 in-processing.
- b. STATTB should reflect a CONTROL_CODE error.
Command - D 8AF2 23 <cr>

5. Invalid COUNTRY_CODE in LC02TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 03 : Command - D 26B3 <cr> 03 <cr> <cr>
LC02TB[17] = 21 : Command - D 26B4 <cr> 21 <cr> <cr>
LC02TB[18] = 3F : Command - D 26B5 <cr> 3F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 3 (INVALID)
Network_Code = 2
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-2 in-processing.
- b. STATTB should reflect a COUNTRY_CODE error.
Command - D 8AF2 23 <cr>

6. Invalid NETWORK_CODE in LC02TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = A1 : Command - D 26B4 <cr> A1 <cr> <cr>
LC02TB[18] = 3F : Command - D 26B5 <cr> 3F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = A (INVALID)
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-2 in-processing.
- b. STATTB should reflect a NETWORK_CODE error.
Command - D 8AF2 23 <cr>

7. Invalid CONTROL_CODE in LC03TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 11 : Command - D 2BB9 <cr> 11 <cr> <cr>
LC03TB[17] = 21 : Command - D 2BBA <cr> 21 <cr> <cr>
LC03TB[18] = 3F : Command - D 2BBB <cr> 3F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 1 (INVALID)
Country_Code = 1
Network_Code = 2
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-3 in-processing.
- b. STATTB should reflect a CONTROL_CODE error.
Command - D 8AF2 23 <cr>

8. Invalid COUNTRY_CODE in LC03TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 06 : Command - D 2BB9 <cr> 06 <cr> <cr>
LC03TB[17] = 21 : Command - D 2BBA <cr> 21 <cr> <cr>
LC03TB[18] = 3F : Command - D 2BBB <cr> 3F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 6 (INVALID)
Network_Code = 2
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-3 in-processing.
- b. STATTB should reflect a COUNTRY_CODE error.
Command - D 8AF2 23 <cr>

9. Invalid NETWORK_CODE in LC03TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = F1 : Command - D 2BBA <cr> F1 <cr> <cr>
LC03TB[18] = 3F : Command - D 2BBB <cr> 3F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = F (INVALID)
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-3 in-processing.
- b. STATTB should reflect a NETWORK_CODE error.
Command - D 8AF2 23 <cr>

10. Invalid CONTROL_CODE in LC04TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 11 : Command - D 30BF <cr> 11 <cr> <cr>
LC04TB[17] = 21 : Command - D 30C0 <cr> 21 <cr> <cr>
LC04TB[18] = 3F : Command - D 30C1 <cr> 3F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 1 (INVALID)
Country_Code = 1
Network_Code = 2
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-4 in-processing.
- b. STATTB should reflect a CONTROL_CODE error.
Command - D 8AF2 23 <cr>

11. Invalid COUNTRY_CODE in LC04TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 07 : Command - D 30BF <cr> 07 <cr> <cr>
LC04TB[17] = 21 : Command - D 30C0 <cr> 21 <cr> <cr>
LC04TB[18] = 3F : Command - D 30C1 <cr> 3F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 7 (INVALID)
Network_Code = 2
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-4 in-processing.
- b. STATTB should reflect a COUNTRY_CODE error.
Command - D 8AF2 23 <cr>

12. Invalid NETWORK_CODE in LC04TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 81 : Command - D 30C0 <cr> 81 <cr> <cr>
LC04TB[18] = 3F : Command - D 30C1 <cr> 3F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 8 (INVALID)
Host_Code = 13 (Local Host 13)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. An error should have occurred during Channel-4 in-processing.
- b. STATTB should reflect a NETWORK_CODE error.
Command - D 8AF2 23 <cr>

Phase II Stage 1: Internal data transfer from LCNTTB to an outgoing network port. There are two tests in this stage of testing.

1. LCNTTB to OUTFRAME_CHA_TB:

Set-up using Network Monitor:

- a. Place 133-bytes of data (HEX=85) in LCNTTB, set:
All bytes = CC : Command - F 8082 8107 CC <cr>
- b. Fill DATA_PACKET Header bytes in LCNTTB as follows:
LCNTTB[0] = 11 : Command - D 8082 <cr> 11 <cr> <cr>
LCNTTB[1] = 21 : Command - D 8083 <cr> 21 <cr> <cr>
LCNTTB[2] = 00 : Command - D 8084 <cr> 00 <cr> <cr>
LCNTTB[3] = 00 : Command - D 8085 <cr> 00 <cr> <cr>
LCNTTB[4] = 00 : Command - D 8086 <cr> 00 <cr> <cr>
To examine : Command - D 8082 85 <cr>
This will set the DATA_PACKET Header to:
Destination_Address = UNID#1 CH#1
Source_Address = UNID#2 CH#1
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 133-byte DATA_PACKET, set:
LCNTNE = 85 : Command - D 85B6 <cr> 85 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- d. Prepare OUTFRAME_CHA_TB for reception, set:
OUTFRAME_CHA_NE = 00 : Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00 : Command - D 31DB <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

- a. A valid I-Frame should be located in OUTFRAME_CHA_TB.
Command - D 2C95 87 <cr>

2. LCNTTB to OUTFRAME_CHB_TB:

Set-up using Network Monitor:

- a. Place 133-bytes of data (HEX=85) into LCNTTB, set:
All bytes = CC : Command - F 8082 8107 CC <cr>
- b. Fill DATA_PACKET Header bytes in LCNTTB as follows:
LCNTTB[0] = 31 : Command - D 8082 <cr> 31 <cr> <cr>
LCNTTB[1] = 21 : Command - D 8083 <cr> 21 <cr> <cr>
LCNTTB[2] = 00 : Command - D 8084 <cr> 00 <cr> <cr>
LCNTTB[3] = 00 : Command - D 8085 <cr> 00 <cr> <cr>
LCNTTB[4] = 00 : Command - D 8086 <cr> 00 <cr> <cr>
To examine : Command - D 8082 85 <cr>
This will set the DATA_PACKET Header to:
Destination_Address = UNID#1 CH#1
Source_Address = UNID#2 CH#1
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 133-byte DATA_PACKET, set:
LCNTNE = 85 : Command - D 85B6 <cr> 85 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- d. Prepare OUTFRAME_CHB_TB for reception, set:
OUTFRAME_CHB_NE = 00 : Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00 : Command - D 3727 <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID : Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHB_TB.
Command - D 31E1 87 <cr>

Phase II Stage 2: Internal data transfer from each incoming network channel to NTLCTB. There are two tests within this stage of testing.

1. NT01TB to NTLCTB:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT01TB, set:
All bytes = CC : Command - F 21FD 2284 CC <cr>

- b. Fill I-Frame Header bytes in NT01TB as follows:

NT01TB[0] = 21 : Command - D 21FD <cr> 21 <cr> <cr>
NT01TB[1] = 00 : Command - D 21FE <cr> 00 <cr> <cr>
NT01TB[2] = 21 : Command - D 21FF <cr> 21 <cr> <cr>
NT01TB[3] = 14 : Command - D 2200 <cr> 14 <cr> <cr>
NT01TB[4] = 00 : Command - D 2201 <cr> 00 <cr> <cr>
NT01TB[5] = 00 : Command - D 2202 <cr> 00 <cr> <cr>
NT01TB[6] = 00 : Command - D 2203 <cr> 00 <cr> <cr>

To examine : Command - D 21FD 87 <cr>

This will set the I-Frame Header to:

UNID Address = To-UNID#2 From-UNID#1
Control = 00
Destination_Address = UNID#2 CH#1
Source_Address = UNID#1 CH#4
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)

- c. To indicate reception of 135-byte I-Frame, set:

NT01NE = 87 : Command - D 2745 <cr> 87 <cr> <cr>
NT01NS = 00 : Command - D 2743 <cr> 00 <cr> <cr>

- d. Prepare NTLCTB for reception, set:

NTLCNE = 00 : Command - D 8AEE <cr> 00 <cr> <cr>
NTLCNS = 00 : Command - D 8AEC <cr> 00 <cr> <cr>

- e. To keep outgoing S-Frame in first 135-bytes of OUTFRAME_CHA_TB, set:

OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr> <cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr> <cr>

- f. In order not to eliminate the preset commands, set:

INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>

- g. Set Program Counter to start of Procedure MAIN.

Command - R (space) PC (space) 0000 (space) 4E35 <cr>

- h. Start UNID : Command - GO <cr>

Display Results:

- a. A valid DATA_PACKET should be located in NTLCTB

Command - D 85BA 80 <cr>

- b. A valid S-Frame should be located in OUTFRAME_CHA_TB

Command - D 2C95 87 <cr>

2. NT02TB to NTLCTB:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT02TB, set:
All bytes = CC : Command - F 2749 27D0 <cr>
- b. Fill I-Frame Header bytes in NT02TB as follows:
NT02TB[0] = 23 : Command - D 2749 <cr> 23 <cr> <cr>
NT02TB[1] = 00 : Command - D 274A <cr> 00 <cr> <cr>
NT02TB[2] = 21 : Command - D 274B <cr> 21 <cr> <cr>
NT02TB[3] = 34 : Command - D 274C <cr> 34 <cr> <cr>
NT02TB[4] = 00 : Command - D 274D <cr> 00 <cr> <cr>
NT02TB[5] = 00 : Command - D 274E <cr> 00 <cr> <cr>
NT02TB[6] = 00 : Command - D 274F <cr> 00 <cr> <cr>
To examine : Command - D 2749 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#3
Control = 00
Destination_Address = UNID#2 CH#1
Source_Address = UNID#3 CH#4
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT02NE = 87 : Command - D 2C91 <cr> 87 <cr> <cr>
NT02NS = 00 : Command - D 2C8F <cr> 00 <cr> <cr>
- d. Prepare NTLCTB for reception, set:
NTLCNE = 00 : Command - D 8AEE <cr> 00 <cr> <cr>
NTLCNS = 00 : Command - D 8AEC <cr> 00 <cr> <cr>
- e. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHB_TB, set:
OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr> <cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. A valid DATA_PACKET should be located in NTLCTB
Command - D 85BA 80 <cr>
- b. A valid S-Frame should be located in OUTFRAME_CHB_TB
Command - D 31E1 87 <cr>

Phase II Stage 3: Internal Network-to-Network Data Transfer.
There are two tests within this stage of testing.

1. NT01TB to OUTFRAME_CHB_TB:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT01TB, set:
All bytes = CC : Command - F 21FD 2284 CC <cr>

- b. Fill I-Frame Header bytes in NT01TB as follows:

NT02TB[0] = 31 : Command - D 21FD <cr> 31 <cr> <cr>
NT02TB[1] = 00 : Command - D 21FE <cr> 00 <cr> <cr>
NT02TB[2] = 31 : Command - D 21FF <cr> 31 <cr> <cr>
NT02TB[3] = 14 : Command - D 2200 <cr> 14 <cr> <cr>
NT02TB[4] = 00 : Command - D 2201 <cr> 00 <cr> <cr>
NT02TB[5] = 00 : Command - D 2202 <cr> 00 <cr> <cr>
NT02TB[6] = 00 : Command - D 2203 <cr> 00 <cr> <cr>

To examine : Command - D 21FD 87 <cr>

This will set the I-Frame Header to:

UNID Address = To-UNID#3 From-UNID#1
Control = 00
Destination_Address = UNID#3 CH#1
Source_Address = UNID#1 CH#4
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)

- c. To indicate reception of 135-byte I-Frame, set:

NT01NE = 87 : Command - D 2745 <cr> 87 <cr> <cr>
NT01NS = 00 : Command - D 2743 <cr> 00 <cr> <cr>

- d. Prepare OUTFRAME_CHB_TB for reception, set:

OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr> <cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr> <cr>

- e. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHA_TB, set:

OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr> <cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr> <cr>

- f. In order not to eliminate the preset commands, set:

INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>

- g. Set Program Counter to start of Procedure MAIN.

Command - R (space) PC (space) 0000 (space) 4E35 <cr>

- h. Start UNID : Command - GO <cr>

Display Results:

- a. A valid I-Frame should be located in OUTFRAME_CHB_TB
Command - D 31E1 87 <cr>
b. A valid S-Frame should be located in OUTFRAME_CHA_TB
Command - D 2C95 87 <cr>

2. NT02TB to OUTFRAME_CHA_TB:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT02TB, set:
All bytes = CC : Command - F 2749 27D0 CC <cr>
- b. Fill I-Frame Header bytes in NT02TB as follows:
NT02TB[0] = 13 : Command - D 2749 <cr> 13 <cr> <cr>
NT02TB[1] = 00 : Command - D 274A <cr> 00 <cr> <cr>
NT02TB[2] = 14 : Command - D 274B <cr> 14 <cr> <cr>
NT02TB[3] = 34 : Command - D 274C <cr> 34 <cr> <cr>
NT02TB[4] = 00 : Command - D 274D <cr> 00 <cr> <cr>
NT02TB[5] = 00 : Command - D 274E <cr> 00 <cr> <cr>
NT02TB[6] = 00 : Command - D 274F <cr> 00 <cr> <cr>
To examine : Command - D 2749 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#1 From-UNID#3
Control = 00
Destination_Address = UNID#1 CH#4
Source_Address = UNID#3 CH#4
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT02NE = 87 : Command - D 2C91 <cr> 87 <cr> <cr>
NT02NS = 00 : Command - D 2C8F <cr> 00 <cr> <cr>
- d. Prepare OUTFRAME_CHA_TB for reception, set:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr> <cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr> <cr>
- e. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHB_TB, set:
OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr> <cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. A valid I-Frame should be located in OUTFRAME_CHA_TB
Command - D 2C95 87 <cr>
- b. A valid S-Frame should be located in OUTFRAME_CHB_TB
Command - D 31E1 87 <cr>

Phase III Stage 1: Data transfer from Local Channel-1 (LC01TB) to each of the two outgoing network channels. There are two tests in this stage of testing.

1. LC01TB to OUTFRAME_CHA_TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 10 : Command - D 21AE <cr> 10 <cr> <cr>
LC01TB[18] = 8F : Command - D 21AF <cr> 8F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte datablock, set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. Prepare OUTFRAME_CHA_TB for reception, set on Network Monitor:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- f. In order not to eliminate the preset commands, set:
Local Monitor:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter for Local and Network side of UNID to start of Procedure MAIN.
Command : Network Monitor -
R (space) PC (space) 0000 (space) 4E35 <cr>
Local Monitor -
R (space) PC (space) 0000 (space) 588D <cr>
- h. Start Network then Local side of UNID: Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHA_TB
Network Monitor: Command - D 2C95 87 <cr>

2. LC01TB to OUTFRAME_CHB_TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>

- b. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 30 : Command - D 21AE <cr> 30 <cr> <cr>
LC01TB[18] = 8F : Command - D 21AF <cr> 8F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command D 219D 80 <cr>

This will set the destination address as:

Control_Code = 0
Country_Code = 1
Network_Code = 3
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)

- c. To indicate reception of 128-byte datablock, set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>

- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>

- e. Prepare OUTFRAME_CHB_TB for reception, set on Network Monitor:
OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr><cr>

- f. In order not to eliminate the preset commands, set on Local Monitor:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>

- g. Set Program Counter for Network and Local side of UNID to start of Procedure MAIN.
Command : Network Monitor -
R (space) PC (space) 0000 (space) 4E35 <cr>
Local Monitor -
R (space) PC (space) 0000 (space) 588D <cr>

- h. Start Network then Local side of UNID: Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHB_TB
Network Monitor: Command - D 31E1 87 <cr>

Phase III Stage 2: Data transfer from Local Channel-2 (LC02TB) to each of the two outgoing network channels. There are two tests in this stage of testing.

1. LC02TB to OUTFRAME_CHA_TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 10 : Command - D 26B4 <cr> 10 <cr> <cr>
LC02TB[18] = 8F : Command - D 26B5 <cr> 8F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block, set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. Prepare OUTFRAME_CHA_TB for reception, set on Network Monitor:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- f. In order not to eliminate the preset commands, set on Local Monitor:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter for both Network and Local side of UNID to start of Procedure MAIN.
Command : Network Monitor -
R (space) PC (space) 0000 (space) 4E35 <cr>
Local Monitor -
R (space) PC (space) 0000 (space) 588D <cr>
- h. Start Network then Local side of UNID: Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHA_TB
Command - D 2C95 87 <cr>

2. LC02TB to OUTFRAME_CHB_TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- b. Fill Destination Address bytes in LC02TB on Local Monitor as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 30 : Command - D 26B4 <cr> 30 <cr> <cr>
LC02TB[18] = 8F : Command - D 26B5 <cr> 8F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 3
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block, set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. Prepare OUTFRAME_CHB_TB for reception, set on Network Monitor:
OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr><cr>
- f. In order not to eliminate the preset commands, set on Local Monitor:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter for Network and Local side of UNID to start of Procedure MAIN.
Command : Network Monitor -
R (space) PC (space) 0000 (space) 4E35 <cr>
Local Monitor -
R (space) PC (space) 0000 (space) 588D <cr>
- h. Start Network then Local side of UNID: Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHB_TB
Command - D 31E1 87 <cr>

Phase III Stage 3: Data transfer from Local Channel-3 (LC03TB) to each of the two outgoing network channels. There are two tests in this stage of testing.

1. LC03TB to OUTFRAME_CHA_TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 10 : Command - D 2BBA <cr> 10 <cr> <cr>
LC03TB[18] = 8F : Command - D 2BBB <cr> 8F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block, set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. Prepare OUTFRAME_CHA_TB for reception, set on Network Monitor:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- f. In order not to eliminate the preset commands, set on Local Monitor:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter for both Network and Local side of UNID to start of Procedure MAIN.
Command : Network Monitor -
R (space) PC (space) 0000 (space) 4E35 <cr>
Local Monitor -
R (space) PC (space) 0000 (space) 588D <cr>
- h. Start Network then Local side of UNID: Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHA_TB
Network Monitor: Command - D 2C95 87 <cr>

2. LC03TB to OUTFRAME_CHB_TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- b. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 30 : Command - D 2BBA <cr> 30 <cr> <cr>
LC03TB[18] = 8F : Command - D 2BBB <cr> 8F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 3
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block, set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. Prepare OUTFRAME_CHB_TB for reception, set on Network Monitor:
OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr><cr>
- f. In order not to eliminate the preset commands, set on Local Monitor:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter for Network and Local side of UNID to start of Procedure MAIN.
Command : Network Monitor -
R (space) PC (space) 0000 (space) 4E35 <cr>
Local Monitor -
R (space) PC (space) 0000 (space) 588D <cr>
- h. Start Network then Local side of UNID: Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHB_TB
Network Monitor: Command - D 31E1 87 <cr>

Phase III Stage 4: Data transfer from Local Channel-4 (LC04TB) to each of the two outgoing network channels. There are two tests in this stage of testing.

1. LC04TB to OUTFRAME_CHA_TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 10 : Command - D 30C0 <cr> 10 <cr> <cr>
LC04TB[18] = 8F : Command - D 30C1 <cr> 8F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte datablock, set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. Prepare OUTFRAME_CHA_TB for reception, set on Network Monitor:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- f. In order not to eliminate the preset commands, set on Local Monitor :
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter for Network and Local side of UNID to start of Procedure MAIN.
Command : Network Monitor -
R (space) PC (space) 0000 (space) 4E35 <cr>
Local Monitor -
R (space) PC (space) 0000 (space) 588D <cr>
- h. Start Network then Local side of UNID: Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHA_TB
Network Monitor: Command - D 2C95 87 <cr>

2. LC04TB to OUTFRAME_CHB_TB:

Set-up using Local Monitor:

- a. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- b. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 30 : Command - D 30C0 <cr> 30 <cr> <cr>
LC04TB[18] = 8F : Command - D 30C1 <cr> 8F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C2 <cr> F0 <cr> <cr>
To examine : Command D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 3
Host_Code = 08 (Local Host 08)
Port_Code = FF0 (Not used in this test.)
- c. To indicate reception of 128-byte data block, set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- d. Prepare LCNTTB for reception, set:
LCNTNE = 00 : Command - D 85B6 <cr> 00 <cr> <cr>
LCNTNS = 00 : Command - D 85B4 <cr> 00 <cr> <cr>
- e. Prepare OUTFRAME_CHB_TB for reception, set on Network Monitor:
OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr><cr>
- f. In order not to eliminate the preset commands, set on Local Monitor:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter for Network and Local side of UNID to start of Procedure MAIN.
Command : Network Monitor -
R (space) PC (space) 0000 (space) 4E35 <cr>
Local Monitor -
R (space) PC (space) 0000 (space) 588D <cr>
- h. Start Network then Local side of UNID: Command - GO <cr>

Display Results:

A valid I-Frame should be located in OUTFRAME_CHB_TB
Network Monitor: Command - D 31E1 87 <cr>

Phase III Stage 5: Data transfer from Network Channel-A (NT01TB) to the four outgoing local channels. There are four tests in this stage of testing.

1. NT01TB to Local Channel-1:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT01TB, set:
All bytes = CC : Command - F 21FD 2284 CC <cr>
- b. Fill Destination Address bytes in NT01TB as follows:
NT01TB[0] = 21 : Command - D 21FD <cr> 21 <cr> <cr>
NT01TB[1] = 00 : Command - D 21FE <cr> 00 <cr> <cr>
NT01TB[2] = 21 : Command - D 21FF <cr> 21 <cr> <cr>
NT01TB[3] = 14 : Command - D 2200 <cr> 14 <cr> <cr>
NT01TB[4] = 00 : Command - D 2201 <cr> 00 <cr> <cr>
NT01TB[5] = 00 : Command - D 2202 <cr> 00 <cr> <cr>
NT01TB[6] = 00 : Command - D 2203 <cr> 00 <cr> <cr>
To examine : Command 21FD 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#1
Control = 00
Destination_Address = UNID#2 CH#1
Source_Address = UNID#1 CH#4
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT01NE = 87 : Command - D 2745 <cr> 87 <cr> <cr>
NT01NS = 00 : Command - D 2743 <cr> 00 <cr> <cr>
- d. To keep outgoing S-Frame in first 135-bytes of OUTFRAME_CHA_TB, set:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN:
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID on Network Side: Command - GO <cr>

Display Results:

- a. A valid 128-byte data block should exit Local Channel-1
Command - Viewed on Local Monitor.
- b. A valid S-Frame should be located in OUTFRAME_CHA_TB
Command - D 2C95 87 <cr>

2. NT01TB to Local Channel-2:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT01TB, set:
All bytes = CC : Command - F 21FD 2284 CC <cr>
- b. Fill Destination Address bytes in NT01TB as follows:
NT01TB[0] = 21 : Command - D 21FD <cr> 21 <cr> <cr>
NT01TB[1] = 00 : Command - D 21FE <cr> 00 <cr> <cr>
NT01TB[2] = 22 : Command - D 21FF <cr> 22 <cr> <cr>
NT01TB[3] = 13 : Command - D 2200 <cr> 13 <cr> <cr>
NT01TB[4] = 00 : Command - D 2201 <cr> 00 <cr> <cr>
NT01TB[5] = 00 : Command - D 2202 <cr> 00 <cr> <cr>
NT01TB[6] = 00 : Command - D 2203 <cr> 00 <cr> <cr>
To examine : Command 21FD 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#1
Control = 00
Destination_Address = UNID#2 CH#2
Source_Address = UNID#1 CH#3
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT01NE = 87 : Command - D 2745 <cr> 87 <cr> <cr>
NT01NS = 00 : Command - D 2743 <cr> 00 <cr> <cr>
- d. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHA_TB, set:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN:
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID on Network Side: Command - GO <cr>

Display Results:

- a. A valid 128-byte data block should exit Local Channel-2
Command - Viewed on Local Monitor.
- b. A valid S-Frame should be located in OUTFRAME_CHA_TB
Command - D 2C95 87 <cr>

3. NT01TB to Local Channel-3:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT01TB, set:
All bytes = CC : Command - F 21FD 2284 CC <cr>
- b. Fill Destination Address bytes in NT01TB as follows:
NT01TB[0] = 21 : Command - D 21FD <cr> 21 <cr> <cr>
NT01TB[1] = 00 : Command - D 21FE <cr> 00 <cr> <cr>
NT01TB[2] = 23 : Command - D 21FF <cr> 23 <cr> <cr>
NT01TB[3] = 11 : Command - D 2200 <cr> 11 <cr> <cr>
NT01TB[4] = 00 : Command - D 2201 <cr> 00 <cr> <cr>
NT01TB[5] = 00 : Command - D 2202 <cr> 00 <cr> <cr>
NT01TB[6] = 00 : Command - D 2203 <cr> 00 <cr> <cr>
To examine : Command 21FD 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#1
Control = 00
Destination_Address = UNID#2 CH#3
Source_Address = UNID#1 CH#1
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT01NE = 87 : Command - D 2745 <cr> 87 <cr> <cr>
NT01NS = 00 : Command - D 2743 <cr> 00 <cr> <cr>
- d. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHA_TB, set:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN:
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID on Network Side: Command - GO <cr>

Display Results:

- a. A valid 128-byte data block should exit Local Channel-3
Command - Viewed on Local Monitor.
- b. A valid S-Frame should be located in OUTFRAME_CHA_TB
Command - D 2C95 87 <cr>

4. NT01TB to Local Channel-4:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT01TB, set:
All bytes = CC : Command - F 21FD 2284 CC <cr>
- b. Fill Destination Address bytes in NT01TB as follows:
NT01TB[0] = 21 : Command - D 21FD <cr> 21 <cr> <cr>
NT01TB[1] = 00 : Command - D 21FE <cr> 00 <cr> <cr>
NT01TB[2] = 24 : Command - D 21FF <cr> 24 <cr> <cr>
NT01TB[3] = 12 : Command - D 2200 <cr> 12 <cr> <cr>
NT01TB[4] = 00 : Command - D 2201 <cr> 00 <cr> <cr>
NT01TB[5] = 00 : Command - D 2202 <cr> 00 <cr> <cr>
NT01TB[6] = 00 : Command - D 2203 <cr> 00 <cr> <cr>
To examine : Command 21FD 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#1
Control = 00
Destination_Address = UNID#2 CH#4
Source_Address = UNID#1 CH#2
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT01NE = 87 : Command - D 2745 <cr> 87 <cr> <cr>
NT01NS = 00 : Command - D 2743 <cr> 00 <cr> <cr>
- d. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHA_TB, set:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN:
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID on Network Side: Command - GO <cr>

Display Results:

- a. A valid 128-byte data block should exit Local Channel-4
Command - Viewed on Local Monitor.
- b. A valid S-Frame should be located in OUTFRAME_CHA_TB
Command - D 2C95 87 <cr>

Phase III Stage 6: Data transfer from Network Channel-B (NT02TB) to the four outgoing local channels. There are four tests in this stage of testing.

1. NT02TB to Local Channel-1:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT02TB, set:
All bytes = CC : Command - F 2749 27D0 CC <cr>
- b. Fill Destination Address bytes in NT02TB as follows:
NT02TB[0] = 23 : Command - D 2749 <cr> 23 <cr> <cr>
NT02TB[1] = 00 : Command - D 274A <cr> 00 <cr> <cr>
NT02TB[2] = 21 : Command - D 274B <cr> 21 <cr> <cr>
NT02TB[3] = 34 : Command - D 274C <cr> 34 <cr> <cr>
NT02TB[4] = 00 : Command - D 274D <cr> 00 <cr> <cr>
NT02TB[5] = 00 : Command - D 274E <cr> 00 <cr> <cr>
NT02TB[6] = 00 : Command - D 274F <cr> 00 <cr> <cr>
To examine : Command 2749 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#3
Control = 00
Destination_Address = UNID#2 CH#1
Source_Address = UNID#3 CH#4
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT02NE = 87 : Command - D 2C91 <cr> 87 <cr> <cr>
NT02NS = 00 : Command - D 2C8F <cr> 00 <cr> <cr>
- d. To keep outgoing S-Frame in first 135-bytes of OUTFRAME_CHB_TB, set:
OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN:
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID on Network Side: Command - GO <cr>

Display Results:

- a. A valid 128-byte data block should exit Local Channel-1
Command - Viewed on Local Monitor
- b. A valid S-Frame should be located in OUTFRAME_CHB_TB
Command - D 31E1 87 <cr>

2. NT02TB to Local Channel-2:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT02TB, set:
All bytes = CC : Command - F 2749 27D0 CC <cr>
- b. Fill Destination Address bytes in NT02TB as follows:
NT02TB[0] = 23 : Command - D 2749 <cr> 23 <cr> <cr>
NT02TB[1] = 00 : Command - D 274A <cr> 00 <cr> <cr>
NT02TB[2] = 22 : Command - D 274B <cr> 22 <cr> <cr>
NT02TB[3] = 33 : Command - D 274C <cr> 33 <cr> <cr>
NT02TB[4] = 00 : Command - D 274D <cr> 00 <cr> <cr>
NT02TB[5] = 00 : Command - D 274E <cr> 00 <cr> <cr>
NT02TB[6] = 00 : Command - D 274F <cr> 00 <cr> <cr>
To examine : Command 2749 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#3
Control = 00
Destination_Address = UNID#2 CH#2
Source_Address = UNID#3 CH#3
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT02NE = 87 : Command - D 2C91 <cr> 87 <cr> <cr>
NT02NS = 00 : Command - D 2C8F <cr> 00 <cr> <cr>
- d. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHB_TB, set:
OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN:
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID on Network Side: Command - GO <cr>

Display Results:

- a. A valid 128-byte data block should exit Local Channel-2
Command - Viewed on Local Monitor.
- b. A valid S-Frame should be located in OUTFRAME_CHB_TB
Command - D 31E1 87 <cr>

3. NT02TB to Local Channel-3:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT02TB, set:
All bytes = CC : Command - F 2749 27D0 CC <cr>
- b. Fill Destination Address bytes in NT02TB as follows:
NT02TB[0] = 23 : Command - D 2749 <cr> 23 <cr> <cr>
NT02TB[1] = 00 : Command - D 274A <cr> 00 <cr> <cr>
NT02TB[2] = 23 : Command - D 274B <cr> 23 <cr> <cr>
NT02TB[3] = 31 : Command - D 274C <cr> 31 <cr> <cr>
NT02TB[4] = 00 : Command - D 274D <cr> 00 <cr> <cr>
NT02TB[5] = 00 : Command - D 274E <cr> 00 <cr> <cr>
NT02TB[6] = 00 : Command - D 274F <cr> 00 <cr> <cr>
To examine : Command 2749 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#3
Control = 00
Destination_Address = UNID#2 CH#3
Source_Address = UNID#3 CH#1
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT02NE = 87 : Command - D 2C91 <cr> 87 <cr> <cr>
NT02NS = 00 : Command - D 2C8F <cr> 00 <cr> <cr>
- d. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHB_TB, set:
OUTFRAME_CHB_NE = 00: Command - D 371A <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3718 <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN:
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID on Network Side: Command - GO <cr>

Display Results:

- a. A valid 128-byte data block should exit Local Channel-3
Command - Viewed on Local Monitor.
- b. A valid S-Frame should be located in OUTFRAME_CHB_TB
Command - D 31E1 87 <cr>

4. NT02TB to Local Channel-4:

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into NT02TB, set:
All bytes = CC : Command - F 2749 27D0 CC <cr>
- b. Fill Destination Address bytes in NT02TB as follows:
NT02TB[0] = 23 : Command - D 2749 <cr> 23 <cr> <cr>
NT02TB[1] = 00 : Command - D 274A <cr> 00 <cr> <cr>
NT02TB[2] = 24 : Command - D 274B <cr> 24 <cr> <cr>
NT02TB[3] = 31 : Command - D 274C <cr> 31 <cr> <cr>
NT02TB[4] = 00 : Command - D 274D <cr> 00 <cr> <cr>
NT02TB[5] = 00 : Command - D 274E <cr> 00 <cr> <cr>
NT02TB[6] = 00 : Command - D 274F <cr> 00 <cr> <cr>
To examine : Command 2749 87 <cr>
This will set the I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#3
Control = 00
Destination_Address = UNID#2 CH#4
Source_Address = UNID#3 CH#1
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
NT02NE = 87 : Command - D 2C91 <cr> 87 <cr> <cr>
NT02NS = 00 : Command - D 2C8F <cr> 00 <cr> <cr>
- d. To keep outgoing S-Frame in first 135-bytes of
OUTFRAME_CHB_TB, set:
OUTFRAME_CHB_NE = 00: Command - D 371A <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3718 <cr> 00 <cr><cr>
- e. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- f. Set Program Counter to start of Procedure MAIN:
Command - R (space) PC (space) 0000 (space) 4E35 <cr>
- g. Start UNID on Network Side: Command - GO <cr>

Display Results:

- a. A valid 128-byte data block should exit Local Channel-4
Command - Viewed on Local Monitor.
- b. A valid S-Frame should be located in OUTFRAME_CHB_TB
Command - D 31E1 87 <cr>

Phase IV Stage 1: Data transfer from Local Channel-1 to each of the three remaining local channels via loopback cable. There are three tests in this stage of testing.

1. LC01TB to LC02TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels One and Two.
- b. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- c. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 22 : Command - D 21AE <cr> 22 <cr> <cr>
LC01TB[18] = 0F : Command - D 21AF <cr> 0F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 20 (Local Host 32)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC02TB:
Command - D 26A3 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

2. LC01TB to LC03TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels One and Three.
- b. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- c. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 22 : Command - D 21AE <cr> 22 <cr> <cr>
LC01TB[18] = 0F : Command - D 21AF <cr> 0F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 20 (Local Host 32)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC03TB:
Command - D 2BA9 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

3. LC01TB to LC04TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels One and Four.
- b. Place 128-bytes of data (HEX=80) into LC01TB, set:
All bytes = CC : Command - F 219D 221D CC <cr>
- c. Fill Destination Address bytes in LC01TB as follows:
LC01TB[16] = 01 : Command - D 21AD <cr> 01 <cr> <cr>
LC01TB[17] = 22 : Command - D 21AE <cr> 22 <cr> <cr>
LC01TB[18] = 0F : Command - D 21AF <cr> 0F <cr> <cr>
LC01TB[19] = F0 : Command - D 21B0 <cr> F0 <cr> <cr>
To examine : Command - D 219D 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 20 (Local Host 32)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC01NE = 80 : Command - D 269F <cr> 80 <cr> <cr>
LC01NS = 00 : Command - D 269D <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC04TB.
Command - D 30AF 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

Phase IV Stage 2: Data transfer from Local Channel-2 to each of the three remaining local channels via loopback cable. There are three tests in this stage of testing.

1. LC02TB to LC01TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Two and One.
- b. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- c. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 24 : Command - D 26B4 <cr> 24 <cr> <cr>
LC02TB[18] = 3F : Command - D 26B5 <cr> 3F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 43 (Local Host 67)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC01TB.
Command - D 219D 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

2. LC02TB to LC03TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Two and Three
- b. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- c. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 24 : Command - D 26B4 <cr> 24 <cr> <cr>
LC02TB[18] = 3F : Command - D 26B5 <cr> 3F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 43 (Local Host 67)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC01TB.
Command - D 219D 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

3. LC02TB to LC04TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Two and Four.
- b. Place 128-bytes of data (HEX=80) into LC02TB, set:
All bytes = CC : Command - F 26A3 2723 CC <cr>
- c. Fill Destination Address bytes in LC02TB as follows:
LC02TB[16] = 01 : Command - D 26B3 <cr> 01 <cr> <cr>
LC02TB[17] = 24 : Command - D 26B4 <cr> 24 <cr> <cr>
LC02TB[18] = 3F : Command - D 26B5 <cr> 3F <cr> <cr>
LC02TB[19] = F0 : Command - D 26B6 <cr> F0 <cr> <cr>
To examine : Command - D 26A3 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 43 (Local Host 67)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC02NE = 80 : Command - D 2BA5 <cr> 80 <cr> <cr>
LC02NS = 00 : Command - D 2BA3 <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC04TB.
Command - D 30AF 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

Phase IV Stage 3: Data transfer from Local Channel-3 to each of the remaining three local channels via loopback cable. There are three tests in this stage of testing.

1. LC03TB to LC01TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Three and One.
- b. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- c. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 2A : Command - D 2BBA <cr> 2A <cr> <cr>
LC03TB[18] = 5F : Command - D 2BBB <cr> 5F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = A5 (Local Host 165)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC01TB.
Command - D 219D 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

2. LC03TB to LC02TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Three and Two.
- b. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- c. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 2A : Command - D 2BBA <cr> 2A <cr> <cr>
LC03TB[18] = 5F : Command - D 2BBB <cr> 5F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = A5 (Local Host 165)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC02TB
Command - D 26A3 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

3. LC03TB to LC04TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Three and Four.
- b. Place 128-bytes of data (HEX=80) into LC03TB, set:
All bytes = CC : Command - F 2BA9 2C29 CC <cr>
- c. Fill Destination Address bytes in LC03TB as follows:
LC03TB[16] = 01 : Command - D 2BB9 <cr> 01 <cr> <cr>
LC03TB[17] = 2A : Command - D 2BBA <cr> 2A <cr> <cr>
LC03TB[18] = 5F : Command - D 2BBB <cr> 5F <cr> <cr>
LC03TB[19] = F0 : Command - D 2BBC <cr> F0 <cr> <cr>
To examine : Command - D 2BA9 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = A5 (Local Host 165)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC03NE = 80 : Command - D 30AB <cr> 80 <cr> <cr>
LC03NS = 00 : Command - D 30A9 <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC04TB.
Command - D 30AF 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

Phase IV Stage 4: Data transfer from Local Channel-4 to each of the three remaining local channels via loopback cable. There are three tests in this stage of testing.

1. LC04TB to LC01TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Four and One.
- b. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- c. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 2C : Command - D 30C1 <cr> 2C <cr> <cr>
LC04TB[18] = 3F : Command - D 30C2 <cr> 3F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C3 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = C3 (Local Host 195)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC01TB.
Command - D 21A5 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

2. LC04TB to LC02TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Four and Two.
- b. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- c. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 2C : Command - D 30C1 <cr> 2C <cr> <cr>
LC04TB[18] = 3F : Command - D 30C2 <cr> 3F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C3 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = C3 (Local Host 195)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC02TB.
Command - D 26A3 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

3. LC04TB to LC03TB:

Set-up using Local Monitor:

- a. Connect loopback cable to Local Channels Four and Three.
- b. Place 128-bytes of data (HEX=80) into LC04TB, set:
All bytes = CC : Command - F 30AF 312F CC <cr>
- c. Fill Destination Address bytes in LC04TB as follows:
LC04TB[16] = 01 : Command - D 30BF <cr> 01 <cr> <cr>
LC04TB[17] = 2C : Command - D 30C1 <cr> 2C <cr> <cr>
LC04TB[18] = 3F : Command - D 30C2 <cr> 3F <cr> <cr>
LC04TB[19] = F0 : Command - D 30C3 <cr> F0 <cr> <cr>
To examine : Command - D 30AF 80 <cr>
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = C3 (Local Host 195)
Port_Code = FF0 (Not used in this test.)
- d. To indicate reception of 128-byte data block, set:
LC04NE = 80 : Command - D 35B1 <cr> 80 <cr> <cr>
LC04NS = 00 : Command - D 35AF <cr> 00 <cr> <cr>
- e. Prepare LCLCTB for reception, set:
LCLCNE = 00 : Command - D 3AB7 <cr> 00 <cr> <cr>
LCLCNS = 00 : Command - D 3AB5 <cr> 00 <cr> <cr>
- f. In order not to eliminate the preset commands, set:
INIT_L_TAB : Command - D 3ABB <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 588D <cr>
- h. Start UNID : Command - GO <cr>

Display Results:

- a. The 128-byte data block should be present in LC03TB.
Command - D 2BA9 80 <cr>
- b. A data packet should be present in LCLCTB.
Command - D 35B5 80 <cr>

Phase V Stage 1: Local-to-Local data transfer from the Zilog MCZ 1/25 Computer System connected to UNID Channel-1 back to the Zilog thru UNID Channel-1. There is only one test in this stage of testing.

TEST - To UNID Local Channel-1 destined for Local Channel-1:

Set-up:

- a. Start-up UNID#2 (Procedures can be found in Appendix A).
- b. Fill Destination Address bytes in Zilog_Application program as follows:
Destination_Address_One = 01
Destination_Address_Two = 21
Destination_Address_Three = 3D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 13 (Local Host 19)
Port_Code = DE0 (H19 Terminal)
- c. Recompile and Link Zilog_Application program. Link must start at location 5000.
- d. Rename Zilog_Application to Zilog_Host and rename Zilog_Application.Map to Zilog_Host.Map.
- e. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_1.
- f. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_2.

Display Results:

- a. Test_Msg_1 should appear on H19 Terminal.
- b. Test_Msg_2 should appear on H19 Terminal.

Phase V Stage 2: Local-to-Local data transfer from the Zilog MCZ 1/25 Computer System connected to UNID Channel-2 back to the Zilog thru UNID Channel-2. There is only one test in this stage of testing.

TEST - To UNID Local Channel-2 destined for Local Channel-2:

Set-up:

- a. Start-up UNID#2 (Procedures can be found in Appendix A).
- b. Fill Destination Address bytes in Zilog_Application program as follows:
Destination_Address_One = 01
Destination_Address_Two = 24
Destination_Address_Three = 6D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = 46 (Local Host 70)
Port_Code = DE0 (H19 Terminal)
- c. Recompile and Link Zilog_Application program. Link must start at location 5000.
- d. Rename Zilog_Application to Zilog_Host and rename Zilog_Application.Map to Zilog_Host.Map.
- e. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_1.
- f. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_2.

Display Results:

- a. Test_Msg_1 should appear on H19 Terminal.
- b. Test_Msg_2 should appear on H19 Terminal.

Phase V Stage 3: Local-to-Local data transfer from the Zilog MCZ 1/25 Computer System connected to UNID Channel-3 back to the Zilog thru UNID Channel-3. There is only one test in this stage of testing.

TEST - To UNID Local Channel-3 destined for Local Channel-3:

Set-up:

- a. Start-up UNID#2 (Procedures can be found in Appendix A).
- b. Fill Destination Address bytes in Zilog_Application program as follows:
Destination_Address_One = 01
Destination_Address_Two = 2A
Destination_Address_Three = AD
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = AA (Local Host 170)
Port_Code = DE0 (H19 Terminal)
- c. Recompile and Link Zilog_Application program. Link must start at location 5000.
- d. Rename Zilog_Application to Zilog_Host and rename Zilog_Application.Map to Zilog_Host.Map.
- e. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_1.
- f. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_2.

Display Results:

- a. Test_Msg_1 should appear on H19 Terminal.
- b. Test_Msg_2 should appear on H19 Terminal.

Phase V Stage 4: Local-to-Local data transfer from the Zilog MCZ 1/25 Computer System connected to UNID Channel-4 back to the Zilog thru UNID Channel-4. There is only one test in this stage of testing.

TEST - To UNID Local Channel-4 destined for Local Channel-4:

Set-up:

- a. Start-up UNID#2 (Procedures can be found in Appendix A).
- b. Fill Destination Address bytes in Zilog_Application program as follows:
Destination_Address_One = 01
Destination_Address_Two = 2C
Destination_Address_Three = 8D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2
Host_Code = C8 (Local Host 200)
Port_Code = DE0 (H19 Terminal)
- c. Recompile and Link Zilog_Application program. Link must start at location 5000.
- d. Rename Zilog_Application to Zilog_Host and rename Zilog_Application.Map to Zilog_Host.Map.
- e. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_1.
- f. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_2.

Display Results:

- a. Test_Msg_1 should appear on H19 Terminal.
- b. Test_Msg_2 should appear on H19 Terminal.

Phase VI Stage 1: External Network-to-Network, via loopback, data transfer. There is only one test within this phase of testing.

TEST - OUTFRAME_CHA_TB to NT02TB (Channel-A to Channel-B)

Set-up using Network Monitor:

a. Place 135-bytes of data (HEX=87) into OUTFRAME_CHA_TB,
set: All bytes = CC : Command - F 2C95 2D1C CC <cr>

b. Fill Destination Address bytes in OUTFRAME_CHA_TB as follows:

OUTFRAME_CHA_TB[0]=23: Command - D 2C95 <cr> 23 <cr><cr>
OUTFRAME_CHA_TB[1]=00: Command - D 2C96 <cr> 00 <cr><cr>
OUTFRAME_CHA_TB[2]=21: Command - D 2C97 <cr> 21 <cr><cr>
OUTFRAME_CHA_TB[3]=34: Command - D 2C98 <cr> 34 <cr><cr>
OUTFRAME_CHA_TB[4]=00: Command - D 2C99 <cr> 00 <cr><cr>
OUTFRAME_CHA_TB[5]=00: Command - D 2C9A <cr> 00 <cr><cr>
OUTFRAME_CHA_TB[6]=00: Command - D 2C9B <cr> 00 <cr><cr>

To examine : Command D 2C95 87 <cr>

This will set the outgoing I-Frame Header to:

UNID Address = To-UNID#2 From-UNID#3
Control = 00
Destination_Address = UNID#2 CH#1
Source_Address = UNID#3 CH#4
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)

c. To indicate reception of 135-byte I-Frame, set:

OUTFRAME_CHA_NE = 87 : Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00 : Command - D 31DB <cr> 00 <cr><cr>

d. Prepare NT02TB for reception, set:

NT02NE = 00 : Command - D 2C91 <cr> 00 <cr> <cr>
NT02NS = 00 : Command - D 2C8F <cr> 00 <cr> <cr>

e. To keep S-Frame in first 135-bytes of OUTFRAME_CHB_TB,
set:

OUTFRAME_CHB_NE = 00: Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr><cr>

f. In order not to eliminate the preset commands, set:

INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>

g. Set Program Counter to start of Procedure MAIN.

Command - R (space) PC (space) 0000 (space) 4E35 <cr>

Display Results:

a. A valid I-Frame should be located in NT02TB

Command - D 2749 87 <cr>

b. A valid S-Frame should be located in OUTFRAME_CHA_TB

Command - D 2C95 87 <cr>

Phase VI Stage 2: External Network-to-Network, via loopback, data transfer. There is only one test within this phase of testing.

TEST - OUTFRAME_CHB_TB to NT01TB (Channel-B to Channel-A)

Set-up using Network Monitor:

- a. Place 135-bytes of data (HEX=87) into OUTFRAME_CHB_TB,
set: All bytes = CC : Command - F 31E1 3268 CC <cr>
- b. Fill Destination Address bytes in OUTFRAME_CHA_TB as follows:
OUTFRAME_CHB_TB[0]=21: Command - D 31E1 <cr> 21 <cr><cr>
OUTFRAME_CHB_TB[1]=00: Command - D 31E2 <cr> 00 <cr><cr>
OUTFRAME_CHB_TB[2]=21: Command - D 31E3 <cr> 21 <cr><cr>
OUTFRAME_CHB_TB[3]=34: Command - D 31E4 <cr> 34 <cr><cr>
OUTFRAME_CHB_TB[4]=00: Command - D 31E5 <cr> 00 <cr><cr>
OUTFRAME_CHB_TB[5]=00: Command - D 31E6 <cr> 00 <cr><cr>
OUTFRAME_CHB_TB[6]=00: Command - D 31E7 <cr> 00 <cr><cr>
To examine : Command D 31E1 87 <cr>
This will set the outgoing I-Frame Header to:
UNID Address = To-UNID#2 From-UNID#1
Control = 00
Destination_Address = UNID#2 CH#1
Source_Address = UNID#3 CH#4
Sequence_Number = 00 (Not implemented in software)
Spare_01 = 00 (Not implemented in software)
Spare_02 = 00 (Not implemented in software)
- c. To indicate reception of 135-byte I-Frame, set:
OUTFRAME_CHB_NE = 87: Command - D 3729 <cr> 00 <cr><cr>
OUTFRAME_CHB_NS = 00: Command - D 3727 <cr> 00 <cr><cr>
- d. Prepare NT01TB for reception, set:
NT01NE = 00 : Command - D 2745 <cr> 00 <cr> <cr>
NT01NS = 00 : Command - D 2743 <cr> 00 <cr> <cr>
- e. To keep S-Frame in first 135-bytes of OUTFRAME_CHA_TB,
set:
OUTFRAME_CHA_NE = 00: Command - D 31DD <cr> 00 <cr><cr>
OUTFRAME_CHA_NS = 00: Command - D 31DB <cr> 00 <cr><cr>
- f. In order not to eliminate the preset commands, set:
INIT_N_TAB : Command - D 372D <cr> C9 <cr> <cr>
INIT_U_SHTAB : Command - D 8B06 <cr> C9 <cr> <cr>
- g. Set Program Counter to start of Procedure MAIN.
Command - R (space) PC (space) 0000 (space) 4E35 <cr>

Display Results:

- a. A valid I-Frame should be located in NT01TB
Command - D 21FD 87 <cr>
- b. A valid S-Frame should be located in OUTFRAME_CHA_TB
Command - D 2C95 87 <cr>

AD-A138 481

PROTOCOL STANDARDS AND IMPLEMENTATION WITHIN THE
DIGITAL ENGINEERING LABO..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P W PHISTER
DEC 83 AFIT/GE/EE/83D-58-PT-1 F/G 17/2

4/4

UNCLASSIFIED

NL



END
DATE
FILMED
3694
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Phase VII Stage 1: Host-to-Host data transfer via Two-UNID Network. There is only one test in this stage of testing.

TEST - From Zilog MCZ 1/25 (UNID#2) to INTEL/432 (UNID#1)

Set-up:

- a. Start-up UNID#1 (Procedures can be found in Appendix A).
- b. Start-up UNID#2 (Procedures can be found in Appendix A).
- c. Fill Destination Address bytes in Zilog_Application program as follows:
Destination_Address_One = 01
Destination_Address_Two = 11
Destination_Address_Three = 3D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1 (Destination UNID = 1)
Host_Code = 13 (INTEL/432 will act as Host 19)
Port_Code = DE (H19 Terminal)
- d. Recompile and Link Zilog_Application program. Link must start at location 5000.
- e. Rename Zilog_Application to Zilog_Host and rename Zilog_Application.Map to Zilog_Host.Map.
- f. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for rest_Msg_1.
- g. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_2.

Display Results:

- a. Test_Msg_1 should appear on INTEL/432 H19 Terminal.
- b. Test_Msg_2 should appear on INTEL/432 H19 Terminal.

Phase VII Stage 2: Host-to-Host data transfer via Two-UNID Network. There is only one test in this stage of testing.

TEST - From INTEL/432 (UNID#1) to Zilog MCZ 1/25 (UNID#2)

Set-up:

- a. Start-up UNID#1 (Procedures can be found in Appendix A).
- b. Start-up UNID#2 (Procedures can be found in Appendix A).
- c. Fill Destination Address bytes in INTEL/432 as follows:
Destination_Address_One = 01
Destination_Address_Two = 21
Destination_Address_Three = 3D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2 (Destination UNID = 1)
Host_Code = 13 (Zilog MCZ will act as Host 19)
Port_Code = DE (H19 Terminal)
- d. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System.

Display Results:

- a. Test_Msg_1 should appear on Zilog MCZ H19 Terminal.
- b. Test_Msg_2 should appear on Zilog MCZ H19 Terminal.

Phase VIII Stage 1: Host-to-Host data transfer via Two-UNID Network using the 18-pair cable. There is only one test in this stage of testing.

TEST - From Zilog MCZ 1/25 (UNID#2) to INTEL/432 (UNID#1)

Set-up:

- a. Start-up UNID#1 (Procedures can be found in Appendix A).
- b. Start-up UNID#2 (Procedures can be found in Appendix A).
- c. Connect 18-pair cable between UNID#1 and UNID#2.
- d. Fill Destination Address bytes in Zilog_Application program as follows:
Destination_Address_One = 01
Destination_Address_Two = 11
Destination_Address_Three = 3D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1 (Destination UNID = 1)
Host_Code = 13 (INTEL/432 will act as Host 19)
Port_Code = DE (H19 Terminal)
- e. Recompile and Link Zilog_Application program. Link must start at location 5000.
- f. Rename Zilog_Application to Zilog_Host and rename Zilog_Application.Map to Zilog_Host.Map.
- g. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_1.
- h. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_2.

Display Results:

- a. Test_Msg_1 should appear on INTEL/432 H19 Terminal.
- b. Test_Msg_2 should appear on INTEL/432 H19 terminal.

Phase VIII Stage 2: Host-to-Host data transfer via Two-UNID Network using the 18-pair cable. There is only one test in this stage of testing.

TEST - From INTEL/432 (UNID#1) to Zilog MCZ 1/25 (UNID#2)

Set-up:

- a. Start-up UNID#1 (Procedures can be found in Appendix A).
- b. Start-up UNID#2 (Procedures can be found in Appendix A).
- c. Connect 18-pair cable between UNID#1 and UNID#2.
- d. Fill Destination Address bytes in INTEL/432 as follows:
Destination_Address_One = 01
Destination_Address_Two = 21
Destination_Address_Three = 3D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2 (Destination UNID = 1)
Host_Code = 13 (Zilog MCZ will act as Host 19)
Port_Code = DE (H19 Terminal connected to Zilog MCZ)
- e. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System.

Display Results:

- a. Test_Msg_1 should appear on Zilog MCZ H19 Terminal.
- b. Test_Msg_2 should appear on Zilog MCZ H19 Terminal.

Phase IX Stage 1: Host-to-Host data transfer via Three-UNID Network. There is only one test in this stage of testing.

TEST - From Zilog MCZ 1/25 (UNID#2) to INTEL/432 (UNID#1)

Set-up:

- a. Start-up UNID#1 (Procedures can be found in Appendix A).
- b. Start-up UNID#2 (Procedures can be found in Appendix A).
- c. Start-up UNID#3 (Procedures can be found in Appendix A).
- d. Fill Destination Address bytes in Zilog_Application program as follows:
Destination_Address_One = 01
Destination_Address_Two = 11
Destination_Address_Three = 3D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 1 (Destination UNID = 1)
Host_Code = 13 (INTEL/432 will act as Host 19)
Port_Code = DE (H19 Terminal)
- e. Recompile and Link Zilog_Application program. Link must start at location 5000.
- f. Rename Zilog_Application to Zilog_Host and rename Zilog_Application.Map to Zilog_Host.Map.
- g. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_1.
- h. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System for Test_Msg_2.

Display Results:

- a. Test_Msg_1 should appear on INTEL/432 H19 Terminal.
- b. Test_Msg_2 should appear on INTEL/432 H19 Terminal.

Phase IX Stage 2: Host-to-Host data transfer via Three-UNID Network. There is only one test in this stage of testing.

TEST - From INTEL/432 (UNID#1) to Zilog MCZ 1/25 (UNID#2)

Set-up:

- a. Start-up UNID#1 (Procedures can be found in Appendix A).
- b. Start-up UNID#2 (Procedures can be found in Appendix A).
- c. Start-up UNID#3 (Procedures can be found in Appendix A).
- d. Fill Destination Address bytes in INTEL/432 as follows:
Destination_Address_One = 01
Destination_Address_Two = 21
Destination_Address_Three = 3D
Destination_Address_Four = E0
This will set the destination address as:
Control_Code = 0
Country_Code = 1
Network_Code = 2 (Destination UNID = 1)
Host_Code = 13 (Zilog MCZ will act as Host 19)
Port_Code = DE (H19 Terminal)
- e. Execute Zilog_Host program on Zilog MCZ 1/25 Computer System.

Display Results:

- a. Test_Msg_1 should appear on Zilog MCZ H19 Terminal.
- b. Test_Msg_2 should appear on Zilog MCZ H19 Terminal.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM																				
1. REPORT NUMBER 6778 AFIT/EE/83-58	2. GOVT ACCESSION NO.	3. REPORT'S CATALOG NUMBER																				
4. TITLE (and Subtitle) PROTOCOL STANDARDS AND IMPLEMENTATION WITHIN THE DIGITAL ENGINEERING LABORATORY COMPUTER NETWORK (DELNET) USING THE UNIVERSAL NETWORK INTERFACE DEVICE (UNID)		5. TYPE OF REPORT & PERIOD COVERED MASTER OF SCIENCE THESIS JUNE 1982 - DECEMBER 1983																				
		6. PERFORMING ORG. REPORT NUMBER																				
7. AUTHOR(s) PAUL W. PHISTER, JR.		8. CONTRACT OR GRANT NUMBER(s)																				
9. PERFORMING ORGANIZATION NAME AND ADDRESS AIR FORCE INSTITUTE OF TECHNOLOGY (AFIT) ELECTRICAL ENGINEERING DEPARTMENT (EM) WRIGHT-PATTERSON AFB, OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS																				
11. CONTROLLING OFFICE NAME AND ADDRESS AIR FORCE INSTITUTE OF TECHNOLOGY (AFIT) ELECTRICAL ENGINEERING DEPARTMENT (EM) WRIGHT-PATTERSON AFB, OH 45433		12. REPORT DATE DECEMBER 1983																				
		13. NUMBER OF PAGES 621																				
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED																				
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE																				
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED																						
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)																						
18. SUPPLEMENTARY NOTES Approved for Release by NSA on 07-12-2017 <i>[Signature]</i> 10 FEB 84 Dean for Research and Technological Development Air Force Institute of Technology (AFIT) Wright-Patterson AFB OH 45433																						
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <table border="0" style="width: 100%;"><tr><td>APPLICATION LAYER</td><td>DELNET</td><td>PRESENTATION LAYER</td><td>RS-449</td><td>1.21</td></tr><tr><td>COMPUTER NETWORKS</td><td>ISO REFERENCE MODEL</td><td>ROUTING</td><td>SESSION LAYER</td><td>1.25</td></tr><tr><td>DATAGRAM</td><td>NETWORK LAYER</td><td>RS-232</td><td>UNID</td><td></td></tr><tr><td>BATA LINK LAYER</td><td>PHYSICAL LAYER</td><td>RS-422/423</td><td>VIRTUAL CIRCUIT</td><td></td></tr></table>			APPLICATION LAYER	DELNET	PRESENTATION LAYER	RS-449	1.21	COMPUTER NETWORKS	ISO REFERENCE MODEL	ROUTING	SESSION LAYER	1.25	DATAGRAM	NETWORK LAYER	RS-232	UNID		BATA LINK LAYER	PHYSICAL LAYER	RS-422/423	VIRTUAL CIRCUIT	
APPLICATION LAYER	DELNET	PRESENTATION LAYER	RS-449	1.21																		
COMPUTER NETWORKS	ISO REFERENCE MODEL	ROUTING	SESSION LAYER	1.25																		
DATAGRAM	NETWORK LAYER	RS-232	UNID																			
BATA LINK LAYER	PHYSICAL LAYER	RS-422/423	VIRTUAL CIRCUIT																			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) See Reverse																						

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

(Block 20 Continued)

Development of the Air Force Institute of Technology's Digital Engineering Laboratory Network (DELNET) was continued with the development of an initial draft of a protocol standard for all seven layers as specified by the International Standards Organization's (ISO) Reference Model for Open Systems Interconnections. This effort centered on the restructuring of the Network Layer to perform Datagram routing and to conform to the developed protocol standards and actual software module development of the upper four protocol layers residing within the DELNET Monitor (Zilog MCZ 1/25 Computer System). Within the guidelines of the ISO Reference Model the Transport Layer was developed utilizing the Internet Header Format (IHF) combined with the Transport Control Protocol (TCP) to create a 128-byte Datagram. Also a limited Application Layer was created to pass the Gettysburg Address through the DELNET. This study formulated a first draft for the DELNET Protocol Standard and designed, implemented, and tested the Network, Transport, and Application Layers to conform to these protocol standards.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

**DAT
FILM**