Table of Contents

Chapter 1

INTRODUCTION

1.1 DOCUMENT DEFINITION

This document is a specification of the architecture, functional characteristics, and services provided by the software required to support the Local Area Network Controller Subsystem (LACS).

1.2 OVERVIEW

The LACS is an integrated Network Controller that is capable of connecting a DPS 6 system to different types of Local Area Networks (LANs). It is a fully buffered micro-processor based programmable communications subsystem that connects to the Level 6 Megabus and comprises of the following components:

- Local Area Controller (LAC) Motherboard

- Adapters (upto four)

- Trunk Couplers (TCs)

- RF Modems or tranceivers

Different types of adapters, trunk couplers, and RF Modems will be provided to support different types of Media Access Control (MAC) and Physical Layer functionalities. The types of MAC and physical layer support that is planned for the LACS are:

- IEEE 802 Baseband CSMA/CD
- IEEE 802 Broadband Token Bus (second release)
- IEEE 802 Token Ring (second release)

The LACS is also suitable for supporting adapters for connections to PBXs (DMI/ISDN) and RS232 HDLC.

The LAC motherboard will contain software to support the following ISO layer protocols:

- Transport Layer. ISO class 4 transport services will be provided (second release).

- Network Layer. The LACS will provide Connectionless Network Services (second release). The ISO "Protocol for Providing Connectionless-mode Network Service" (ISO 8473), commonly known as the Internet Protocol will be used to provide connectionless network services. Both the "Inactive Network Layer Protocol" subset (NULL Network subset) and the "non-segmenting protocol" subset will be supported.

- Sub-network Layer. Since the LACS only provides connectionless link services (LLC Type 1), connection oriented subnetwork services will be provided for users requiring reliable "Link Connection" services. ISO Class 4 Transport Protocols will be used for this purpose.

- Data Link Layer. IEEE 802 Data Link protocols will be supported. Type 1 (Connectionless) Logical Link Control (LLC) services will be provided as defined in IEEE 802.2. The MAC sub-layer of the Data Link Layer will be implemented on daughter (adapter) boards as described above.

In order to support the LACS, a software driver will be written for the DPS 6 to provide the interface to the different layers. In addition to this, System Management facilities will be provided in the DPS 6 and LACS to help control and administer the local LACS as well as the entire network.

1.3 SCOPE

This specification describes the software components that support the LACS and the interfaces to these components. Services provided by these software components can be divided into two major categories: Data Transfer Services and LACS System Management Services. Software components, therefore, include:

- Driver logic in the DPS 6 that provides the interface to allow software in the DPS 6 access any of the layers in the LACS.

- System Management and layer management logic in the DPS 6.

- LLC, Network, Subnetwork, and Transport logic in the LAC.

- System Management and layer management logic in the LAC.

This specification describes the above software at a higher architectural and functional level. Details of the design are provided in the corresponding component specifications.

This specification describes software that has been planned for implementation that is phased out over two releases. The first release supports primary (DPS-6 to DPS-6 or DN-8) traffic only. The second release will allow the DPS-6 to also support secondary network (terminal) traffic. The term "first" or "initial" and "second" or "next" release will be used to specify which release a particular functionality will be supported. The term "future" will be used for extensions which are possible to be made but not planned for or committed. Mention of releases is done mainly to give a feel for the evolution of the product and should not be taken as a definitive statement. Refer to the LACS Product Functional Specification for a definitive statement of what functionality is planned for which release.

This specification only describes DPS 6 software components that operate under the MOD 400 Operating system. DPS 6 software providing similar functions but operating in other operating systems such as Octanet 8 is outside the scope of this specification.

Besides the LAC software mentioned above, the LAC will contain kernel operating system software and software interface routines. The software interface routines reside in LAC RAM and are so called because they isolate the LLC software from the hardware characteristics of the megabus and MAC adapters.

LAC kernel software is based on a communications oriented O.S. provided by Bridge Communications Inc.. Kernel software is described in Reference [11]. Software interface routines are described in the LACS Hardware EPS-1 [Reference [3]].

1.4 USERS OF LACS SERVICES

This section describes some of the users of LACS services.

Type 1 LLC provides minimum connectionless data transfer services over the LAN. This is suitable for use by the DSA P-P Network Services since the DSA Transport Services provide the necessary error recovery mechanisms.

The Connection Oriented [CO] Subnetwork services are required by the DSA X.25 services. The purpose of the CO subnetwork layer is to add error detection and recovery mechanisms to the connectionless LLC services.

Transport Services provide reliable connection oriented data transfer services. When used with the Internet Network layer, it also allows the user to communicate with peers that are not located on the same LAN. Transport Services are suitable for use by the MOD 400 Distributed Resource Management Option [DRMO]. It is also suitable for communicating with other vendor products that support ISO class 4 transport protocols. It will also be used, by the ISO VTP and ISO session services in the DPS-6 to access terminals and other secondary network devices over a LAN. These devices will be accessed via Bridge NIUs.

System Management services available to Administrative applications are provided via an SRGIO interface modelled after IEEE 802.1 Part B [System Management]. These services will be used by the DSA Node ADministrator (NAD) and T & V Routines.

1.5 REFERENCE DOCUMENTS

This specification uses the following documents as reference. When the documents are superseded by later or approved revisions, the latest revision shall apply.

Document No.	Title
1. 60149743	Local Area Network Controller PFS
2. 58082015	Local Area Network Global Functional Specification.
3. 60149766	EPS-1 (Hardware) Local Area Controller Subsystem (LACS) Rev G.
4. IEEE 802.1	(Part A) Local and Metropolitan Area Network Standard Overview and Architecture Rev B June 1983.
5. Draft IEEE 802.1	(Part B) Systems Management , Revision I, Sept 1985
6. ANSI/IEEE Std 802.2-1985	IEEE Standards for Local Area Networks: Logical Link Control.
7. ANSI/IEEE Std 802.3-1985	IEEE Standards for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
8. ANSI/IEEE Std 802.4-1985	IEEE Standards for Local Area Networks: Token Passing Bus Access Method and Physical Layer Specifications
9. ANSI/IEEE Std 802.5-1985	IEEE Standards for Local Area Networks: Token

Ring Access Method and Physical Layer Specifications

10. ISO 7498-1984
Information Processing Systems -- Open Systems Interconnection -- Basic Reference Model, October 15,1984

11. ISO/DIS 8348
Information Processing Systems -- Data Communications -- Network Service Definition

12. ISO/DIS 8348/DAD 1
Addendum to the Network Service Definition Covering Connectionless-mode Transmission. April 1984.

13. ISO/DIS 8348/DAD 2
Addendum to the Network Service Definition Covering Network Layer Addressing. April 1984.

14. ISO/DP 8473
Information Processing Systems -- Data Communications -- Protocol for providing the Connectionless-mode Network Service. May 1984

15. ISO/TC 97/SC 6 N3141
Internal Organization of the Network Layer April 1984.

16. ISO/DIS 8072
Information Processing Systems -- Data Communications -- Transport Service Definition

17. ISO/DIS 8073
Information Processing Systems -- Data Communications -- Transport Protocol Specification.

18. ISO/DP 8073/DAD 1
Information Processing Systems -- Data Communications -- Addendum to Transport Protocol Specification to enable Class Four operations over Connectionless Mode Network Service.

19. OS-0016-00 ESPL Software Technical Reference Manual, Vol. 1, Kernel and Support Software (Bridge Communications, Inc.)

20. CZ05-00 GCOS 6 MOD 400 System Programmer's Guide Vol I

21. CZ06-00 GCOS 6 MOD 400 System Programmer's Guide Vol II

22. DSA 70 Distributed Systems Administration & Control ARCHITECTURE

23. DSA 71 Distributed Systems Administration & Control ADMINSTRATIVE EXCHANGE PROTOCOL

24. DSA 72a Distributed Systems Administration & Control NETWORK CONTROL LANGUAGE

25. DSA 76 Distributed Systems Maintainability

26. Engineering Component Specification, LAN Data Structures, July , 1985

27. Engineering Component Specification, LACS Driver Interface Services, P.Stapera July , 1985

28. Engineering Component Specification, LACS Driver Megabus Services, P.Stapera July , 1985

29. Engineering Component Specification, LAN Configuration Services, L.Vivaldi July , 1985

30. CCITT Recommendation X.409, Message Handling Systems: Presentation Transfer Syntax and Notation, September,1984

1.5 ABBREVIATIONS AND DEFINITIONS

TBW

Chapter 2

ARCHITECTURE

2.1 REFERENCE MODEL

Figure 2.1 illustrates the architectural scope of this document. The model is described in terms of the ISO Open Systems Interconnection (OSI) Reference Model (RM) and the IEEE 802 model for the Data Link Layer and for System Management.

```
 ------------      -----------------
|            |    |TR  :TRANS LAYER|
|            |    |LM  :[ISO CL4]  |
|  SYSTEM    |    |___:____________|
|            |           |
|MANAGEMENT  |           V
|            |    |---:------------|
|            |    |    :CL NETWORK |
|            |    |NET:  LAYER     |
|            |    |LM  :NULL OR IP |      ----------------
|            |    |___:____________|     |    :CO SUB-     |
|            |           |               |SN  :NETWK LAYER |
|            |           |               |LM  : [CL4 TR]   |
|            |           |               |___:____________|
|            |           |                       |
|            |           V                       V
|            |    |---:-------------------------------------|
|            |    |LLC:    TYPE 1  LLC                      |
|            |    |LM :                                     |
|            |    |___:_____________________________________|
|            |                       |
|            |                       V
|            |    |---:-------------------------------------|
|            |    |MAC:    MEDIA ACCESS CONTROL             |
|            |    |LM : [CSMA/CD, TOKEN BUS OR TOKEN RING]  |
|____________|    |___:_____________________________________|
```

Fig. 2.1 LACS Reference Model

The LACS will implement the following ISO layer protocols:

- Transport Layer. ISO class 4 transport services will be provided in the second release. ISO Class 4 Transport provides the following services:

  * Connection Oriented Data Transfer services with full error detection and recovery. Transport manages a normal and expedited data stream each of which are subject to independent flow control regulations.

  * Segmenting and reassembly. Service Data Units [SDUs] supplied by the user can be segmented into multiple

Transport Protocol Data Units [TPDUs] and reassembled at the receiving transport entity. This allows the user's sdu size to be independent of the optimum size of the pdu for the communication medium.

* Multiplexing and de-multiplexing of two or more connections on a single network path. Transport also allows for a pair of user entities to have more than one connection between them.

* Splitting and recombining. Class 4 transport allows the simultaneous use of two or more network paths to support the same transport connection. This is restricted to network paths out of the same LACS. Support of multiple network paths for the same connection through two or more LACS requires the transport entity to exist in the DPS-6. When using connectionless network services, it is the network layer's responsibility to chose the path out of the system to transmit an N_SDU. The splitting function is therefore transparent to the sending transport entity. The receiving transport entity, however, has the responsibility of maintaining the order of T_SDUs received for a connection.

- Network Layer. The LACS will provide Connectionless Network Services in the second release. The network entity is responsible for selecting the appropriate path [LAN] out of the controller to reach the remote endpoint. The ISO "Protocol for Providing Connectionless-mode Network Service" [ISO 8473], commonly known as the Internet Protocol will be used to provide connectionless network services. Both the "Inactive Network Layer Protocol" subset [NULL Network subset] and the "non-segmenting protocol" subset will be supported in the second release.

NULL Network protocol implies that the link SDU does not carry any protocol information [the network header only contains 1 octet specifying the Network Protocol ID of 0 for this subset]. The NULL Network Protocol only allows point-to-point connections to systems on ANY of the LANs attached to this controller. The Internet Protocol with an "Active" Network Layer Protocol [i.e. the link SDU containing network protocol information], allows relaying to systems not directly connected to any of the LANs that the

LACS is connected to. The Internet Protocol (IP) will itself be implemented in stages. The "non-segmenting protocol" subset implies that the LACS network entity does not perform any segmentation and reassembly. Also the LACS will act only as an endpoint. Relaying capabilities through the LACS and between two LACS as well as segmentation and re-assembly will be supported in a future release.

- Sub-network Layer. Since the LACS only provides connectionless link services (LLC Type 1), connection oriented subnetwork services will be provided for users requiring reliable "Link Connection" services. According to the ISO Network Layer model, this is equivalent to providing a connection oriented Sub-Network Dependent Convergence Protocol (SNDCP) using connectionless LLC services. The SNDCP services are equivalent to LLC Type 2 or HDLC LAP-B services required by the X.25 PLP (Network) layer. SNDCP services will primarily be used by the DSA X.25 layer. Class 4 transport protocols will be used to provide these services. Using class 4 transport protocols (which are needed for other purposes) instead of Type 2 LLC reduces the implementation effort needed to provide the required services. The transport protocol is only used to provide the connection oriented data transfer services. Other "Transport" services such as segmenting/re-assembly are not used. The format of the SAP addresses are also different from TSAP addresses. (See section 2.3.3 on addressing).

- Data Link Layer. IEEE 802 Data Link protocols will be supported. IEEE 802 divides the Data Link Layer into two sub-layers: Logical Link Control (LLC) and Media Access Control (MAC). Type 1 (Connectionless) LLC services will be provided as defined in IEEE 802.2. In the initial release, only the CSMA/CD MAC protocols as defined in IEEE 802.3 will be supported. The Token Bus (IEEE 802.4) and the Token Ring (IEEE 802.5) MAC protocols will be supported in the next release. The MAC protocols are implemented on daughter boards. The LACS can support upto four adapters each supporting the same or different MAC protocols. (The actual number maybe smaller depending on type and mix of protocols.)

In addition to the Data Transfer Services described above, Management facilities must be provided to support local area network (LAN) operation. Management facilities provide capabilities to start up, carry traffic, reconfigure, restart and close down the LAN.

2.1.1 Management Reference Model

Management services are modelled after the IEEE 802.1 Part B System Management document. 802.1 Part B describes a model for management of the 802 LLC and MAC Data Link sub-layers only. It is expected that the same model will be extended by ISO to describe management of all the OSI layers. Figure 2.2 is an extension of the 802.1 model to include the layers that are supported by the LACS.

```
                                          |-------------|
                         4-SMDSI          |             |
                   |----------------------|             |
 [TSAP]            v                      |             |
|-[*]...[*]---[*]--|------|               |   SYSTEM    |
|      TRANSPORT   :XPORT | 4-LMI         |             |
|  Protocol Entity : LME  |<----->|       |             |
|------------------:------|               |             |
                         3-SMDSI          |             |
                    |---------------------| MANAGEMENT  |
   [NSAP]           v                     |             |
|---[*]........[*]-|------|               |             |
|       NETWORK    :Netwrk| 3-LMI         |             |
| Protocol Entity  : LME  |<----->|       |             |
|------------------:------|               |             |
                         2b-SMDSI         |             |
                    |---------------------| APPLICATION |
   [LSAP]           v                     |             |
|---[*]........[*]-|------|               |             |
|       LLC        : LLC  | 2b-LMI        |             |
| Protocol Entity  : LME  |<----->|       |             |
|------------------:------|               |             |
                                          |  PROCESS    |
        [MACSAP]                          |             |
|---------[*]------|------|               |  [SMAP]     |
|       MAC        : MAC  | 2a-LMI        |             |
| Protocol Entity  : LME  |<----->|       |             |
|------------------:------|               |             |
                                          |             |
        [PSAP]                            |             |
|---------[*]------|------|               |             |
|       PHY        : PHY  | 1-LMI         |             |
| Protocol Entity  : LME  |<----->|       |             |
|------------------:------|               |-------------|
```

LME = Layer Management Entity
LMI = Layer Management Interface
SMDSI = System Management Data Services Interface

FIGURE 2.2 MANAGEMENT MODEL

According to the model, management functions are distributed between the System Management Application Process (SMAP) and the Layer Management Entity (LME) components within each layer. The SMAP contains the application level management functions such as initialization and closedown, statistics collection and logging/reporting etc. These functions are outside the scope of OSI definition. Included also in the SMAP is the System Management Entity (SME). The SME is responsible for supporting the peer protocols between the SMAPs in different OSI systems and for interfacing with the LMEs. 802.1 and ISO are responsible for defining the architecture and functions of the SME and LMEs, the Layer Management Interface (LMI) between the SME and each LME, and the peer protocols between the SMEs in each system.

The LME is responsible for managing the parameters of the layer which are of interest to the SMAP; i.e. it keeps statistics and responds to request to update operational parameters such as addresses, timer values etc. The LME is also responsible for responding to requests for performing actions such as updating states, perform loopback tests etc. and for reporting events to the SMAP. The LMI provides an interface for the SME to request the LME to Get and Set parameter values and to perform actions, and for the LME to respond to these requests. It also provides an interface for the LME to send event indications to the SME.

Management of remote systems is performed by the SME in one system exchanging Protocol Data Units (PDUs) with the SME in another system. If the PDU is a Request PDU (Get, Set or Action), then the remote SME is responsible for routing it to the appropriate LMEs, collecting the responses and returning a Response PDU. A local SME may also generate an Event Indication PDU, as a result of an LME Event Indication, to notify the remote SMAP of local events.

802.1 allows for SME to SME protocol exchange to be conducted by directly using the Data services of LLC or any of the layers above it. The services of each layer Protocol Entity are accessed through the System Management Data Services Interface (SMDSI) for that layer.

Management services described in this specification are limited directly to the management of the LACS and its software layers. Management functions that are provided are:

- LACS Loading/Dumping
- LACS Initialization
- LACS Object Control and Status
- LACS Statistics
- LACS Testing

Other SMAP functions are performed by Administrative Applications such as the DSA Node ADministrator [NAD] and T&V Routines. These are outside the scope of this specification. Exchange of PDUs between the Applications is done at the session level following the DSA Administrative Exchange Protocol [AEP]. This is also outside the scope of this document. No protocol exchange is supported via any of the LACS layers in the first release. Management exchange maybe required between the DPS-6/LACS and Bridge Server products in the second release [TBD].

A System Management Server in the DPS-6 provides an interface to the Administrative Applications to access the services described above. This is called the System Management Interface [SMI].

2.2 SOFTWARE STRUCTURE

Figure 2.3 describes the software components required in the DPS-6 and on the LACS board to implement the functions described in the Reference Models. The figure also shows some potential users of LACS services.

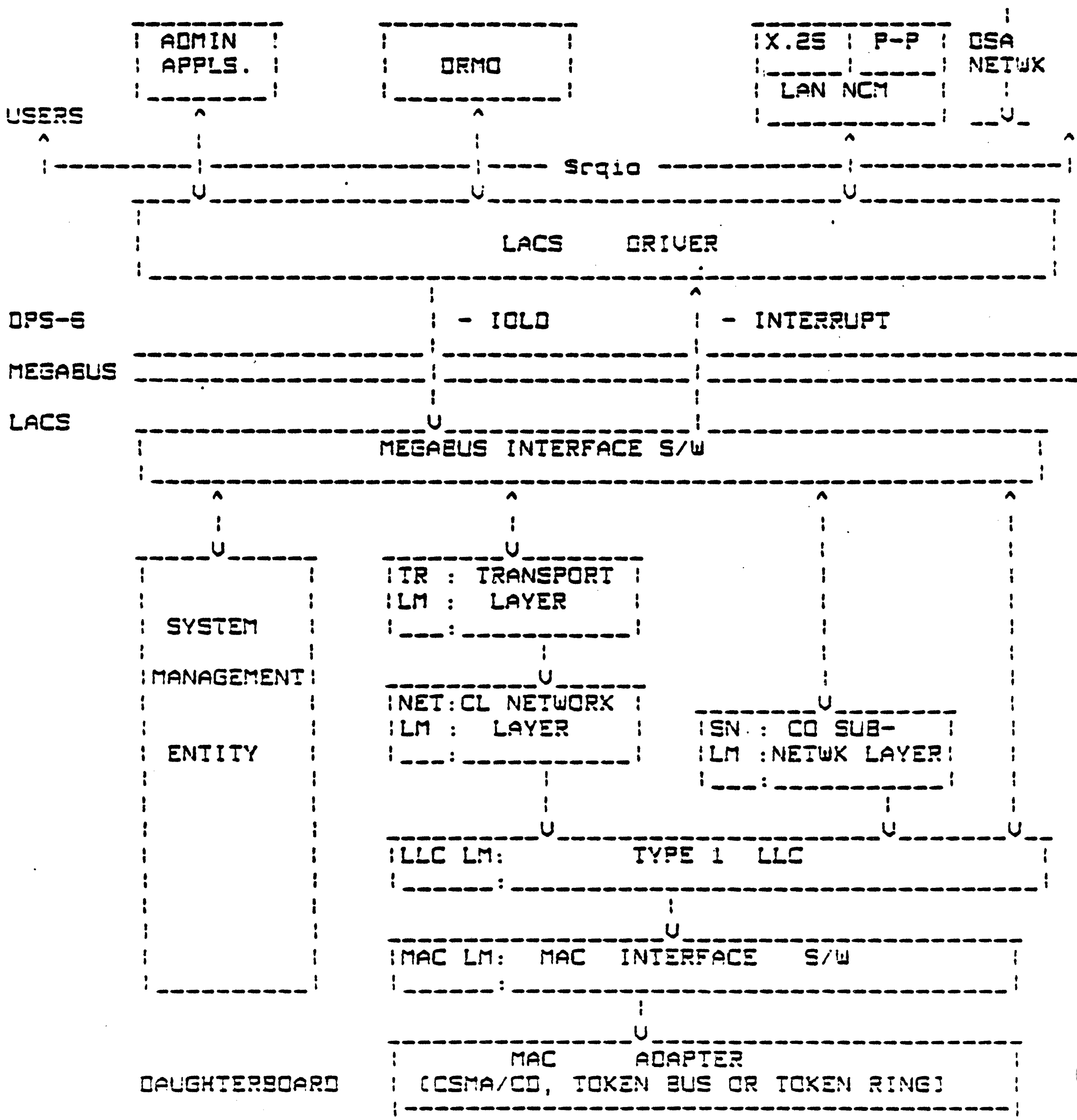

FIGURE 2.3 LACS SOFTWARE STRUCTURE

As can be seen from figure 2.3 the current implementation of LAN services is divided between the DPS 6, LACS motherboard and LACS daughterboards.

The LAC daughterboards provide MAC and physical layer services. These are mainly implemented by hardware and firmware and are described in more detail in the hardware EPS. A different daughterboard will be implemented for each type of MAC and physical layer, viz., broadband token bus, token ring and baseband CSMA/CD. The LACS motherboard is capable of handling more than one type of daughterboard [i.e. LAN] or multiple LANs of the same type.

The LACS motherboard provides the Transport, Connectionless Network, Connection Oriented Sub-network, and Type 1 LLC services. In addition it provides layer management services for each layer entity and System Management services which route management requests and information to and from the various layer management entities. The structure of the various components providing System Management services is described in more detail in section 2.2.2.

The LACS Driver in the DPS 6 is primarily responsible for providing an interface to different types of users of LACS services and control the activities of each of the controllers attached to the DPS-6. It is possible to access either the transport, sub-network, LLC, or System Management services on the LACS. The LACS driver provides a $RQIO MCL interface as well as a subroutine call [Lnj] interface to access LACS services. The Lnj interface can only be used by other system [i.e. SS Task Group] routines [such as the NTD Driver]. The LACS Driver structure is described in more details in section 2.2.1.

Because LAN services are divided into different hardware modules additional support services need to be provided to deal with passing information across hardware boundaries. These support services are the Megabus Services sub-component of the LACS Driver, and the Megabus and MAC interface services in the LACS. These support services isolate the remaining software from the specifics of the hardware [megabus or type of MAC].

The OS Kernel services in the LAC provide a basic executive environment for the software on the motherboard. Kernel services include process [task] management, interprocess communications [mail services], buffer management, and clock [timer] services.

LACS interface services are described in more detail in the Hardware EPS. OS Kernel Services are described in Ref 11. The functions of the other modules shown in fig 2.3 are described in detail in other sections in this specifications. The remaining sections of this chapter define the structure of the LACS Driver [2.2.1], System Management components [2.2.3], and some basic concepts and nomenclatures used in this specification.

2.2.1 LACS Driver Structure

Figure 2.4 shows the various components of the LACS Driver.

```
                           --------------------
                           |   application    |
                           |__________________|
                                   ^
                                   | - $rqio
       |---------------------------|---------------------------------|
       |                           V                                 |
       |  ----------------------------------------------------------  |
       |  |                                                        |  |
       |  |        lacs driver interface services [ldis]           |  |
       |  |________________________________________________________|  |
       |      |      ^             |      ^             |     ^       |
       |jmp - |      | - lnj  jmp - |      | - lnj  jmp - |     | - lnj |
       |   ___V______|___      ___V______|___      ___V_____|___      |
LACS   |  |CL 4 TR +     |    |LLC type 1   |    | sys mgr     |      |
DRVR   |  | SN layer     |    |  layer      |    |  layer      |      |
       |  |   servers    |    |  server     |    |  server     |      |
       |  |    [ls]      |    |   [ls]      |    |   [ls]      |      |
       |  |______________|    |_____________|    |_____________|      |
       |      |      ^             |      ^             |     ^       |
       |lnj - |      | - jmp  lnj - |      | - jmp  lnj - |     |- jmp  |
       |   ___V______|______________V______|______________V_____|___   |
       |  |          lacs driver megabus services [ldms]           |  |
       |  |________________________________________________________|  |
       |                    |                   ^                     |
       |--------------------|-------------------|---------------------|
                            | - iold            | - interrupt
MEGA-  ---------------------V-------------------|----------------------
BUS    ----------------------------------------------------------------
                                   ^
                                   | - lcb
                        -----------V---------------
                        |                         |
                        |          lacs           |
                        |_________________________|
                                   ^
                                   | - pdu
         --------------------------V---------------------------------
LAN      ------------------------------------------------------------
```

FIGURE 2.4 LACS DRIVER STRUCTURE

The LACS Driver in the DPS 6 is primarily responsible for providing an interface to different types of users of LACS services and control the activities of each of the controllers attached to the DPS-6. Ther are two types of users; Data Transfer Services users and System Management Service users. Data Transfer Service users can be further divided into users of either the transport, sub-network, or LLC services on the LACS. The LACS driver provides a SRQIO MCL interface as well as a subroutine call (Lnj) interface to access LACS services. The Lnj interface can only be used by other system (i.e. SS Task Group) routines (such as the NTD Driver). The Driver is responsible for validating the request and mapping the information in the IORB to that required by the appropriate layer instance on the LACS. Information between the DPS-6 and the LACS is passed in a LAN Control Block (LCB) the contents of which is layer specific. The Driver also services interrupts for the completion of the LCB requests. In addition, the System Manager server in the LACS is responsible for loading and initializing the LACS and controlling the activities of all the LACS controllers attached to the DPS-6

LACS Driver services are therefore divided into the following logical components:

1. LACS Driver Interface Services (ldis). This component provides common interface services to both Data Service and System Management Service users. See the LACS Driver Interface Services Component specification for details of the functions performed by this component.

2. Layer Servers. The layer servers perform functions that are specific to the layer to which the services are mapped to. The Layer Servers can be considered extensions of the layer entities on the LACS. Three types of layer servers are identified for this implementation.

    1. Transport and Subnetwork layer servers. These servers provide support for the connection oriented services that are provided by the Transport and CO Sub-network layer instances on the LACS.

    2. LLC Type 1 server. This provides support for the connectionless data transfer services provided by the LLC sublayer on the LACS.

3. System Manager layer server. This performs loading, initializing and other management of the LACS as well as an interface to administrative applications in the DPS-6.

3. LACS Driver Megabus Services [ldms]. This component provides common services required by all layer servers for interfacing with the megabus. The ldms is responsible for issuing I/O orders [I/O and IOLD] across the megabus and receiving interrupts from the LACS and dispatching them to the appropriate Layer Server. The ldms also provides flow control on the total orders outstanding on the LACS.

2.2.2 System Management Structure

Figure 2.5 shows the various components involved in providing Management services.

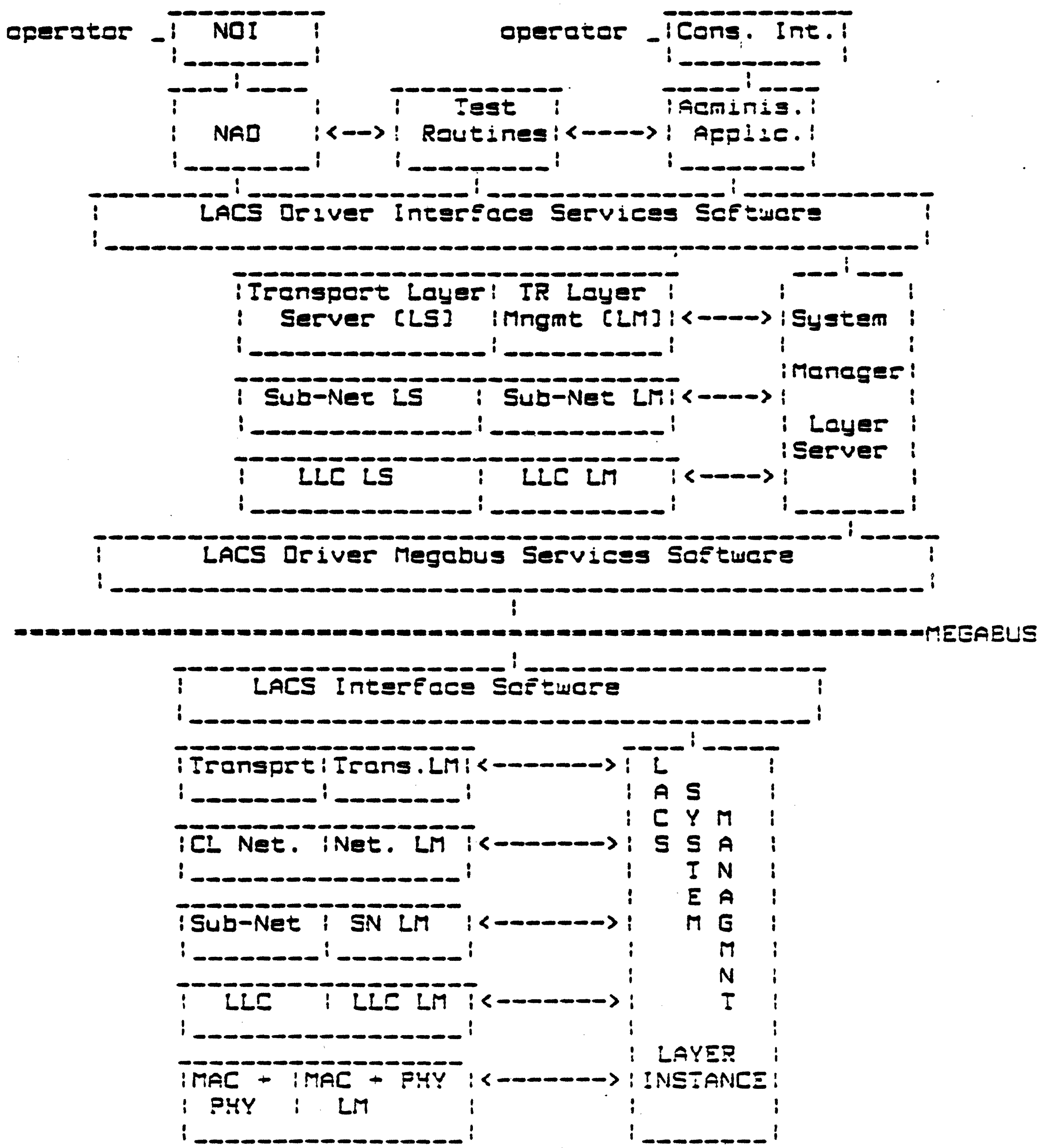

Figure 2.5 System Management Components

System and Layer Management functions are distributed between the DPS-6 and the LACS. The System Management component in the DPS-6 is called the System Management Layer Server, or SM server. The SM component in the LACS is called the LACS SM layer instance.

The System Manager layer server in the DPS-6 is responsible for loading and initializing each LACS controller attached to the DPS-6. In addition it provides an interface to an Administrative Application or Test Routines. The only Administrative Application that is planned to use System Management services is the DSA NAD. In the future, a non-DSA Administrative Application (ADAP) may be implemented. This application would be more in line with OSI and IEEE 802 Management architecture.

The System Manager Server is reponsible for routing the requests to the appropriate layer management routine. This may be part of the layer server in the DPS-6 or the layer instance in the LACS. If the operation is to be performed in the LACS then the request is routed via the System Management Layer Instance in the LACS. The System Management components in the LACS and DPS-6 are also capable of receiving event indications from the layer management components. If the Application has requested notification of events then the event indication is passed on to the application. Certain event indications from a layer management component on the LACS may also be routed to the layer server on the DPS-6.

2.3 CONCEPTS

This section describes OSI, IEEE 802, and Honeywell Distributed Systems Architecture (DSA) concepts and terminology that are used in this specification.

2.3.1 Layer Entities and Layer Instances

A Layer Entity is an active element in the layer which co-operates with peer entities by using one or more protocols. There can be several Instances of layer entities within a layer. Each such instance is called a layer entity instance or layer instance for short. For example, the ISO class 4 transport services can be considered an instance of a transport layer entity, while the Connectionless Network services and the SNDCP sub-network services are two instances of network layer entities.

A layer instance in each LACS is identified by a layer number [0-7] and a layer instance number [0-7]. Layer number 0 is reserved for System Management. Layer number 1 is reserved for MAC [adapter] interface software. Layer numbers 2-7 have the same definition as the OSI RM.

The LACS is capable of supporting multiple instances of layer entities at each layer. This maybe the same type [i.e. supporting the same protocol] or different types. In the structure defined in Fig. 2.3 there is one instance of the transport layer, two instances of the network layer [Connectionless and SNDCP] and one instance of LLC for each MAC adapter.

2.3.2 Service Access Points [SAPs]

According to the OSI RM a Service Access Point [SAP] "is the point at which a pair of entities in adjacent layers use or provide services" i.e. a SAP is the interface point through which the service requester accesses the services of the lower layer service provider.

The name of the SAP is associated with the layer providing the service. Thus transport layer services are accessed through a TSAP, network layer services are accessed through an NSAP, SNDCP services are accessed through an SNSAP[1] and Link Layer services through an LSAP.

----

1. SNSAP is not an ISO defined term. A formally defined SAP between the network and the SNDCP layer is required in this implementation because the services are split between the DPS 6 and the LACS.

2.3.3 Addressing

Associated with each SAP is a SAP address. SAP addresses are used by the service requester as well as the service provider to identify the corresponding service requester entities that are engaged in a peer to peer dialogue.

A layer entity is typically capable of providing services to more than one entity in the layer above: i.e. it must support more than one SAP. A layer entity is also typically capabable of using the services of different entity instances in the layer below it as long they can provide the minimum services required by the layer entity. For e.g., the ISO Class 4 Transport Entity in the LACS can provide services to DRMO and to the ISO Session Control Entity in the DPS-6. In turn it could use the Connectionless Network entity in the LACS and any other type of network entity in the future. Each layer entity instance is therefore responsible for mapping the SAP addresses associated with it to the SAP address of the lower layer entity instance that it uses to provide the service.

In the LACS environment the following definitions and conventions apply to the addresses of SAPs associated with different layer entities and to their mappings to lower layer entity SAP addresses:

- The LSAP address is defined in IEEE 802.2 as consisting of an LLC address field (8bits) plus the MAC address (16 or 48 bits). MAC addresses (and therefore LSAP addresses) must be unique for the LAN.

  There is one instance of the LLC protocol entity for each MAC adapter. Each LLC instance must be capable of supporting multiple LSAPs (maximum theoretical limit is 128 per adapter, of which 64 are reserved for assignment by IEEE). By convention an LSAP must be defined for each network entity. If the full structure shown in figure 2.3 is supported by a system then we need to define three LSAPs per adapter, one each for the Connectionless Network entity in the LACS, the DSA X.25 SNDCP sub-network entity in the LACS and the DSA P-P entity in the DPS-6. The following pre-defined values, from the locally defined range will be used for the LLC subfield for each type of LLC user:

IP=X"10", DSA P-P=X"20", DSA X.25 = X"30". If 802 assigns a Universally defined value for IP this will be used instead of the privately defined value. This convention eliminates the need to co-ordinate the LLC subfield value.

Mapping of LSAP address to MACSAP adresses is implicit.

- SNSAP addresses are identical to the LSAP address used for sending and receiving the sub-network layer protocol data units (PDUs). The SNDCP sub-network layer must be capable of supporting multiple users (SAPs). Each user has its own set of SNSAPs; one per adapter that it wishes to communicate on. In the structure shown in Fig. 2.3 there is only one user; the DSA X.25 Network Entity. The LLC subfield of the LSAP will be used in SN_PDUs (such as Connect Request PDU) that require a SAP_ID.

- NSAP addresses must be unique in the network. The format defined in ISO DIS 8348/DAD 2 must be followed. NSAP addresses following ISO 8348/DAD 2 conventions can vary in length upto a maximum length of 20 octets. ISO 8348/DAD 2 breaks up the NSAP address into domains. The definition of the semantics for the Domain Specific Part is left to the person or entity having authority over the Domain. The format described below will be used in the second release when assigning NSAP addresses for communicating between HIS systems and HIS and Bridge Servers. The network entity should also be capable of supporting other ISO standard or defacto standard (e.g. MAP) formats.

According to ISO 8348/DAD 2 the NSAP Address syntax is as shown:

AFI	IDI	DSP
(1 octet)	(Variable)	(upto 15 octets)

where:

AFI:= Authority and Format Identifier. The AFI is an integer in the range 0-99 (one octet). The values assigned for "Local" (48-51) must be used for privately defined NSAP addresses. For these values of AFI the IDI is defined to be NULL. [Other AFI and IDI values are used for ISO and CCITT defined NSAP addresses]. We will use a binary representation [hex digits] to specify NSAP addresses in our configuration file so the AFI must be X"49".

IDI:- Initial Domain Identifier. This field is NULL for local AFI values [48-51].

DSP:- Domain Specific Part. This can be defined to be specific to HIS and Bridge Communications Inc. The structure used by MAP will be followed with some modifications. This is described below.

The format of the DSP is as follows.

Pvt. IDI [1 octet]	SN Addr [2 octets]	SNPA [11 octets]	NSAP Sel [1 octet]

where:

Pvt. IDI:- Is a Privately defined IDI that identifies the semantics used for the rest of the DSP. A value of X"8" will be used.

SN Addr:- Sub-Network Address. This defines the sub-network [LAN] through which the NSAP can be accessed. A value of FFFF implies that the address is a group NSAP address.

SNPA:- Sub-Network Point of Attachment. This is the sub-network specific address and must be unique for the sub-network. There is a one-to-one mapping of the SNPA to the LSAP [LLC + MAC] address. The SNPA should not, however, contain the MAC address since this would cause the NSAP address to change if the hardware address changes [e.g. replacing one Bridge Server with another].

NSAP Sel:= NSAP Selector. Identifies the user of the Network entity. A pre-defined value [0] must be used by all systems to identify the Class 4 Transport entity as the Network Service user. A Selector value of zero is the default value that must be assumed when the NULL Network protocol is used. The selector field can be considered a local logical address of the SAP.

The total NSAP address length is 16 octets.

The NSAP structure described above allows for efficient network routing. This scheme, however, requires a system such as the DPS-6 that supports multiple controllers, each supporting multiple attachments to LANs, to be addressed by different NSAP addresses, one for each LAN attachment. Controller and System Group NSAP concepts are defined so that the controller or system can be viewed as a single SAP. For example, the Group Controller SAP concept allows the Transport entity to access network services through a single SAP. The following convention will be used by DPS-6 systems to define the SNPA:

Octet 1-2: Is the controller identifier [0-F]. FFFF=Group.

Octets 3-4 Is the adapter identifier [0-F]. FFFF=Group.

Octets 5-11: Identifies the system. This must be the same for all NSAPs supported by a system.

An NSAP address with the sub-network field FFFF and octets 1-4 of the SNPA FFFFFFFF is to be considered as a System Group NSAP address. An NSAP address with the sub-network field FFFF and octets 3-4 of the SNPA FFFF is to be considered as a Controller Group NSAP address.

The Connectionless Network Entity is responsible for selecting the path to be taken to send a packet to a remote NSAP. The path defines the local LSAP to use as well as the LSAP address of the station on the LAN which is the next hop in the route. The NULL Network Protocol only allows point-to-point connections to systems on ANY of the LANs attached to this controller.

For the structure defined in Fig. 2.3, there is only one transport entity so one needs to define only one NSAP per adapter and one Controller Group NSAP per controller[2].

The network layer needs the following mapping information in order to determine the local and remote LSAP to use when sending a packet to a remote NSAP. The subnetwork # is checked to see if the remote NSAP is attached to any of the subnetworks to which the controller is attached to. If it is not (true only when IP protocols used) then the subnetwork # is used to determine the local LLC instance and the remote LSAP address to use to send the N_PDU. If the remote NSAP is attached to one of the "local" subnetworks then the SNPA is mapped to the local LLC instance and the remote LSAP address. The reverse mapping is done for incoming N_PDUs. If mapping information does not exist then the remote LSAP address passed by LLC and the remote NSAP address are used to dynamically create the mapping information. In the case of the NULL Network subset the N_PDU does not contain any NSAP addresses. In this case the SNPA of the remote NSAP is assumed to be FFFFFFFF plus the remote LSAP passed by LLC. The subnetwork # is set to be the subnetwork # on which the N_PDU was received. As mentioned above, the NSAP selector is defaulted to 0.

- The TSAP address identifies the transport service user entities (DRMO, ISO session etc). The transport entity has to map the local and remote TSAP addresses to the local and remote NSAP addresses to be passed to the network layer entity. The TSAP address is composed of a Transport User selector field (2 octets) plus the NSAP address. This allows an implicit TSAP address to NSAP address mapping. There is one TSAP per user per adapter. In most cases, however, the user does not want to chose the TSAP to use to provide the services for a connection. The LACS Driver can provides this independence by supporting "Group" TSAPs. A group TSAP is a set of TSAPs owned by the user. The user

----

2. It is necessary to define NSAPs for the DSA X.25 and P-P Network entities. This is outside the scope of this document.

may request services through a single Group TSAP. A TSAP containing the System Group NSAP address is a Group TSAP. The driver will use the remote TSAP address to map transport connection requests to the appropriate LACS. If the remote TSAP can be accessed by more than one controller then the one supporting the fewest connections will be chosen.

2.3.3.1 Object Structure

The System Management function administers the LAN layers by operating on the parameters which control and describe the operation of each of the layers. These parameters are viewed slightly differently in a DSA environment than in a pure IEEE 802 environment. In a DSA environment, information about some aspect or entity of a system are logically grouped together into objects. Objects of the same type are in turn grouped together into classes of objects. Management operations are performed on attributes of objects or object classes.

In an IEEE 802 environment control is exercised over parameters belonging to entities (components) within a layer or sub-layer instance. The LACS system management function provides services which span both perspectives and are from herein referred to as objects. Objects defined for a Honeywell IEEE802 LAN are :

- Controller(CT).

  Describes the LACS board; maintained by the L6 system management function.

- System Management Administrative Function(SMAF).

  Describes the System management administrative function; maintained in L6.

- Physical Line (PL).

  Describes the characteristics of the adapter boards' hardware and firmware. Described by attributes and statistical information available from both the MAC and Physical layers. The Physical Line is maintained in the LACS.

- Physical Connection [PC].

  Describes the physical connection onto the LAN. The Physical Connection object is not supported for the LAN.

- Logical Line [LL].

  Describes the characteristics of a LSAP. Described by the set of attributes for an LSAP. Maintained in the LACS board.

- Link Connection [LK].

  Describes the characteristics of the link connection between two LSAPs. Described by a set of statistics maintained by the LLC layer on the link connection. Maintained in the LACS. The LK object is not initially supported by the LLC layer [type II services not supported].

The attributes of each object are listed in Appendix A.1.

DSA objects and IEEE 802 components have a set of states which describe their present operational capabilities. The current states identified for IEEE802 components still lack any detailed definition, so the DSA state representation has been used to describe each of the objects. DSA defines seven possible states which can describe any object. These states are used for administration and should not be confused with the substates used for operational control. The seven DSA states defined for the objects of a LAN are as follows:

- IN-USE - The object is fully operational and is currently being used by a higher layer object.

- ENABLED - The object is fully operational but is not currently being used by a higher layer object.The Enabled state is not currently supported.

- DISABLED - The object is not available for use by a higher layer object due to a lower layer supporting object being unavailable. The Disabled state is not currently supported.

- LOCKED The object is unavailable for service to other objects. Entering the LOCKED state is an abortive operation.

- TEST - The object is under exclusive control of a test function.

- SHUTDOWN - The object is undergoing a graceful transition to the LOCKED state. The Shutdown state is not currently supported.

- NONEXISTENCE - This is not truly a representative state but is used to describe objects with which there is not an associated data structure.

A summary of the DSA states to be implemented is shown in Appendix A.2.

Chapter 3

LACS SERVICES FUNCTIONAL DESCRIPTION

]

3.1 LACS DRIVER SERVICES

This section provides a basic description of the services available to LACS users. Since different users require different types of services, the LACS Driver has the responsibility of exposing Transport, sub-network, LLC and System Management services. Following MOD 400 conventions, these services are provided via $RQIO interface. A LNJ interface will also be provided in the second release for users that are in the SS group such as the NTD Driver. This interface may be designed to resemble the Gate Manager Interface used between DSA/ISO layers. The lnj interface is FFS.

The LACS Driver provides a mapping from the logical view provided via the $RQIO interface and the ISO oriented services provided by the LACS. Familiarity with MOD 400 $RQIO interface is assumed.

The terms **Local User** and **Remote User** are used throughout this specification.

A Local User is a generic term to define the user of LACS services. Each local user is identified by an lrn. There is one Local User for each SAP exposed through the Driver. Local users can be broadly divided into two categories: users of management services and users of data transfer services. The type of data transfer service in turn depends on the layer entity providing the service. The Driver maps the requests made via the SRGIO interface to the appropriate layer instance on the LACS.

Note that an application[3] task or task group can access one or many layer instances. It must, however, use different lrns to access each layer entity service.

Users of data transfer services communicate with peer entities. These peer entities are refered to as Remote Users.

LACS Driver services can be divided into three parts:

- SAP Services.
- Data Transfer Services.
- Management Services.

SAP Services allow the User to create an association with the system and to activate/deactivate use of services through the SAP. They can be used by all types of Users.

Data Transfer Services can in turn be divided into:

- Connectionless (CL) Services
- Connection Oriented (CO) Services

Whether a user has access to CL or CO Services or both depends on the type of services provided by the layer instance. Currently the transport and subnetwork layer entities provide connection oriented services only whereas the LLC layer entity provides connectionless services only. The IEEE 802.2 standard allows for providing connectionless and connection oriented services at an LSAP. Only Connectionless LLC services are supported by the LACS.

----

3. The term application is used in the context of a MOD400 application i.e. a requester of SRGIO services and not in the context of an application layer entity as defined by ISO.

There are two major differences between Connectionless and the Connection Oriented services provided by the LACS.

Firstly, CO services guarantees the delivery of data. Guaranteed delivery means that the entity is capable of detecting and recovering from errors in most cases. In case of irrecoverable errors the user is notified via a reset or disconnection of the connection. Connectionless services do not provide any error detection and recovery mechanisms.

Secondly, in a CL environment, neither the Driver nor the layer entity keeps any context related to the remote SAP, i.e. no separate queues are maintained. Thus, for example, when issuing a read request for connectionless data, the user cannot request to receive only from a specific remote SAP but rather has to request for data destined for its SAP regardless of the remote source SAP address. The user is informed of the source SAP address when the read is posted back. In a CO case the user can make requests specific to a connection once the connection is established.

****Note:****

The remainder of the chapter only describes services obtained through the $RQIO interface. The way services are requested and provided via a LNJ interface is TBD.

****End Note:****

The LACS Driver provides an interface to the following SAP Service primitives:

- Associate User
- Activate Local SAP
- Activate Remote SAP
- Deactivate Local SAP
- Deactivate Remote SAP (second release only)
- SAP Event Indication (CL Data arrival, Credit available, SAP deactivated, Connect Indication)

The LACS Driver provides an interface to the following Connectionless Data Transfer Service primitives:

- Read Connectionless Data
- Write Connectionless Data

The LACS Driver provides an interface to the following Connection Oriented Data Transfer Service primitives:

- Connect Request
- Connect Response
- Read CO Data
- Read Expedited CO Data
- Write CO Data
- Write Expedited CO Data
- Connection Event Indication (CO Data Arrival, CO Credit available, Disconnect Indication)
- Disconnect Request

The LACS Driver supports the following System Management Service primitives:

- Management Request
- Management Event Indication

3.1.1 Service Description

All Users must issue an Associate User call before they can issue any other call. The Associate User call is a Monitor Call unlike all other calls which are $RQIO calls. The user specifies a symbolic name and is assigned an lrn in return. This lrn must be used in all subsequent $RQIO calls. The Driver also registers the user's Task Group ID with the exec as a user of the lrn. If the Task Group is abnormally aborted the exec will deactivate the SAP (and disassociate the lrn) on behalf of the user.

Users must then issue Activate Local SAP and Activate Remote SAP calls. The Activate Local SAP call enables the SAP component in the corresponding layer entity thus "activating" the availability of services through the SAP.

The Activate Remote SAP call must be issued for each remote user with which the local user wishes to communicate. The user specifies a symbolic name and is assigned a logical address in return[4]. This logical address must be passed in all subsequent requests requiring a Remote SAP address. This scheme is consistent with the use of lrns to reference local SAPs.

Note 1: The initial release provides management services for local entities only. A Management Service user, therefore, should not issue an Activate Remote SAP call.

Note 2: In the first release, Data Transfer service users are required to issue Activate Remote SAP calls for all SAPs that they expect to communicate with before any services to or from that SAP can be provided. Also it is necessary to provide configuration information for all Remote SAPs of interest. In the second release, it is only necessary for the user to issue Activate Remote SAP calls for those SAPs that it is initiating a conversation to [i.e. for which it issues a connect request or the first connectionless data transfer request]. If addressing information for the remote SAP is not configured locally it should be possible to obtain them via System Management Directory services. It should also be possible, in the second release, for a Remote SAP to initiate a conversation [i.e. issue a connect request or the first connectionless data transfer request] even though it has not been "activated" locally. Logical addresses are assigned dynamically and associated with Remote SAPs which

----

4. A 4 octet logical address is returned instead of the actual SAP address since SAP addresses are variable in length and can be quite long; e.g. an NSAP can be as large as 20 octets.

are not locally configured. (There will be a limit to the number that can active at a given time]. This association is broken when the Remote SAP is De-activated or when the Local SAP for which the association is made is De-activated.

Once SAPs have been activated the user can issue management or data transfer service calls depending on the type of user.

In the case of a Data Tranfer Service User, if the SAP supports CL services then the CL Service calls can be made in any order. If the SAP supports CO services then the user must first issue a Connect Request to establish a connection to a remote or a SAP Event Indication call to receive a Connect Indication call from any remote SAP. The Connect Response call must be made to respond to a returned Connect Indication Event. The Connect Response call is made to inform the service provider whether to accept or reject the remote connect request. Once a connection has been established, the user may issue any other CO call.

Both normal and expedited data transfer services are supported when CO services are used. The expedited data transfer service is optional requiring negotiation with the peer transport entity. It allows for the transfer of a limited number of octets of data to bypass, and given higher priority than, the normal data transfer flow.

Once a Disconnect Request call has been issued or an event indication returned, indicating that the connection has been disconnected because of a remote disconnect request or for internal reasons, no other calls can be accepted on the connection. The user can initiate and accept multiple connections through a SAP to the same or different remote SAPs.

The Deactivate Local SAP call should be issued when all user activity is terminated. It allows the executive and Driver/LACS to clean up all the resources associated with the user. This call is also made by the executive if it detects an abnormal abort of the Task Group associated with the user.

The Deactivate Remote SAP call should be issued when all user activity to the Remote SAP is terminated. If the logical address for the remote sap was dynamically created then its association with the remote SAP is broken and the logical address can be re-used to reference another Remote SAP.

The interfaces to the system management services, in both the LS and the LACS, is IEEE 802 oriented. It is based on the structure of system management PDUs described in the standard for System Management, IEEE802.1, Part B. The LACS Driver System Manager Server supports three types of PDUs, a request SM PDU, a response SM PDU, and an event SM PDU. A user issues a request PDU to the System Manager and the System Manager responds to that request with a response PDU. The request and response PDU contain routing and operation information. The routing information identifies what layer and object the operation is on. The layer is identified by fields describing the layer, the sublayer and the layer instance. The object is identified by the layer internal selector. This field is defined to contain a set of DSA-like selection parameters; name, class, type, venue, and state of an object. The operation information specifies what operation to perform. The following operations are supported by the system management layer server:

GET — Read a specified attribute or group of attributes associated with this layer.

SET — Set an object attribute or group of attributes to the specified value(s).

ACTION — Perform the action on the layer entity or object. Action provides the ability to control an object's state or perform layer specific operations. The following common actions have been defined for LAN operations:

- UPDATESTATE: Update the state of an object or component.
- CREATE: Create an object or component. This is applicable only from a LACS board perspective. In initial releases all objects and components are created during initialization and configuration of the LACS.
- LIST: List the selection parameters (name, class, type, venue) for the specified class of objects or components.

- DUMP: Dump memory of the LACS board.
- LOAD: Load LACS memory from the given LACS bound unit. Start execution from the given address.

In addition to the SM Interface, the interface between the LACS system management function and the layer management functions will be implemented to insure an IEEE 802.1 compatibility.

The primitives are described in detail in the next section. The following subsections describe aspects that the user has to take into consideration which are not directly related to the service.

3.1.2 Buffer Management

The Local User can only request transfer of complete SDUs via the Write and Read Data calls. The user may, however, supply multiple buffers in a Read or Write Request. If the buffers supplied in a Read Request are not large enough to contain the complete sdu then no data is transfered. A status is returned with the actual size of the sdu.

The user may specify a maximum sdu size in the Activate Local SAP call (for CL data sdus) and in the Connect Request or Connect Response call (for CO data sdus). If this size exceeds a maximum size set up administratively for the SAP then the smaller size is returned to indicate the actual maximum size to use. An "ideal" maximum sdu size is also returned. In the CL case this represents the optimal size to use for maximum performance through the LACS and on the LAN. In the CO data case, this represents the maximum size which will not cause the layer entity to segment the sdu into multiple pdus. The maximum sdu size applies to both transmitted and received data. Any "Write" call with data exceeding the maximum sdu size will be rejected. In addition, the layer entity instance on the LACS will discard any SDU that it receives which is larger than this size. It is therefore incumbent on the user to negotiate the maximum sdu (i.e. user's PDU size) that both the local and remote users send.

If the layer instance on the LACS supports segmenting, and the SDU is larger than the PDU then the layer instance may request partial transfers of SDUs between the DPS 6 and the LACS. The amount transfered for transmit depends on the PDU size and the amount of credit available. Incoming PDUs are transferred immediately (if a read is pending) and acknowledged. The Read Request is not posted back until the complete SDU is transferred.

3.1.3 Flow Control

Data Transfer Services are flow controlled in order to prevent one user from using more than its allocated share of LACS resources. The LACS uses free memory for the LACS control blocks (lcbs) containing information regarding each SRGIO call, for inter-process messages and for the data. Since they are all related to the number of outstanding requests, flow control is achieved by limiting this number. Separate limits are kept depending on the type of request, for each SAP and for each connection. Flow control mechanisms are described below for each type of request.

3.1.3.1 Flow control of Indication Calls

Each User (SAP) can have only one SAP Event Indication call pending at any given time. There can only be one Connection Event Indication call pending per connection at any given time.

3.1.3.2 Flow Control of Write Data Calls

Flow of data from the local user to the remote user is controlled by the use of Write CRedits (WCRs). WCR counts are established for each SAP and for each connection for CO data. A WCR for the SAP (CL_WCR) represents the number of additional Write CL Data calls that the user can issue on the SAP. There are two WCR counts for the connection: one for normal data (CO_WCR) and one for expedited data (ECO_WCR). They represents the number of additional Write CO Data and Write Expedited CO Data calls that the user can issue.

The initial value for CL_WCR is returned in the Activate Local SAP call. The initial values for CO_WCR and ECO_WCR are returned in the SAP Event (Connect Indication) or the Connect Request (confirm status) call.

Each time the user issues any Write Data call he should decrement the corresponding xxx_WCR count by one. One of the output parameters returned with the call specifies how much additional credit is allocated to the user. This value should be added to the xxx_WCR. Only zero or a positive number of additional credits can be allocated i.e. credit once allocated cannot be revoked. If a xxx_WCR count is zero then the corresponding Write request cannot be issued. It is possible for the user's xxx_WCR to run down to zero and for the user to not have any pending Write calls which could potentially increase the count. If this happens and more credit becomes available then a corresponding event indication call is returned indicating the availability of credit.

Mechanisms must be provided in the LACS Driver and the layer entity in the LACS to determine the initial credit value and to determine the amount of additional credit to allocate. These mechanisms must be modularized as much as possible so that they can be replaced by other mechanisms in the future. The following mechanism will be used in the initial release.

Flow_Control_Principles

The following principles are followed in determining the flow control mechanism:

1. Principle of fairness. A limit should be put on the total number of write orders that are outstanding on the controller in order to guarantee that controller resources are available for other operations (particularly read operations). Limits should be put on the number of write orders that each user can have outstanding so that one user does not use up all the resources of the controller for write operations.

2. Principle of over-subscribing. Since all users would not reach their maximum limit at the same time, the sum total of the credits allocated to each user can exceed the maximum allocated for write operations on the controller. If the maximum allocated for the controller is reached then the driver should queue up write orders in the DPS-6.

3. Principle of simplicity. The overhead involved in providing flow control should be as low as possible.

Mechanisms_for_managing_Write_CL_Data_call_Credits_[CL_WCR]

A value for the maximum number of pending write requests [CL and normal CO] is set up administratively for each SAP [SAP_MPW] and for the Controller [CT_MPW]. The layer entity is responsible for allocating the initial credit available on the LACS, as well as allocating additional credit, for the SAP. The initial credit is set to be SAP_MPW. Additional credit allocated is simply SAP_MPW minus the sum of credit already allocated and the credit used i.e. the total number of write requests [CL and CO] being operated on, for the SAP, by the layer entity in the LACS. If this amount is negative, a value of zero is returned.

The Driver is responsible for passing the initial CL_WCR value and the additional credits, that is supplied by the layer entity, to the user. The initial value is passed to the user via the Activate Local SAP call. The Driver must keep track of the total credit available in the LACS for the SAP. If the user has zero credit and issues a Write CL Data request, then the request is rejected. If the user HAS credit and issues a Write CL Data request but there is no credit available for the SAP[5] or for the controller then the request is queued in the DPS-6. The amount of additional credit allocated to the user is supplied by the layer entity and passed on to the user in the Write CL Data or the SAP Event Indication call.

----

5. This is only possible when both CL and CO services are provided through the SAP. In the CL only case, the user's credit value would also be zero.

Mechanisms_for_managing_Write_CO_Data_call_Credits_(CO_WCR)

The layer entity is responsible for allocating the initial credit available on the LACS, as well as allocating additional credit, for the connection. The initial credit is passed to the LACS Driver in the Connect Request or the SAP Event (Connect Indication) LCB. The initial credit is set to be the smaller of the following values: the available credit for the SAP and the credit allocated by the remote transport entity in the Connect Request (CR) or the Connect Confirm (CC) PDU. Additional credit is allocated in the Write CO Data or the CO Event Indication LCB. This is determined to be the difference between the available credit and the sum of the credit used and the credit allocated for the connection. The "available credit" is again the smaller value between the credit available on the connection and the credit available on the SAP. If the additional credit is computed to be a negative value it is set to zero.

The Driver is responsible for passing the initial CO_WCR value and the additional credits that are supplied by the layer entity to the user. The initial value is passed to the user via the Connect Request or the Connect Indication call. Additional credit is allocated via the Write CO Data call or the CO Event Indication IORB. The Driver must keep track of the total credit available in the LACS for the connection and for the SAP. If the user has zero credit and issues a Write CO Data request, then the request is rejected. If the user has credit and issues a Write CO Data request but there is no credit available for the SAP or controller, then the request is queued in the DPS-6.

Mechanisms_for_managing_Write_Expedited_CO_Data_call Credits_(ECO_WCR)

The transport protocol only allows one expedited data sdu to be unacknowledged at any given time. The initial ECO_WCR returned via the Connect Request or the Connect Indication IORB is therefore set to one. One additional credit is returned when each Write Expedited CO Data call is returned to the user. The corresponding Write Expedited CO LCB is issued to the LACS layer entity even if there is no credit available on the SAP or controller.

3.1.3.3 Flow Control of Read Data Calls

In the case of Read Data, one needs to limit the number of Read Data calls (CL and CO) that the user has pending as well as the number of sdus that are buffered in the LACS for the user. Two sets of credit values are required: read order credits(xxx_RCR) and credits for buffering sdus (xxx_SDU_CR).The maximum number of Read CL Data calls that the user can have pending (CL_RCR) is established when the local SAP is activated. The maximum number of Read CO Data calls (CO_RCR) and Read Expedited CO Data calls (ECO_RCR) are established when the connection is established. The user should not have more than the corresponding xxx_RCR number of Read Data calls pending.

Flow of data from the remote user to the local user is controlled by the use of sdu credits. An SDU Credit value (CL_SDU_CR) is established for each SAP for connectionless data and for each connection for normal CO data (CO_SDU_CR) and for expedited CO Data (ECO_SDU_CR).

Mechanisms must be provided in the LACS Driver and the layer entity in the LACS to determine the initial Read Order and SDU credit values. These mechanisms must be modularized as much as possible so that they can be replaced by other mechanisms in the future. The following mechanisms will be used in the initial release. These mechanisms follows the same principles that are used for the Write flow control mechanisms.

Mechanisms_for_managing_CL_Read_Order_and_SDU_Credits (CL_RCR_and_CL_SDU_CR)

The initial CL_RCR and CL_SDU_CR values are established as follows. Each SAP has a maximum Read Credit (SAP_MRCR) value set up administratively. There is also a maximum number of pending

Read Data calls (CL and normal CO data) value for the controller (CT_MPR) set up administratively. The user may propose a CL_RCR value in the Activate Local SAP call. If this value is non-zero and smaller than the SAP_MRCR then the user value is used, else the SAP_MRCR value is used to determine the initial CL Read Credit. This value is also used as CL_SDU_CR. The negotiated CL_RCR and CL_SDU_CR values are returned in the Activate Local SAP call.

In the CL data case, the layer entity discards incoming SDUs if the number of SDUs that are already queued equals CL_SDU_CR. The layer entity also sets a limit to the length of time any SDU is allowed to be queued (maximum age of SDU). The sdu is discarded once this limit is exceeded. The maximum age of a CL SDU is also negotiated in the same way as CL_RCR when an Activate Local SAP call is made. The layer entity also rejects any Read CL Data lcb which causes the number of pending Read CL lcbs on the SAP to exceed CL_RCR or which causes the total number of Read lcbs on the SAP (CL or CO) to exceed SAP_MPR.

The LACS Driver rejects any Read CL Data calls if the number of pending Read CL Data calls for the SAP (not counting the new call) equals CL_RCR. If the total number of Read Data calls (CL and normal CO) exceeds SAP_MPR then the call is queued on the SAP queue in the DPS-6. If it exceeds CT_MPR then it is queued on the controller queue.

Mechanisms_for_managing_CO_Read_Order_and_SDU_Credits (CO_RCR_and_CO_SDU_CR)

The initial CO_RCR and CO_SDU_CR values are established as follows. The user may propose a CO_RCR value in the Connect Request or Connect Response call. If this value is non-zero and smaller than the SAP_MRCR then the user value is used, else the SAP_MRCR value is used to determine the maximum CO Read Credit (CO_RCR). This value is also used as CO_SDU_CR. The negotiated CO_RCR and CO_SDU_CR values are returned in the Connect Request or Connect Response call. The actual credit allocated to the peer transport entity is either CO_SDU_CR or the available credit on the SAP whichever is smaller. The initial value is specified in the CR or CC PDUs. Additional credit is allocated in the Ack

PDUs.

In the CO data case, the layer entity does not acknowledge receipt of an PDU to its peer entity until it has been transfered to the user's buffer in the DPS 6 specified in the Read CO Data lcb. Subsequently it is not possible for the layer entity to receive more PDUs than the allocated credit (if it does it is a protocol violation). Other than this fact, the behavior of the layer entity and the Driver is the same as for the connectionless case except that now limits for the connection or SAP or the Controller should not be exceeded.

As for the Write Expedited CO Data case the initial values for ECO_RCR and ECO_SDU_CR are set to one. These values are also returned in the Connect Request or Connect Response calls.

3.1.4 Error Handling

This subsection describes the different mechanisms that are required in the DPS-6 Driver and the LACS to handle different types of error situations.

Errors can be divided into the following classes:

- Catastrophic Errors
    * Unreportable Catastrophic Errors
    * Reportable Catastrophic Errors
- Fatal Operation Errors
    * Fatal Operation errors that must be reported to the initiator of the operation.
    * Fatal operation errors that must be reported to the initiator of the operation and as an event to System Management.
- Non-Fatal Operation Errors.
- Recoverable Errors.

- Protocol Errors.

- DPS6 System Errors.

3.1.4.1 Catastrophic Errors

Unreportable catastrophic errors are errors where the integrity of the LACS controller is corrupted sufficiently to warrant halting the LACS controller. Furthermore the controller cannot be trusted to even report the error. These are typically hardware errors such as parity error while executing instructions, LACS internal bus errors, etc.. The approach taken in handling such errors is to immediately halt all LACS processing in order to preserve the state of the LACS at the time of the errors occurrence. The DPS6 detects a LACS fatal unreportable catastrophic error through the absence of any LACS response or as a result of exceeding the allowed limits for a nak'd IO instruction. There is a possibility that the LACS will not support a dump of its' memory due to the severity of a nonreportable catastrophic errors.

Reportable catastrophic errors are errors where the integrity of the LACS system is corrupted sufficiently to halt the LACS but it is possible to make an effort to report the error. These are typically caused by software errors where the LACS system is not set up correctly. An example of such an error is the kernel returning a "mailbox id unknown" error when attempting to send a message to another process. Any process detecting a reportable catastrophic error must record the location and type of the error in a common area associated with the process and report it to the "layer" management [LM] process it is associated with. An error event indication is sent to LACS system management. The LACS system management process will attempt to report to the SM layer server in the DPS6. In the event that the system mangement process failed to report it to the DPS6, it will request the Controller "layer management" to halt the LACS. The controller LM process will also save an indication of the type of error in a known System Area and set up the controller to NAK all I/O orders except for an Output LACS Control order. The LACS Driver recognizes the fact that the controller is DOWN when repeated efforts to issue IO orders fail or upon receipt of a completed SM event LC3 stating the LACS is DOWN. If the LACS Driver detects a LACS controller is DOWN, it reports this event to the DPS-6 resident system management server. The SM server in turn sends a message to the operator console, sends event indications to any administrative applications that are interested, and deactivates all the SAPs. The system or network administrator can now take action such as dumping, reloading and restarting the LACS.

3.1.4.2 Fatal operation errors

Fatal operation errors are errors which cause the process to abort the particular operation that it is performing and require that the layer server deactivate a SAP or disconnect a connection. An example of such an error is a parity or non-existent memory error when attempting to perform DMA operations between DPS-6 and LACS memory for a CO Write operation. Depending on the type of error, the error is reported via the LCB associated with the error and the layer management process associated with the process, or to the LACS System Management entity, or to all. Errors reported to System Management only are errors which cannot be reported via the LCB, [e.g. DMA errors on the LCB itself]. System Management in turn reports the error to the SM server in the DPS6. The SM server then notifies the layer server in the DPS-6 of the error. If the LCB is available to report the error, then the DPS6 layer server is notified of the error through the LCB. The LACS layer management may still report the error to system management but only for the purpose of notifying a system management application. The action taken by the layer server on receipt of the error depends on the type of service provided. In the case of connectionless service, the associated IORB is posted back with an appropriate status and the SAP deactivated. In the case of connection oriented service, the error may cause the layer server to disconnect the affected connection,and possibly deactivate the SAP. This again depends on the type of operation and is described in more detail in the component specification for the layer server.

3.1.4.3 Non-Fatal operation errors

Non-fatal operation errors are errors which cause the process to abort the particular operation that it is performing. An example of such an error is a write request to a non-existent remote SAP. The error is reported via the LCB associated with the error, to the layer management process associated with the process, and possibly to the LACS System Management entity.

Errors reported to the SM server are in turn reported to a system management application. The layer server layer management process may record the error as part of statistical count or ignore it. It continues to serve all other requests.

3.1.4.4 Recoverable errors

Recoverable errors are errors which are temporary in nature typically caused by a temporary lack of controller resources (e.g. nak'd IOLD, no free memory etc.). The process (in the DPS-6 or LACS) experiencing the error should retry the operation after a certain time period. In general, the layer managment function should record these in a statistical record. If the error occurs repeatedly then it can be treated as a fatal or non-fatal operation error depending on the operation being performed when the error is detected.

3.1.4.5 Other types of errors

There are two other types of errors which do not fall in the classes described above. These are protocol errors and DPS-6 system errors. The effect of protocol errors such as transport protocol errors are described in the corresponding protocol specification. DPS-6 system errors such as "non-existant resource" error on the megabus when attepting to issue an I/O order, causes traps. The effect of this is the same as for other DPS-6 controllers.

3.1.4.6 Requirements of each Process

Each process on the LACS is required to have two blocks pre-assigned: one to use to send an error event message to system management or to layer management, and one for setting an Alarm (timer) to retry an operation which caused a recoverable error.

Each layer instance in the LACS (including the megabus interface software) must maintain a table of errors that it could possibly have to deal with and the error class it belongs to. This not only modularizes the error handling routine but also makes it possible to change the error class associated with an error. For example, errors treated normally as layer operation errors could be treated as catastrophic errors in debug mode so as to trap the error as soon as possible. Each layer server layer management function must handle any reported errors . This will be used to report layer operation errors which are to be reported to the layer server via System Management. Errors that need to be reported to the layer server in the DPS6 must be coordinated with the layer instance in the LACS. System management will report the error to the layer servers' layer management function when it receives an error event indication with the proper class (This is accomplished by a call to the routine whose address is in the layer instance table). It is the responsibility of the DPS6 layer server layer managemnt function to take the appropriate action. The DPS6 layer server layer management function may also report errors to system management by directly calling system management.

Appendix B gives a list of different error conditions detected and the class of error that they fall under.

3.1.5 Quality of Service

Several primitives defined in the ISO and IEEE 802 Service Definition Standards contain a quality of service parameter. In general quality of service applies to the following characteristics of data transfer:

- transit delay
- throughput of succesfully transfered sdus.
- error rate
- priority
- probability of failure (i.e. of losing the connection or path)

ISO and IEEE 802 allow for a quality of service parameter to be associated with each connection or with each Connectionless Data transfer. How the quality of service parameter is to be used is not clearly defined in ISO or IEEE 802.

Mechanisms for providing quality of service in the DPS-6/LACS environment are FFS. A possible mechanism could be to give each user (SAP) a priority level from 0-7 (0 being the highest priority]. The task levels at which the user's requests are processed by the Driver can then be assigned according to this priority level. Furthermore, in case of LCBs being queued up in the DPS-6 because of the maximum limit being reached on the controller, could be queued according to the user's priority level. The layer instance on the LACS strictly process orders as they are received. Their only responsibility would be to pass on the priority level to the layer below and where applicable in the PDU.

3.2 Service Primitive Definition

This section gives a detailed description of the service primitives. Note that the primitives are described in a conceptual manner. The intent is to explain what parameters are expected from the user to provide a service and the effect of providing the service. The actual format (syntax) of the call and its associated IORB is described in the component specifications. Note also that parameters and status required and returned by the executive are not described here. Whenever a status value other than "successful" is returned it should be assumed that the requested operation was not performed unless where it is specifically specified that the status is a non-fatal status.

3.2.1 SAP Service Primitives

3.2.1.1 Associate User

Description:

The user must issue an Associate_User MCL before it can issue any other request. The main purpose of the call is to assign an lrn to the user and to register ownership of the lrn with the executive. This allows the executive to free up resources in the event of an abnormal abort of the task group associated with the user.

Input Parameters:

Symbolic Name

Output Parameters:

lrn

Status:

successful
Symbolic Name unknown
lrn associated with another task group

Comments:

The lrn represents the logical address of the Local SAP.

3.2.1.2 Activate Local SAP

Description:

This call activates the availability of services through the SAP.

Input Parameters:

lrn
Proposed Maximum SDU size
Proposed CL Receive SDU Credit
Proposed Maximum Age of SDU

Output Parameters:

Type of SAP
Maximum SDU size
Ideal SDU size
CL Read Order Credit
CL Write Credit

Maximum Age of SDU
Maximum number of connections supported

Status:

successful
lack of resources
unable to load lacs
sap already activated

Effect When Generated:

If this is the first call since the system is initialized then all LACS controllers in the system are loaded and initialized. In all cases the appropriate layer entity in the LACS is called to activate the SAP.

The Activate Local SAP call may cause the layer entity to activate the SAP of the entity in the layer below it that is required to provide services through the SAP.

Comments:

The "Type of SAP" output parameter specifies the layer and types of services [CL, CO or both] available to the user through this SAP.

The "maximum SDU size" output parameter is the smaller value between the maximum size that is proposed by the user and the maximum size that can be supported by the layer providing the service. If the user parameter is null then the value supported by the layer is used. The user data buffers passed in any "Write" call cannot exceed this size. It is also the maximum size of SDU that the user can expect to receive. The layer entity will discard any SDU that it receives which is larger than this size.

See section 3.1.5 on flow control for an explanation of the CL Receive SDU Credit and the Read order credit parameters. If the proposed Receive SDU credit is zero then the maximum Read Credit value set up for the SAP is used. A zero read order credit output parameter indicates that CL services are not provided.

The "ideal SDU size" is the ideal SDU size for optimum performance.

The Maximum Age of the SDU is the time a CL SDU is buffered in the LACS awaiting a read order from the DPS-6. It is expressed as the number of 10 msec intervals to hold the SDU. The value returned is the smaller of the number proposed and the number setup for the SAP.

3.2.1.3 Activate Remote SAP

Description:

This call returns the logical address of the Remote SAP to the user. The address is represented by a non-zero number. This address must be used by the user in subsequent requests that require the Remote SAP address.

Input Parameters:

lrn
Symbolic Name of Remote SAP

Output Parameters:

Logical Remote SAP address

Status:

Successful
Symbolic Name Unknown
cannot access Remote Sap through this local SAP

Effect When Generated:

The Driver obtains the logical address associated with the symbolic name of the Remote user from the layer management entity [LME] of the corresponding layer instance and returns it to the user. The symbolic name to address mapping has to be set up currently by configuration. In future it should be possible for the LME to request System Management to access this information via a remote directory service[6]. This call may cause the layer instance to call the layer below it to activate the remote SAP address to which this remote SAP maps to.

Comments:

In order to standardize the interface the returned address is a fixed size logical address (4 octets, right justified, zero filled). The ISO Network and Transport protocols allow for variable size addresses. The LACS layer instance will perform the necessary mapping.

3.2.1.4 Deactivate Local SAP

Description:

This call can be issued by the user when it wishes to terminate all activities through the specified SAP. It is also issued by the executive if the user task group aborted and the lrn associated with the SAP is still registered with the executive.

Input Parameters:

lrn

Output Parameters:

None

Status:

successful
SAP not active

----

6. This may be a central directory service or a distibuted on where the symbolic name is broadcast over the LAN and the system owning the SAP responding with the actual address

Effect When Generated:

This call causes all activity on the SAP to terminate. All active connections are terminated and pending request blocks if any are returned. The SAP goes into an "inactive" state. All remote SAPs associated with this local SAP are also de-activated.

Comments:

None

3.2.1.5 Deactivate Remote SAP

Description:

This call must be issued by the user when it wishes to terminate all activities to the specified remote SAP.

Input Parameters:

lrn
remote SAP logical address

Output Parameters:

None

Status:

successful
SAP not active

Effect When Generated:

This call causes all activity to the remote SAP to terminate. All active connections are terminated and pending request blocks if any are returned. The association between the local and remote SAP is terminated. If the logical address for the remote SAP was assigned dynamically it can be re-used for assignment to another remote SAP that attempts to initiate a conversation and that is not known to [configured by] the system.

Comments:

None

3.2.1.6 SAP Event Indication

Description:

The user must issue this primitive to enable the Driver [and the LACS] to inform the user of the events the user wishes to receive notification of. Only one Event Indication request should be outstanding at any given time.

Input Parameters:

lrn
Event Request Mask
Buffer [for connect Indication user data]

- Data Arrival
- Additional Write credit available
- SAP Deactivated
- Connect Indication

Output Parameters:

Event Indication Mask
Event Info

- Length of data [If Data Arrival Event]
- Amount of additional credit [If Additional Wr. Credit Event]
- SAP Deactivated Reason Code [If SAP Deactivated Event]

- [If Event Indication is Connect Indication]
  * connection identifier
  * logical remote SAP address
  * remote SAP dynamically Activated
  * expedited data option
  * Quality of service
  * Write Credit
  * Buffer [containing user data [max 32 octets]]

Status:

SAP not active

Effect When Generated:

The user specifies the events it wishes to be notified of via the Event Request Mask. If the requested event occurs [or had occured and was unreported], the request is returned with the corresponding flag set in the Indication Mask. Only one event is reported at a time. If the User issues a request while one is outstanding the old request is posted back with no indication. The Event Mask in the new Request becomes effective.

If the Data Arrival flag is set then this primitive is returned if data arrives for this SAP and there is no pending Read CL Data Request. The user is only notified once about the arrival of an SDU.

The Write Credit available Event is posted when the user's credit had gone down to zero [and hence could not issue any write requests] and now additional credit is available but there are no outstanding write calls through which this fact can be reported.

The SAP Deactivated event is returned if the SAP is deactivated for internal reasons.

The Connect Indication Event is returned when a Connect Request PDU is received from a Remote SAP. Connection parameters are passed in the Event Info fields. The connection identifier passes by the layer instance in the LACS is modified before being passed to the user so that the Driver can determine when subsequent calls are made as to which controller supports this connection. The "Remote SAP dynamically activated" parameter acts as a flag to the user that the a new logical address is being passed. The connect request was received from a Remote SAP that was not known to the LACS and a new entry for it was created. The user must respond to the connect indication with a Connect Response or a Disconnect Request order to indicate if it wishes to accept or reject the remote Connect request.

Comments:

3.2.2 Connectionless Data Transfer Service Primitives

3.2.2.1 Read Connectionless Data

Description:

The Read Connectionless Data primitive allows the user to read any Connectionless sdu destined for the specified SAP.

Input Parameters:

lrn
buffer list [to receive data]

Output Parameters:

buffer list [containing data]
residual range of each buffer
actual total size of data
remote SAP [source] address
remote SAP dynamically Activated
Quality of Service

Status:

successful
insufficient buffer space
SAP deactivated
Max # of pending Reads exceeded

Effect When Generated:

The request is passed on to the LACS (if not limited by flow control). If there is any data available for the user and this is the only outstanding request then the data is passed to the user in the specified buffers. Otherwise the request is queued. When data is available the first request on the queue for the SAP is used to process the data.

Comments:

The "Remote SAP dynamically activated" parameter acts as a flag to the user that the a new logical address is being passed. The SDU was received from a Remote SAP that was not known to the LACS and a new entry for it was created.

The LACS driver allows the user to specify a list of buffers into which data is to be moved. The LACS fills the first buffer on the list completely before using the second. It is possible for more than one buffer to not be filled completely if the sdu is smaller than the total space provided. A residual range is returned for each buffer in the list.

If the sdu is larger than the available space in the buffers then no data is moved and the order is returned with an "insufficient buffer space" status. If another read order is queued and this too does not have sufficient space, then it is also returned with the same status. This is continued till a read order with sufficient buffer space is issued. The actual size of the sdu parameter allows the user to issue another request with sufficient buffer space.

The SAP Event Indication primitive allows the user to be notified of arrival of data without having the resources (buffers) of a Read CL Data committed. Having a Read CL Data outstanding, however, speeds up the reception of data (avoids the turn-around time of receiving the Event Indication and then issuing the Read).

The SDU age is reset if a SAP [data arrival] Event primitive is returned or a Read Data primitive is returned with an "insufficient data space" status. This gives the user more time to issue the appropriate read order to receive the data.

3.2.2.2 Write Connectionless Data

Description:

This primitive is used to transmit a connectionless SDU to a remote SAP.

Input Parameters:

lrn
logical Remote SAP Address
Buffer List [of buffers containing SDU]
Service Class

Output Parameters:

Additional Write credits allocated

Status:

successful
Local SAP not active
insufficient LACS resources
Write Credits exceeded
Remote SAP unknown or not activated

Effect When Generated:

The LACS Driver passes the request to the LACS [if not limited by flow control considerations]. The LACS posts back the request as completed as soon as the data is copied into the LACS.

Comments:

3.2.3 Connection Oriented Data Transfer Service Primitives

3.2.3.1 Connect Request

Description:

This primitive is used when the Local User wishes to initate a connection to a remote SAP.

Input Parameters:

lrn
logical remote SAP address
expedited data option
quality of service
Proposed Maximum SDU size
Proposed CO Receive SDU Credit
buffer (containing user data)

Output Parameters:

connection identifier
responding address
expedited data option
quality of service
Maximum SDU size
Ideal maximum SDU size
CO Read Order Credit
CO Write Credit
buffer (containing remote user-data)

Status:

succesful
Local SAP not active
Remote SAP unknown or not activated
Disconnect Indication

Effect When Generated:

The LACS driver passes the request to the layer instance supporting the local SAP (specified by the lrn). If the lrn is associated with a Group SAP (i.e. more than one controller supports this SAP) then the Driver picks the controller through which the remote SAP can be accessed. If more than one, then the controller with the fewest connections is chosen. The layer instance in the LACS does not return the LCB until a Connect Confirm or Disconnect Indication PDU is received for this connection. The connection identifier passes by the layer instance in the LACS is modified before being passed to the user so that the Driver can determine, when subsequent calls are made, as to which controller supports this connection.

Comments:

User Data is limited to 32 octets.

The Receive SDU Credit input parameter is used to determine the amount of initial credit to assign the remote transport entity for this connection. The amount of credit assigned by the remote transport for its receive flow is used to determine the amount of Write Credit returned to the user as an output parameter.

The negotiated maximum PDU size is used to determine the ideal maximum SDU size. SDUs smaller than this size will not be segmented. The DPS-6 server and the layer instance in the LACS are capable of supporting SDUs upto the Maximum SDU size returned as a output parameter.

3.2.3.2 Connect Response

Description:

This call is made in response to a previously returned (Connect Indication) SAP Event call if the user wishes to confirm the Connect Request from the remote SAP.

Input Parameters:

lrn
connection identifier
expedited data option
Proposed Maximum SDU size
Proposed CO Receive SDU Credit
buffer [containing user data]

Output Parameters:

expedited data option
quality of service
Maximum SDU size
Ideal maximum SDU size
CO Read Order Credit
CO Write Credit

Status:

succesful
invalid connection identifier [connection terminated]

Effect When Generated:

The Driver passes the request to the layer instance in the controller that is supporting the connection. The layer instance sends a Connect Confirm PDU to the remote SAP. Credit parameters are used in the same way as for Connect Request call. Negotiated parameters are returned as output parameters.

Comments:

Comments made for the Connect Request order apply for the connect response also.

It is possible for the remote transport entity to timeout and issue a disconnect request before the user issues a connect response. The invalid connection identifier status is returned in this case[7].

3.2.3.3 Read CO Data

Description:

This call is issued to receive normal data received by the layer instance for this connection.

Input Parameters:

lrn
connection identifier
buffer list

Output Parameters:

buffer list (containing data)
residual range of each buffer
actual total size of data

Status:

successful
insufficient buffer space
invalid connection identifier (connection disconnected)
SAP deactivated
Max # of pending Reads exceeded

Effect When Generated:

----

7. Once a connection identifier is freed up (because the connection is terminated) it should not be re-assigned for a reasonable length of time. This would reduce the chances of not detecting errors caused by a user issuing orders on an already disconnected connection.

Same as for Read CL Data except that the order is for the connection.

Comments:

Same as for Read CL Data except that they apply to the connection rather than the SAP.

3.2.3.4 Read Expedited CO Data

Description:

This call is issued to receive expedited data received by the layer entity for this connection. It is valid only if this option is supported by the connection. If supported, the user should have one Read Expedited CO Data outstanding for the length of the connection. The data arrival event does not apply to expedited data.

Input Parameters:

lrn
connection identifier
buffer

Output Parameters:

buffer [containing data. Maximum 16 octets]
residual range of buffer

Status:

successful
insufficient buffer space
invalid connection identifier [connection disconnected]
expedited option not supported
SAP deactivated
Max # of pending Expedited Reads exceeded

Effect When Generated:

Same as for Read CO Data.

Comments:

It is not necessary to return the actual size of the data since the maximum Expedited SDU size is 16 octets. The user must allocate at least 16 octets of buffer space. If not the order is rejected immediately by the Driver.

3.2.3.5 Write CO Data

Description:

This call is issued to transmit normal SDU on the specified connection.

Input Parameters:

lrn
connection identifier
Buffer List (of buffers containing SDU)

Output Parameters:

Additional Write Credit

Status:

successful
invalid connection identifier (connection disconnected)
SAP deactivated
Write Credit exceeded

Effect When Generated:

The Driver passes the request to the layer instance in the controller supporting the connection (if not limited by flow control restrictions). If the SDU has to be segmented then the layer instance transfers only that much of the SDU as can fit in the number of PDUs that it has credit to send. The order is posted back as soon as the complete SDU is transferred to the LACS.

Comments:

The amount of additional credit to give the user is determined by the amount of credit available on the connection.

3.2.3.6 Write Expedited CO Data

Description:

This call is issued to transmit expedited SDU on the specified connection. It is valid only if this option is supported by the connection.

Input Parameters:

lrn
connection identifer
expedited data buffer (maximum 16 bytes)

Output Parameters:

Additional Expedited Write Credit

Status:

successful
invalid connection identifier (connection disconnected)
expedited option not supported
SAP deactivated
Expedited Write Credit exceeded

Effect When Generated:

The driver passes the request to the layer instance in the LACS. Normally the layer instance returns the order with an additional credit of one as soon as the data is transfered to the

LACS. However, if a subsequent order is issued before the previous expedited PDU is acknowledged then it is held in the LACS until the previous order is acknowledged (i.e. the LACS will only hold two expedited PDUs per connection at any given time). This scheme ensures that the user's expedited data credit does not go down to zero.

Comments:

None

3.2.3.7 Connection Event Indication

Description:

The user must issue this primitive to enable the Driver (and the LACS) to inform the user of the events on the connection that the user wishes to receive notification. Only one Event Indication request should be outstanding at any given time.

Input Parameters:

lrn
connection identifier
Event Request Mask

- Normal Data Arrival
- Additional Normal Data Write credit available
- Disconnect Indication

Output Parameters:

Event Indication Mask
Event Info

- Length of Data arrived
- Amount of additional credit

- Reason code for disconnect

Status:

None

Effect When Generated:

The user specifies the events it wishes to be notified of via the Event Request Mask. If the requested event occurs (or had occured and was unreported), the request is returned with the corresponding flag set in the Indication Mask. If the User issues a request while one is outstanding the old request is posted back with no indication. The Event Mask in the new Request becomes effective.

If the Data Arrival Flag is set then this primitive is returned if data arrives for this connection and there is no pending Read CO Data Request. The user is only notified once about the arrival of an SDU.

The Write Credit available Event is posted when the user's credit for normal CO data write has gone down to zero (and hence could not issue any write requests) and now additional credit is available but there are no outstanding write calls through which this fact can be reported.

Comments:

3.2.3.8 Disconnect Request

Description:

This call is issued when the user wishes to terminate the connection. This call causes any data normal or expedited that is queued by either transport entity to be discarded.

Input Parameters:

lrn
connection identier

Output Parameters:

None

Status:

succesful
invalid connection identifier (already disconnected)
disconnected locally but not acknowledged by remote

Effect When Generated:

The Driver passes the request to the layer instance in the LACS. A Disconnect Request PDU is issued to the remote transport. The order is not returned until the request is confirmed or a timeout occurs.

Comments:

None

3.2.4 Management Service Primitives

3.2.4.1 Management Request

Description:

This call is issued when the SM service user wishes to perform an operation on an object or class of objects under the System Manager's domain.

Input Parameters:

lrn
buffer list

Output Parameters:

buffer list

Status:

TBW

Effect When Generated:

The buffers pointed to by the buffer list contain request information. The request information includes the layer, sublayer and layer internal selector on which the operation is to be performed and the operation code. Operations supported are Get, Set and Action. The information is encoded in IEEE 802.1 PDU syntax notation. See the System Management component specification for a description of the PDU format.

The buffers are also used to return the response to the request.

Comments:

3.2.4.2 Management Event Indication

Description:

This call is issued by the SM user to provide a mechanism for SM to return Event Indication information to the user.

Input Parameters:

lrn
buffer list

Output Parameters:

buffer list

Status:

TBW

Effect When Generated:

TBW

Comments:

3.3 SERVICES PROVIDED BY LAYER INSTANCES

This section describes the mechanisms through which services are provided by each layer instance.

Each layer instance must provide SAP and data transfer services to the layer above it and obtain these services from the layer below it. In addition, the Layer Management Entity within each layer instance must provide layer management services to the System Management layer instance in the LACS. Each layer instance must, therefore, be capable of providing services to the LACS Driver in the DPS-6 as well as other layer instances in the LACS. While the information exchanged is the same, the manner in which information is exchanged depends on whether the service requestor is in the LACS or DPS-6.

Information Exchange Mechanisms when Requestor is in the LACS

When the service requestor is another layer instance in the LACS, information regarding the service is passed as messages.

Parameters for Request primitives are passed as a message from the service requestor to the service provider. If the request requires a confirmation (confirm primitive), as is the case for a connect.request, then the block used for the request message block is used to return the confirmation. If the request does not require a confirmation, as is the case for a data.request, then the service provider is responsible for releasing the resources (data buffers etc.) associated with the request or passing them on to the layer below.

Parameters for Indication type primitives such as incoming data.indication are passed as a message from the service provider to the requestor. The service requestor is responsible for releasing the resources (data buffers etc.) associated with the indication or passing them on to the layer above.

Information_Exchange_Mechanisms_when_Requestor_is_in_the DPS-6

When the service requestor is in the DPS-6, information regarding the service is passesd in a LAN Control Block [LCB]. The LACS Driver creates the LCB in DPS-6 RAM area and passes a pointer to it by issuing an "Output LCB Pointer" IOLD command [FC = 09/0D] across the megabus.

Parameters for Request primitives issued by the service requestor are mapped by the Driver into LCB parameters and passed via an IOLD command to the service provider. The layer instance providing the service must transfer the LCB into LACS RAM in order to read all the parameters. If the request involves data transfer then the data also must be transfered to LACS RAM. If the request requires a confirmation [confirm primitive], as is the case for a connect.request, then the LCB is used to return the confirmation. Confirmation parameters are DMAed from the LACS to the DPS-6 LCB area and an interrupt is issued to indicate completion of the request. If the request does not require a confirmation, as is the case for a data.request, then the LCB is released [i.e. a completion interrupt issued] as soon as all the pertinent information is transferred to the LACS.

Because of the nature of the megabus interface, the Driver must also issue LCBs on behalf of the service requestor in order to provide a mechanism for the layer instance to pass parameters for Indication type primitives. If the Indication requires a response, as is the case for a connect.indication, then the service requestor must issue a response call which gets transformed into another LCB call. If the Indication does not require a response, as is the case for a data.indication, then the service provider releases the resources [data buffers etc.] associated with the indication [and updates acknowledgement counters if necessary] as soon as the necessary information is transfered to the LCB and a completion interrupt is issued.

3.3.0.1 IOLD Command Format

The IOLD command contains a 32 bit "Address" word, a 16 bit "Range" word, and a 10 bit "Channel Number".

"Address" word format

The "Address" word contains the byte address of the start of the LCB area in the DPS-6.

"Range" word format

The format of the "range" word is as shown below:

```
Bits     _0_________3_4_________7_8______________________15_
         ! Fun Code  ! RFU        ! LCB length in bytes     !
         |-----------|------------|-------------------------|
```

where:

Fun Code = The type of function requested through this LCB.

The format of the "Channel Number" is as shown below:

```
Bits     _0_________3_4_______6_7_______9_
         ! CTL ADDR  ! LAYER # !LYR INS # !
         |-----------|---------|----------|
```

The IOLD Dispatcher Interface Software routine that receives the interrupt for the IOLD, uses the Layer and Layer Instance numbers in the Channel Number to find the mailbox table associated with the layer instance. It then uses the Function Code in the range word to dispatch the IOLD information to the appropriate mailbox in the layer instance.

3.3.0.2 LCB Format

The general format of the LCB is shown below in Fig. 3.3.1. For more details see the data structure component specification.

Interrupt Control Word
Function Code
Indicators
Buffer Descriptor Area
Function Specific Input/Output Parameters
Controller Status
Function Specific Status
Completion Word

Fig. 3.3.1 LAN Control Block

Parameter Description

Interrupt Control Word:

This word contains the CPU number and interrupt level of the CPU to be interrupted on completion of this request.

Bits 0-7 : RSU and MBZ (The LCB image in LAC RAM will contain the 6 LSBs of the channel number in bits 2-7 of this word).

Bits 8-9 : CPU number.

Bits 10-15 : Interrupt Level

Function Code Word:

This word specifies the function to be performed.

Indicators:

The following indicators are defined.

- Buffer descriptor area contains the description of DPS-6 data buffers (address, range etc.). This is used when one to four buffers are used to pass data.

- Buffer descriptor area contains a pointer to a buffer descriptor. This is used when more than four buffers are used.

- Buffer descriptor area contains data. This may be used when the sdu size is small (upto 24 octets?). This avoids the overhead of doing two DMA transfers: of the LCB and Data.

Buffer Descriptor Area:

The contents of this area depends on the indicators field as described above.

Function Specific Input/Output Parameters:

The Function specific parameters vary for each function within the major function code types specified in the Range word. The function specific input and output parameters are described in the corresponding component specifications.

Controller Status:

This word specifies status related to the controller. This status may be returned for any function. The format of this status word is as follows:

RFU MBZ (8 bits)	INV FC	MEM EXH	RAMNE	RAMP	MY	NEM	LSB	MR

where:

INV FC = Invalid Function Code
MEM EXH = RAM Memory Exhaust
RAMNE = RAM location non-existent
RAMP = RAM Parity error
MY = L6 Memory Yellow
NEM = Non-existent L6 memory
LSB = L6 Bus parity error
MR = L6 Memory Red

Except for the MY status which is a warning all other errors are fatal for the operation, i.e. the action requested was not completed.

MEM EXH error occurs when the I/O Dispatcher does not have any message blocks available for this channel to dispatch the LCB pointer from the IOLD order to the appropriate process.

INV FC status is returned if the I/O Dispatcher routine cannot find a mailbox to send the order to or if the function specific code in the LCB is invalid.

All other errors may occur while doing DMA transfers from or to L6 memory of the LCB or data buffers.

Function Specific Status:

The interpretation of these status codes is given in the description of each primitive. The general format of this word is: the left byte specifies the status code and the right byte the reason for this status if one exists.

Completion Word:

This is the last word written by DMA Controller at completion of the LCB.

Bit 0 : Completion Bit

Bits 1-15 : RFU

Appendix A

OBJECT ATTRIBUTES AND STATES

A.1 OBJECT ATTRIBUTES

LAN ATTRIBUTES

Attributes are those parameters and variables which are used to control and describe the various functions of each layer. There are two types of attributes which can be defined, those that are static and those which can be changed dynamically. Static parameters are read only parameters. They can be set only through CLM directives which are executed at system initialization. Dynamic attributes can be changed due to internal conditions in each layer or through System Management set requests to a layer management entity. At the initalization of the system dynamic parameters will be set to default values or to a set of values defined by the application administrator in the LAN management configuration file. Parameters are described using X409 syntax. Each parameter is classed according to whether it is readable or writable. Default values are assigned to parametrers which are required for create requests.

SYSTEM MANAGEMENT ATTRIBUTES

Static Parameters defined in CLM directives

SM MGR NAME [SS] Implicit IA5 string

Name of the LAN Manager. Must be 8 ASCII characters. Under ordinary conditions there should only be one system manager interface. During testing operations, the T&V routines will require a dedicated interface with a unique name, T&VMGR.

Static Parameters defined by LACS Hardware or loaded Bound Units

ManufacturerId [0] Implicit IAS string
Manufacturer name and country. The manufacturer name consists of the character string 'Honeywell Informations Systems,USA'.

StationTypeId [1] Implicit Octet string
Manufacturer specific station type. TBS

OPT SUPP [99] Impicit Integer

2 octets.Options supported. Null for initial implementation. Default value of zero.

Dynamic Parameters defined in LAN Manager Configuration file or events occurring during LACS operation.

LMGR STATE [99] Implicit Sequence [[

[0] State — Implicit Octet string
1 octet. State of System Manager. Default value set to indicate locked - 03. At CLM set to LOCKED.

[1] Substate — Implicit Octet string
1 octet. Null for initial implementation. Default value of zero. At CLM set to RESET

]]

MAX LCB [99] Implicit Integer

16 bit integer.Maximum number of LCBs allowed for LAN manager. Default value of 99.

CONTROLLER ATTRIBUTES

The LACS board is described by a set of parameters located in the LAN Control Table. The complete set of these parameters in the LAN Control Table comprise the Controller object. In addition to parameters describing the controller, the controller object will contain parameters describing the station management as defined in IEEE 802. 1. The following attributes will be defined for each controller :

Controller attributes

Static Parameters defined in CLM directives

CT NAME [SS] Implicit IA5 string

8 octets. Name of this controller. The name of the controller is based on the megabus slot it located in, LANCT0 - LANCTF. Read only.

CT ADDR [SS] Implicit Integer

32 bit integer. Megabus address of this controller. Read only.

Static Parameters defined by LACS Hardware or loaded Bound Units

HW REV [SS] Implicit Octet string

2 octets. Hardware revision for this controller.Identifies the LACS board. Read only.

FW REV [SS] Implicit octet string

2 octets. Firmware revision for this controller. Not initially supported. Read only

SW REV [SS] Implicit octet string

2 octets. Software revision for this controller.Not initially supported. Read only

PRAM SZ [SS] Implicit Integer

16 bit integer. Size of available procedure

RAM in controller. Read only

DBUF SZ [99] Implicit integer

16 bit integer. Size of available data buffer RAM in controller. Read only

Dynamic Parameters defined in LAN Manager Configuration file or events occurring during LACS operation.

CT STATE [99] Implicit sequence {

Present state of controller.

[0] State	Implicit Octet string 1 octet. State of Controller. Default value set to indicate locked - 03.	
[1] Substate	Implicit Octet string 1 octet. Default set to the reset state - 00..	

}

MAX LCB [99] Implicit integer

16 bit integer. Maximum number of LCBs allowed for controller. Default value set to 99.

MAXMIO [99] Implicit integer
16 bit integer. Maximum number ofretries to do io or iold instructions on the megabus. Default value set to 4.

STATISTICS

CT TBI [99] Implicit integer

16 bit integer. Total number of LCBs that have been issued to the controller.

CT TBN [99] Implicit integer

16 bit integer. Total number of LCBs that have been NAK'D by the controller.

CT TBF [99] Implicit integer

16 bit integer. Total number of LCBs that have been queued by the controller because of flow control reasons.

PHYSICAL LINE ATTRIBUTES

The adapter board on each controller is described by a set of parameters located in the LAN Adapter Table. The complete set of these parameters is the LAN Adapter Table comprise the Physical Line object. The Physical Line object is a DSA concept which encompasses aspects of both the MAC and Physical layers in the IEEE 802 standard. In order to remain as faithful as possible to? both DSA and 802, the attributes of the Physical Line have been partitioned into MAC attributes and Physical layer attributes. The attributes of the Physical Line describe attributes which are unique to a particular protocol as well as attributes common to more than one protocol. The following parameters describe the attributes of the MAC and Physical layers :

L6 Dependent Attributes

Static Parameters defined due to LACS hardware or in Configuration File directives

ADAPT NAME [99] Implicit octetstring

8 octets. Name of the adapter. Name is assigned during CLM based on the controller, the adapter is located and its position assigned on the controller. The name is of the form, ADPT00 - ADPT03 through ADPTF0 - ADPTF3. Read only.

2 octets. Layer instance identifier. It is a value related to the index of the layer instance table pointer in the layer table. The value is between 0 and 7. Read only

HW ID [99] Implicit Octetstring

2 octets. Hardware identifier of adapter. Read only. TBS

CSMA [99]

ADAPT FWREV [99] Implicit octetstring

2 octets. Adapter firmware revision number. Not initially supported. Read only.

Dynamic Parameters defined in LAN Manager Configuration file or events occurring during LACS operation.

ADAPTER STATE [99] Implicit sequence {

State of adapter, summary of MAC and Physical. Initialized at CLM to LOCKED and RESET.

[0] State Implicit Octet string
1 octet. State of the Adapter. Initial value set to indicate locked - 03.
[1] Substate Implicit Octet string
1 octet. Initially set to the reset state - 01..

}

MAC Layer Attributes

Static Parameters defined in Configuration File directives

MAC ADDR [99] Implicit integer

48 bit integer. MAC address. MAC addresses only 16 bits long will be right justified and a 32 MSB zeroed. Default value will be zero (if that makes any sense for MAC layer management). Read only.

MAC ADDLN [99] Implicit integer

16 bit integer. Length of MAC address (16 or

48]. Default will be a value of 48. Read only.

Physical Layer Attributes

Static Parameters defined in CLM directives

MODEM TYPE [SS] Implicit octetstring

2 octets. Type of modem for this adapter. Not initially supported.

Dynamic Parameters defined by events occurring during LACS operation. They are all initialized to zero.

PCEPD
TX FRM [SS] Implicit Integer

16 bit integer. Number of frames sent correctly (not aborted due to errors). Read only.

PCEDA [SS] Implicit Integer
16 bit integer. Number of octets sent.. Read only.

PCRPD
RCV FRM [SS] Implicit Integer

16 bit integer. Number of frames received correctly.. Read only.

PCRDA [SS] Implicit Integer
16 bit integer. Number of octets received.. Read only.

PCROR
OVR ERR [SS] Implicit Integer

16 bit integer. Number of overruns.. Read only.

PCRFC
FCS ERR [SS] Implicit Integer

16 bit integer. Number of FCS, checksum or parity errors.?. Read only.

PCRAB [99] Implicit Integer
16 bit integer. Number of abort sequences received.. Read only.

PCXHW [99] Implicit Integer
16 bit integer. Number of hardware errors signalled by the controller.: Read only.

PCRER [99] Implicit Integer
16 bit integer. Number of frames received incorrectly (total of below errors).. Read only.

PCEER [99] Implicit Integer
16 bit integer. Number of frames sent incorrectly (aborted due to error). Read only.

PCRAL
ALN ERR [99] Implicit Integer

16 bit integer. Number of misaligned packets discarded due to CRC errors.. Read only.

PCRSC
F ERR [99] Implicit Integer

16 bit integer. Number of correct packets discarded due to lack of resources. Read only.

PCRLN [99] Implicit Integer
16 bit integer. Number of received PDUs discarded due to length error.. Read only.

PCEBY [99] Implicit Integer
16 bit integer. Number of transmit defferals due to medium busy.. Read only.

PCEST [99] Implicit Integer
16 bit integer. Accumulated number of

transmit defferals due to medium busy. Read only.

PCECO [99] Implicit Integer
16 bit integer. Number of send aborted due to collision detection on the cable.. Read only.

PCRCL [99] Implicit Integer
16 bit integer. Number of collisions spoiling reception.

LOCAL LOGICAL LINE ATTRIBUTES

The Logical Line represents an interface to the physical line. A Logical Line in a DPS6 is represented locally by a logical resource number (LRN). A remote Logical Line is represented by a sub-LRN. The 802 concept of a Logical Line is represented by a Link Service Access Point (LSAP), which can be remote or local. The following set of parameters describe the attributes associated with a local LSAP or Logical Line object :?

Static Parameters defined in CLM directives

L6/LACS Dependent Attributes

LSAP NAME [99] Implicit IAS string

8 octets. Name of LSAP. A LSAP name is assigned through a create LSAP operation. Read only.

LSAP LRN [99] Implicit integer

8 bit integer. LRN associated with LSAP. Read only.

SAPLAY [99] Implicit integer
8 bit integer. Layer the SAP is defined in. Value of 0-7. Read only.

LAYINS [99] Implicit integer
8 bit integer. Layer instance the SAP is defined in. Value of 0-7. Read only.

IEEE802 Defined Attributes

LSAP ADDR [99] Implicit integer

8 bit integer. LSAP Address. Read only.

Static Parameters defined by LACS Hardware or loaded Bound Units

LLC SW REV [99] Implicit octetstring

2 octets. LLC software revision number. Read only. Not initially supported.

Dynamic Parameters defined in LAN Manager Configuration file or events occurring during LACS operation.

L6/LACS Dependent Attributes

LSAP STATE [99] Implicit sequence {

- Present state of the LSAP.

[0] State — Implicit Octet string
1 octet. State of the LSAP. Default value set to indicate locked - 03.

[1] Substate — Implicit Octet string
1 octet. Default set to the reset state - 00..

}

LOGADD [99] Implicit integer

16 bit integer. Initial value is zero until activate local SAP request, LACS LLC LME function determines logical address while activating SAP.

MAX TX BUF BYTES [99] Implicit integer

16 bit integer. Maximum number of bytes of data buffer allowed for this LSAP on transmit operations. Default value set to 99.

MAX RCV BUF BYTES [99] Implicit integer

16 bit integer. Maximum number of

bytes of data buffer allowed for this LSAP on receive operations. Default value set to SS.

IEEE 802 Defined Attributes

LSAP TYPE VECTOR [SS] Implicit bitstring

8 bits. Supported service type. Initially only type one service is to be supported.

Bit

- 01 Type I
- 02 Type II
- 03 Type III
- 04 Type IV

MAX LPDU SZ [SS] Implicit integer

16 bit integer. Maximum L_PDU size in bytes. Default value set to SS.

R MXKC [SS] Implicit integer.

Maximum number of connections allowed for this SAP. Default value of zero for connectionless SAPs and default value of 15 fo connection oriented SAPs.

LSAP Statistics:

The following counters are maintained per LSAP for type 1 operations:

R CUKC [SS] Implicit integer

8 bit integer. Current number of connections for this SAP. Read only.

R TCBI [SS] Implicit integer

16 bit integer. Total number of LCBs that have been issued to the SAP. Read only.

C TCBN [SS] Implicit integer

16 bit integer. Total number of LCBs that

have been NAK'D by the SAP. Read only.

C TCBF [S9] Implicit integer

16 bit integer. Total number of LCBs that have been queued by the SAP because of flow control reasons.

LKEPD [S9] Implicit Integer
16 bit integer. Number of unnumbered information protocol data units [PDU] sent. Read only.

LKEDA [S9] Implicit Integer
16 bit integer. Number of information octets sent. Read only.

LKRPD [S9] Implicit Integer
16 bit integer. Number of unnumbered information PDU's received correctly. Read only.

LKRDA [S9] Implicit Integer
16 bit integer. Number of information octets received. Read only.

LKEPK Total number of frames sent [data, control, XID, and TEST]. Read only.

LKRPK Total number of frames received [data, control, XID, and TEST]. Read only.

LKICB [S9] Implicit Integer
16 bit integer. Number of frames with incorrect field contents [bad addr., cmd-code]. Read only.

LKRTS [S9] Implicit Integer
16 bit integer. Number of TEST response PDU's received. Read only.

LKQTS [S9] Implicit Integer
16 bit integer. Number of TEST request PDU's

received. Read only.

LKRID [99] Implicit Integer
16 bit integer. Number of XID response PDU's received. Read only.

LKGID [99] Implicit Integer
16 bit integer. Number of XID response PDU's received.

REMOTE LOGICAL LINE ATTRIBUTES

Static Parameters defined in CLM directives

L6/LACS Dependent Attributes

LSAP NAME [99] Implicit IAS string

8 characters. Name of LSAP. Read only

LOGADD [99] Implicit integer
16 bit integer. Initial value is zero until activate local SAP request, LACS LLC LME function determines logical address while activating SAP.

IEEE 802 Defined Attributes

RLSAP ADDR [99] Implicit integer

54 or 24 bit integer. LSAP Address. Consists of 8 bit remote LSAP address and a remote MACSAP address of either 16 or 48 bits. Read only.

RLSAP ADDRLN [99] Implicit integer

16 bit integer. Length of remote LSAP address. Must be a value of 16 or 48 bits. Read only.

A.2 OBJECT STATES

This Appendix lists the DSA states that are supported for the LAN type objects. The conditions under which these states are reached are also described.

SUPPORTED DSA STATES

	CONTROLLER	SYSTEM MGT. ADMIN.FCT.	PHYSICAL LINE	LOGICAL LINE	LINK CONNECTION
IN-USE	To Be Supported	To Be Supported	To Be Supported	To Be Supported	To Be Supported in Future
ENABLED	For Future Study	For Future Study	For Future Study	For Future Study	For Future Study
DISABLED	Not Applicable	For Future Study	For Future Study	For Future Study	For Future Study
LOCKED	To Be Supported	To Be Supported	To Be Supported	For Future Study	Not Applicable
DOWN	To Be Supported	Not Applicable	To Be Supported	Not Applicable	Not Applicable
TEST	To Be Supported	For Future Study	To Be Supported	For Future Study	Not Applicable
SHUTDOWN	For Future Study	Not Applicable	For Future Study	For Future Study	For Future Study

DSA STATES SUPPORTED ON 802 LAN

* Type I link connections exist for statistical purposes only .?

CONTROLLER OBJECT

IN-USE STATE

DESCRIPTION
The controller is fully operational and able to service user request for controller processes. This requires that the LACS board has been fully initialized [LACS software has been loaded and LAACS kernel and processes are running].

CAUSE OF TRANSITION
The controller object may transition into the IN-USE state from the LOCKED state. It will transition from the LOCKED state due to an operator command or during system configuration.

NEXT STATES

LOCKED	Result of a request by operator.
DOWN	Result of a hardware fault detected on the LACS board.

AFFECTS ON STATES OF OTHER OBJECTS
NONE

ENABLED STATE
For future study.

DISABLED STATE
Not applicable to the controller object.

LOCKED STATE

DESCRIPTION
A controller in the LOCKED state is not available for service by any non-administrative users. While residing in the LOCKED state , the controller can be in one of four substates which correspond to the operational capabilities

of the LACS board.

CAUSE OF TRANSITION
The controller can be transitioned into the LOCKED state from the IN-USE or DOWN states as a result of an operator command. It can also enter the LOCKED state from TEST state under control of the test program.

SUBSTATES

RESET	LACS board hardware and firmware has been reset. LACS firmware functions are operational.
HALT	LACS processes have been halted. LACS hardware and firmware functions have not been reset and LACS firmware functions are operational.
LOADED	LACS board memory has been loaded.
STARTED	The Kernel software and LACS software is executing.
OPERATIONAL	LACS initialization of the kernal and software processes has been completed. In addition all SAPS and LAN data structures have been initialized.

NEXT STATES

IN-USE	Due to operator command or system configuration.
DOWN	Due to detected hardware failure.
TEST	Due to test program.

AFFECTS ON STATES OF OTHER OBJECTS
A controller object transitioning to a LOCKED state will cause all physical lines and logical lines which are mapped to it to also enter the LOCKED state. A controller in a substate other than OPERATIONAL implies that all other objects (upward mapping) are LOCKED or non-existent.

DOWN STATE

DESCRIPTION
A hardware fault has been detected in the controller.

CAUSE OF TRANSITION
LACS board firmware has detected a fault associated with the controller.

NEXT STATES

LOCKED	Due to operator command.
NON-EXISTENCE	

AFFECTS ON STATES OF OTHER OBJECTS

A controller object transitioning to a DOWN state will cause all physical lines and logical lines which are mapped to it to also enter the LOCKED state.

TEST STATE

DESCRIPTION

A controller object in the TEST state is under control of a test program.

CAUSE OF TRANSITION

A test program transitions a controller to the TEST state from the LOCKED state due to a test program request.

NEXT STATES

LOCKED	Due to test program.
DOWN	Due to test program.

AFFECTS ON STATES OF OTHER OBJECTS

Specific to nature of test program.

SHUTDOWN STATE

For future study.

NON-EXISTENCE

The controller object is initially created in the locked state at system initialization.

PHYSICAL LINE OBJECT

IN-USE STATE

DESCRIPTION

A physical line is in the IN-USE state when a LAN adapter has been fully initialized.

CAUSE OF TRANSITION

The physical line can transition into the IN-USE state from the LOCKED state. The transition from the LOCKED state will occur at the successful completion of LACS initialization or due to an operator command.

NEXT STATES

LOCKED	Due to an operator command.

DOWN Due to a detected hardware failure in the LACS adapter.

AFFECTS ON STATES OF OTHER OBJECTS
The physical line can transition into the IN-USE only from the LOCKED state. This does not affect the state of any other object.

ENABLED STATE
For future study.

DISABLED STATE
For future study.

LOCKED STATE

DESCRIPTION
A physical line in the LOCKED state is unavailable for service to higher layer users.

CAUSE OF TRANSITION
The physical line will transition to the LOCKED state from the IN-USE , DOWN , and TEST state due to a command from the operator.

SUBSTATES

RESET	LACS adapter hardware has been reset.
OPERATIONAL	LACS adapter is fully operational.

NEXT STATES

IN-USE	Due to operator command or during system configuration.
DOWN	Due to operator command.
TEST	Due to command from test program.

AFFECTS ON STATES OF OTHER OBJECTS
None. While the physical line is locked all requests from higher layers will be rejected. Only administrative requests will be accepted.

DOWN STATE

DESCRIPTION
The DOWN state of a physical line describes an adapter with which a fault has been detected by LACS firmware.

CAUSE OF TRANSITION
Detection of a fault in the adapter.

NEXT STATES
LOCKED Due to operator command.

AFFECT ON STATES OF OTHER OBJECTS
None. See LOCKED state.

TEST STATE

DESCRIPTION
The physical line in the TEST state indicates that it is unavailable for service and dedicated to a testing operation.

CAUSE OF TRANSITION
The physical line will transition from the LOCKED state to the TEST state under control of test program.

SUBSTATES
Substates to be defined by specific test requirements.

NEXT STATES
LOCKED Due to operator or test program command.
DOWN Due to a command from a test program.

AFFECT ON STATES OF OTHER OBJECTS
Specific to test program.

SHUTDOWN
For future study.

NON-EXISTENCE
The physical line object will be created in LOCKED state at system initialization or due to a Create Physical Line request.

SYSTEM MANAGEMENT ADMINISTRATIVE FUNCTION [SMAF]

IN-USE STATE

DESCRIPTION
A system management administrative function is in the IN-USE state when a LAN adapter has been fully initialized and an activate SAP request has been received from a user of system

management services.

CAUSE OF TRANSITION

The system management administrative function can transition into the IN-USE state from the LOCKED state. The transition from the LOCKED state will occur at the successful completion of LACS initialization or due to an operator command.

NEXT STATES

LOCKED Due to an operator command.

AFFECTS ON STATES OF OTHER OBJECTS

The system management administrative function can transition into the IN-USE only from the LOCKED state. This does not affect the state of any other object.

ENABLED STATE

For future study.

DISABLED STATE

For future study.

LOCKED STATE

DESCRIPTION

A system management administrative function in the LOCKED state is unavailable for service to higher layer users.

CAUSE OF TRANSITION

The system management administrative function will transition to the LOCKED state from the IN-USE , DOWN , and TEST state due to a command from the operator.

NEXT STATES

IN-USE	Due to operator command or during system configuration.
TEST	Due to command from test program.

AFFECTS ON STATES OF OTHER OBJECTS

None. While the system management administrative function is locked all requests from users other than T&V routines will be rejected.

DOWN STATE

Not Applicable

TEST STATE

For future study.

CAUSE OF TRANSITION
The physical connection will transition from the LOCKED state to the TEST state under control of test program.

SUBSTATES
Substates to be defined by specific test requirements.

NEXT STATES

LOCKED	Due to operator or test program command.
DOWN	Due to a command from a test program.

AFFECT ON STATES OF OTHER OBJECTS
Specific to test program

SHUTDOWN
For future study.

NON-EXISIENCE
The physical connection object will be created in LOCKED state at system initialization or due to a Create Physical Line request.

LOGICAL LINE OBJECT

IN-USE STATE

DESCRIPTION
A logical line is IN-USE when an LSAP has been initialized due to an associate monitor call.

CAUSE OF TRANSITION
The logical line will transition to the IN-USE state from the LOCKED state due to an associate monitor call.

NEXT STATES
LOCKED Due to an operator request.

AFFECTS ON STATES OF OTHER OBJECTS
NONE?

ENABLED STATE

For future study.

DISABLED STATE
For future study.
LOCKED STATE

DESCRIPTION
A logical line in the DISABLED state is unavailable for service to the higher layer mapped to the logical line.

CAUSE OF TRANSITION
A logical line will transition to the LOCKED state due to an operator command.

SUBSTATES

RESET	LSAP counters and values are reset.
HALTED	LSAP is 'INACTIVE' but defined.
OPERATIONAL	LSAP is operational.

NEXT STATES

IN-USE	Due to an associate monitor call.

AFFECTS ON STATES OF OTHER OBJECTS
The transitioning of the logical line to the DISABLED state will cause any link connections to become DISABLED.

DOWN STATE
Not applicable to the logical line object.

TEST STATE
For future study.

SHUTDOWN STATE
For future study.

NON-EXISTENCE
The logical line is created in the LOCKED state at system initialization or due to a Create Logical Line request.

LOGICAL LINK OBJECT
For future study.