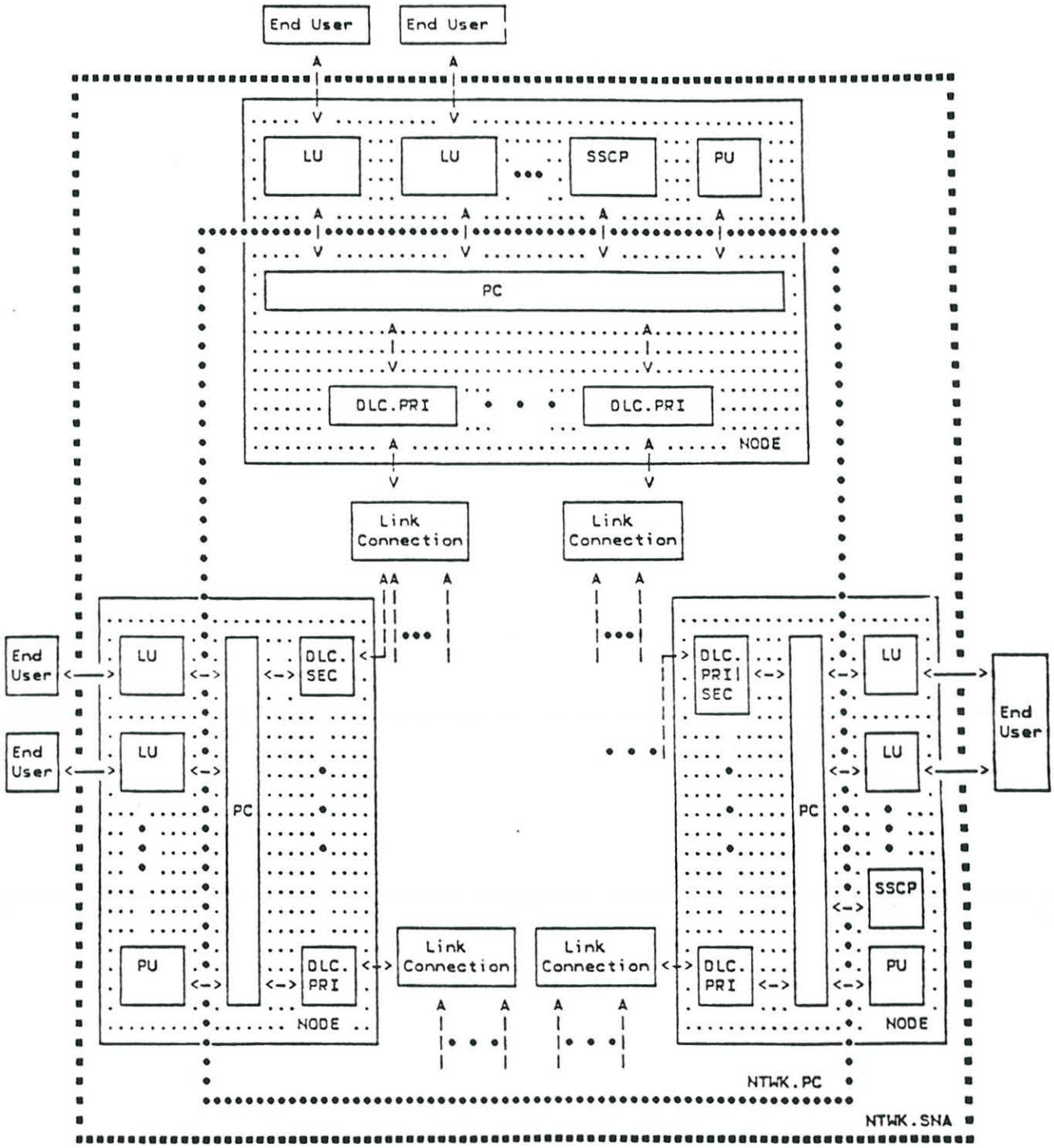
IBM LOGICAL UNIT TYPE 6.2

AN OVERVIEW

V. Decreuse

Rapporteurs: Mr. B.C. Hamshere
Miss M. West

1.0 SNA NETWORK OVERVIEW



1.1 THE OBJECTS AND THEIR BASIC PROPERTIES

The Network Addressable Units (NAU) are logical entities representing the ports through which end users may access the communication facilities.They own a unique name in the naming space and either a unique address or collection of unique addresses in the addressing space of the SNA network.There are tree NAU types:

- The Logical Unit (LU) permits one or many end users to access the network services.The end user interactions rely on relationships established between their underlying LUs.

- The Physical Unit (PU) is the node entity responsible for managing the physical components of the node related to the network,by exchanching informations with an overall Control Point to which is associated an operator either human or programmed.

- Control Point is either a System Services Control Point (SSCP) or a Single Node Control Point (SNCP).

  - SSCP is responsible for the overall management of the physical and logical network resources of a "Domain" being assumed that a SNA network may be spread across several "Domains" each one being ruled by its SSCP.SSCP is aimed at providing domain and networkwide Configuration and Directory services.

  - The purpose of SNCP is to coordinate the node resources by shielding the LUs from the physical configuration of the network,driving the PU in the management of the physical configuration of the node and providing an interface with the node operator. SNCP,as asociated with a particular peripheral node PU,the so called PU Type 2.1 (Formerly PU convergent),is aimed at providing Directory and Configuration services at node level only.

1.2 THE RELATIONSHIPS BETWEEN THE OBJECTS.

The basic logical pipe allowing two entities to speak with each other is referred to as a SESSION.

- There are for session types:

  - The SSCP-PU session by means of which the SSCPs and their subordinate PUs exchange commands and responses in a sollicited or unsollicited mode.Those sessions are usually set up at network activation time.

  - The SSCP-LU session by means of which the SSCPs and their subordinate LUs exchange control commands and responses.Those sessions too are usually set up at network activation time.

  - The LU-LU session permitting two end users to communicate with each other provided that the SSCP-LU sessions have been previously set up.

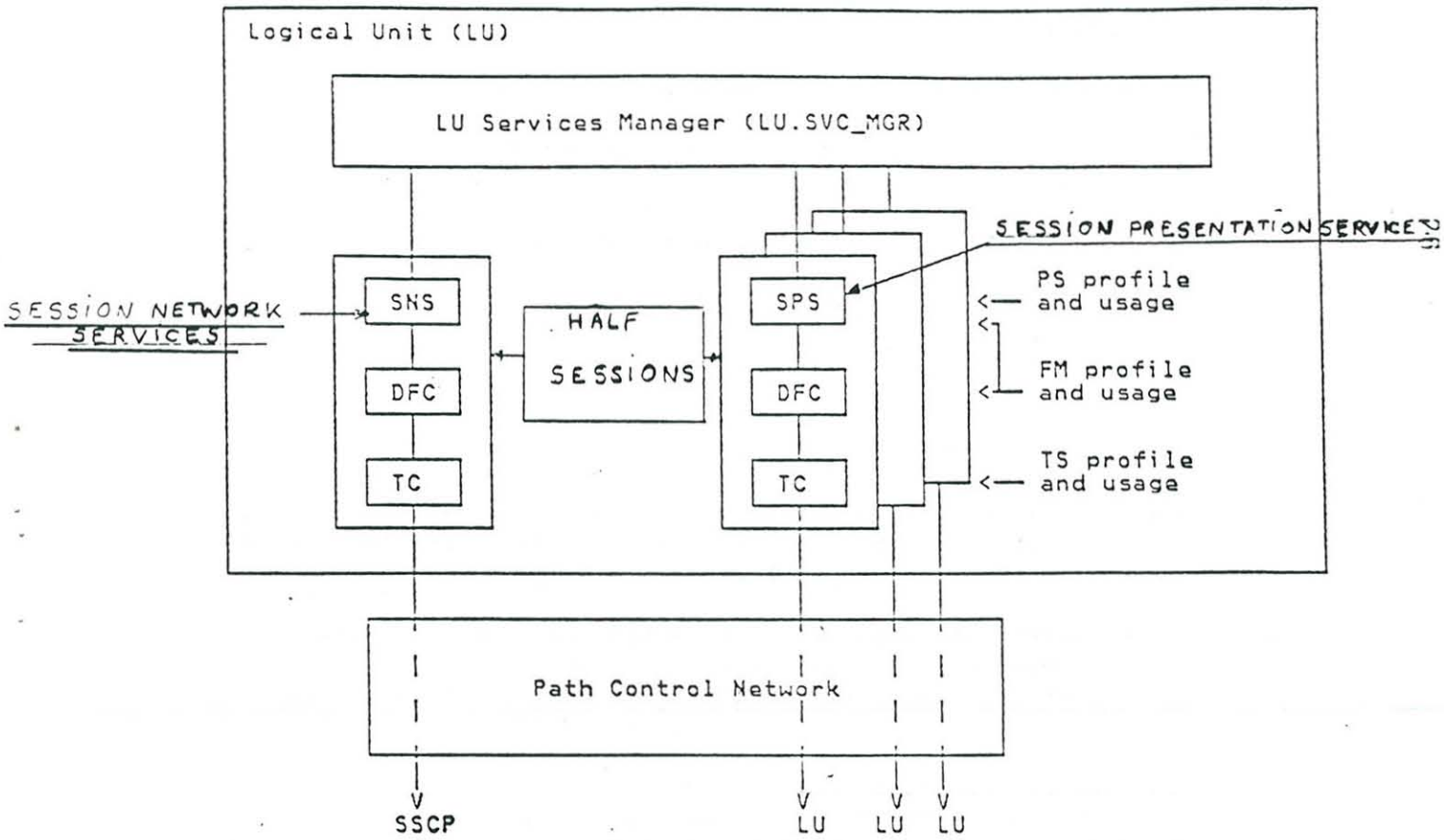
  - The SSCP-SSCP sessions by means of which SSCPs belonging to different domains communicate with each other thereby allowing their ruled LUs to set up the so called "Cross Domain Sessions".

- LU-LU Session set up mechanisms.

  LU-LU session is established between a Primary LU and a secondary LU. The session is set up upon request of either the primary LU (Acquisition mode) or the secondary LU (Acceptation mode) or a third party (Acceptation mode) while asking the involved ruling SSCP(s) to provide Directory and Configuration services. The BIND command which carries information related to the dialogue rules,the end to end data flow control characteristics,the sizes of exchanged messages.....etc is sent by the primary LU to its secondary LU partner.Some session characteristics may be negotiated between the two partners.

  The primary LU is also responsible for managing recovery at session level.

2.0 LU ARCHITECTURE BEFORE LU 6.2 ADVENT.



2.1 LU FUNCTIONAL STRUCTURE.

- LU SERVICE MANAGER SERVICES.

  LU service manager provides network services to the end user:

  - Session Services enable the SSCP and the LU to activate and deactivate LU-LU session.
  - Configuration Services control the network physical configuration.
  - Network Operator Services for activating the LU.
  - Maintenance and Management Services allow a LU and its SSCP to conduct various tests.

  LU service manager provides end user services to the end user:

  - Provides a protocol boundary to the end user.
  - Provides presentation services and synchronization services to the end user.

- HALF SESSION SERVICES.

  - Transmission Control deals with message sizes,end to end data flow control (end to end pacing) according to the chosen Transmission Subsystem profile.(See the BIND command)
  - Data Flow Control deals with request/response mode,HDX/FDX send /receive mode,Bracket and Chaining rules,data flow control techniques and requests according to the chosen Function Management profile.(See the BIND command).
  - Presentation Services deal with character string usage (for instance SNA character string or SCS),code repertoire (EBCDIC,ASCII..) according to the chosen Presentation Service profile.(See the BIND command).

2.2 LU TYPES PRIOR THE ADVENT OF LU 6.2

LU TYPE	PRESENTATION SERVICES CHARACTERISTICS
0	ANY DESIRED OPTION.IS THE MORE VERSATILE AND FLEXIBLE LU TYPE.
1	SCS OR STRUCTURED FIELDS BASED DATA STREAMS.FMH USAGE:NONE OR ONE OR MORE FMH1,FMH2,FMH3.ONE OR MULTIPLE MEDIA SUPPORT.
2	SNA 3270 DATA STREAM.NO FMH.DISPLAY SUPPORT.
3	SNA 3270 DATA STREAM.NO FMH.ONE PRINTER SUPPORT.
4	SCS BASED DATA STREAMS.FMH:NONE OR FMH1 ,FMH2,FMH3.DATA PROCESSING AND WORD PROCESSING MEDIA SUPPORT.USED FOR DATA COMMUNICATION BETWEEN TWO TERMINALS (PEER TO PEER SESSIONS) OR BETWEEN APPL. PROGRAMS AND SINGLE OR MULTIPLE DEVICE TERMINALS.
6	DATA STREAMS ARE USER DEFINED.FMH:FMH4, FMH5,FMH6,FMH7 AND FMH10 ARE OPTIONAL. FUNCTION SHIPPING,ASYNCHRONOUS PROCESS., DISTRIBUTED TRANSACTION PROCESSING IN DIPE ENVIRONMENT.

3.0 FROM THE TRANSACTION TO THE CONVERSATION.

3.1 THE TRANSACTION.

- SYSTEM TRANSACTION VS USER TRANSACTION

  - The system transaction is the basic interaction entity between a terminal and a processing system.It usually relies on an input message sent by the terminal conveying a transaction code and user data.The transaction code allows the system to schedule a first program according to a specified running environment(Execution priority, resource protection level....).A reply is usually sent to the requestor at the end of the processing.

  - The user tansaction usually consists of a sequence of system transactions which are aimed at achieving a given logical piece of work.

- Relationships between the transaction and the system.

  The system is only aware from a data integrity or protection mechanisms standpoint of the system transaction entity.The user is responsible for ensuring,by designing his own additional logic,the data integrity along the user transaction.

- Relationships between the transaction and the SNA session.

  The interoperability between two LUs rely on the SNA session which provides the user with connectivity facilities(Role of the Path Control or "Common Network") and interworking facilities based on the utilization of end to end dialogue protocols.

  The SNA session does not know what is the transaction entity,which is system matter,and is only concerned with the dialogue between two partners LUs.It provides the two partners with dialogue and synchronization facilities which are defined at session set up time.The session is serially reusable by the transactions relying on it.

  From a dialogue point of view,the most suitable protocol for the interactive transactional environment has been referred to as "Bracketing" whereby two partners,using logical HDX communication mode,can manage the potential contentions by defining a "First Speaker" and a "Bidder"and speak in a two way alternate mode by using indicators and commands aimed at providing a " data token" management facility.

It would be desirable to map the user transaction onto the bracket entity from a logical point of view.Unfortunatly,when looking at the IBM existing systems it appears that for many reasons related to basic design or migration considerations such a mapping was not workable.For instance CICS (Customer Information Control System)encapsulates the system transaction inside a bracket while IMS (Information Management System) as a queuing system is not very concerned with this problem.

As a conclusion,it seems that the SNA defined Bracket Protocol, though being the most suited to an interactive transactional environment,was not of great interest in the past to the involved systems which used it only in order to manage the inherent contentions to the logical HDX communication mode.

- The evolution (Historical perspective).

  - First step.

    The session is set up between a host (application) primary LU and a set terminal-human operator which is the secondary LU.The operator's responsibility in the communication may be emphasized to the extent that,as a human being,it looks like a very complex and flexible program capable of handling unexpected events and situations. The operator,in fact,rules the relationships between the two entities communicating across the SNA session.

  - Second step.

    The session is set up between a host primary LU and a programmed secondary LU homed by either a host or a cluster controller.The communicating programs ask,according to their processing needs,for unique and sophisticated session facilities leading the architecture to define more and more session types (Session types "explosion"). This approach obviously impacts in a tremendous way the system and application designers,the system and application programmers and finally leads to waste a lot of data processing resources.

  - Third step.

    The advent of the intersystem coupling (ISC) allowing a given logical piece of work to be executed in a cooperative way in different places will emphasize the needs of efficient and powerful protocolssuitable for the so called "Advanced Program To Progam Communication" (APPC) environment.

    Functional capabilities like Function Shipping and mainly Distributed Transaction Processing claim for the definition of architected and powerful protocols allowing the programs to cooperate in a standardized ,efficient and reliable way to achieve a distributed Unit of Work.

Historically,a first CICS implementation was achieved leading to the definition of the so called LU type 6.LU type 6 protocols are aimed at meeting the above stated requirements by relying on the one hand on dialogue and synchronization protocol elements provided by the underlying session,on the other hand on pure application defined protocol elements conveyed by a set of new Function Management Headers (FMHs).The LU 6 encapsulates the dialogue between two cooperating TPs in a bracket thereby defining the embryo of a new entity:the"Conversation".

A more global approach encompassing a lot of various requirements will lead in a further step the architecture to define in a formal way the "conversation entity".

4.0 THE LU TYPE 6.2 GENESIS.

4.1 SOME BASIC PRINCIPLES.

- Some mechanisms such as Synchronization/Resynchronization,parallel session tuning....have a networkwide significance and therefore cannot rely on various system implementations defining concepts which are not required to be close to each other.

  As a consequence,the System Network Architecture has to design some corresponding architected mechanisms all the involved systems will have to comply with permitting them to cooperate in a consistent way.

  Such an evolution can be viewed as a migration of functions from the processing part of the application layer to its communication counterpart.

- The relationships between two cooperating TPs claim for new functional capabilities which are beyond the scope of the session.A new entity ,the"Conversation" is to be defined which will permit,while still relying on serially reusable sessions,to take into account the unique properties related to a program to program dialogue.

  As a consequence,the transaction code which till now was a pure application object,will become a System Network Architecture being. The System Network Architecture will define a transaction common naming space to the extent that some new architected system functions will rely on system service transaction programs communicating with each other by means of conversations thereby requiring some unique,networkwide significant,transaction codes or transaction program names (TPN).

- The architecture has to stop the LU type "explosion phenomenon" what means that it is requested to simplify the structure of the various data streams,presentation services,commands,FMHs.....etc.

4.2 THE CONVERGENCE EFFORT.

The System Network Architecture has to meet the requirements of both the small and large systems.The"Convergence Effort"will rely on:

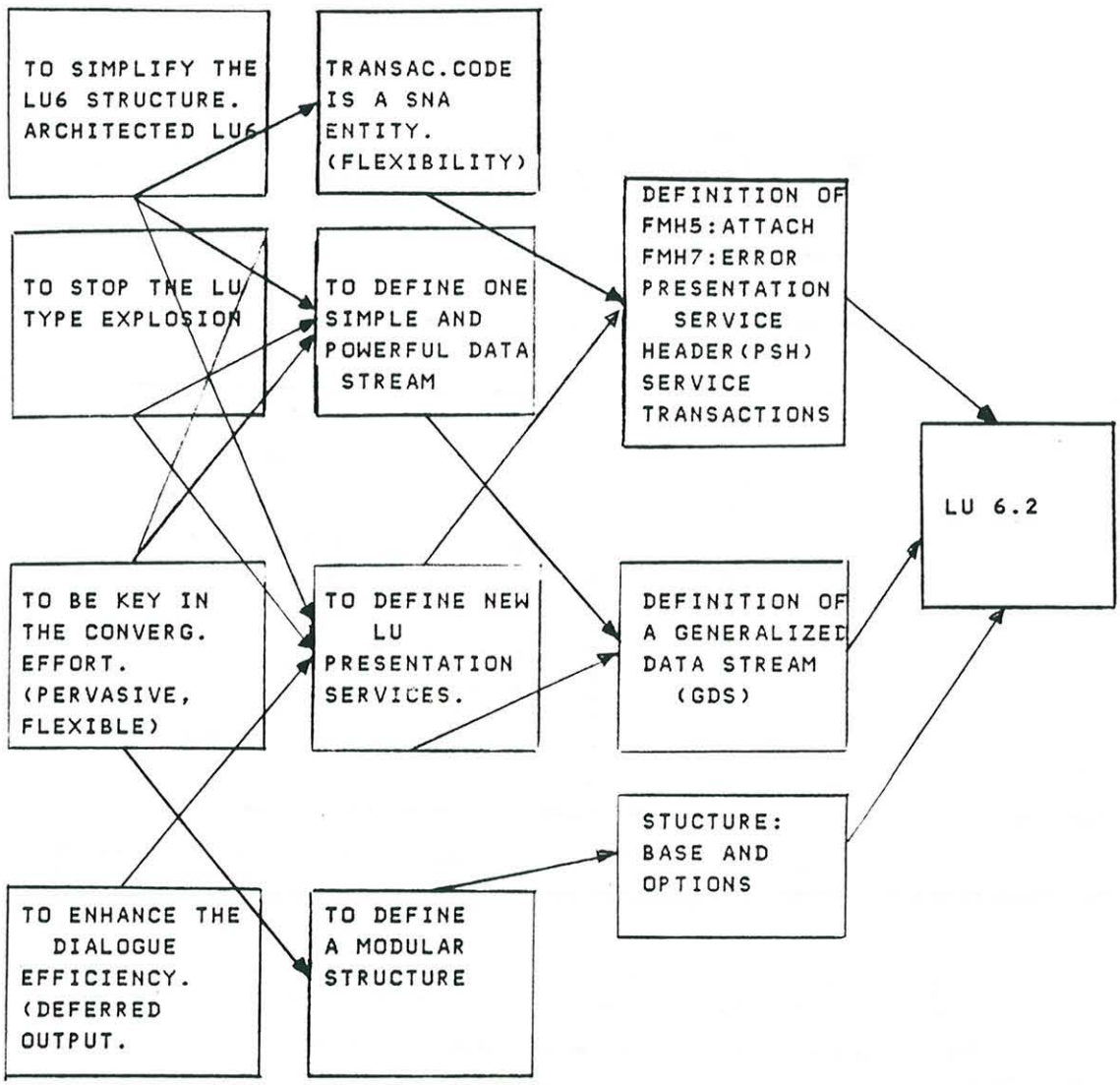
- The definition of a new PU type,the PU type 2.1,enabling the small systems to speak directly with each other,i.e without involving any SSCP mediated service,while being capable of participating to large

system networks.

- The definition of a new LU type,the so called "LU type 6.2" which is required to be as pervasive and flexible as possible thereby implying it has a modular structure (Base functional set and towers).The LU 6.2 will be the cornerstone of the communication inside small system networks as well as large system networks.As a pervasive LU it has to meet complex functional requirements (those related to the distributed interactive processing environment) while keeping a basic functional set as simple as possible so as to be suitable for the small system environment.

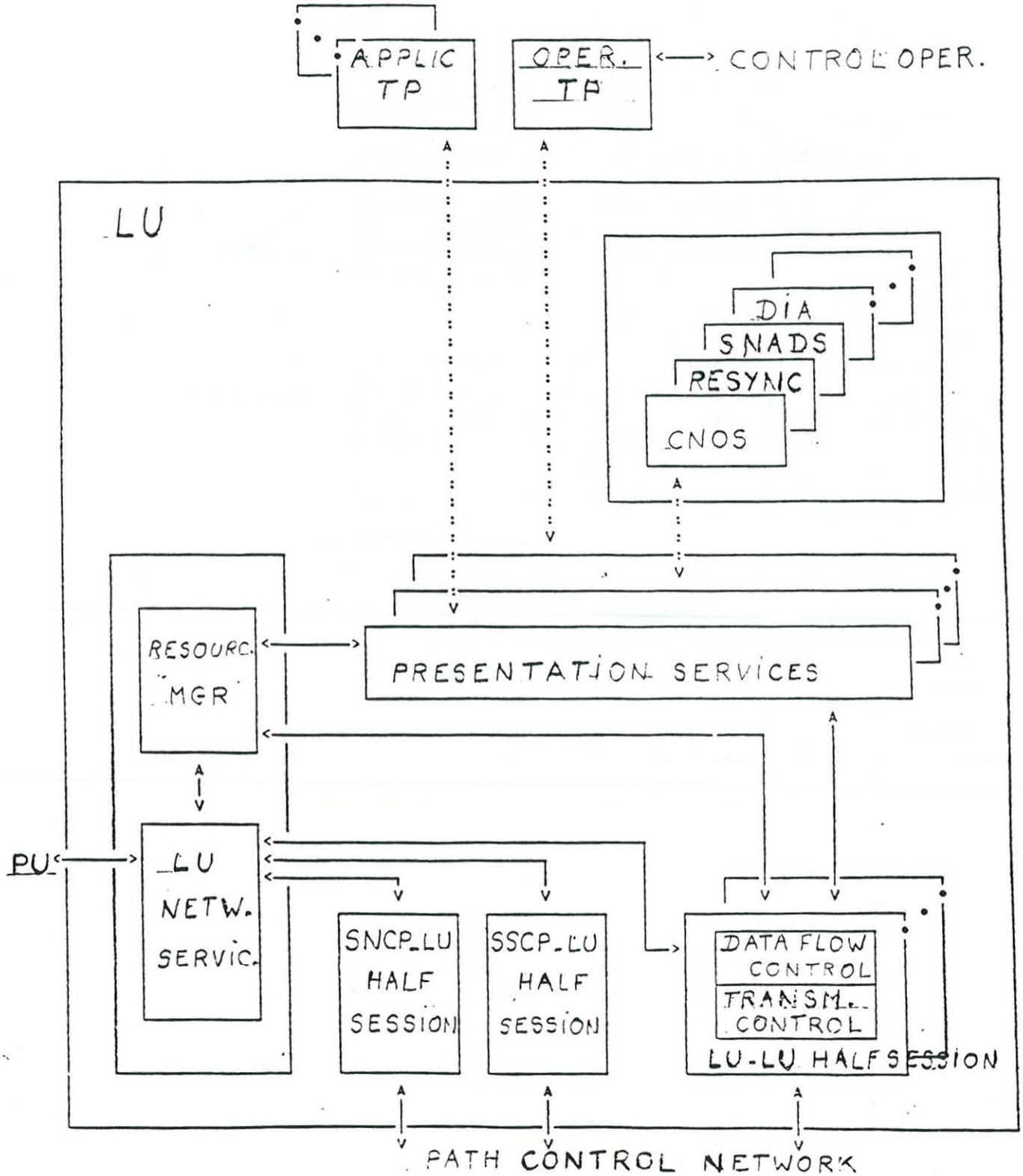
- The definition of architected system services like SNA Distribution Services (SNADS),Document Interchange Architecture (DIA).....etc.

4.3 THE REAPPRAISED LU:THE LU 6.2



5.0 LU 6.2 DESCRIPTION: AN OVERVIEW.

5.1 BASIC LU ARCHITECTURE.



- PRESENTATION SERVICES.

  MANAGE TPs AND CONTROL CONVERSATION LEVEL COMMUNICATION BETWEEN TPs

  - LOADS AND CALLS THE TPs.
  - MAINTAINS THE CONVERSATION PROTOCOL STATE BY MANAGING FINITE MACHINE STATES (FSM).
  - COORDINATES SPECIFIC PROCESSING FOR EACH VERB.
  - PERFORMS SOME MAPPING OF TP DATA.
  - CONVERTS LOGICAL RECORDS PROVIDED BY THE TP (MAPPED CONVERSATION) IN GDS FORMAT AND CONVERSELY.
  - BUFFERS CONVERSATION DATA FROM THE TP FOR EFFICIENT USE BY HALF SESSION.
  - TRUNCATES AND PURGES DATA WHEN ERRORS ARE REPORTED OR DETECTED BY THE TP.
  - GENERATES AND ISSUES FM HEADERS FOR ATTACH (FMH5) AND ERROR DESCRIPTION.

- HALF SESSION.

  CONTROLS SESSION LEVEL COMMUNICATION BETWEEN LUs

  - PROVIDES THE NETWORK WITH APPROPRIATE SIZED MESSAGE UNITS.
  - BUILDS AND ENFORCES CORRECT PROTOCOL RELATED INDICATOR SETTINGS.
  - CREATES CHAINSAND ENFORCES CHAINING SA THE UNIT OF LU TO LU RECOVERY(THE CHAIN IS THE UNIDIRECTIONAL MESSAGE TANSFER ENTITY).
  - ENFORCES BRACKET PROTOCOL.
  - GENERATES AND ENFORCES SEQUENCE NUMBERING TO DETECT LOST OR DUPLICATE MESSAGE(S)
  - ENFORCES PROTOCOLS ACCORDING TO THE COMMON AGREED UPON RULES BY THE TWO PARTNERS AT SESSION SET UP TIME(BIND COMMAND AND BIND RESPONSE).
  - ENCIPHERS AND DECIPHERS DATA WHEN NECESSARY.

- RESOURCE MANAGER.

  MANAGES PRESENTATION SERVICES AND CONVERSATIONS.

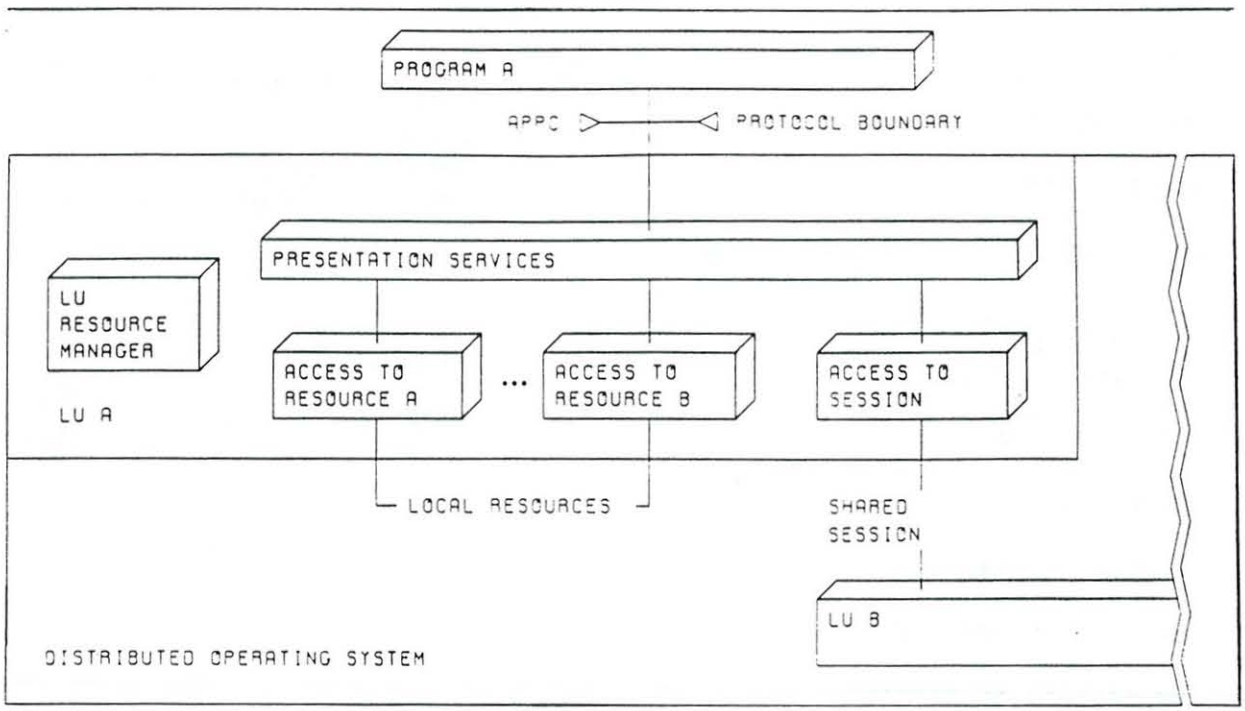
  - CREATES AND DESTROYS INSTANCES OF PRESENTATION SERVICES.
  - CREATES AND DESTROYS CONVERSATION RESOURCES AND CONNECTS THEM TO PRESENTATION SERVICES AND HALF SESSION.
  - MAINTAINS IN STORAGE THE CONTROL BLOCK STRUCTURES.
  - CHOOSES THE SESSION TO BE USED BY A CONVERSATION AND MANAGES CONTENTION FOR THE SESSION.
  - REQUESTS LU NETWORK SERVICES TO ACTIVATE AND DEACTIVATE SESSIONS.
  - PROVIDES SERVICES FOR SUPPORT OF THE SYNCPOINT LOG.

- LU NETWORK SERVICES.

  LNS MANAGE THE SESSIONS.

  - COORDINATES SESSION INITIATION IN CONCERT WITH THE CONTROL POINT.
  - SENDS AND RECEIVES THE BIND COMMAND.
  - SUPPLIES AND NEGOTIATES BIND SESSION PARAMETERS.
  - NOTIFIES RESOURCE MANAGER OF SESSION OUTAGE.
  - CREATES AND DESTROYS HALF SESSION INSTANCES AND CONNECT THEM TO THE PATH CONTROL INSTANCES.

5.2 LU 6.2 AS A DISTRIBUTED OPERATING SYSTEM.



5.3 LU 6.2 PROTOCOL BOUNDARY.

A set of architected verbs aimed at exercising the appropriate protocols is referred to as a "protocol boundary" rather than as an application progam interface (API),in order to distinguish them from the functionally similar interfaces that products provide for the use of their application programs.

- THE VERBS.(MAPPED CONVERSATIONS).

  .BACKOUT

  .GET_TYPE

  .MC_CONFIRM

  .MC_CONFIRMED

  .MC_ALLOCATE

  .MC_DEALLOCATE
  TYPE(FLUSH)
  TYPE(SYNC_LEVEL)
  TYPE(ABEND)
  TYPE(LOCAL)

  .MC_FLUSH

  .MC_GET_ATTRIBUTES

  .MC_POST_ON_RECEIPT

  .MC_PREPARE_TO_RECEIVE

  .MC_RECEIVE_AND_WAIT

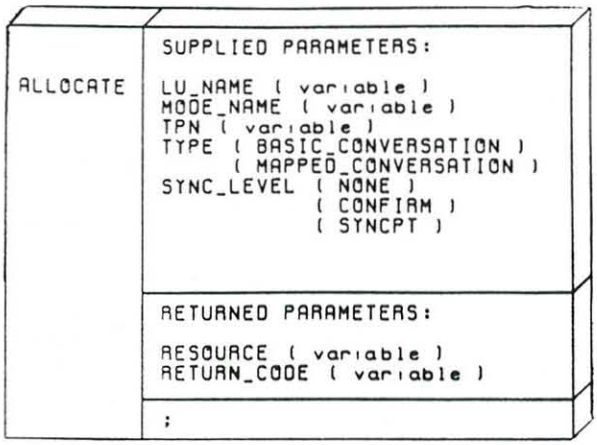
  .MC_REQUEST_TO_SEND

  .MC_SEND_DATA

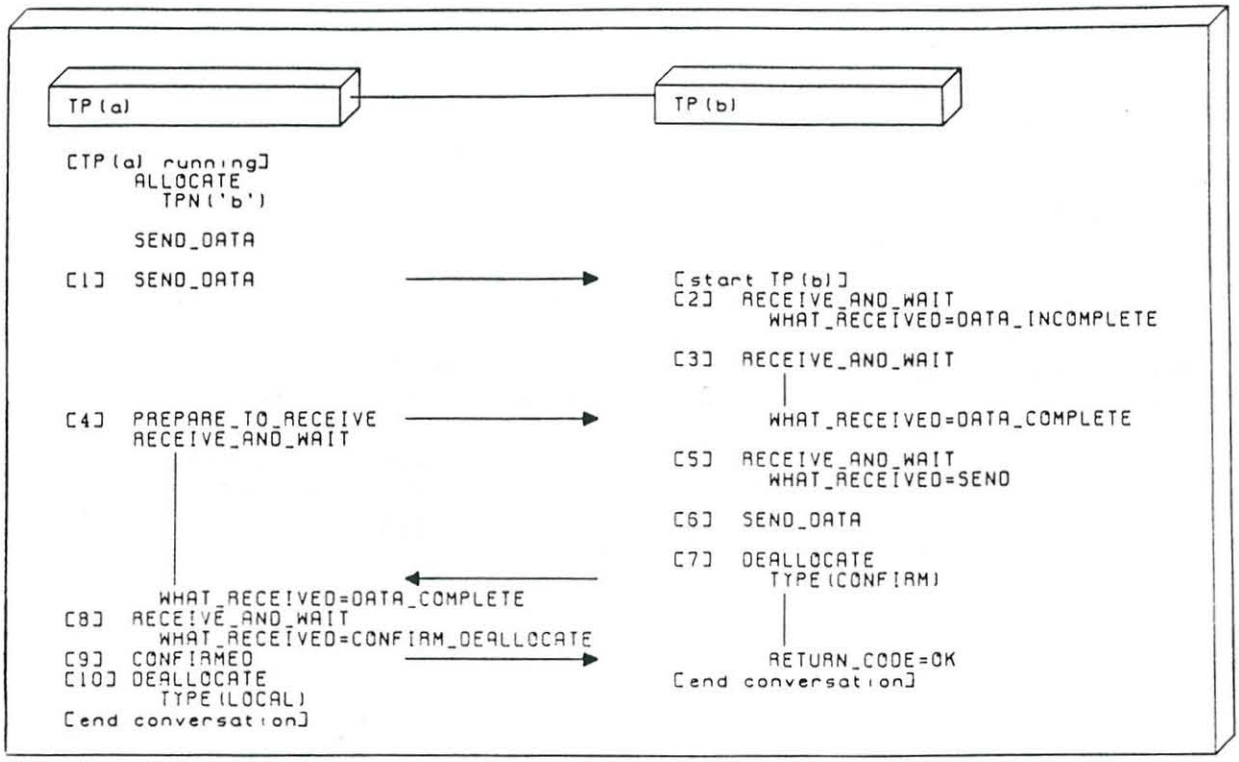
  .MC_SEND_ERROR

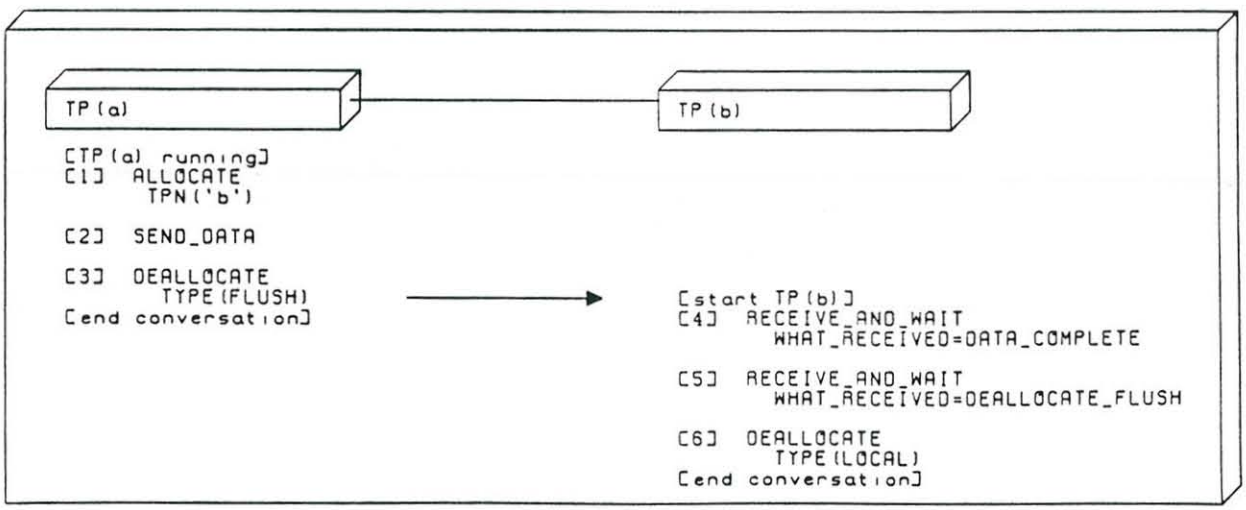
  .SYNCPT

  .WAIT



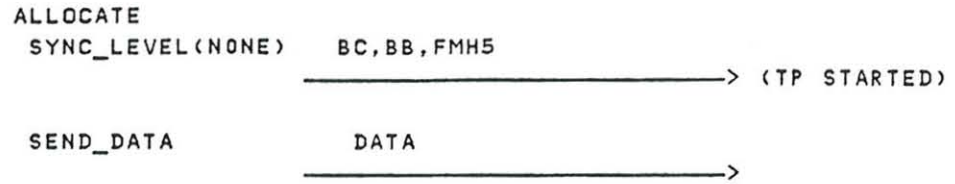
- THE CONVERSATION.



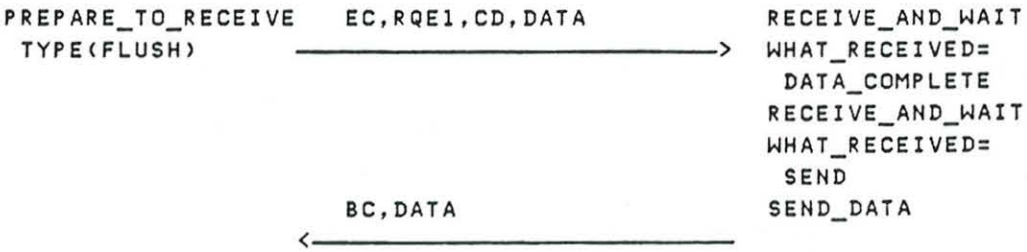


5.4 THE PROTOCOLS AND DATA FLOWS.

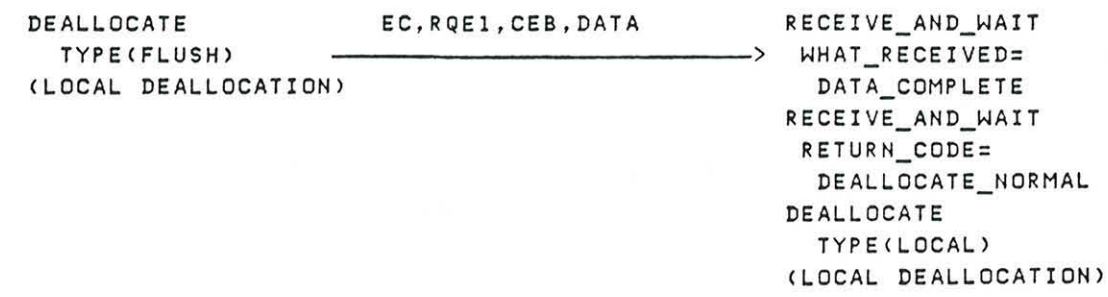
.START CONVERSATION WITHOUT CONFIRMATION



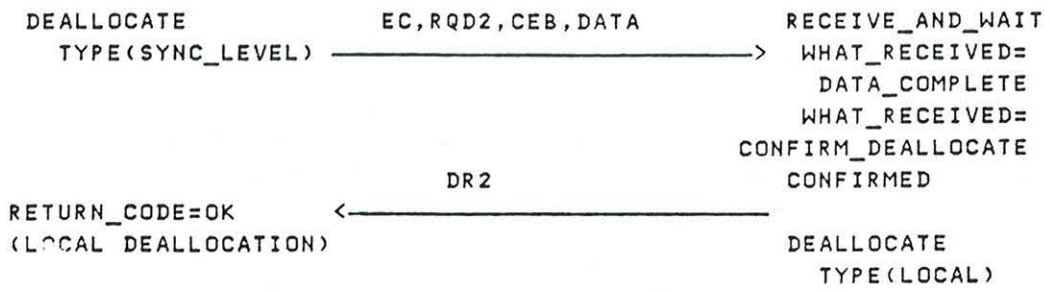
.CONVERSATION TURNAROUND WITHOUT CONFIRMATION



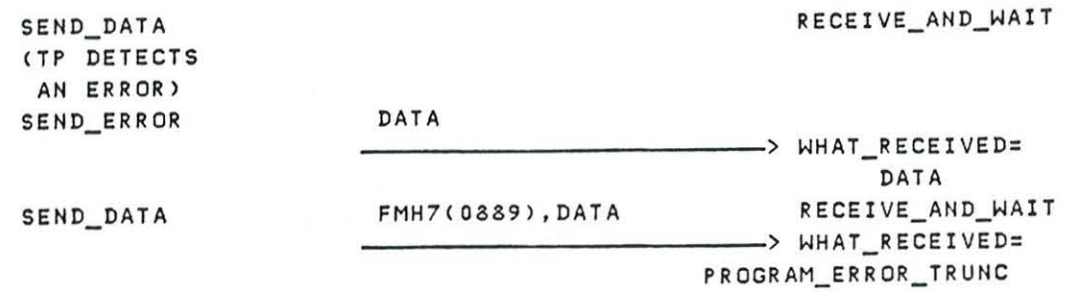
.FINISH CONVERSATION WITHOUT CONFIRMATION



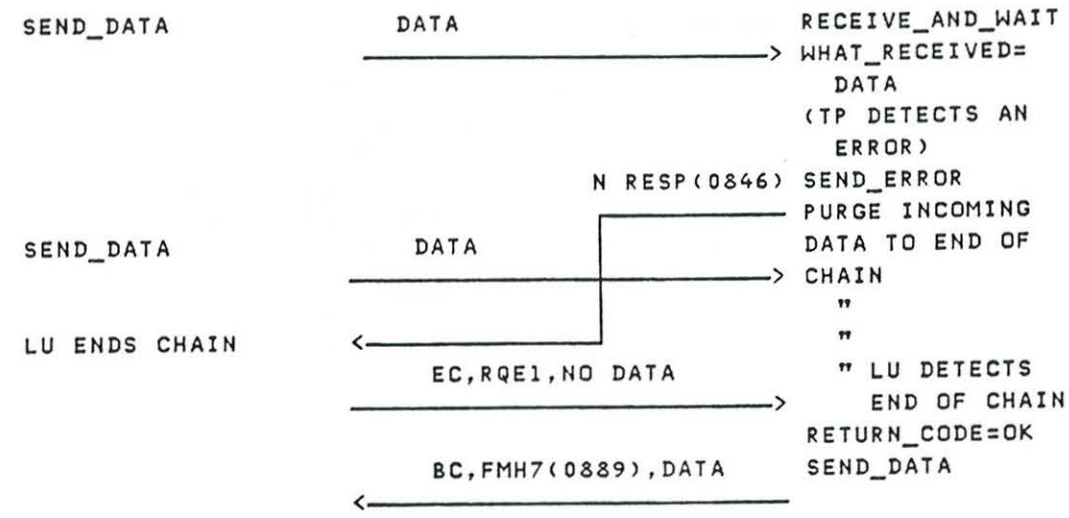
.FINISH CONVERSATION WITH CONFIRMATION



.SEND_ERROR ISSUED BY SENDER



.SEND_ERROR ISSUED BY RECEIVER



RETURN_CODE=
PROGRAM_ERROR_PURGING
RECEIVE_AND_WAIT

DISCUSSION

Professor Randell asked is the most important impact of LU 6.2 performance?

Mr. Decreuse replied yes, performance and pervasiveness. We want an LU type suited to all environments.

Professor Pyle asked that if before LU 6.2, type 0 was any option? Also is this superceded by LU 6.2?

Mr. Decreuse replied, existing LU's will remain.

Professor Pyle asked if LU 6.2 is for small and large environments? Also will it work for a personal computer?

Mr. Decreuse replied that IBM can give no statement on that point.

Professor Pyle stated if LU 6.2 is pervasive and for all environments, then personal computers are the most important on which it must work.

Mr. Decreuse made no comment.

Professor Randell asked how effective could this be for a very light-weight protocol for remote procedure calls?

Mr. Decreuse replied that the concept of deferred output enables the LU to minimise action of user messages between control entities. The program issues verbs and the corresponding data communicates in a buffer, until the buffer is full or the program asks to send the buffer's contents. The sending program will wait until it receives a response or confirmation. An exceptional response is sent if failure occurs. Thus exchanges are minimised and piggy-backing is used. The light-weight protocol enhances performance.