MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

MESSAGE LINK PROTOCOL (MLP) AND ITS INTERFACES

Anton F. Hackl

IPP R-38 October 1982

The Message Link Protocol MLP ist established on top of X.25 packet level and comprises the following functions:

- connection of a pair of processes
- selection of communication modes
- synchronisation of processes
- end-to-end recovery between processes
- transport of messages between processes

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

0. Introduction 0

1. Interface: Network Machine - MLP 1

  1.1 As we see the network 1
  1.2 Exported Network Services 1
  1.3 Imported Network Services 3

2. Interface: MLP - User 5

  2.1 Exported MLP services 5
  2.2 Imported MLP Services 7

3. MLP Protocols 10

  3.1 Establishing a message link 10

    3.1.1 ML-Establishment without conflict 10
    3.1.2 ML-Establishment with conflicts 14

  3.2 Closing the message link 19
  3.3 Choosing the mode of communication 21
  3.4 Data Transferring 23
  3.5 Interrupt Protocol 24

4. MLP-Automaton 25

RESY Report Editor System, Version 2.1, Jan 1981, IPP

0. Introduction

This paper deals with a level 4/5 protocol named "Message Link Protocol" (MLP) investigated and implemented at the computer centre of the Max-Planck-Institute for Plasma Physics, Garching, FRG.

It formally describes the functions of the MLP and shows the interfaces to the levels below and above (i.e. to the network machine and to the user of the MLP).

Hint:
The "user" may not only be a user-written application program but also a higher level protocol

In order to make efficiently use of a network service protocol like CCITT recommendation X.25, there must be a connection between a pair of users (local and remote).This connection is called "message link" , because user data ("messages") is transferred via this logical link.
(The user is not urged to install a message link only with a remote partner, he also can communicate via MLP with a process on its local site without using the network service at all).

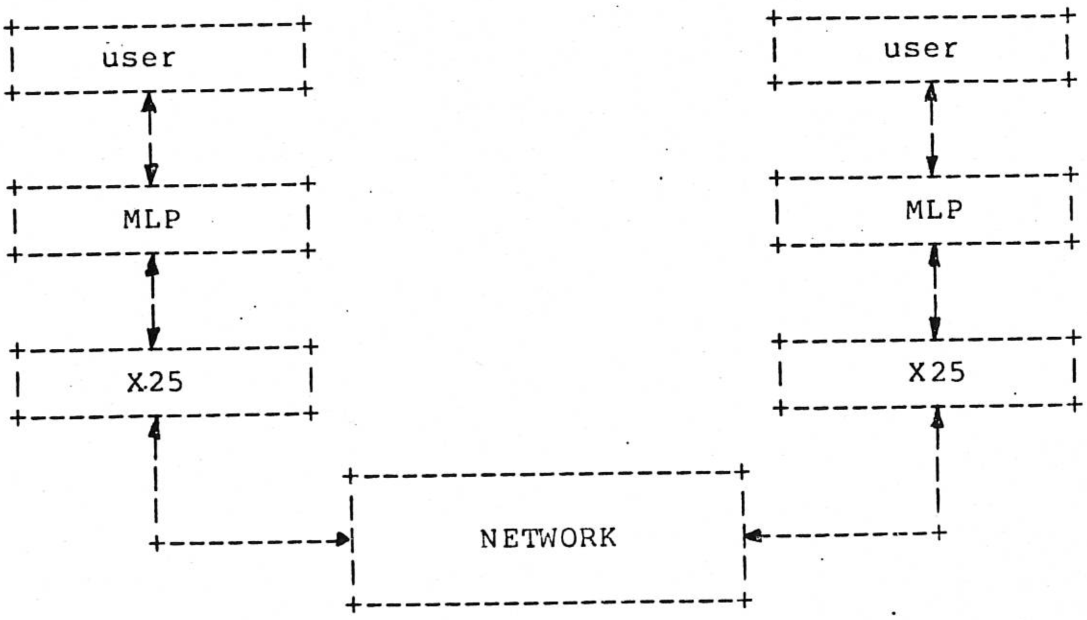


Fig. 1 : Overall Structure

1. Interface: Network Machine - MLP

1.1 As we see the network

Independently of the chosen network one can see the network as a module which offers services to its users. Partly these services however, are tightly connected to the underlying network protocol like -in our case- X.25, because there must be a mapping function which enables the correspondence between exported network services and the internal network primitives like types of packets etc. (one may think at the X.25 Interrupt procedure which is not part of all networks).
On the other hand there have to be services with overall significance like installing a (network) link, transferring data, or resetting the link.

1.2 Exported Network Services

The aim of this section is to enumerate the exported network services and shortly characterize the meanings of them.

Independently of the implementation one might choose the following syntax for the services of the network machine:

ID (par 1, ... ,par n)

ID is thereby a symbolic name for the service
par i (i=1,...,n) the parameters of the service

(1) RESERVE (MLN)
This service has only local significance (i.e. no data transfer takes place) and means that the user of the service intends to make use of the network facilities. Primarily, the action taking place is making the user's name (i.e. MLN) known to the (local) network machine.
(MLN = Message Link Name)

(2) FREE (MLN)
The opposite of RESERVE.
All resources occupied by the user MLN are freed and the user's name is made unknown.

(3) INIT (MLN , sender , receiver , type)
The user wishes to install a virtual circuit. In the case of a switched virtual channel this service is mapped onto the X.25 CALL procedure. The parameters "sender" and "receiver" are X.25 DTE addresses. The parameter "type" indicates the type of channel: permanent or switched.

(4) TERM (LCN)
Opposite of INIT.
The user finishes the data transfer period and therefore no longer a X.25 logical channel has to be reserved for him. Only for switched channels this service is mapped onto the X.25 CLEAR procedure. (LCN = logical X.25 channel number)

(5) XFER (LCN , data)
Over the installed logical X.25 channel the user would like to transfer data. If the service ends with no success (showing by the imported sevice DATACK), wait for RESUME (see 1.3.).

(6) RCV (LCN , @BUF , LBUF)
This operation is issued to receive a prior (by DATA, see 1.3) signalled message in the indicated buffer with length LBUF.

(7) RESET (LCN , code)
Due to a detected error on the message link the user intends to reset "his" line.
This service is mapped onto the X.25 RESET procedure.

(8) IRPT (LCN , data)
The user has data (one byte only) which should be transferred outside the normal flow control mechanism of X.25. (i.e. with the X.25 interrupt procedure).
The user himself has to take care of synchronisation.

```
 +--------------------------------------------------+
 |                       MLP                        |
 +--------------------------------------------------+
   |     |     |     |      |      |      |       |
RESERVE  |     |     |      |      |      |       |
   |   FREE    |     |      |      |      |       |
   |     |   INIT    |      |      |      |       |
   |     |     |   TERM     |      |      |       |
   |     |     |     |    XFER     |      |       |
   |     |     |     |      |     RCV     |       |
   |     |     |     |      |      |    RESET     |
   |     |     |     |      |      |      |     IRPT
   |     |     |     |      |      |      |       |
 +-v-----v-----v-----v------v------v------v-------v-+
 |               X.25 Network Machine               |
 +--------------------------------------------------+
```

Fig. 2 : Exported Network Services

1.3 Imported Network Services

As we have seen the exported network services act only in one direction , namely from MLP to network machine downwards.
Naturally it is necessary to install services in the opposite direction from network machine to MLP , for instance if a logical X.25 channel was ordered from a remote host, if a channel has to be cleared, if data is received or if a reset has to be done on a channel.
In all these situations the network machine is urged to react and deliver information to the attached network user.
This information transfer upwards is called "imported service" (from the standpoint of the deliverer).

(1) READY (LCN, MLN , success)
If a X.25 channel was "called" (by INIT from the local or the remote user) , the network machine informs the user.
The parameter "success" is a boolean variable , which -if false- means that there is no free channel (as response to INIT).

(2) CLEAR (LCN, MLN , code)
In the case that the channel was cleared by the remote partner or by the network, this is the information for the network user.

(3) DATA (LCN)
If there was data received , the user is signalled with this service.

(4) DATACK (LCN, success)
After having issued the service XFER the user must wait for this data acknowledge before intending to transfer more data. The parameter 'success' indicates the result of the previous issued XFER.

(5) RESUME (LCN)
Due to temporary shortage of buffers for instance, the user must not issue the exported service XFER until not receiving this signal RESUME. (The user was first notified from the shortage by the parameter 'success' in DATACK).

(6) RESET (LCN , code)
Due to a X.25 reset (or restart) the associated user is informed. The parameter "code" signals the cause of the reset.
(In the case of a restart all users are informed).

(7) IRPT (LCN , data)
Outside the normal flow control mechanism the remote partner has transferred data (one byte) , which is given to the receiver (=MLN).

(8) RESTART (LCN-group)
This service is necessary to signal the user the re-starting of the network-machine. All connections for this group of channels are thus closed.

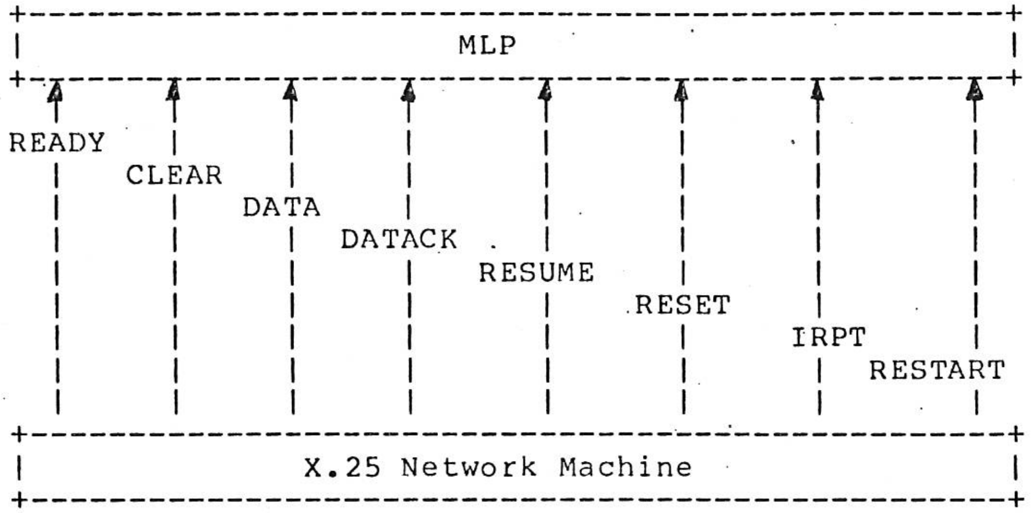


Fig. 3 : Imported Network Services

The formalized connection between exported and imported network services -which is referenced later on- is shown in the following simple figure:

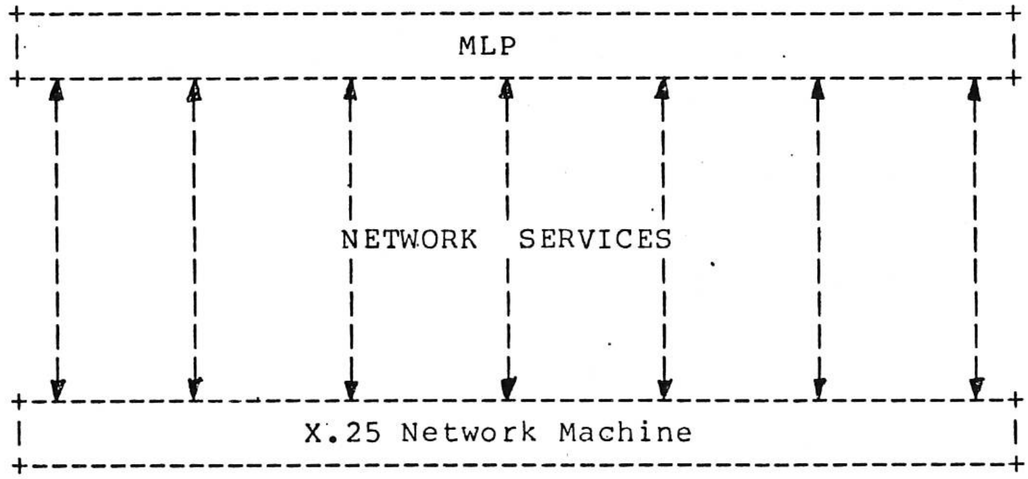


Fig. 4 : Network Services

2. Interface: MLP - User

As the network machine the message link layer (level 4/5 protocol, transport service) also offers services to its users.
This services comprises the set up, handling, and clearing of the message link. It also gives means of selecting the communication modes, synchronizing of users and end-to-end recovery.
All these services are available to the user at the interface MLP - User , which is implementation-dependent and need not be the same on both ends of the message link.
It is invisible to the user that there exist layers below MLP, because modularity and layering of protocols is desirable.
Each level of protocol should be independent of the details of the lower level protocol(s) that support it. The primary effect of this guideline is that MLP control information is not imbedded within X.25 control packets.
In the following chapters all the exported and imported services of the MLP are explained without details of the level 4/5 protocol.

2.1 Exported MLP services

(1) ALLOCATE (PI)
This service has only local significance (i.e. no data transfer takes place) and means that the user (= PI , process identifier) of the service intends to make use of the MLP facilities.
Primarily, the action taking place is making known the user's name for further actions. After using this service the MLP is able to identify the user.

(2) RELEASE (PI)
The opposite of ALLOCATE.
All resources occupied by the user PI are freed and the user's name is made unknown.
All messages for an unknown user are discarded.

(3) ECONN (code, PIs, PIr, [Password,billing specs])

code	active or passive ECONN request
PI	process identifier (s=sender, r=receiver) PIr must contain the DTE address of the receiving host.
Password billing specs	optional parameters

The ECONN service is used to install a message link between PIs and PIr with support by an X.25 logical channel (permanent or switched)
Two kinds of ECONN requests are available:

a) Active ECONN
The PIs issues a ECONN request: The MLP initiates to "call" an X.25 channel and then it tries to install a message link.

b) Passive ECONN
The PIs issues a ECONN request: no attempt is made to install a message link but the MLP is waiting for an incoming ECONN request.

Hint:
Are passive ECONN requests ordered on both sides, no message link can be installed , that means: one partner has to be active.

(4) EDISC (PI, code)
code reason for closing the connection

The EDISC service is provided to close a message link. MLP implementation will ensure that data sent by PI prior to the EDISC request are delivered to the remote host and made available to the remote process.

(5) EOPEN (PI , CM)
PI name of sending process
CM communication mode

This service is used to "open" a (new) communication mode, e.g. identifying the types of higher level protocols (>5) before starting with data transfer.
In the first implementation one might choose only the "Virtual Terminal Protocol" and/or the "Communication Variable Protocol".
(see ADN ....) The nesting of this service is for further study.

(6) ECLOSE (PI , code)
A previous "opened" communication mode is closed. (In the case of nested communication modes the last opened one is cleared , i.e. the youngest in the stack).

(7) ETRANS (PI , @MSG , LMSG , success)
The message exchange operation ETRANS is issued if a user wants to transmit a message of length LMSG stored in the buffer @MSG. This operation should cause copying the message from the user space to the MLP address space (because the user space is then not tied up during message transmission and may be rolled out or paged out). The parameter 'success' signals the success of the service: if there was no possibility to take over the message into the area of MLP-process because of temporary shortage of buffers, the user has to wait for 'CONTINUE' (see 2.2.).

(8) EREC (PI , @BUF , LBUF , success)
This operation is issued to receive a message in the indicated buffer @BUF with length LBUF. If a message is waiting in the receive queue of MLP, it is copied into the buffer and the user continues. If there is no message, the user is notified by the parameter 'success' and he has to wait for the service 'MSG' (see

2.2.). It is implementation- -dependent, whether the user goes into wait state or not.

(9) EIRPT (PI , data)
This service is used to send interrupt-data to the remote host outside the normal flow control mechanism.

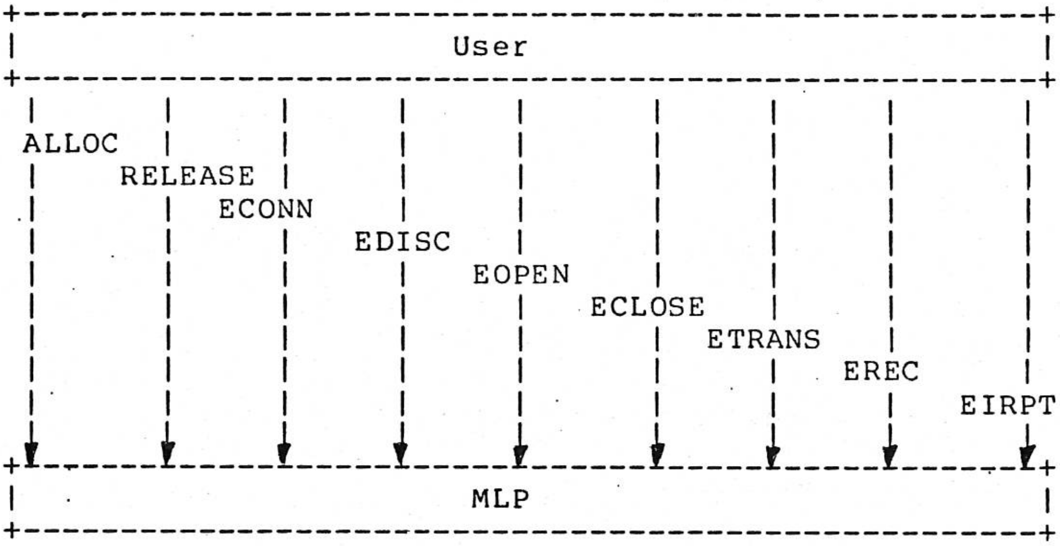


Fig. 5 : Exported MLP Services

2.2 Imported MLP Services

As with the imported network services the imported MLP services act also upwards, namely from MLP to User, i.e. the MLP process issues them and the user process gets the associated information.
Mainly these services are responses to previously issued exported MLP services. Beneath they indicate the arrival of information (such as messages, interrupts, "close" requests,...) from the remote host.
The implementation may vary from host to host: it is not strictly necessary to introduce all these services explicitely (for instance by supervisor call) , but one may also imagine that it is possible to integrate part of the imported services in the exported services as return code for instance.
What is illustrated in this chapter is the logical function of the interface User-MLP in the upward direction.

(1) ICONN (PI , code)
This is the response to a ECONN service from the local user respectively the indication of a remote ECONN request.
The result (positive or negative) is given by the parameter "code"

(2) IDISC (PI , code)
The IDISC primitive indicates the shut down of a

message link (either response or remote indication) The parameter "code" shows the reason of the disconnection.

(3) IOPEN (PI , CM , code)
This operation is issued if a remote EOPEN was ordered or as response to a local EOPEN.
Has it been impossible to open a new communication mode, "code" signals the reason.

(4) ICLOSE (PI , CM , code)
This is the response to a local ECLOSE or the indication of a remote ECLOSE.

(5) MSG (PI)
In the case that the MLP receives a message the associated user PI is informed and he has to get his message with EREC.

(6) IIRPT (PI , data)
If an interrupt-message has been received at MLP and both ends of the message link are already synchronized the MLP issues IIRPT and tells the user the reason of the interrupt.

(7) CONTINUE (PI)
This service is used if a temporary shortage of buffers is cleared and the user had tried to send messages in the meantime. In this way the user tries only once to deliver a message if there is a shortage in the MLP-process.

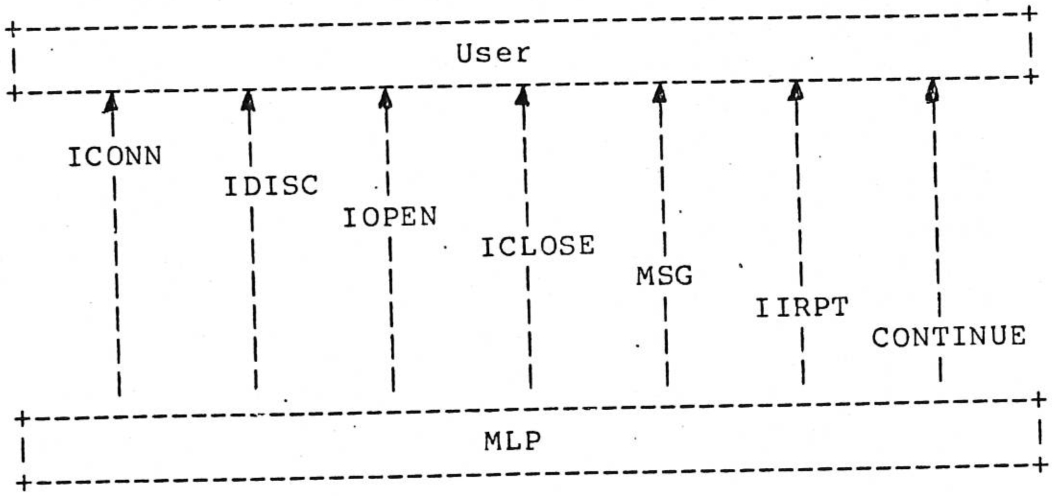


Fig. 6 : Imported MLP Services

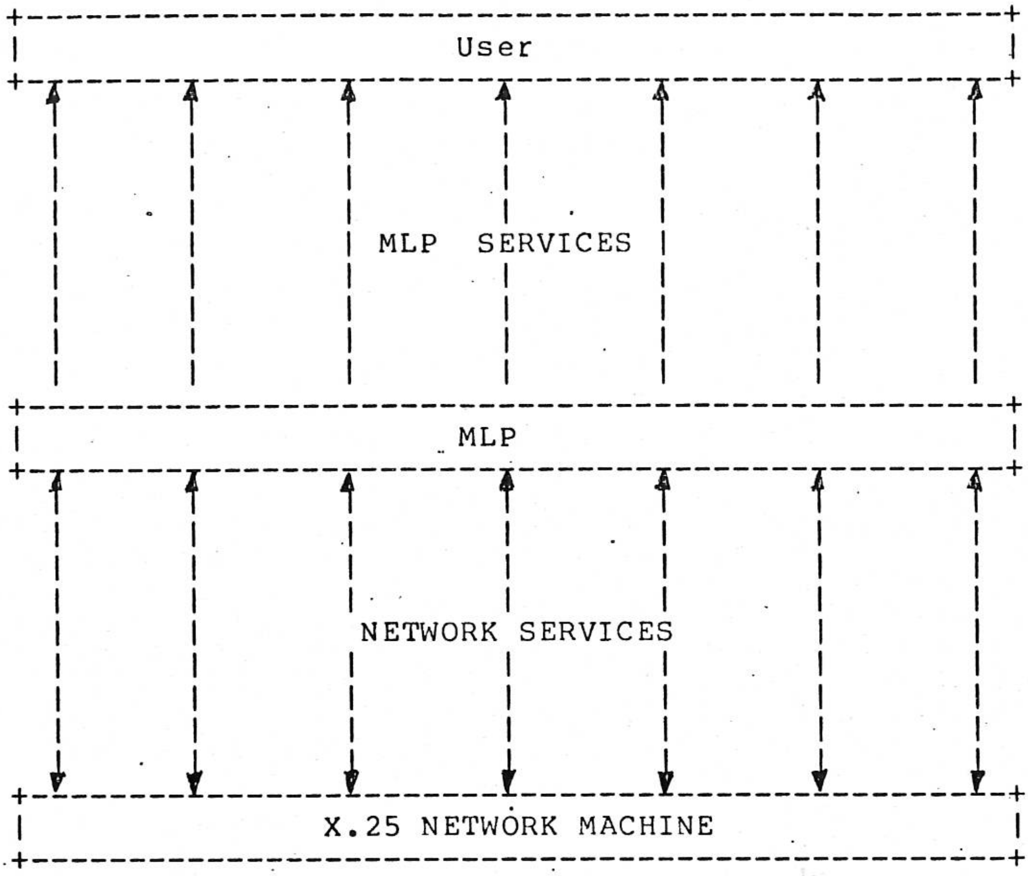


Fig. 7: Overall service structure

3. MLP Protocols

It is possible to distinguish between five kinds of protocols inside the MLP:

- Establishing of a message link
- Closing a message link
- Choosing the mode of communication
- Data transferring
- Interrupt protocol

These subprotocols are explained in this section by means of figures showing the actions of the protocols in conjunction with time characteristics. These figures are not as exactly as necessary for implementation because some relevant details are not incorporated, but they give an overview over the flow of exported and imported services as defined in the above chapters.
Because of simplicity the services are not given in the exact notation: the parameters are omitted or replaced by a symbolic variable which is explained in the figure.

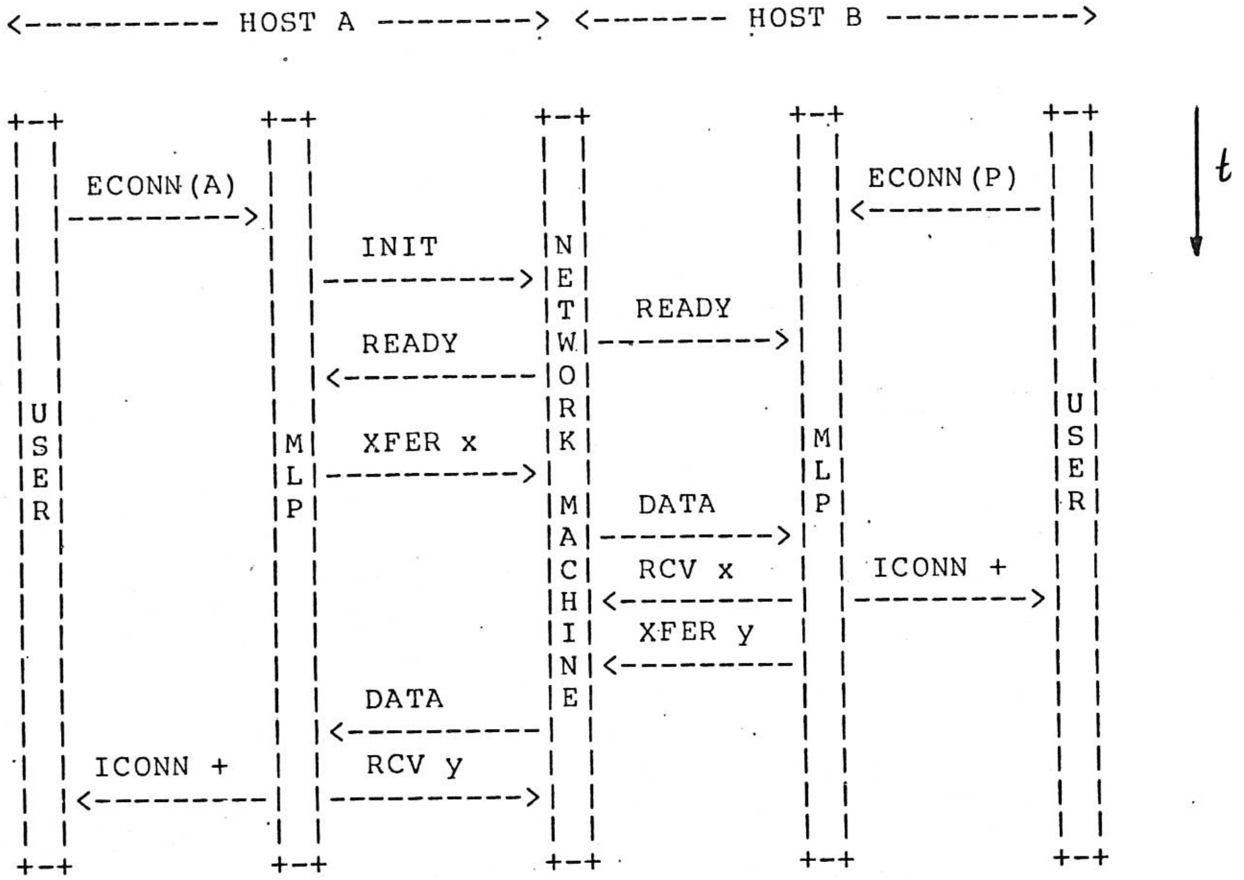
3.1 Establishing a message link

A message link is the logical connection between a pair of users of the MLP for the sake of transferring messages. These users need not be located in different hosts (geographically separated) but they also can be in the same host (e.g. connecting a locally hooked up terminal with an output processor). In this case installing a message link means to establish a path between two "processes" in the same host over which the processes can communicate with each other. This procedure uses only the MLP services as defined above and no underlying service is attached.
A more complex way to establish a message link is over a network service like X.25:
The users which wish to communicate are located in geographically separated (network-) hosts. The difficulty herein is that first of all before installing a message link there must be a network resource called logical channel in X.25 notation connected to the MLP processor over which the "connect request" is to be transmitted.

3.1.1 ML-Establishment without conflict

The establishment of a message link without conflict is only guaranteed in the case that one partner issues an active ECONN and the other a passive ECONN.
It should be predetermined which user is passive, because if both use this MLP-service no ML could be established.
Under the presumption that host B has already issued a passive connect request ECONN with specified parameters, host A initiates the establishment of a message link by an active ECONN (with specified parameters).
First of all, MLP has to cater for a network connection

between host A and host B. This is done by the network service INIT. Thereby, if no problems arise, the remote user of the network (=MLP) is informed about the "called" channel and also the initiator of INIT by the service READY.
With this signal host A is able to transmit over the installed logical channel the message "connect request" to the remote partner, who waits for this message. If all the parameters match - delivered within the connect message - the remote partner sends back a positive acknowledgement and the message link between these partners is installed . (see fig. 3.1.)
But if the relevant parameters don't match a negative acknowledgement is sent back and no message link could be installed. The occupied network connection has to be released. (see fig. 3.2.)
Besides it is up to the users to try it once more with modified parameters or with another partner.



ECONN(A)	active ECONN
ECONN(P)	passive ECONN
ICONN +	message link established
x	connect request
y	positive connect confirmation

Fig. 3.1. : Message Link establishment
- matching parameters
- only one active ECONN

```
<---------- HOST A --------> <------- HOST B ---------->

+-+          +-+            +-+          +-+          +-+
| |          | |            | |          | |          | |
| | ECONN(A) | |            | |          | | ECONN(P) | |
| |--------->| |            | |          | |<---------| |
| |          | |   INIT     |N|          | |          | |
| |          | |----------->|E|          | |          | |
| |          | |            |T| READY    | |          | |
| |          | |   READY    |W|--------->| |          | |
| |          | |<-----------|O|          | |          | |
|U|          | |            |R|          | |          |U|
|S|          |M|   XFER x   |K|          |M|          |S|
|E|          |L|----------->| |          |L|          |E|
|R|          |P|            |M| DATA     |P|          |R|
| |          | |            |A|--------->| |          | |
| |          | |            |C| RCV x    | | ICONN -  | |
| |          | |            |H|<---------| |--------->| |
| |          | |            |I| XFER z   | |          | |
| |          | |            |N|<---------| |          | |
| |          | |   DATA     |E|          | |          | |
| |          | |<-----------| |          | |          | |
| | ICONN -  | |   RCV z    | |          | |          | |
| |<---------| |----------->| |          | |          | |
| |          | |   TERM     | |          | |          | |
| |          | |----------->| |          | |          | |
| |          | |            | | CLEAR    | |          | |
| |          | |   CLEAR    | |--------->| |          | |
| |          | |<-----------| |          | |          | |
| |          | |            | |          | |          | |
+-+          +-+            +-+          +-+          +-+
```

ECONN(A)	active ECONN
ECONN(P)	passive ECONN
ICONN -	message link not established
x	connect request
z	negative connect confirmation

Fig. 3.2. : Message Link establishment
- mismatching parameters
- only one active ECONN

3.1.2 ML-Establishment with conflicts

Conflicts arise if only one message link should be installed and both participants (=user of the MLP) issue an active ECONN.
What happens thereby?
Both ends of the planned ML simultaneously cater for a network connection between the locally separated hosts, they initiate the establishment of a X.25 logical channel.
In this way two network connections exist but this is one too much because each side uses its "own" channel: the connect request message divergates.
This problem cannot be detected by the network for it should not have any information about higher protocols.
The MLP however has to detect and resolve this conflict which is named CONTENTION.

MLP-Strategy for detecting and resolving CONTENTION:

(1) Detecting CONTENTION
MLP is able to detect CONTENTION if it already has sent a connect request message over the installed "own" logical channel and there is also a connect request message in the opposite direction and over another than the "own" channel.
If time characteristics are included the detection of the conflict is much more complicated: suppose the remote host (host B) is quicker than host A. Then the connect request message from host B could arrive at host A even if no "own" logical channel was installed.
But host A knows about the installation of the channel which is used by host B for transmission. Therefore perhaps it would be desirable to insert into the "call user data" of the X.25 call request packet the port number MLN of the called MLP and give it with the READY signal to the receiving MLP. This technique enables the MLP to detect CONTENTION earlier.
Once detected CONTENTION has to be resolved.

(2) Resolving CONTENTION
An arbitrary criterion for resolving contention is the network address of the hosts which is known by the MLP when contention is detected:
The MLP with the lower network address ignores the incoming connect indication message from the remote side, the MLP with the higher network address cancels his own connect request (i.e. he doesn't wait for answer) , sends back a connect confirmation message over this logical channel which has delivered the connect indication and clears the "own" installed X.25 switched channel. (see figures 3.3 , 3.4. , 3.5.)

```
<---HOST A (higher DTE@)---> <----------HOST B---------->

+-+            +-+            +-+            +-+            +-+
| |            | |            | |            | |            | |
| | ECONN(A)   | |            | |            | |            | |
| |----------->| |            | |            | | ECONN(A)   | |
| |            | | INIT a     | |            | |<-----------| |
| |            | |----------->| |            | |            | |
| |            | |            | | READY a    | |            | |
| |            | |            | |----------->| |            | |
| |            | |            | | INIT b     | |            | |
| |            | | READY a    | |<-----------| |            | |
| |            | |<-----------| |            | |            | |
| |            | | READY b    | |            | |            | |
| |            | |<-----------| | READY b    | |            | |
| |            | |            | |----------->| |            | |
| |            | | XFER x     | |            | |            | |
| |            | |----------->|N| XFER y     | |            | |
| |            | |            |E|<-----------| |            | |
| |            | |            |T|            | |            | |
| |            | |            |W| DATA       | |            | |
| |            | |            |O|----------->| |            | |
| |            | | DATA       |R| RCV x      | |            | |
| |            | |<-----------|K|<-----------| | (ignore)   | |
|U| ICONN +    | | RCV y      | |            | |            |U|
|S|<-----------|M|----------->|M|            |M|            |S|
|E|            |L| XFER z     |A|            |L|            |E|
|R|            |P|----------->|C|            |P|            |R|
| |            | |            |H| DATA       | |            | |
| |            | |            |I|----------->| |            | |
| |            | | TERM a     |N| RCV z      | | ICONN +    | |
| |            | |----------->|E|<-----------| |----------->| |
| |            | |            | | CLEAR a    | |            | |
| |            | | CLEAR a    | |----------->| |            | |
| |            | |<-----------| |            | |            | |
| |            | |            | |            | |            | |
| |            | |            | |            | |            | |
+-+            +-+            +-+            +-+            +-+
```

a,b	switched X.25 channels
x,y	connect request
z	positive connect confirmation

Fig. 3.3. : Message link establishment: CONTENTION
- matching parameters
- two active ECONN
- two X.25 channels
- one message link

```
<---HOST A (higher DTE@)---> <---------HOST B---------->

+-+        +-+         +-+        +-+        +-+
| |        | |         | |        | |        | |
| |ECONN(A)| |         | |        | |        | |
| |------->| |         | |        | |ECONN(A)| |
| |        | |INIT a   | |        | |<-------| |
| |        | |-------->| |        | |        | |
| |        | |         | |READY a | |        | |
| |        | |         | |------->| |        | |
| |        | |         | |INIT b  | |        | |
| |        | |READY a  | |<-------| |        | |
| |        | |<--------| |        | |        | |
| |        | |READY b  | |        | |        | |
| |        | |<--------| |READY b | |        | |
| |        | |         | |------->| |        | |
| |        | |XFER x   | |        | |        | |
| |        | |-------->|N|XFER y  | |        | |
| |        | |         |E|<-------| |        | |
| |        | |         |T|        | |        | |
| |        | |         |W|DATA    | |        | |
| |        | |         |O|------->| |        | |
| |        | |DATA     |R|RCV x   | |        | |
| |        | |<--------|K|<-------| |(ignore)| |
|U|ICONN - | |RCV y    | |        | |        |U|
|S|<-------|M|-------->|M|        |M|        |S|
|E|        |L|XFER z   |A|        |L|        |E|
|R|        |P|-------->|C|        |P|        |R|
| |        | |         |H|DATA    | |        | |
| |        | |         |I|------->| |        | |
| |        | |TERM a   |N|RCV z   | |ICONN - | |
| |        | |-------->|E|<-------| |------->| |
| |        | |         | |CLEAR a | |        | |
| |        | |         | |------->| |        | |
| |        | |         | |TERM b  | |        | |
| |        | |CLEAR a  | |<-------| |        | |
| |        | |<--------| |        | |        | |
| |        | |CLEAR b  | |CLEAR b | |        | |
| |        | |<--------| |------->| |        | |
+-+        +-+         +-+        +-+        +-+
```

a,b	switched X.25 channels
x,y	connect request
z	negative connect confirmation

Fig. 3.4. : Message link establishment: CONTENTION
- mismatching parameters
- two active ECONN
- two X.25 channels
- one message link

```
<---HOST A (higher DTE@)---> <---------HOST B---------->

+-+          +-+           +-+          +-+          +-+
| |          | |           | |          | |          | |
| | ECONN(A) | |           | |          | | ECONN(A) | |
| |--------->| |           | |          | |<---------| |
| |          | | INIT a    | | INIT b   | |          | |
| |          | |---------->| |<---------| |          | |
| |          | |           |X|(COLL.)   | |          | |
| |          | |           | |          | |          | |
| |          | | READY a   | | CLEAR b  | |          | |
| |          | |<----------| |--------->| |          | |
| |          | |           | | READY a  | |          | |
| |          | |           | |--------->| |          | |
| |          | |           | | INIT c   | |          | |
| |          | |           | |<---------| |          | |
| |          | |           |N| READY c  | |          | |
| |          | | XFER x    |E|--------->| |          | |
| |          | |---------->|T|          | |          | |
| |          | | READY c   |W| DATA     | |          | |
| |          | |<----------|O|--------->| |          | |
| |          | |           |R| RCV x    | |          | |
| |          | |           |K|<---------| |(ignore)  | |
| |          | |           | | XFER y   | |          | |
| |          | | DATA      | |<---------| |          | |
|U|          | |<----------| |          | |          |U|
|S|          |M| RCV y     |M|          |M|          |S|
|E| ICONN +  |L|---------->|A|          |L|          |E|
|R|<---------|P| XFER z    |C|          |P|          |R|
| |          | |---------->|H|          | |          | |
| |          | |           |I| DATA     | |          | |
| |          | | TERM a    |N|--------->| |          | |
| |          | |---------->|E| RCV z    | | ICONN +  | |
| |          | |           | |<---------| |--------->| |
| |          | | CLEAR a   | | CLEAR a  | |          | |
| |          | |<----------| |--------->| |          | |
| |          | |           | |          | |          | |
| |          | |           | |          | |          | |
+-+          +-+           +-+          +-+          +-+
```

a,b,c	switched X.25 channels
x,y	connect request
z	positive connect confirmation

Fig. 3.5. : Message link establishment: CONTENTION
- matching parameters
- two active ECONN
- two X.25 channels
- one message link
- X.25 Call Collision

Besides CONTENTION there can be another problem when establishing message links.
Assuming user 1 of host A intends to install a message link to user 1 of host B , and user 2 of host B wants to establish a message link to user 2 of host A. For both message links the data rate is high and the communication takes a long time, so it is useful to allocate a network connection between host A and host B which is named permanent logical channel (X.25 notation). The installation of this permanent channel is done at network subscription time, so no X.25 call packets are exchanged: the network connection over this channel is inherent.
Now it is possible (only if users have the ability to allocate permanent channels) that both sides of the two planned message links allocate the same permanent channel and transfer messages over it.
This conflict is named COLLISION.

```
       +--+ < - - - - - - - M L - - - - - - > +--+
USER 1 |  |                                   |  | USER 1
       +--+                                   +--+
          |  +--+       NETWORK         +--+  |
          +->| M|   +---------------+   |M |<-+
HOST A       | L|<->|PERMANENT CHANN|<->|L |       HOST B
          +->| P|   +---------------+   |P |<-+
          |  +--+                       +--+  |
       +--+                                   +--+
USER 2 |  |                                   |  | USER 2
       +--+ < - - - - - - - M L - - - - - - > +--+
```

In order to make efficiently use of a level 4/5 protocol like MLP it is necessary that the protocol-overhead tends to a minimum. This can be achieved if the number of theoretically possible conflicts is low.
Therefore the MLP should be designed in the way that no collision occurs.
This fact is already guaranteed if no user of the MLP can allocate by the exported network service INIT the network resource "permanent logical channel".
So all the message links using switched virtual circuits are installed for a finite time interval and then released whereas those using permanent channels are installed for an infinite time interval at system generation time and never released.
This assumption makes protocol handling easier and reduces the overhead without loss of efficiency, because there is no user who must transfer his data through the network by permanent X.25 channels.

3.2 Closing the message link

After having finished the data transfer to the remote partner and the communication line is no longer needed one of the connected partners must close the message link.
This is achieved by issuing the exported MLP service EDISC.
The remote partner is informed and has to answer with a confirmation message. After receiving this acknowledgement the associated installed switched logical channel has to be cleared. (see fig. 3.6.)
In the case of a collision (simultaneously issued EDISC) no confirmation message needs to be sent, the eventually resulting "clear collision" at X.25 level 3 is resolved by the network machine (see fig. 3.7.).

```
<---------- HOST A --------> <------- HOST B ---------->

+-+    .     +-+    .     +-+    .     +-+    .     +-+
| |    .     | |    .     | |    .     | |    .     | |
| |    .     | |    .     |N|    .     | |    .     | |
| | EDISC    | |          |E|          | |          | |
| |--------->| |          |T|          | |          | |
| |          | | XFER x   |W|          | |          | |
| |          | |--------->|O|          | |          | |
| |          | |          |R| DATA     | |          | |
| |          | |          |K|--------->| |          | |
|U|          | |          | | RCV x    | | IDISC    |U|
|S|          |M|          |M|<---------|M|--------->|S|
|E|          |L|          |A| XFER y   |L|          |E|
|R|          |P|          |C|<---------|P|          |R|
| |          | | DATA     |H|          | |          | |
| |          | |<---------|I|          | |          | |
| | IDISC    | | RCV y    |N|          | |          | |
| |<---------| |--------->|E|          | |          | |
| |          | |          | |          | |          | |
| |          | | TERM     | |          | |          | |
| |          | |--------->| |          | |          | |
| |          | |          | | CLEAR    | |          | |
| |          | | CLEAR    | |--------->| |          | |
| |          | |<---------| |          | |          | |
| |          | |          | |          | |          | |
+-+          +-+          +-+          +-+          +-+
```

x disconnect message
y disconnect confirmation message

Fig. 3.6. : Closing ML

```
<---------- HOST A --------> <------- HOST B ---------->

+-+     .     +-+     .     +-+     .     +-+     .     +-+
| |     .     | |     .     | |     .     | |     .     | |
| |     .     | |     .     |N|     .     | |     .     | |
| | EDISC     | |           |E|           | | EDSISC    | |
| |---------->| |           |T|           | |<----------| |
| |           | | XFER x    |W| XFER x    | |           | |
| |           | |---------->|O|<----------| |           | |
| |           | | DATA      |R| DATA      | |           | |
| |           | |<----------|K|---------->| |           | |
|U|           | | RCV x     | | RCV x     | |           |U|
|S| IDISC     |M|---------->|M|<----------|M| IDISC     |S|
|E|<----------|L|           |A|           |L|---------->|E|
|R|           |P| TERM      |C| TERM      |P|           |R|
| |           | |---------->|H|<----------| |           | |
| |           | | CLEAR     |I| CLEAR     | |           | |
| |           | |<----------|N|---------->| |           | |
| |           | |           |E|           | |           | |
| |           | |           | |           | |           | |
| |           | |           | |           | |           | |
| |           | |           | |           | |           | |
| |           | |           | |           | |           | |
+-+           +-+           +-+           +-+           +-+
```

x   disconnect message

Fig. 3.7. : Closing ML :Collision

3.3 Choosing the mode of communication

This protocol of MLP is very simple and realized by the MLP services EOPEN, ECLOSE, IOPEN, ICLOSE.
Having established a message link, the two partner-processes have to stipulate the type of data they want to transfer. Only if both processes agree it is guaranteed that both ends of the message link use the same kind of interpreting the received data.
Two modes of communication are available:

- Data in accordance with VTP and MLP
- Data in accordance with MLP only

The meaning of these modes is illustrated in the following figure:

```
+-----------------------------------------+
|               U S E R                   |
+-----------------------------------------+
             |                 |
             |                 | MODE 1
             | MODE 2          |
             |                 |
             |                 |
             |                 |
             |    +------------V----------+
             |    |        V T P          |
+------------V----+-----------------------+
|                 M L P                   |
+-----------------------------------------+
                    |
                    |
+-------------------V---------------------+
|            N E T W O R K                |
+-----------------------------------------+
```

Mode 1 means that the entrance of the user to the MLP is through VTP , and mode 2 uses none protocol above MLP.

Sometimes the nesting of communication modes is desirable and easy to implement. But if nesting is not mandatory the whole protocol can be omitted, because it is able to insert the communication mode as parameter into the ECONN service.

```
<---------- HOST A --------> <------- HOST B ---------->

+-+          +-+           +-+           +-+          +-+
| |          | |           | |           | |          | |
| | EOPEN    | |           |N|           | |          | |
| |--------->| | XFER x    |E|           | |          | |
| |          | |---------->|T|           | |          | |
| |          | |           |W| DATA      | |          | |
| |          | |           |O|---------->| |          | |
| |          | |           |R| RCV x     | | IOPEN    | |
| |          | |           |K|<----------| |--------->| |
|U|          | |           | |           | |          |U|
|S|          |M|           |M|           |M| EOPEN    |S|
|E|          |L|           |A|           |L|<---------|E|
|R|          |P|           |C| XFER x    |P|          |R|
| |          | | DATA      |H|<----------| |          | |
| |          | |<----------|I|           | |          | |
| | IOPEN    | | RCV x     |N|           | |          | |
| |<---------| |---------->|E|           | |          | |
| |          | |           | |           | |          | |
+-+          +-+           +-+           +-+          +-+
```

x   open message

Fig. 3.8 : Open a communication mode

```
<---------- HOST A --------> <------- HOST B ---------->

+-+          +-+           +-+           +-+          +-+
| |          | |           | |           | |          | |
| | ECLOSE   | |           |N|           | |          | |
| |--------->| | XFER x    |E|           | |          | |
| |          | |---------->|T|           | |          | |
| |          | |           |W| DATA      | |          | |
| |          | |           |O|---------->| |          | |
| |          | |           |R| RCV x     | | ICLOSE   | |
| |          | |           |K|<----------| |--------->| |
|U|          | |           | |           | |          |U|
|S|          |M|           |M|           |M| ECLOSE   |S|
|E|          |L|           |A|           |L|<---------|E|
|R|          |P|           |C| XFER x    |P|          |R|
| |          | | DATA      |H|<----------| |          | |
| |          | |<----------|I|           | |          | |
| | ICLOSE   | | RCV x     |N|           | |          | |
| |<---------| |---------->|E|           | |          | |
| |          | |           | |           | |          | |
+-+          +-+           +-+           +-+          +-+
```

x   close message

Fig. 3.9. : Close a communication mode

3.4 Data Transferring

After having established a message link and having chosen the communication mode (explicitly or implicitly) the "owners" of the ML are able to transfer data across this ML to the subscriber-partner.
The quantum which can be delivered to the MLP is called "message", the length of which must not exceed a maximum value of 256 octets.
Sending a message means issuing the exported MLP service ETRANS. The parameter "success" of this service signals the success: if there was no possibility to take over the message into the area of MLP the user has to wait for the imported MLP service CONTINUE and then issuing once more ETRANS with the same message, otherwise the user can go on with data transferring.
Receiving a message generally means issuing the exported MLP service EREC after having been signalled by the imported MLP service MSG.

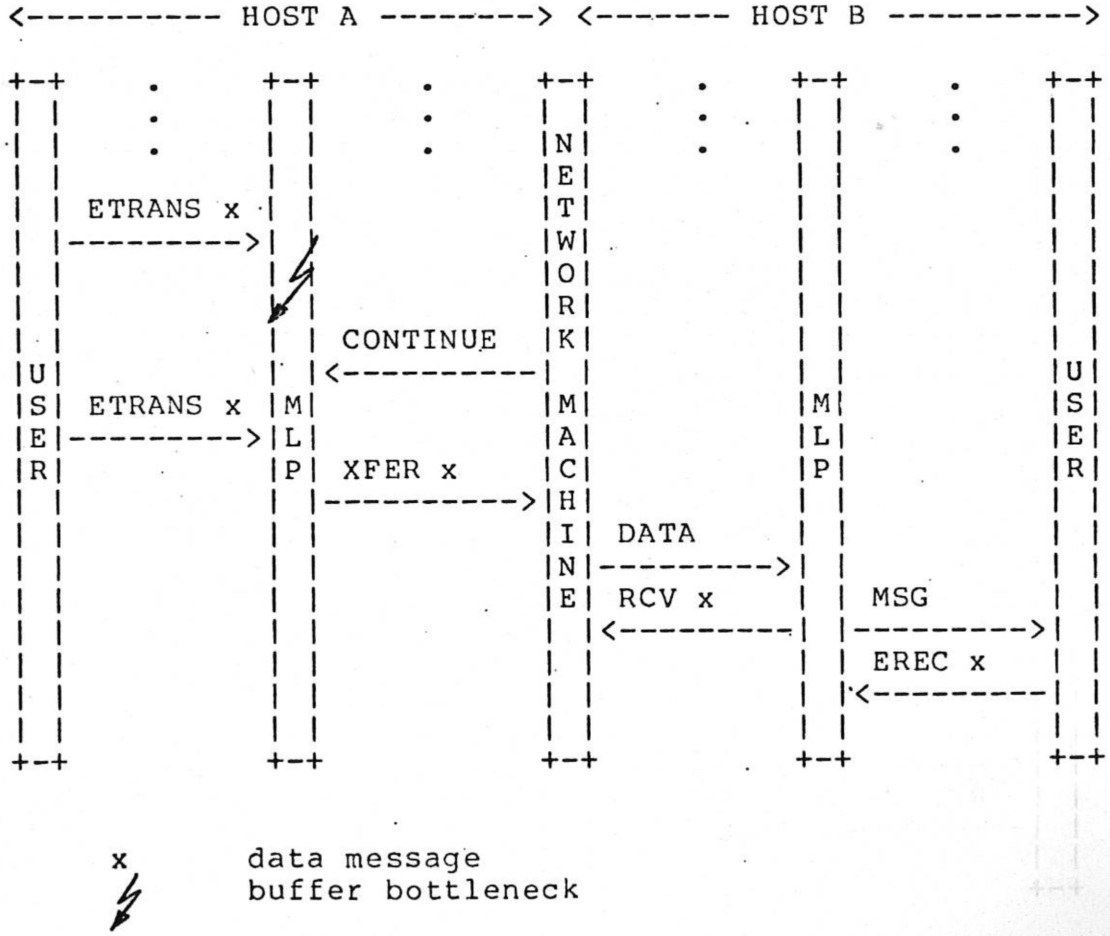


Fig. 3.10: Data transferring and receiving

3.5 Interrupt Protocol

The interrupt mechanism on a message link allows a user to transmit one octet of data to his correspondent outside the normal flow control procedure which governs the packets of data inside the network machine.
This does mean, however, the interrupt packet could be transferred through the network with a higher priority than data packets and is able to overtake them.
To overcome this uncertainity MLP has to provide a tool for synchronizing the data exchange phase of the two correspondents after an interrupt message was sent.
Achieving this MLP has to signal the correct sequence of data-messages to the remote partner by a separate message called "MARK", which is inserted into the stream of data-messages at the same time when the interrupt message is sent. The remote MLP does the same when receiving the interrupt message. Then the processes are synchronized if both have got the mark message from the correspondent: they can continue exchanging messages or do anything other.

```
<---------- HOST A --------> <------- HOST B ---------->

+-+    .     +-+     .     +-+    .     +-+    .     +-+
| |    .     | |     .     | |    .     | |    .     | |
| |    .     | |     .     |N|    .     | |    .     | |
| | ETRANS u | |           |E|          | |          | |
| |--------->| | XFER u    |T|          | | ETRANS v | |
| |          | |---------->|W|          | |<---------| |
| | EIRPT    | |           |O| XFER v   | |          | |
| |--------->| | IRPT      |R|<---------| |          | |
| |          | |---------->|K| IRPT     | |          | |
|U|          | |           | |--------->| | IIRPT    |U|
|S|          |M| DATA      |M|          |M|--------->|S|
|E|          |L|<----------|A| DATA     |L|          |E|
|R| MSG      |P| RCV v     |C|--------->|P|          |R|
| |<---------| |---------->|H| RCV u    | | MSG      | |
| | EREC v   | | XFER x    |I|<---------| |--------->| |
| |--------->| |---------->|N| XFER y   | | EREC u   | |
| |          | | DATA      |E|<---------| |<---------| |
| |          | |<----------| | DATA     | |          | |
| | MSG      | | RCV y     | |--------->| |          | |
| |<---------| |---------->| | RCV x    | | MSG      | |
| | EREC y   | |           | |<---------| |--------->| |
| |--------->| |           | |          | | EREC x   | |
| |          | |           | |          | |<---------| |
+-+          +-+           +-+          +-+          +-+
```

u,v	data messages sent prior to interrupt message
x,y	mark messages for re-synchronisation

Fig. 3.11. : Interrupt Protocol

4. MLP-Automaton

The following state diagramms show the actions taken by MLP in receipt of messages from the remote MLP.
Five major states of MLP are available:

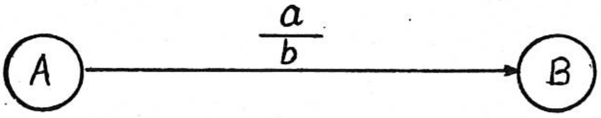
S1	Network channel, but no message link
S2	Connect request message sent
S3	Wait for connect request message
S4	Message link established
S5	Disconnect message sent

There are three substates of S4:

S4/1	Open message sent, wait for confirmation
S4/2	Close message sent, wait for confirmation
S4/3	Wait for mark message

State diagram conventions:

- states are represented by circles containing the state number as explained above
- input a in state A causing output b and transition to state B is represented by



- $\frac{a}{-}$ indicates input a causes no output
- $\frac{-}{a}$ no input but output a
- cont1 indicates contention, lower DTE@
- cont2 indicates contention, higher DTE@

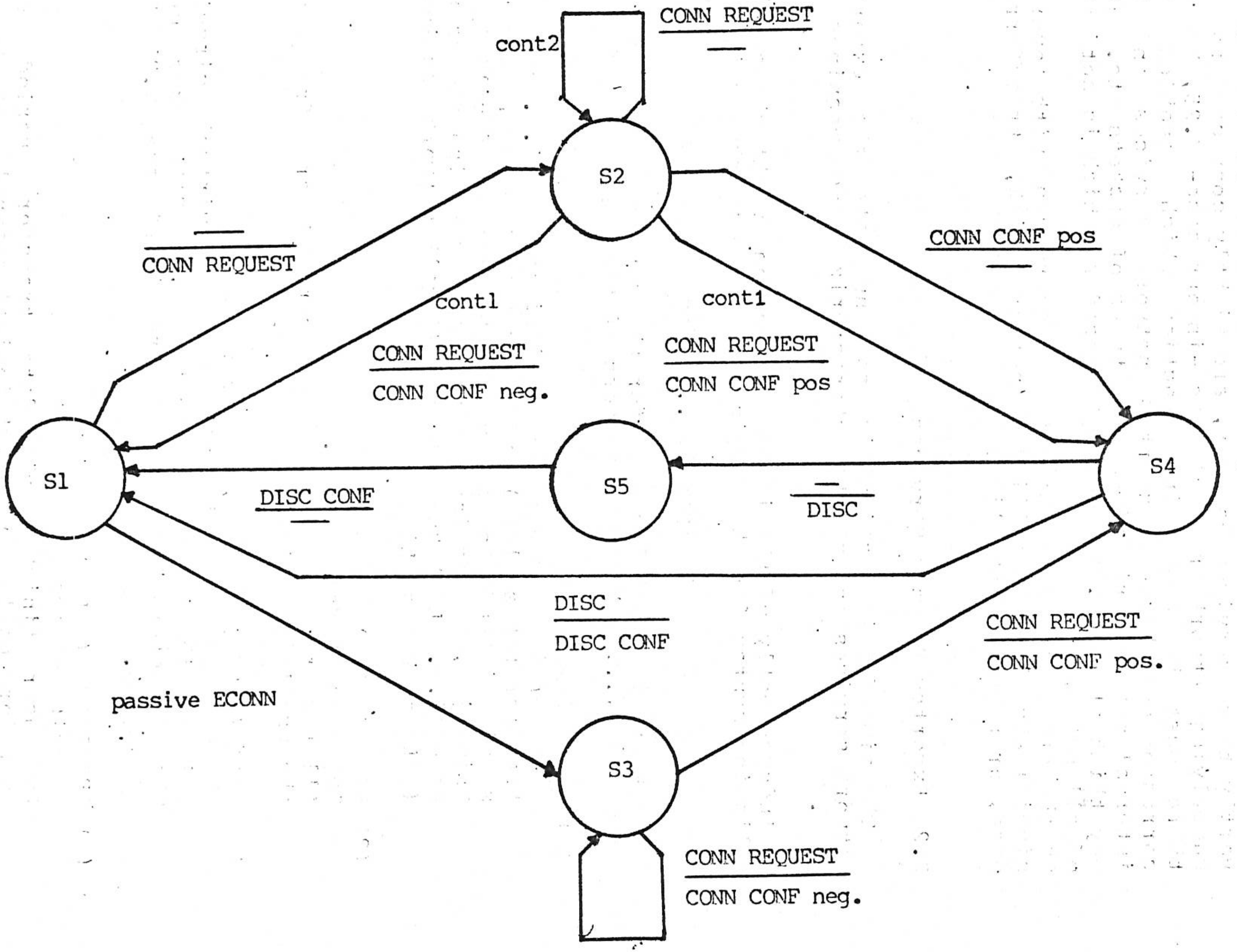


Fig. 4.1.: FINITE STATE MACHINE MLP (ESTABLISHING AND CLOSING A MESSAGE LINK)

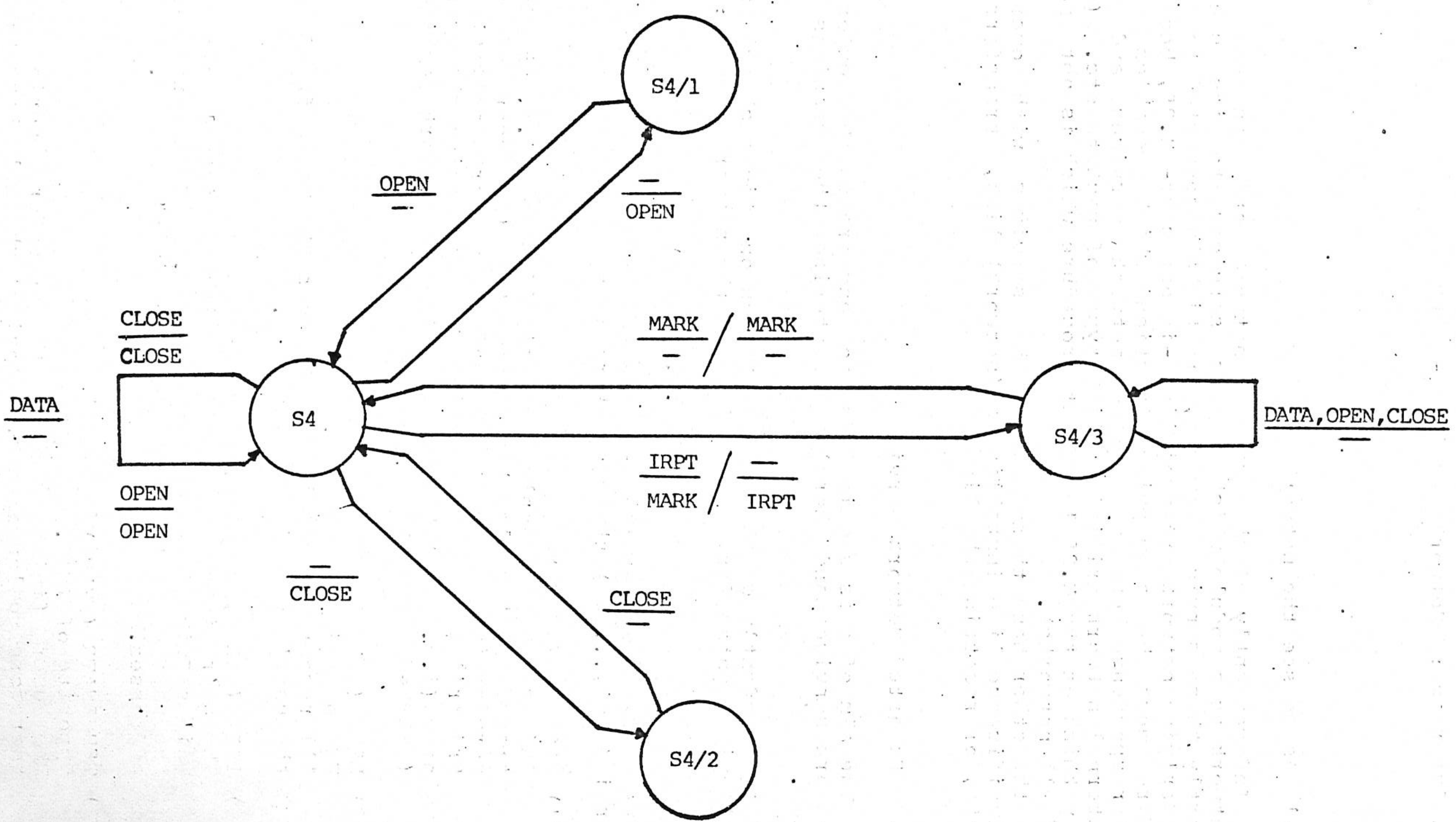


Fig. 4.2.: FINITE STATE MACHINE MLP (DATA TRANSFER, INTERRUPT, CM-PROTOCOL)