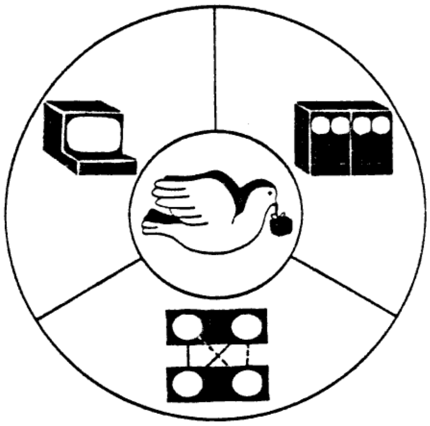


CENTERNET

SYSTEM SPECIFICATIONS

rc COMPUTER
A/S REGNECENTRALEN af 1979

HEADQUARTER: LAUTRUPBJERG 1 - DK 2750 BALLERUP - DENMARK
Phone: + 45 2 65 80 00 - Cables: rcbalrc - Telex: 35 214 rcbaldk

RCSL No : 43-GL11740

Edition : August 1982

Author : Per Høgh (ed.)

(218 printed pages)

Foreword

First edition : RCSL No. 43-GL10190

Second edition : RCSL No. 43-GL11740
The text has been changed on the following points

- chapter 1 : editorial corrections.
- chapter 2 : editorial corrections and correction of the Network Control Centre definition.
- chapter 3 : editorial corrections.
- chapter 4 : updated to describe the RC3502C system.
- chapter 5 : editorial corrections, MLP and SC addressing protocol removed. The NC subsections have been made uniform.
- chapter 6 : editorial corrections, overview system inserted.
- chapter 7 : editorial corrections.
- chapter 8 : editorial corrections. Description of Host Port Protocol changed.
- chapter 9 : editorial corrections.
- chapter 10 : updated to describe the RC3502C system. Subsection concerning remote load and HCF/SCF is incomplete. (Refer to note below).
- chapter 11 : Due to the fact that hardware changes will have major influence on the capacity considerations, this chapter has been removed, and appears as an independent document listed in appendix A.

All documentation are no longer part (as appendices) of this System Specification. Appendix A is structure in a way, which make it easy to find the needed references.

All figures are changes, so they reflect the logical structure of the actual implemented and described software/-hardware.

Note:

The subsections concerning Remote Load and HCF/SCF is incomplete, and will not be available in complete description until January 1983.

Per Høgh
A/S REGNECENTRALEN af 1979, August 1982.

This page is intentionally left blank.

CONTENTS PAGE

CONTENTS (continued) PAGE

CONTENTS (continued) PAGE

CONTENTS (continued) PAGE

CONTENTS (continued) PAGE

CONTENTS (continued) PAGE

CONTENTS (continued) PAGE

CONTENTS (continued) PAGE

1. INTRODUCTION

This specification covers a description of the functional and operational characteristics of the CENTERNET Communication Network.

The network interconnects various types of equipment and mainframes thereby enabling universities and other institutes for education and/or research to share the available computing facilities.

The individual chapters apply to different readers, depending on their different needs:

> Chapter 2 gives a general description of the units constituting CENTERNET. This chapter should be read before any other part of the specification.
>
> In chapter 3 the ISO Reference Model of Open Systems Interconnection is applied as a framework for an architectural description of CENTERNET.
>
> Chapter 4 relates the architecture, given in chapter 3, to the actual implementation including hardware as well as software.
>
> The individual software-modules, including protocols used for peer communication as well as services offered-/utilized by these modules, are outlined in chapter 5.
>
> The network control functions disposed for the operation of CENTERNET are described in chapter 6.
>
> The present possibilities for connection to CENTERNET are depicted in chapters 7 and 8. Chapter 7 concerns the terminal support whereas the connection of an RC8000 Host is given in chapter 8.
>
> Functions performed in this host are collected in chapter 9.
>
> Chapter 10 gives an overview of the tools for network management and maintenance.

For a clarification of terminology, Appendix B can be used as a handbook.

Detailed information about the communication protocols being utilized, the service offered by each module and maintenance documentation may be found in the CENTERNET Documentation (Appendix A.1).

This page is intentionally left blank.

2. GENERAL SYSTEM OUTLINE

A main purpose of the CENTERNET network is to offer the possibility to interconnect various types of equipments and mainframes, and even different network systems offered by other vendors. Figure 2.1 gives an overview of the general structure of CENTERNET.

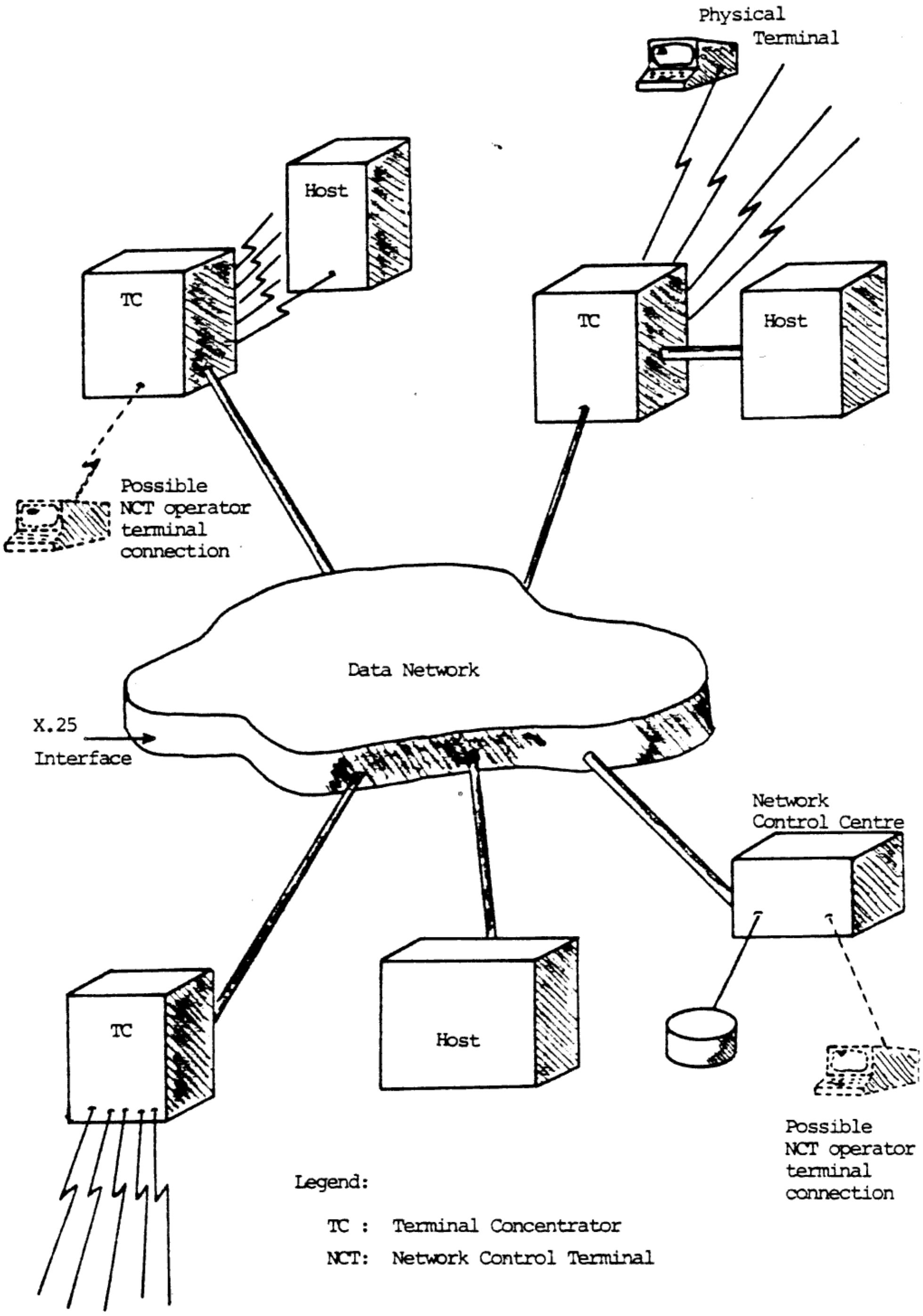


Figure 2.1 CENTERNET General Structure

This figure indicates that CENTERNET is build up based on four units, a Data Network (section 2.1), Terminal Concentrators (section 2.2), Hosts (section 2.3) and a Network Control Center (section 2.4).The last unit is more a logical unit, as the network control function may be/are distributed on several Hosts and Terminal Concentrators.

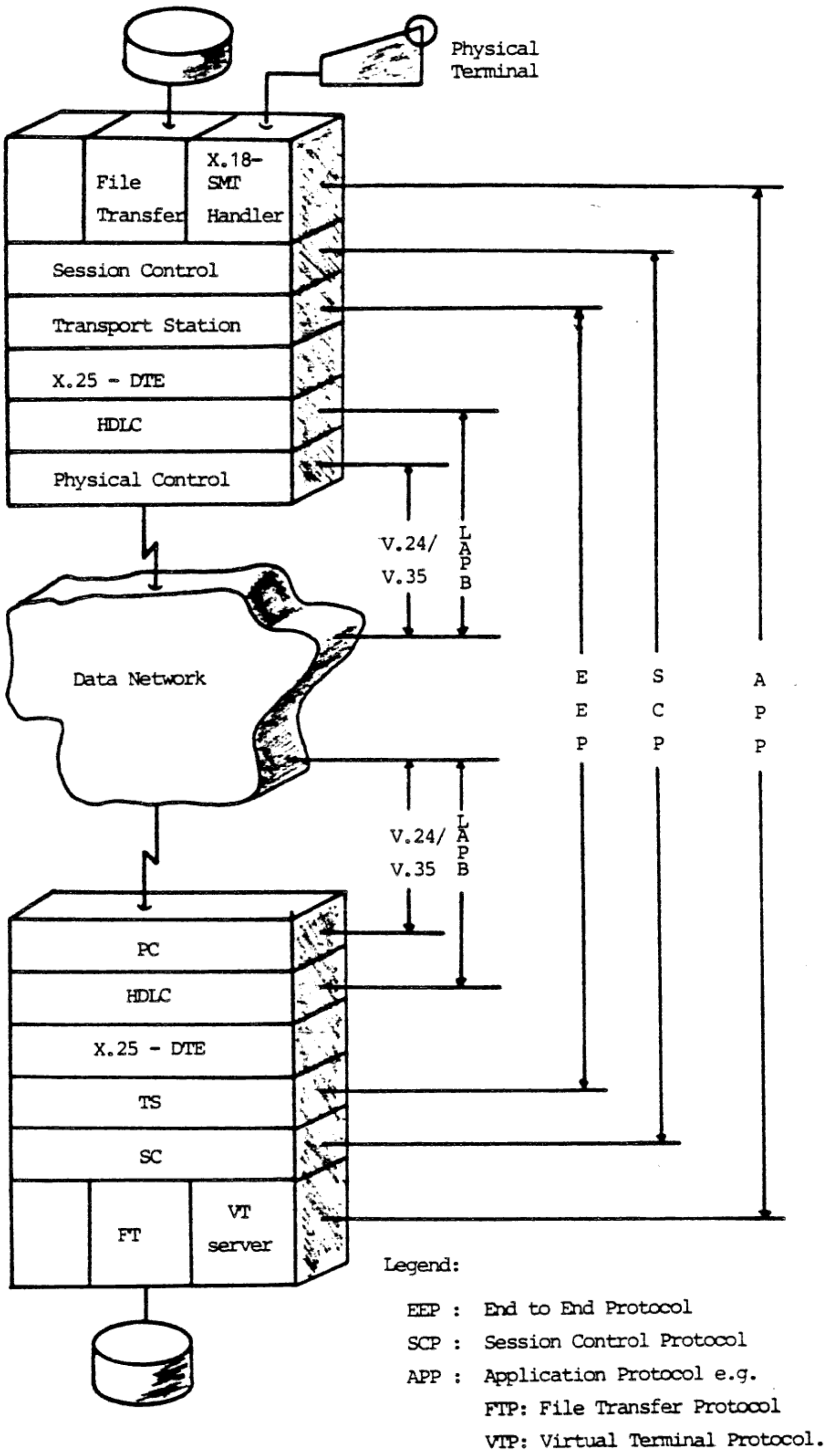


Figure 2.2 CENTERNET Protocol Hierachy.

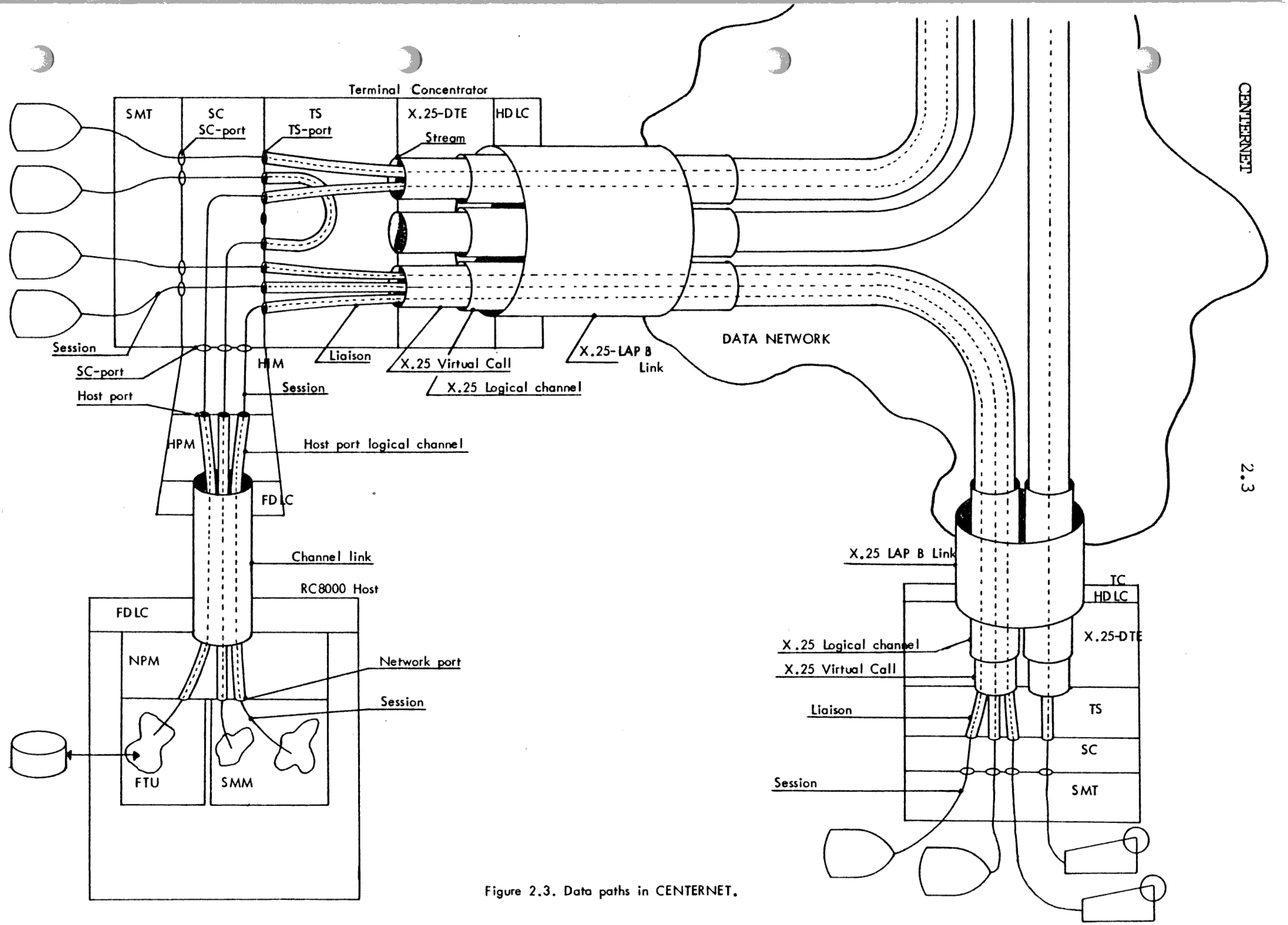


Figure 2.3. Data paths in CENTERNET.

The CENTERNET network can be described utilizing the ISO Reference Model of Open Systems Interconnection - Basic reference model (ISO/DIS 7498, April 1982, ref. (122)) as described in chapter 3. This model establishes a framework for describing a communication network.

In CENTERNET international standard protocols/interfaces have been used in case of existence else widely used protocols have been chosen.

Figure 2.2 shows the module structure of CENTERNET network and between which modules the individual protocols are used. Figure 2.3 shows the 'data paths' in CENTERNET.

2.1 The Data Network

The Data Network is a packet switched network offering an X.25 interface. The reference document used for the X.25 interface is CCITT Recommendation X.25, ref. (102).

At present a private data network (PAXNET, section 5.1 & 5.2) offering the X.25 interface is used.

At the moment the National PTT offers a data network with an X.25 interface connection, this public data network may be used instead of the private network without any changes in the CENTERNET network.

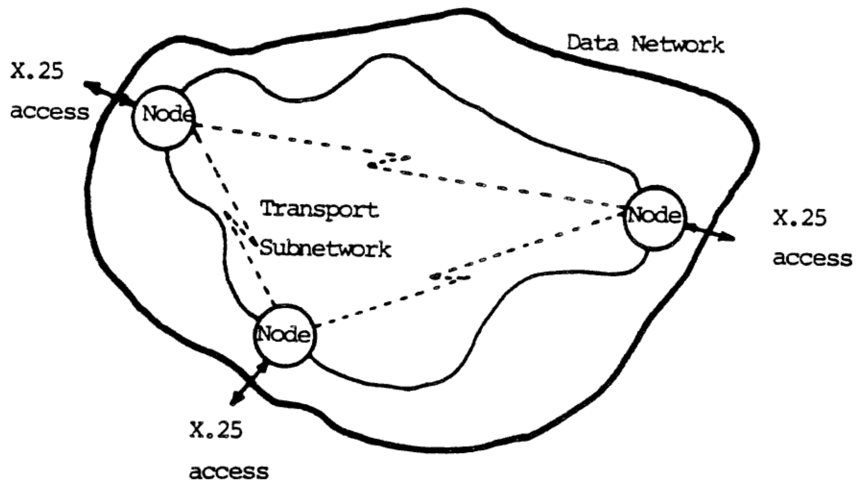


Figure 2.4 The Data Network.

2.2 Terminal Concentrators

The term Terminal Concentrator, TC, covers a set of software/hardware modules, handling different parts of the functions of the TC.

The main purpose of a Terminal Concentrator is to multiplex and connect different types of terminals to the Data Network, but also other functions may be provided. All the functions may be grouped into:

- terminal multiplexing and handling
- file transfer
- remote job entry
- host connection through a host interface.

Figure 2.5 shows an example of the structure of a TC.

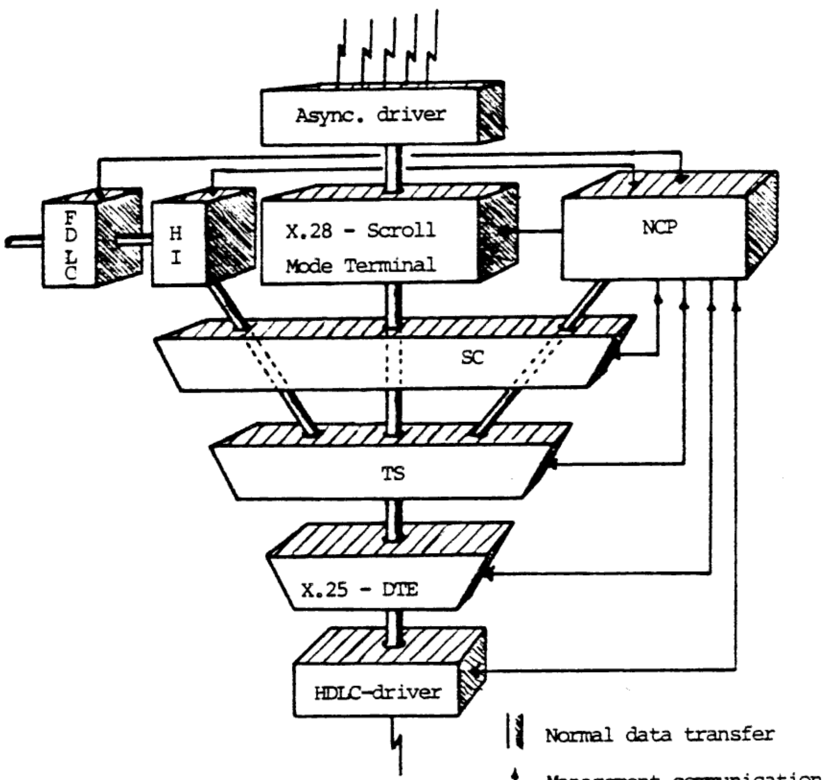


Figure 2.5 Example of Terminal Concentrator.

The HDLC driver, the X.25-DTE-, TS-, SC-, and NCP-modules are always present in a TC. These are the modules supporting the basic network functions. All other modules belong to one of the above-mentioned groups, and may be present when needed. The multiplexing and addressing functions are performed by the Session Control (SC) and the Transport Station (TS). The access to the Data Network is always through the X.25-DTE.

The terminal multiplexing-, the file transfer-, and the remote job entry functions are all typical application functions. Each of these types uses its own application protocol (figure 2.2).

Different types of physical terminals may be connected to the TC, each type via a certain terminal protocol. For each protocol a Terminal Module is defined (see figure 7.1 & 7.2), and these Terminal Modules handle the physical terminals according to the specific protocol and perform the conversion to the network standard terminal protocol, the CENTERNET Virtual Terminal Protocol, ref. (2).

2.3 Host Interfaces

As indicated in figure 2.1 host computers may be connected to CENTERNET in three different ways:

(1) through a Terminal Concentrator
(2) directly to the Data Network
(3) as a terminal to the Terminal Concentrator.

In all three cases software modules/hardware components named Host Interface (HI) are necessary for the connection. Each defined Host Interface is described in a chapter later in this System Specification.

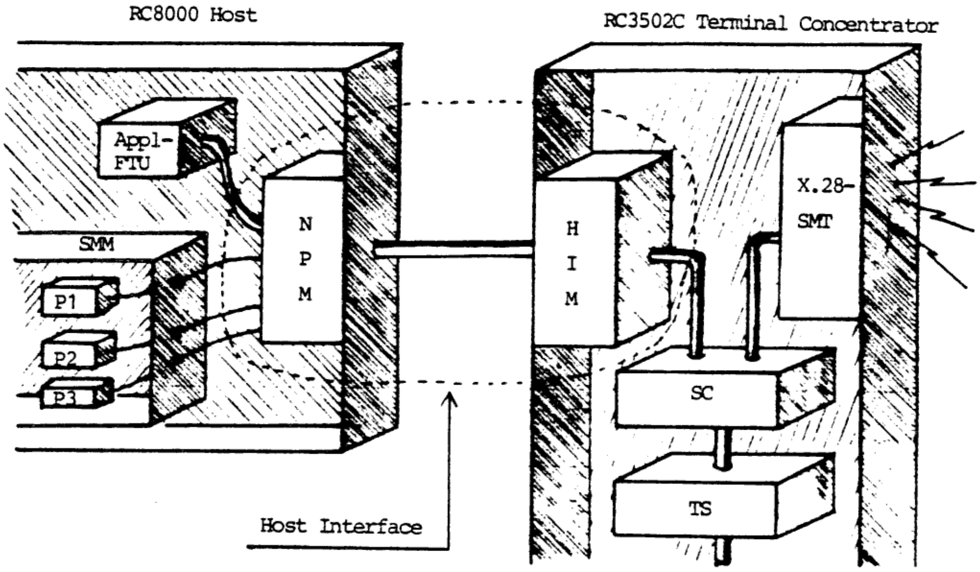


Figure 2.6 RC8000 Host Interface.

In method (1) a high speed line is used between the Terminal Concentrator and the Host, and the Host Interface is located both in the Host and in the TC, as shown in figure 2.6. This figure shows the connection of an RC8000 as a Host Computer. In the Host the module NPM acts as network interface and in the TC the module HIM acts as host interface. The RC8000 Host Interface is described in chapter 8.

A variant of method (1) is to use a gateway between CENTERNET and the network supported by the mainframe vendor. Figure 2.7 shows an example of connecting UNIVAC 1100 as a Host to CENTERNET.

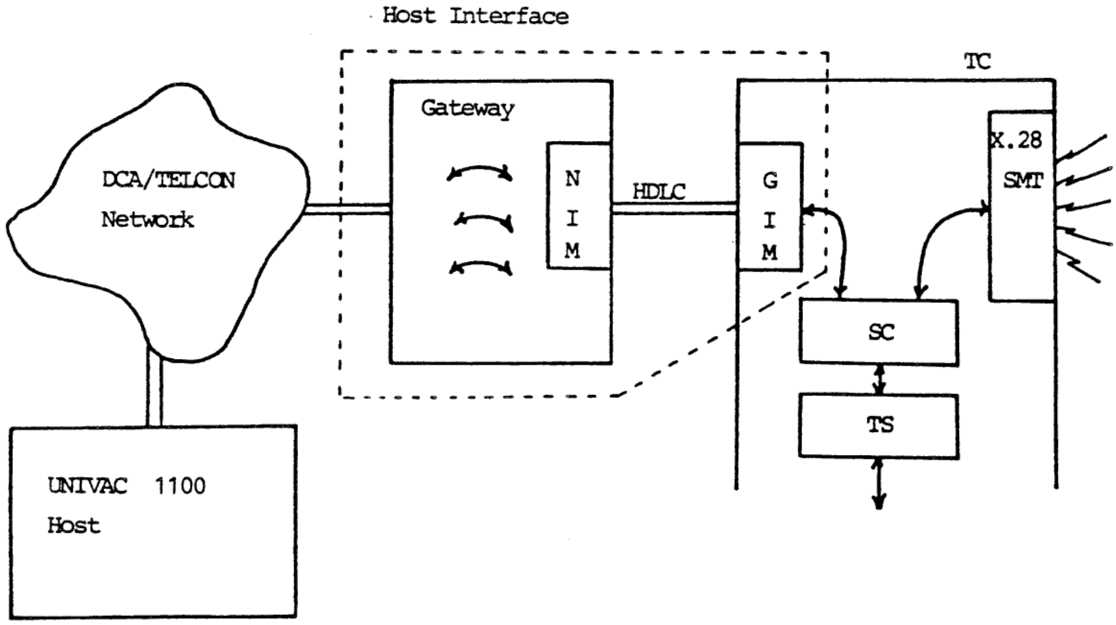


Figure 2.7 UNIGATE as Host Interface.

In method (2) all standard CENTERNET network software modules (protocol handlers, access modules etc.) will have to be placed in the host computer, so the Host Interface is located totally in the Host.

Method (3) is mainly intended for connection of medium/small host computers, because the line will be of a medium/low speed type. The method is based upon the host computer simulating a terminal. The type of the terminal may be any of those supported by the specific TC. The Terminal Concentrator sees the host computer just as a normal terminal. No specific host interface software is used, except that for terminal simulation.

2.4 Network Control

The Network Control System of CENTERNET, as mentioned above, may be distributed on several geographical locations. In this section the NC-system will be described as one logical unit and as a framework for network monitoring, - controlling, and management.

Monitoring comprises the functions of report/statistical information retrieval whereas controlling is the direct interaction in network performance. Management is the administration/manipulation of the monitoring- and control functions, and the retrieved information. Figure 2.8 shows the trisection of the Network Control System.

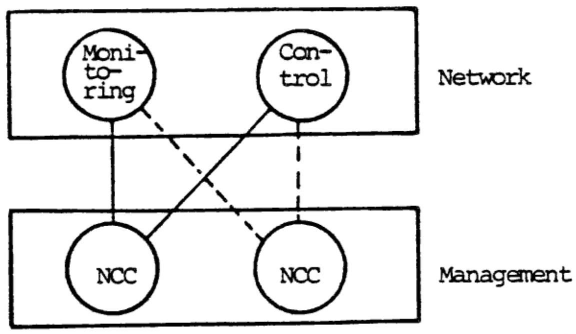


Figure 2.8 Trisection of Network Control System.

The Network Control consists of the following subsystems

- Network Control Centres:
  including:
  - Network Control Centre Nucleus (NCC)
  - Network Control Terminal (NCT)
  - Network Control Utilities
- Network Control Probes (NCP)
- Local Control Probes (LCP).

The Network Control Centre, the 'master', is located in an RC8000. It communicates with the 'slaves' (NCP's) using a Supervisor Protocol. An Network Control Centre consists of a nucleus (NCC), Operator Terminal (NCT), and a number of utilities as indicated in figure 2.9.

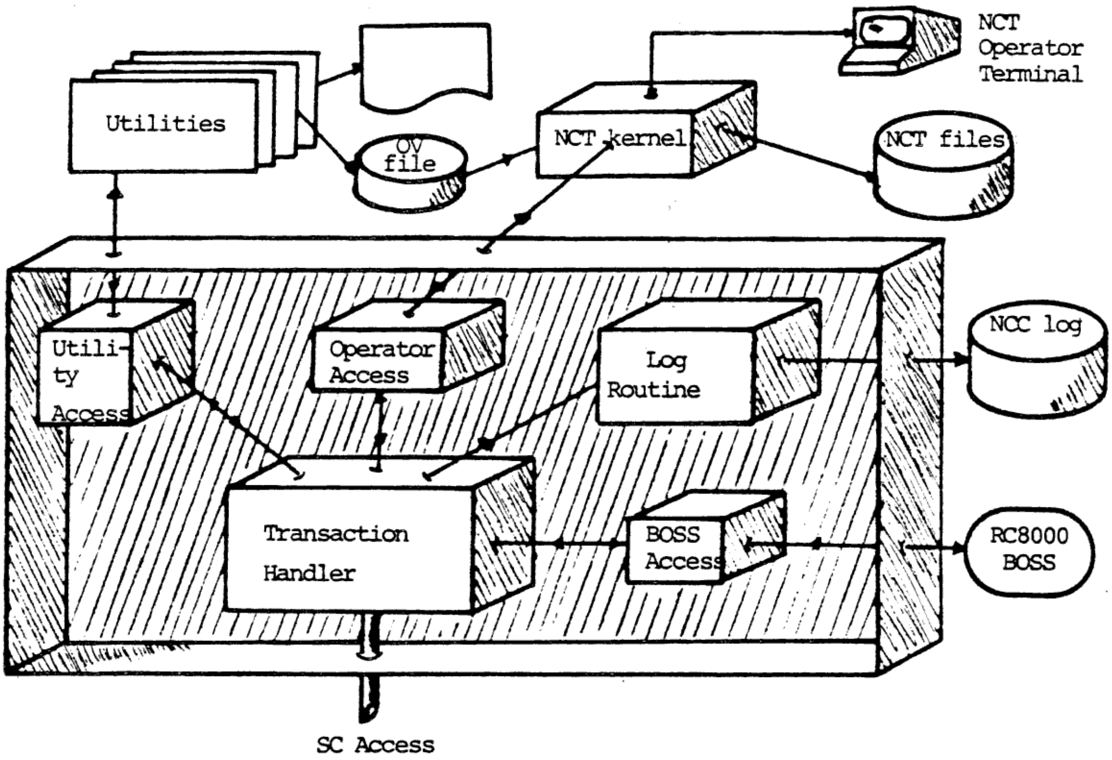


Figure 2.9 Structure of NCC.

An NCP is located in each subsystem of the network (e.g. a terminal concentrator), and performs the control function and the monitoring by accessing the individual program modules via the LCP's (figure 2.10).

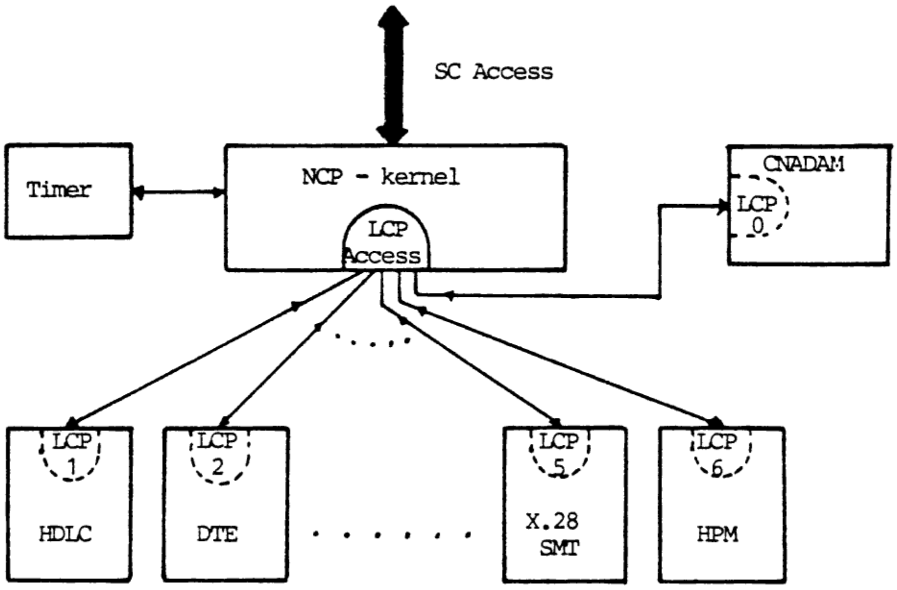


Figure 2.10 Structure and environment of NCP.

The NCC- and NCP- modules may be viewed as special applications. Applications because they exchange information using a normal network service (lettergram) and special because they use preallocated/dedicated Session Control ports.

2.5 Addressing in CENTERNET

The subject of addressing in CENTERNET may be split into two, access-addressing (section 2.5.1) and network user identification (section 2.5.2).

2.5.1 Access-addressing

As stated above in this chapter and in chapter 3 the CENTERNET network is build based upon a layered structure. For each layer/module a service interface is defined. These service interfaces comprise service-primitives and access-addresses. Figure 2.11 shows access-addresses identification in the Terminal Concentrator and the RC8000 Host Interface.

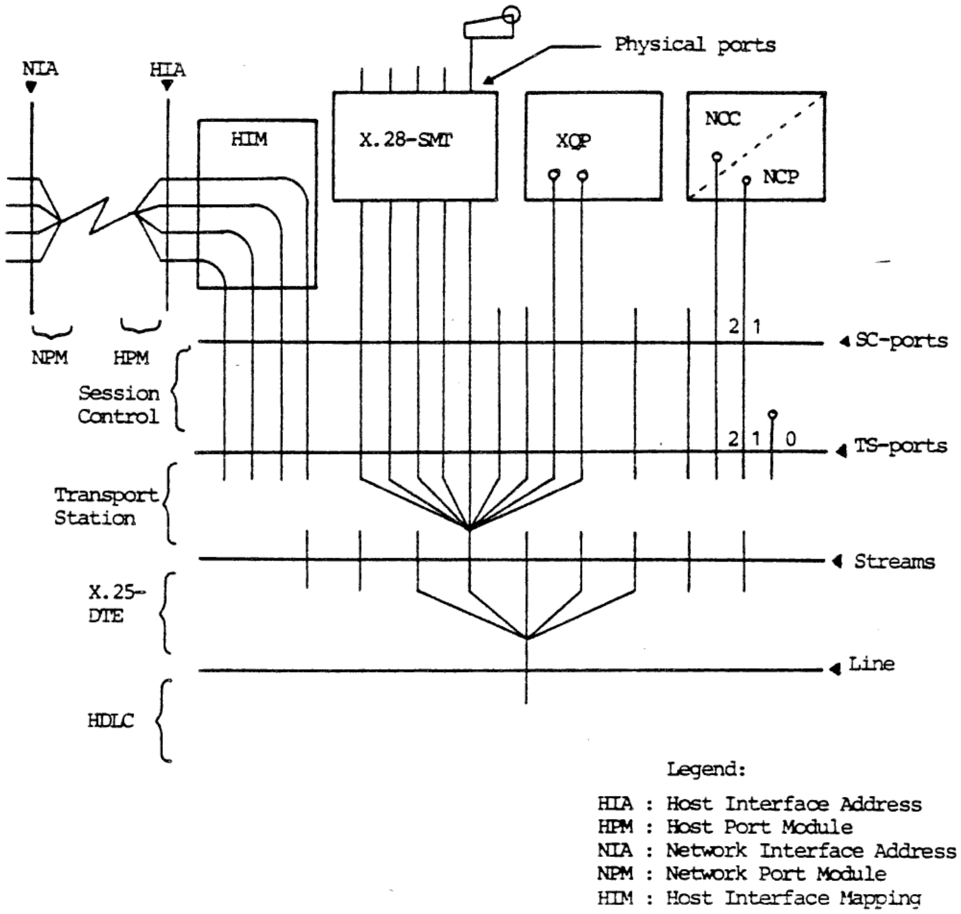


Figure 2.11 Access-addresses in TC and RC8000 HI.

The X.25-DTE module accesses the HDLC module on a line basis, and a line is identified by a number between 1 and 4.

At the X.25-DTE module's service interface a data path is identified by a streamnumber between 0 and 255. Within the X.25-DTE module a one-to-one correspondence between a stream-number and a logical channel number (LCN) exists.

A TS port and a SC port are always equal and identified by the same number, port number. Only one exception exists, TS port zero is reserved for Session Control usage. I.e. the following limits exist:

$0 \leq$ TS port number < 32767
$1 \leq$ SC port number ≤ 32767.

The SC ports 1 and 2 are reserved for Network Control purpose. Port 1 is allocated to the NCP and port 2 to the NCC.

A physical terminal is identified at the X.28-SMT module's "service interface" by a symbolic portid.

This symbolic portid consists of 2 parts, an alfa character string (1 - 7 characters) and a physical port number (3 digits).

In the RC8000 Host Interface two identifications of access-addresses are used. Host Interface Address (HIA) in the RC3502C (at the HPM service interface) and Network Interface Address (NIA) in the RC8000 (at the NPM service interface). Between an NIA and an SC port a one-to-one correspondence exists and an HIA is always equal to the corresponding SC port.

To summarize the following access-address identification and representation are used.

line	: 1,2,3,4	(2 bit)
stream	: 0,1,2 ......N	(8 bit)
TS port	: 0,1,2 32767	(15 bit)
SC port	: 1,2,3 32767	(15 bit)
physical port	: <id> xxx	(4 - 10 alfa numeric char.)
HIA	: 0,1,2 ... M	(16 bit)
NIA	: 0,1,2 ... L	(16 bit)

2.5.2 Network User Identification

In CENTERNET the network users connect to the network at the Session Control Service Interface, and at this interface a network user is uniquely identified by a Network User Identification (NUI). This identification consists of two parts, a Network Unit Identification (NUID) and a Network Application Identification (NAID).

The Network Unit Identification and the TC address are identical and as a one-to-one correspondence exists between a TC address and a DTE address the NUID consists of 1 to 14 digits. The DTE address consists of up to 14 digits according to CCITT Recommendation X.121, (ref. (121)).

In the actual implementation a DTE address consists of 11 digits.

At the SC Service Interface, absolute as well as symbolic addressing of Network Users are permitted. A symbolic address consists of 1 to 10 alfanumeric characters, including two special symbols ('_',' '). The absolute address equals the SC access-address, SC port. I.e. 1 to 5 digits.

Several symbolic addresses may be used as synonym for a Network Unit while on the other hand a symbolic NIAD may cover several application entities.

For example:

NEU
NEUCC } = NEUCC = TC address q

TSO = { application A
application N
application Q }

At the moment the session is established, the SC port number unambiguously identify the Network User at that Session Control. This is true because only one Network User is "connected" to an SC Port and because the session only exists as long as the Network User is "connected".

Based on the abovementioned rules the formal definition of the Network User Identification becomes:

<network user id> ::= <network unit id> , <network application id>

<network unit id> ::= <nu-address> | . <name>
<network application> ::= <na-address> | . <name>

<nu-address> ::= $\{\langle\text{digit}\rangle\}_{1}^{14}$

<na-address> ::= $\{\langle\text{digit}\rangle\}_{1}^{5}$

<name> ::= $\{\langle\text{character}\rangle\}_{1}^{10}$

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

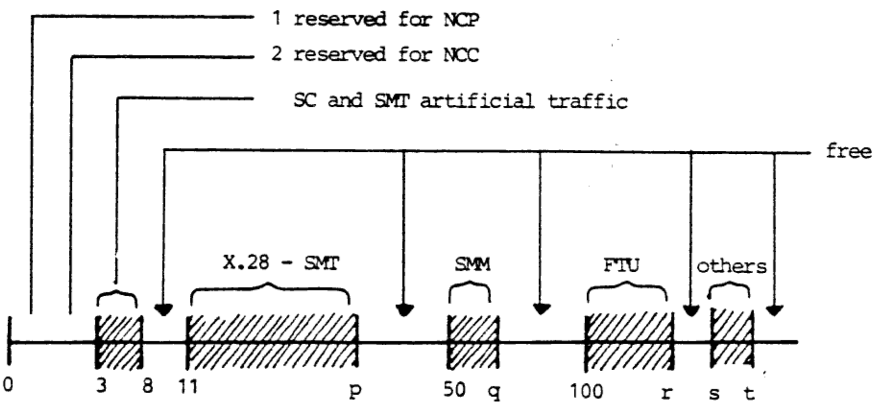
<character> ::= "A"-"Å" | "a"-"å" | " " | "-" | "0"-"9"

Notes:
<na-address> is limited to 1 ≤ <na-address> ≤ 32767

<name> may be more than 10 characters but only the first 10 is significant.

The abosolute address space for network application id is divided into intervals and each interval is assigned to a group of related network users or a host interface, e.g. all terminals connected to the X.28-SMT module belongs to one interval.

Figure 2.12 illustrates the division into intervals.



Artificial traffic from 3 to 8
X.28-SMT from 11 to (p-1)
RC8000 SMM from 50 to (q-1)
RC8000 FTU from 100 to (r-1)
others from s to (t-1).

Figure 2.12 Example of interval outline in TC.

The outline of the intervals is specified in one configuration file defining all application in CENTERNET.

2.6 Basic Terminology

As indiated in figure 2.2 several protocols are involved in the data transfer process. To achieve a consistent terminology this section contains a description of the structure of elements in each protocol and the data paths used to transport these elements.

In Appendix B the different terms used in the CENTERNET System Specifications are depicked.

Figure 2.13 shows the packing of protocol elements from SC to HDLC.

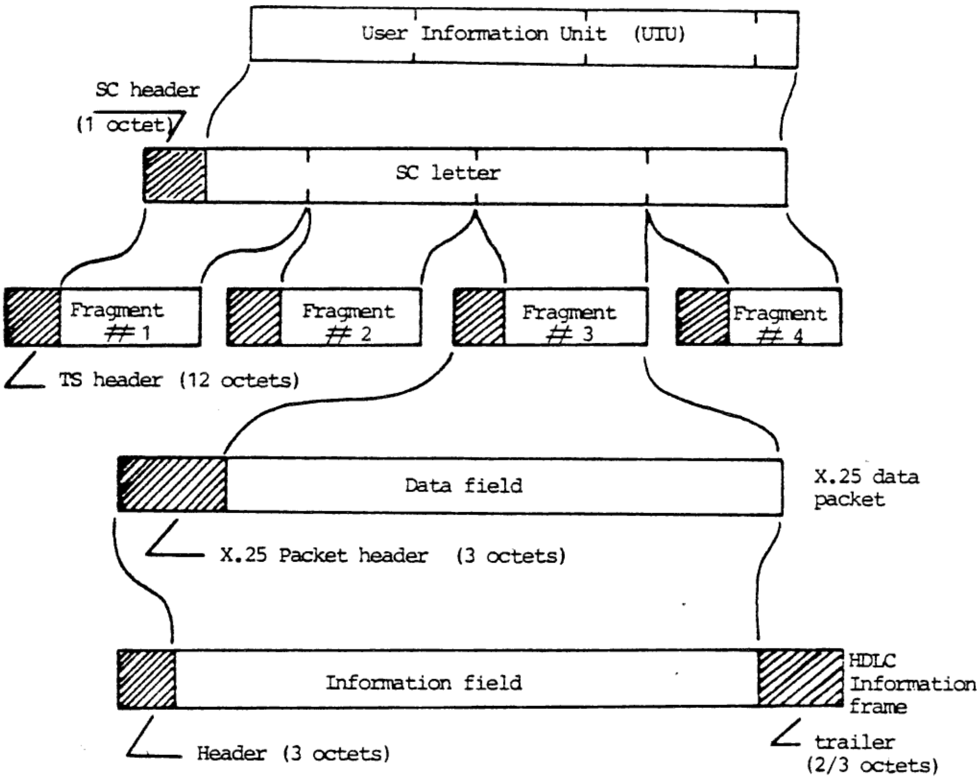


Figure 2.13 General Structure of protocol elements in the basic network protocols.

A SC-user delivers a LETTER/LETTERGRAM (both called User Information Unit) to the Session Control. The SC attach a SC header to the LETTER and is transperant to LETTERGRAMS. The TS divides these units into FRAGMENT's and adds a header, TS-HEADER. These two units constitute the Information Unit exchanged between two TS's (either on a LIAISON or as a LETTERGRAM), and as so forms the DATA-FIELD of an X.25-DATA PACKET. The X.25-DATA-PACKET is the Information Unit exchanged between the X.25-DTE and the X.25-DCE on an X.25-LOGICAL-CHANNEL. The DATA-FIELD is the unit exchanged between two X.25-DTE's on an X.25-VIRTUAL-CALL. The X.25-LOGICAL-CHANNEL is the access to the DATA NETWORK where as the X.25-VIRTUAL-CALL is the data path to another X.25-DTE.

The X.25-DATA-PACKET equals the INFORMATION-FIELD of the LINK-FRAME (X.25 LAP B frame). This is the unit exchanged on the X.25-LAP-B-LINK (or the actual physical circuit).

Figure 2.14 gives an overview of the general structure of some application protocols' information unit. For a detailed description please refer to the individual protocol specification. The data path between two applications is named a SESSION.

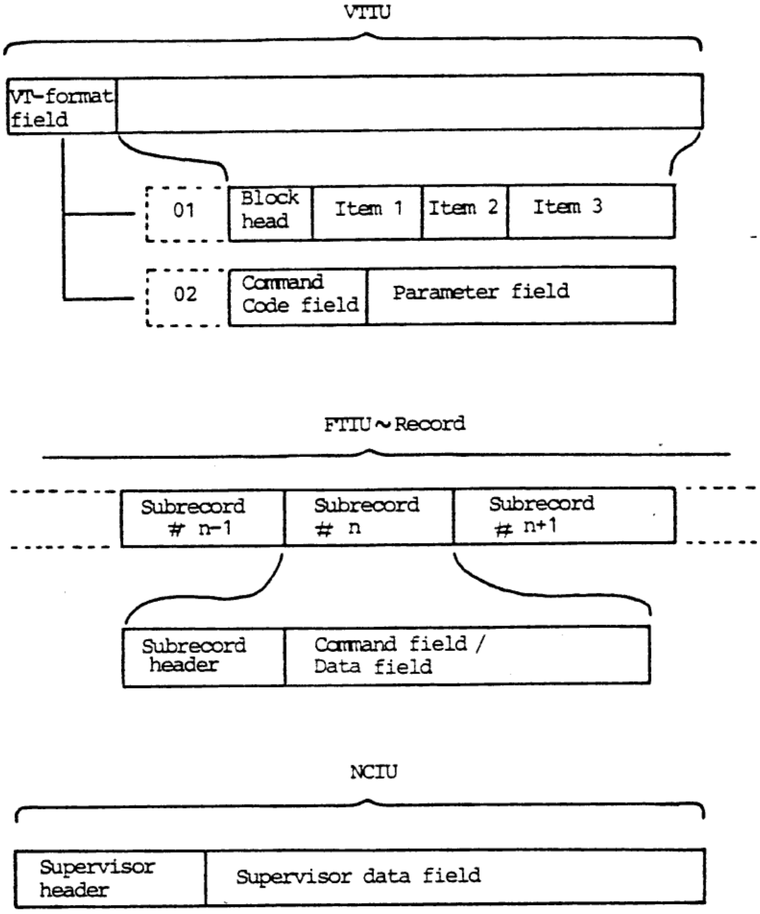


Figure 2.14 General structure of the elements of VT, FT and NC applications protocols.

This page is intentionally left blank.

3. CENTERNET RELATED TO THE ISO MODEL

CENTERNET is constructed as an open network. The logical structure of the network is based upon a layered structure as indicated in figure 3.1. This layered structure is in accordance with the ISO Reference Model of Open Systems Interconnection as presented in ISO/DIS7498. April, 1982 (ref. (122)).

This model defines for each layer the framework for the layer including the set of functions, the layer is responsible for performing.

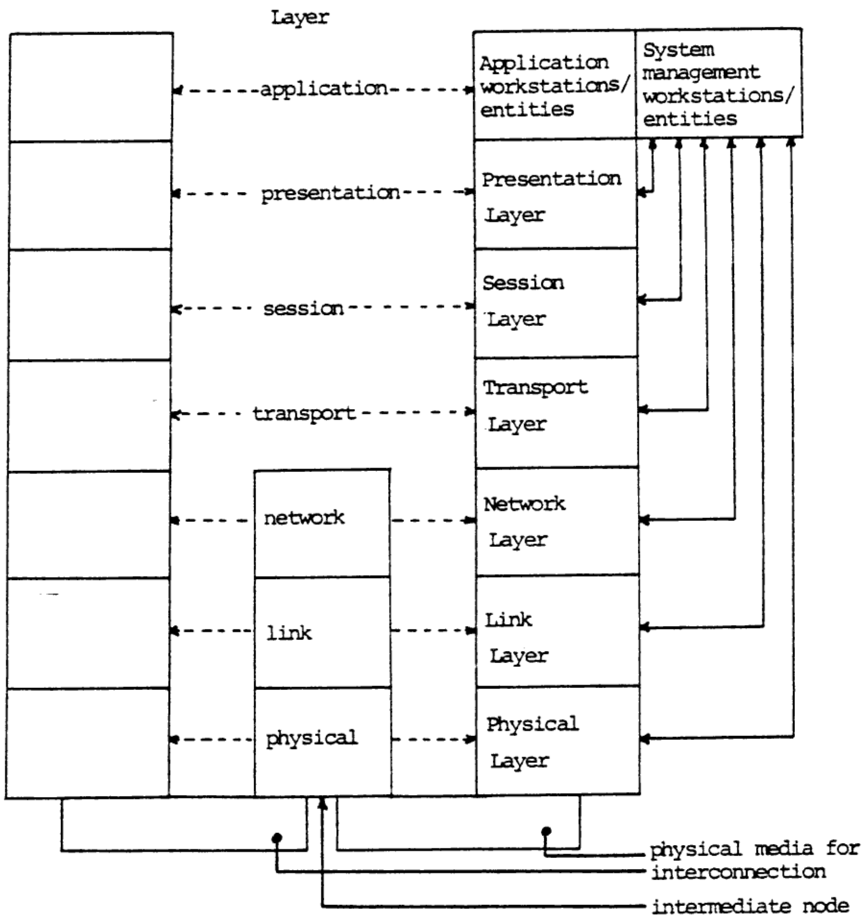


Figure 3.1 The Logical architecture of the CENTERNET network viewed as a relay system.

In the CENTERNET network, the Application Layer consists of the application/system processes in the host computers and different terminal concentrators. Information processing and management is characteristic for the layer.

The purpose of the Presentation Layer in CENTERNET is to provide the set of services which may be selected by the Application Layer to enable it to interpret the meaning of the data exchanged. The layer provides the window/glasses through which arbitrary applications view all other applications that may be interconnected through the Open System CENTERNET. The Presentation Layer serves the Application Layer directly, thus no multiplexing is provided.

The separation into the Application Layer and Presentation Layer may vanish depending on the functions in question. E.g. the functions for file transfer includes also some information processing and a high degree of information management besides the purpose for pure presentation.

The protocols of the Presentation Layer are based on a small number of presentation-image-definitions. These functions cover areas as Virtual Terminals, a Virtual Filestore, Virtual RJE facilities, etc.

The CENTERNET system includes protocols for virtual terminals and file transfer.

The CENTERNET Virtual Terminal Protocol, ref. (2), defines the framework for virtual terminals. The TTY-compatible terminals are connected to CENTERNET utilizing a modified X.28 service, so the terminal handling module and the connection module constitute the Virtual Terminal throughout the whole network (please refer to figure 2.2 and 5.6.3).

A subset of "A Network Independent File Transfer Protocol" (INWG Protocol 86, 12th December 1977, ref. (124)) prepared by the High Level Protocol Group has been selected as the presentation-imagedefinition for the RC8000 disc-to-disc file transfer.

The subset is defined in ref. (4).

In addition the Presentation Layer may include service for RJE terminals/functions.

Thus the Presentation Layer allows for interactive- and batch traffic as well as file transfer.

The Presentation Layer is served by a Session Layer supporting the interactions between cooperating application-entities. Thus the Session Layer may bind two application-entities into a relationship. When established, a relationship may be terminated by an unbind operation requested by the application-entity. The service provided is called session-administration-service.

In CENTERNET, at present, the service provided by the Session Layer includes functions for management of network connections only. During the data transfer the Session Layer may be viewed as a transparent layer.

The Session Layer is connected at a Transport Layer providing a universal transport-service in association with the underlaying services provided by supporting layers. The protocol to be used is a modification of IFIP INWG 96.1, Proposal for an Internetwork End-to-End Transport Protocol, January 1978 (ISO/TC97/SC16 N 24 and SC6 N 1557), ref. (125). The protocol is described in ref. (1).

A session-entity is known to the Transport Layer as a transport-address. Transport-addresses are provided by the Transport Layer and can be used by session-entities to uniquely identify other session-entities through the transport-service, i.e. transport-addresses are the means by which session-entities can communicate using the transport-service.

Because the Session Layer is transparent during the data transfer the Session Layer and Transport Layer may be perceived, in CENTERNET, as one unit providing session management- and data transport services. As a consequence there is a strong binding between a transport-address and a session-address. They are always equal, so a network user interfaces the network at the Session Layer Service.

Because no multiplexing exists at the Presentation Layer, a one-to-one correspondance between the presentation-entities and the session/transport-addresses exists.

The Transport Layer so chosen provides for liaison as well as lettergram service. A lettergram is a self-contained unit not larger than the maximum size of a letter. The lettergram is sent without knowing if the receiver is ready. A delivery confirmation can be given on request to the sender after the lettergram has been taken out of the queue by the receiving end.

Contrary to the lettergram service a liaison service is based on the establishment of credit for transmission by the creation of a liaison through the entire network. One credit for transmission is given to the sender each time the receiver has indicated its intention to receive a new letter. The facility gives the transmitting end continuous knowledge of the receiving end's ability to receive. The sender can then make sure that letters sent, for which credits were granted, will be accepted by the receiver.

To summarize, the Transport Layer in CENTERNET offers two kinds of communication service, the liaison service based upon single endpoint-connections (non-shared-ports) and the lettergram service based upon multi-endpoints-connections ('shared ports').

The Network Layer provides functional and procedural means to exchange network-service-data-units between two transport entities over a network connection.

The network connection is X.25 logical channels and used for virtual calls, as indicated in figure 2.3.

In CENTERNET, the X.25 level 3 'elements of procedure' constitutes the Network Layer.

The Link Layer provides functional and procedural means to establish, maintain, and release, one or several data-links between the DTE and the DCE. A data-link is composed of an X.25 LAP B link, also known as an HDLC line (class BA, option 2,8).

In CENTERNET, the Link Layer utilizes the Single-Link-Procedures (SLP) as specified in X.25 level 2, ref. (120).

The Physical Layer provides mechanical, electrical, functional, and procedural characteristics to establish, maintain, and release data circuits between the DTE and the DCE. The datacircuits consists of V.24/V.35 connections, one for each X.25 LAP B link.

The infrastructure of the Data Network is not part of the CENTERNET network specification, and is described elsewhere. Please refer to section 2.1.

As described above, the ISO Reference Model (OSI) defines a framework for Open Systems, including the CENTERNET network. The model defines the functional and procedural characteristics of each layer. Communication between peer layers are not necessarly based upon one protocol as layer 6/7 is an example of.

To summarize, the CENTERNET network can be viewed as an Open System according to OSI. Network users are connected to the network at the Session Layer utilizing either the session service or the lettergram service. The connection to the Data Network is utilizing an X.25 interface.

4. SYSTEM REALIZATION

The functions described in the previous chapters are realized by a number of hardware and software modules. The set-up and interconnection of these reflects to some extend the logical architecture of the network structure in accordance to the use of the OSI model as a framework for software and hardware architecture.

Generally speaking CENTERNET can be considered to consist of the following main modules:

- Network nodes (NN) with X.25 interfaces.

- Terminal Concentrators (TC), which connect to an NN with X.25 interface and which support the following external connections:

  1. Asynchronous Scroll Mode Terminals (SMT's).

  2. Host interface to RC8000 via a parallel channel adapter.

  Other types of terminals can be supported if appropriate terminal handlers (software modules) are added to the TC.

  The asynchronous ports may likewise be used to connect host computers with asynchronous terminal support, offering a cheap host interfacing to CENTERNET.

- RC8000 Host and Network Control (NC) computer. The RC8000 carries out the following tasks:

  1) Contains all software modules for the Network Control Center.

  2) Contains a File Transfer Utility module used to exchange files between network locations (e.g. in the starting phase between RC8000 host computers).

  3) Functions as a normal host computer in CENTERNET with support of SMT terminals.

Fig 4.1 illustrates the physical layout of the basic elements in CENTERNET.

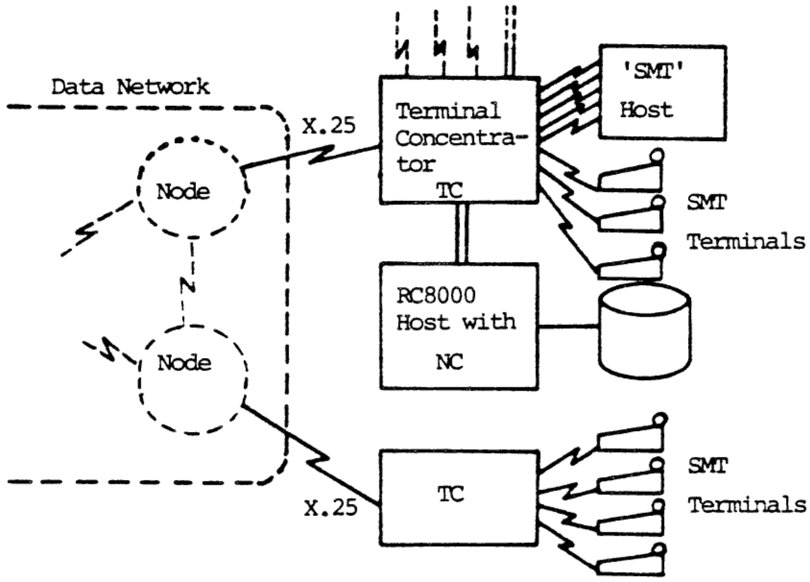


Figure 4.1 Physical Layout of CENTERNET.

The stipulated part of fig. 4.1 shows elements of CENTERNET which are not presently covered by this description.

4.1 X.25 Data Network

(To appear).

4.2 Terminal Concentrator

The physical and logical set-up of a TC is realized through the use of the RC3502C minicomputer system and the protocol and control system as illustrated in chapter 2 and 3. The following describes the hardware structure and basic software structure.

4.2.1 Hardware Structure, TC

The basic unit consists of an RC3502C CPU with the following elements (fig. 4.2):

- 16 bit bitslice CPU including a preprocessor (4 PC boards)
- 64/256 Kbytes RAM memory module (each 1 PC board) with 4-32 K bytes PROM
- Crate and power supply

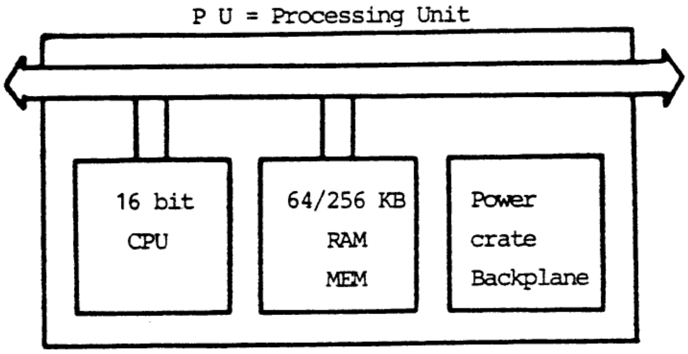


Figure 4.2 Processing Unit of a TC.

The term processing unit (PU) is used for the abovementioned physical unit.

A PU can be equipped with PC boards of the following types:

1) MEM - 64/256 Kbytes RAM memory module
2) IMS - 8 lines asynchronous multiplexor (1 board, max. 9600 bps)
3) COM - 4 lines synchronous HDLC multiplexor (2 boards, max. 64 Kbps)
4) IOM, I/O board for 8 high-speed channels (1 board)
5) MBA, Adaptor board for the INTEL MULTIBUS
6) TES, 64K bytes EPROM memory for image load

The IOM board is used for connection of external equipment including the parallel channel (FPA) to the RC8000 Host Computer.

Fig. 4.3 illustrates an example of a single-PU TC with an RC8000 channel attachment.

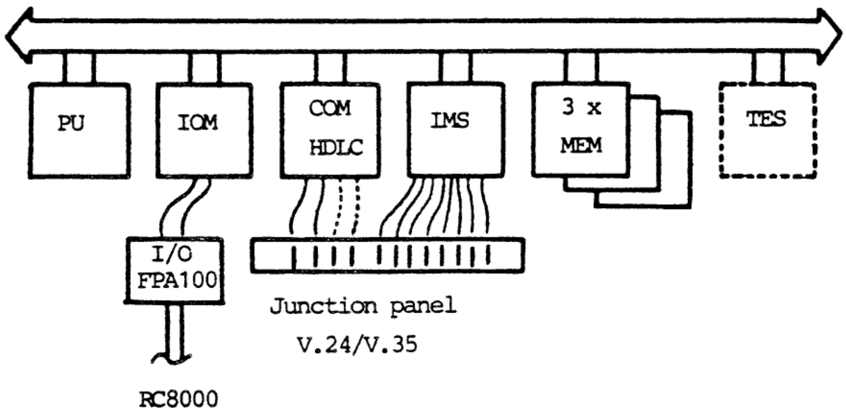


Figure 4.3 Single-PU configuration of a TC.

The following gives a brief description of the major specifications of the RC3502C minicomputer system:

Processing Unit Architecture

- 16 bit arithmetic unit built around 4 AM2901A bitslice chips

- Prefetch hardware

- Dynamic MOS memory with 480ns cycletime

- Address space : 2 Mbytes RAM and 2 Mbytes PROM/EPROM

- 128 interrupt levels

- Appromixmately 120 register sets each with 8 registers and a hardware stack of 8 x 16 bits

- The 120 register sets are dynamically distributed among interupt levels and active processes

- Scheduling is performed in 3 priority classes:

  Class I : Hardware controlled priority (high)

  Class II : Round robin on each priority level (medium)

  Class II : Time-sliced round robin (low)

- Nominal I/O capacity: a) programmed: 2 Mbps
  b) DMA : 15 Mbps

Processing Unit Instruction Set

- Stack-oriented P-code processor
- Format: 1 byte instruction code followed by a number of instruction parameters
- Push-and-pop operations
- Arithmetic operations in twos complement
- Jump, call, return, case jump
- I/O instructions for status handling and for read/write of byte/word or block of bytes/words
- Signal/Wait operations (chained semaphores)
- Addressing modes: absolute, indirect, relative (to address on top of stack, local frame, global frame, and intermediate frame)

For further information please refer to the appropriate reference manuals listed in appendix A.2.

4.2.2 Software Structure, TC

The following gives the basic ideas of the software system in an RC3502C minicomputer. The CENTERNET application modules are described elsewhere.

The basic concept laid down in the software system is very closely related to the process concept which is described shortly in the following:

A processor is a device, which executes instructions. A processor may be implemented in hardware (CPU, I/O controller), software (interpreter, ...), or in a combination of both hardware and software.

A processor is characterized by:

- an instruction set
- an address space, which can be shared with other processors

A program is a sequence of instructions which can be executed by a processor.

An incarnation of a process is a program and a data structure which controls the execution of the program. The data structure contains local variables for the program (a stack) and variables specific for the incarnation (the incarnation description).

One program may be shared by several incarnations.

Incarnations communicate by messages. Incarnations exchange right of access to messages by means of shared variables (semaphores).

The standard procedures Signal and Wait are used to exchange these references to messages.

The RC3502C system is provided with a high-level programming lanquage Real-Time Pascal, which is a real-time version (superset) of standard Pascal. The nucleus of the RC3502C software system consists of a number of modules (all written in Real-Time Pascal):

- MONITOR
- ALLOCATOR
- TIMER
- LINKER
- ADAM
- OPSYS
- OPERATOR, CONSOLE
- LOADER
- PRINTEXCEPT
- S
- Application processes (inclusive drivers).

Fig 4.4 illustrates an example of the process structure in a single-PU configuration.

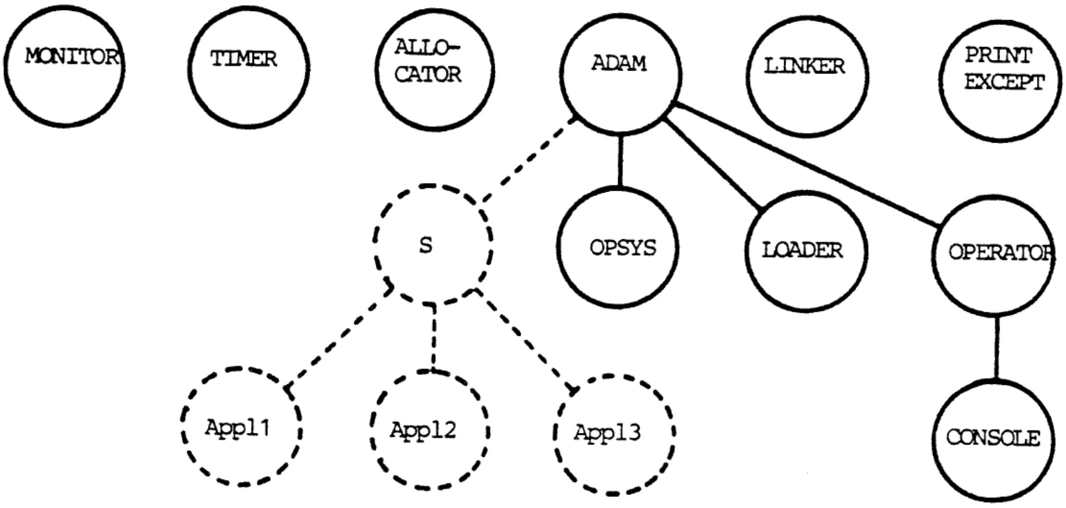


Figure 4.4 Example of process structure in an RC3502C single-PU system.

The following incarnations exist:

MONITOR — schedules the incarnations created in the PU. The MONITOR performs both short term scheduling (round robin, time slicing) and medium term scheduling (START, STOP).

TIMER — performs delay timing and timeout of drivers.

ALLOCATOR — administers allocation and deallocation of RAM memory and I/O channels.

LINKER — performs internal linking (static relocation) and external linking (routine calls) of object programs loaded from an external device. Besides the LINKER performs dynamic linking of processes on request from running incarnations.

ADAM — is the root of the dynamic tree of incarnations. ADAM creates automatically three incarnations:

CONSOLE

OPERATOR

S

ADAM may remove any of the incarnations on request.

LOADER is intended for dynamic load of object programs in a running system.

OPSYS is a command interpreter functioning as an interface between a human operator and ADAM.

OPERATOR, CONSOLE is the interface between a human operator and the running incarnations. CONSOLE performs I/O to the debug console.

OPERATOR processes messages signalled to the operator semaphore.

PRINTEXCEPT prints a list of the dynamic chain of routines calls, when a process incarnation goes into a runtime error (exception).

S is the root of all application incarnations. If a process S exists in the LINKER catalog, an incarnation of S will be created and started. This will be the case when a process S is blasted in PROM or autoloaded.

S may replace OPERATOR with its own NEW_OPERATOR, and OPSYS by its own NEW_OPSYS.

4.3 RC8000 Host Computer

The following gives a very short introduction to the basic architecture of the RC8000 Host Computer. Further information regarding hardware/software structure, I/O facilities and capacities can be obtained from the references given in appendix A.3.

4.3.1 RC8000 Hardware

The RC8000 is built up around a fast, so-called "unified", or common, bus to which the CPU as well as all storage devices, discs, and peripheral processors are coupled. The structure of this bus allows primary storage to be built up of modules with different speeds, allows all units connected to be directly addressable, and facilitates the interconnection of more than one central processing unit, so that true multiprocessing is possible. All slow- and medium-speed peripheral units, including data communication equipment are connected to the central unit via peripheral processors (device controllers). These are true processors with their own central units and memory, and they are able to take over a large part of the timeconsuming work of the central processor, for example, data testing, code conversion, communications supervision, the supervision of the peripherals, etc.

Fig. 4.5 illustrates the basic structure.

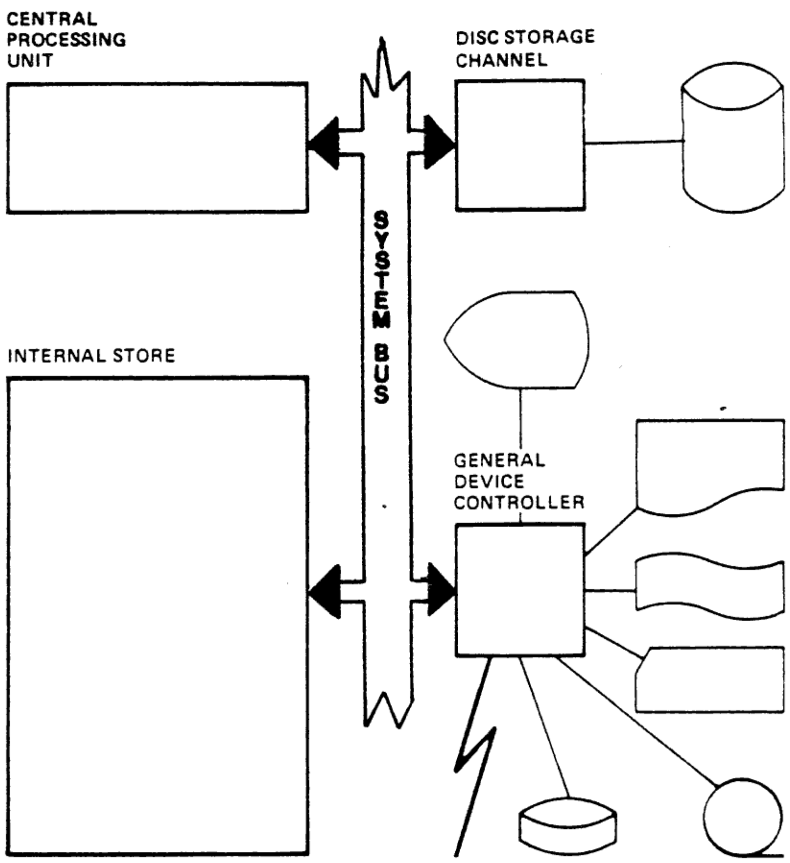


Figure 4.5 Basic Structure of an RC8000 computer.

Presently the RC8000 series of computers comprise the following basic models, listed in order of capacity and giving the average instruction times in microseconds, the maximum memory capacity (24 bit words), and the maximum number of front-ends which can be connected:

RC 8000/15	:	6.2 microsec.,	128 KW,	1 F.E.
RC 8000/35S	:	6.2 -	512 - ,	2 - -
RC 8000/45S	:	2.3 -	4 MW,	8 - -
RC 8000/55S	:	1.2 -	4 - ,	8 - -

All of the above listed models are equipped with semiconductor memory with a word lenght of 24 data bits plus 3 parity bits and with a cycle time of 700 nanoseconds. The data bus throughput is 3.3 million words per second, and the transfer rate via the directly connected disc channel is 1.2 Mbytes per second.

Central Processing Unit

The CPU executes program instructions which are fetched from the internal store at the time of processing. The unit contains registers and circuits for arithmetic logic, general control and interrupt control. The CPU is build around AM2901A bit slices and an optimum of flexibility and speed is achieved by combining microprogramming capabilities with special hardware features (instruction prefetch, cache memory (optional), floating point units).

The units have 24 bit per word single address instruction format with 64 basic instructions. Each instruction has a 12 bit displacement and 16 address modifications including relative, indexed and indirect addressing modes. 12 bit halfwords are the smallest data units which can be addressed directly.

The RC8000 Central Processing Units uses 4 working registers, three of which also function as index registers. This means, that the full instruction set is available for immediate address modifications and the number of empty transfers of registers to the Internal Store is greatly reduced.

The data formats comprise 12 bit half-words and 24 bit fullwords for integer arithmetic. 48 bit double-words are used for floating point and extended range integer arithmetic.

The instruction set is very versatile and includes facilities for half-word operations and word comparison which aids data manipulation. Logical operations permits setting and testing of single bits. Also included is an escape facility, which can cause programmed action on preselected types of instructions. This is a valuable tool for program debugging, emulation of special instructions etc. The various addressing modes allow dynamic relocation of programs. This means, that programs can be executed in any part of the Internal Store and can thus be moved around in the Store when needed.

A program protection system combined with a real-time clock and a powerful interrupt system provides facilities for multiprogramming operation.

4.3.2 RC8000 Software

The RC8000 System Software consists of a multiprogramming MONITOR, Operating systems "s", SOS, and BOSS, utility system, File Processor, data base management system, assembler, editor, sort/merge programs, statistical and mathematical procedures, together with compilers for the high level languages ALGOL and FORTRAN. "s" is a relatively simple, basic operating system, SOS (Swapping On-line System) is especially optimized for interactive on-line terminal applications, and BOSS is an advanced on-line and batch operating system which includes facilities for service bureau administrative data processing (accounting system, file privacy security, etc.).

The following gives a short description of the basic software nucleus: Monitor, basic operating system (s) and I/O system.

4.3.2.1 The Monitor

The Monitor is the software element that implements the RC8000 multiprogramming system. The basic concept within the system is the "process". A process is generally defined as an area in the internal store in which all computational activities pertaining to a certain job are performed. In the multiprogramming system, the attention of the central processor is shared equally among all present processes and they are therefore termed "parallel processes". In this environement, the supervisor program, Monitor, controls the following functions:

- Distribution of computing time among parallel processes
- Initiation, execution and termination of processes
- Communication between processes
- Reservation and initiation of sequential input/output devices
- Backing storage catalog handling

The Monitor program is permanently located in the internal store. When it is activated, it can notbe interrupted by any other program. It can be regarded as an extension of the hardware facilities.

By means of the interval timer and the interrupt system the Monitor allocates 25.6 msec of computing time to the processor in turn. If a process is interrupted after say 10 msec in order to wait for a peripheral, the next process in the queue will be started.

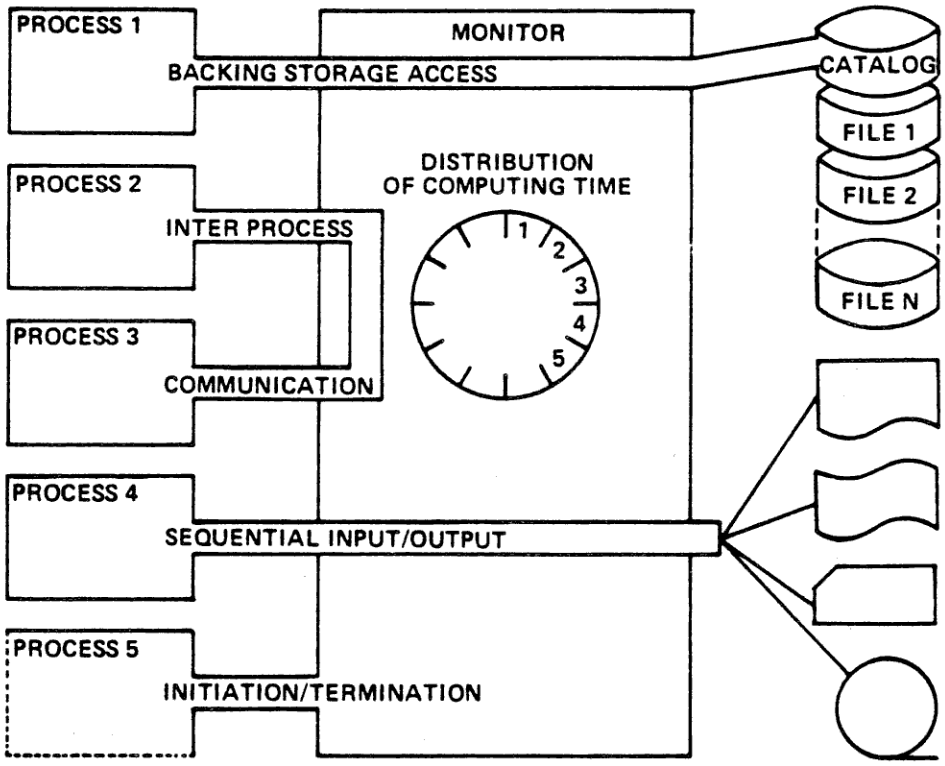


Figure 4.6 Monitor Functions.

Each process has a process description containing a symbolic name, relationship to other processes, limits of process area in the internal store, status and other information necessary for the administration of computer resources. A maximum of 21 processes can exist simultaneously and by means of a protection system it is guaranteed that no process inadvertently operates outside its boundaries.

If communication between parallel processes is wanted, the Monitor is able to mediate the contact by means of five procedures called: send message, wait answer, wait message, send answer and wait event. Each process has a queue within the Monitor in which it can receive messages from other processes. Using the communication procedures it is possible to transfer data from one process to another.

Peripherals are also regarded as a kind of process and they too are identified by a symbolic name. The communication procedures can then be used to initiate data transfer between processes and sequential input/output devices or establish a conversation with a terminal.

Users can retain programs and data permanently on the disc backing storage which is organized as a collection of named data areas. A fixed part of each store is reserved for a catalog describing the names and locations of the data areas. The catalog can be devided into an unlimited number of nested subcatalogs, each with a specific access restriction.

This hierachial structure combined with the program protection system secures privacy of files and at same time allows users to benefit from common program libraries etc. Data areas are allocated after a strategy that allows extensions and reductions to be performed when needed and makes reorganization superfluous.

4.3.2.2 Basic Operating System (s)

The System Process "s" is the key to the dynamic operating system concept of RC8000.

An operating system is a program that controls the execution of other programs, for instance, a batch processing system organizing a sequential execution of programs, a time sharing system for simultaneous programming from a number of terminals or a real-time system for updating a database. Usually an operating system is made for one, and only one, mode of operation.

In contrast to this, the Monitor of RC8000 has no built-in assumptions about program scheduling and resource allocation, it allows any program to initiate other programs in a hierarchial structure and to execute them according to any strategy desired. The functions of the Monitor described in the last subsection provide a general framework for different scheduling strategies.

After the initial system load the System Process "s" owns all computer resources. Users can then, from an arbitrary terminal, reserve a storage area and start a program. "s" will then immediately create the process description for the Monitor and the process is established.

Such a user process will then run in parallel with "s" which acts as a primitive operating system for all the parallel processes A, B and C it has started itself.

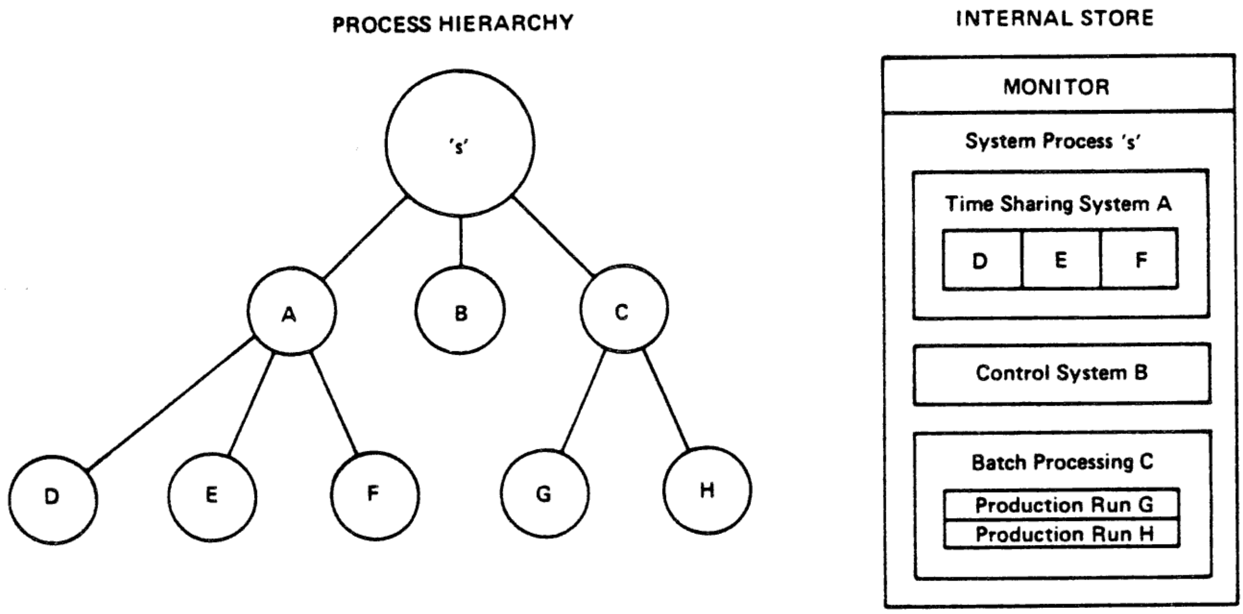


Figure 4.7 Process structure of RC8000.

The three processes A, B and C can be termed "child" processes of "s", they can now create their own child processes, D, E, F, G and H. The child processes can only be allocated a subset of the resources reserved by the parent. The parent acts as operating system for the children, it can start, modify, stop and remove its children according to any strategy desired.

This hierachy of processes can be expanded in both depth and width. In the resulting family tree each parent has complete jurisdiction of its children. In the RC8000, then, the operating system concept becomes varied and dynamic, Operating Systems can be written in a suitable high level language, such as ALGOL8, and implemented just like any other programs, they can be replaced dynamically enabling the system to switch between various modes of operation and several operating systems can be avtive simultaneously.

4.3.2.3 I/O System

The I/O system is one of the keys to the high performance of the RC8000 System. The intelligent peripheral controllers and the data bus structure is the basis for a standardized I/O programming system with an efficient allocation of peripheral resources.

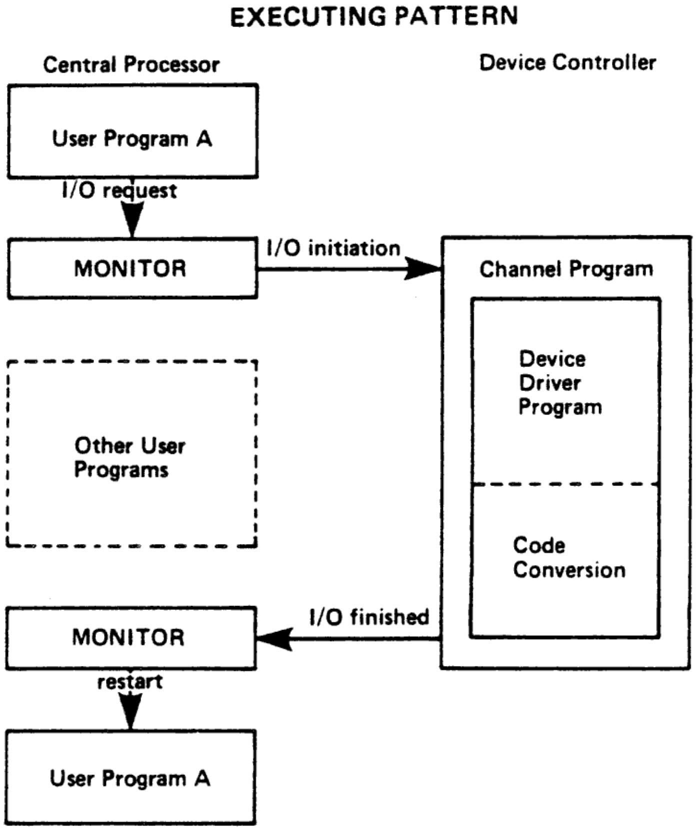


Figure 4.8 Basic I/O concept of RC8000.

To a job process an Input/Output operation is a matter of sending an I/O request to the Monitor stating the required peripheral device and the address at which data is to be read or written. The Monitor administrates an I/O request queue and when the required device is ready the Monitor initiates a data transfer on the data bus by activating a "channel" program. The channel programs reside permanently in the internal store and they are executed by the device controller that controls the requested peripheral devices. Once started, the device controller executes the channel program without further engagement from the central processor. The physical control of the peripherals and the necessary code conversions are performed autonomously by the controllers.

This structure with a main processor and separate peripheral processors can be regarded as a small network, and it is implemented as such. The General Device Controller software includes a Network Control Program, this means that an RC8000 System right from the start is prepared for connection to a data processing network in which the General Device Controller will act as a node. Terminal polling, peripheral operation, data transfer on the system and all the tasks of the

General Device Controller are performed by a software system quite similar to that of the total RC8000 system with its own Monitor and operating systems. But the General Device Controller is an integrated functional unit which cannot be accessed by users. To the users it is a black box which mediates the contact with the RC8000 system.

5. FUNCTIONAL DESCRIPTION - PROTOCOLS

This chapter describes the modules constituting the system outlined in chapter 2 and 3. Each section, except 5.1, is build up using the same mould

- the first subsection (5.-.1) describes the function of the module
- in the second subsection (5.-.2) the access (usage of the service interface) to the below laying module is explained
- in the third subsection (5.-.3) the service interface of the module is defined
- in the fourth subsection (5.-.4) the communication to the NCP is outlined
- the fifth subsection (5.-.5) gives an overview of the logical structure of the module.

References to the different protocol descriptions and detailed service descriptions may be in Appendix A.1.

5.1 THE TRANSPORT SUBNETWORK

This section will appear later.

5.2 CENTERNET X.25-DCE Service

This section will appear later.

5.3 CENTERNET X.25-DTE

The network subscriber is connected to a data network as a Packet Mode Data Terminal Equipment (DTE), the network connection being a Data Circuit - Terminating Equipment (DCE). The interface between the DTE and the DCE is defined by CCITT Recommendation X.25 (ref. (120)).

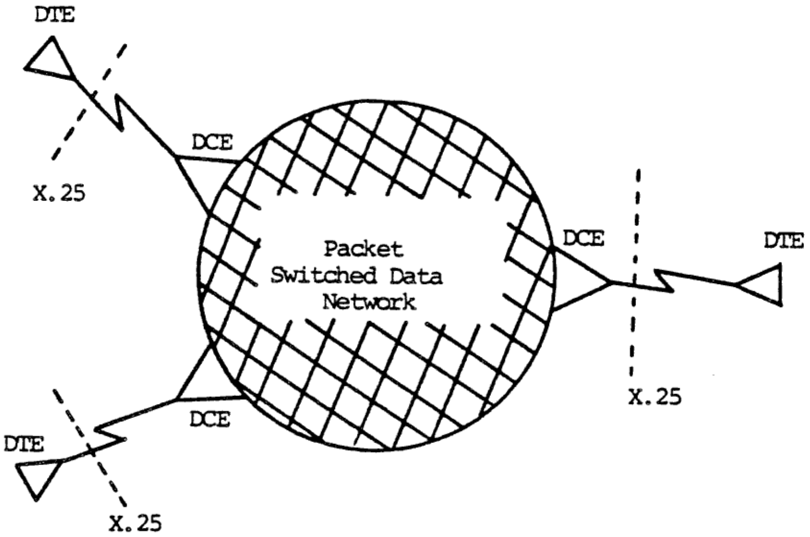


Figure 5.3.1. Hypothetical Reference Connection.

An overview of the X.25 level 3 (X.25-DTE) functions is given in section 5.3.1.2.

The X.25 level 2 is based upon Recommandation X.25 LAPB (ref. (120)) and the X.25 level 1 (PL) is based on V.24/V.35.

5.3.1 DTE Functions

The CENTERNET DTE consists of the layer 1, 2, and 3 as stated above.

The functions of layer 2 is described in section 5.3.1.1 and the X.25-DTE functions (layer 3) in section 5.3.1.2.

5.3.1.1 X.25 DTE Level 2 Functions

As mentioned above the X.25 level 2 is based upon Recommendation X.25 LAPB (ref. (120)). For a detail description of the functions and service interface of the HDLC module please refer to ref. (82), (83), and (80).

5.3.1.2 X.25 DTE Functions

As stated above the CENTERNET DTE Level 3 is based on the CCITT Recommandation X.25 level 3 (ref. (120)) utilizing no optional facilities, except a user defined priority. Some remarks on the X.25-DTE functions are given in this section.

The DTE only uses Virtual Call's (VC). A configuration file contains the default assigned logical channel numbers as a range. The lowest possible is LTC (=1) and the highest is HTC (= 4095). The actual assignment can be changed by the NCC restarting the DTE.

In order to minimize the risk of call collision, the LCN search algorithm will start with the highest numbered logical channel in the READY state and move downward, in order to find a LCN to be used in the Call Request packet.

In case the data network internal makes use of transfer with priority, a priority facility is defined and used in the CENTERNET X.25-DTE. The Facility Marker is used to support this non-X.25 facility as specified in the reference document.

Two priority levels exist, and the X.25-DTE user can at call establishment decide which level to use, as the priority is on a per call basis.

The X.25-DTE transfers to the network the priority facility indication in the facility field of the Call Request packet, and receives the indication in the same field of an Incoming Call packet. Furthermore the X.25-DTE utilizes the priority levels to speed up transmission to the network by indicating the priority level of each packet to the HDLC driver. The HDLC driver searches the output queue for packets of high priority and transfers these packets first, before starting the transmission of the low priority packets.

The maximum data field length is 256 octets in data packets.

If a DTE user wishes to indicate more data to follow, a mark called MORE DATA, M-bit, is used. Two categories of data-packets are defined:

1. a) packets not having the maximum data field length

   b) packets having the maximum data field length and no MORE DATA mark

2. c) packets having the maximum data field length and a MORE DATA mark.

The X.25-DTE module will check the use of the M-bit by comparing the actual data field length and the M-bit setting to see, if the data-packet belongs to any of the abovementioned categories. If not, the packet will be rejected.

If a DTE user wishes to transfer data on more than one level, it uses a DATA QUALIFIER indicator, Q-bit. When only one level is being transmitted, this is set to zero. If two levels are being transmitted, the DATA QUALIFIER in all packets of a complete packet sequence are set to the same value, either zero or one. The use of the Q-bit is on a per call basis. I.e. that an X.25-DTE user indicates, at call establishment, whether or not he will use two levels of data.

The X.25-DTE module will validate the usage of the Q-bit, and in case it is set without indication at call establishment the packet will be rejected.

As stated above the X.25-DTE module offers, in the service interface, the use of the M- and Q-bit, but the module checks the usage.

In order to maintain independency between the individual streams (refer to section 5.3.3) only the X.25-DTE module or the network can initiate a restart procedure. The X.25-DTE users will in case of a restart be informed (refer to section 5.3.3.3).

The X.25-DTE module does not utilized the D-bit in any packets.

As mentioned in the reference document up to the first four octets of the 'call user data field' is reserved for protocol identification. The user is responsible for setting the bits according to the rules, the X.25-DTE will not change nor check the contents of these fields.

Packets must be/are delivered from/to the HDLC module in the Send Sequence Number, P(S), sequence. Otherwise the X.25-DTE will consider the packet outside the window and reset the virtual call.

Furthermore the P(R) and P(S) numbers have only local significans so the interpretation of the P(R) value is only an updating of the lower window edge.

5.3.2 DTE Access

The service offered by the HDLC module is utilized in the following way:

5.3.2.1 Line Control

CONNECT LINE

Before the actual transfer the line must be connected physically and logically. A positive respond should indicate that the line has been brought into Asynchronous Balanced Mode. As line characteristies can differ, these are supplied when connecting is demanded.

Parameters: line-id, line set up timer, retry timer, retry counter, no. of outstanding frames.

DISCONNECT LINE

The line connection will be terminated when this primitive is issued.

Parameter: line-id

RESET LINE (RETURN ALL BUFFERS)

If errors occur, the DTE module must be able to purge the line using this primitive. All messages currently being executed or waiting for retransmission and all pending messages to the identified line must be returned without treatment. The above described actions will bring the line into a well-defined state, known by the DTE. It is therefore able to reestablish the datatransfer when this message has been returned.

Parameter: line-id

5.3.2.2 Data Transfer

The previous primitives will when returned to the DTE module enable this to initiate and terminate the actual datatransfer. Data that should be sent will be given to the HDLC module in a buffer, and when the DTE module is ready to receive data, it will indicate this by handing over an empty buffer to the HDLC module. After the empty buffer has been (partly) filled it is expected to be returned.

TRANSMIT MESSAGE

All X.25(3) commands are transmitted to the DCE using this primitive. As mentioned above a priority can be indicated to speed up data transfer.

Parameters: line-id, priority.

RECEIVE MESSAGE

Controlling the flow is done by deliver one or more buffers to the HDLC module. The DTE module will, as mentioned earlier, when ready to receive a command (packet), hand over a buffer to the HDLC module.

Parameters: line-id, buffer, bufferlength.

5.3.2.3 Line Monitoring

EVENT MESSAGE

The DTE must at given events be able to obtain knowledge of the performance on the line. The returned message contains status information.

Parameters: line-id.

5.3.3 X.25-DTE Service

To support the functions described in section 5.3.1 this module implements X.25 level 3, multiplexing a number of logical channels on one HDLC line.

At the the DTE interface the logical channels will, when brought into the state "data transfer" be referred to as streams. Individual streams are identified by a stream no allocated by the user.

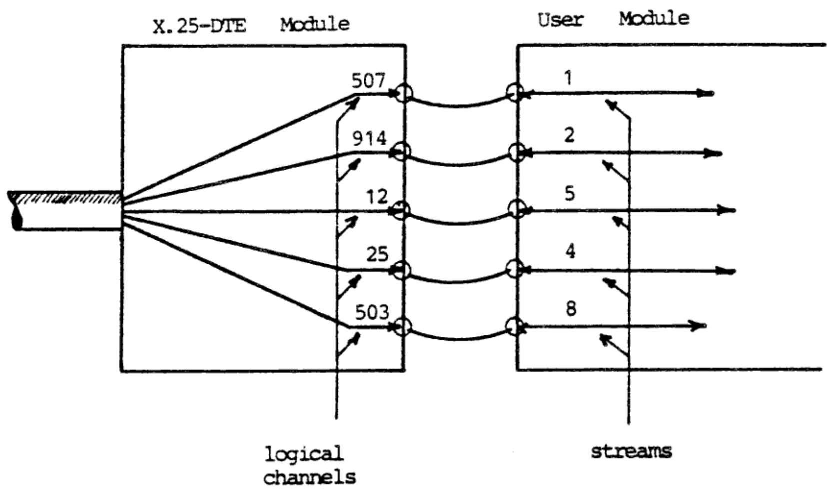


Figure 5.3.2 Mapping of logical channels/streams.

5.3.3.1 Stream Supervisor Primitives

For the control of X.25 call setup and clearing this group offers the subscriber the appropiate primitives.

<u>RECEIVE INCOMING CALL</u>

Enables the user to examine the wanted characteristics of a distant initiated call, before accepting it.

Parameters: control buffer, timer period for validity.

<u>ACCEPT INCOMING CALL</u>

If the user accepts an incoming call this primitive can be used to force the DTE module to establish the virtual circuit.

Parameters: Streamno, reference no. identifying the call. (Given as responseparameter to RECEIVE INCOMING CALL)

<u>REJECT INCOMING CALL</u>

If the user cannot accept the call, this primitive will cause a "clearing" of the not yet established virtual circuit.

Parameter: reference no. identifying the call to be rejected.

<u>CALL REQUEST</u>

Used to establish a call, to a distant DTE, if the call is accepted by this. The codings of the below given parameters concerning facilities and protocol id must follow Recommendation X.25 (ref. (120)).

Parameters: Streamno, called DTE addr., facilities, protocol id, call user data.

<u>RESET REQUEST</u>

If any discrepany is detected during the transfer, this primitive will enable the user to bring the virtual call into a welldefined state. All pending messages will be returned. The user is responsible for supplying the necessary number of input buffers to ensure that no data is lost.

Parameters: streamno, reset diagnostic code.

<u>SYNCHRONIZE STREAM</u>

This primitive is necessary to synchronize the user module and DTE module when errors are reported. It should be used by the former to indicate an acknowledgment of the disturbance.

Parameter: streamno.

CLEAR REQUEST

When a virtual call is to be removed the subscriber can issue this primitive.

Parameter: Streamno.

STREAM STATUS

The DTE module offers the possibility for the user to monitor the status of a given stream. The information returned includes a 16 bit status word and the last transmitted/-received X.25 causes and diagnostic codes.

5.3.3.2 Data Transfer Primitives

For the reception/transfer of subscriber information, a group of four primitives are described. A subdivision of each stream into four substreams designating input/output of the two information units, data packets & interrupt packets, accounts for the number of primitives.

SEND DATA

After a stream has been established, transfer of an X.25 data packet is accomplished by the use of this primitive. If the flow on the virtual call is temporarily stopped, caused by the reception of a RNR command, or disabled due to window closure, queing is done until flow control permits the transfer.

Parameters: Streamno, databuffer, bufferlength, M-bit indication, Q-bit indication.

RECEIVE DATA

Flow control on the input substream of datapackets is performed utilizing this primitive. Controlling the flow is made by the user handing over input buffers in a speed, indicating the users ability to receive data.

Parameters: Streamno, receivebuffer, bufferlength.

Utilizing the two interrupt substreams is done by the below described two primitives.

SEND INTERRUPT

Used to transfer one octet of subscriber information, held outside the flow control of data packets. An interrupt packet will be delivered at or before the point in the stream of data packets at which it was generated.

Parameters: Streamno, interrupt buffer

RECEIVE INTERRUPT

This primitive is used by the subscriber to indicate his readiness to receive an X.25 interrupt packet. If no receive buffer has been given to the level 3 module before an interrupt packet arrives, this will be discarded.

Parameters: Streamno, interrupt receive buffer

5.3.3.3 Status in Answers

The abovementioned primitives are implemented as messages from the userprocess. Every message will be returned with a status showing how it has been treated and the actual state of the stream. Possible indications are:

Channel has been cleared
Channel has been reset
Interrupt/incoming call/data packet lost
Illegal primitive
Timeout
Rejected
etc.

Further information will in connection with some status be available. As examples are:
Clearing/reset causes
Diagnostic codes
etc.

To retrieve this information the user must ask for a stream status.

5.3.4 Network Control Service

The service interface to the NCP offers two types of functions as specified in section 6.3:

control functions (CONTROL)
monitoring functions (EVENT, SENSE and GET STATISTIC).

For each type of function, one or more specific functions are defined. The next two subsections will give a general discription. The exact contents and formats are described in ref. (10) and ref. (11).

As mentioned above in section 5.3 the CENTERNET X.25-DTE descriptive covers level 2 and level 3, so the following two subsections will describe Network Control Service for both the HDLC module and the DTE module.

5.3.4.1 Control Functions

Below is listed the CONTROL functions supported by the HDLC module. For a detailed description please refer to ref. (10).

SET NC MASK
SET LINE TABLE
SET HDLC PARAM
CONNECT LINE
DISCONNECT LINE
SET MODEM SIGNAL.

The NC can open/close HDLC lines, utilizing the commands CONNECT LINE and DISCONNECT LINE. The HDLC module will try to connect the indicated line. If it can not be connected an event indicating this is returned to the NC.

The disconnection is performed at the moment it is requested regardless of actual traffic.

SET LINE TABLE is used to establish a correspondance between a physical line and an HDLC driver incarnation.

The DTE module supports the below listed CONTROL functions.

SET NC MASK
RESTART DTE

RESTART DTE may be used to change the assigned interval of the logical channel number. Furthermore some system parameters may be updated.

SET NC MASK is used (for both module) to activate/deactivate the monitoring functions. I.e, the NC can indicate to the module whether or not statistical information shall be gathered which events that shall be logged, and for the DTE module, whether tracing should be performed or not.

5.3.4.2 Monitoring

The X.25-DTE and HDLC modules support three different monitoring functions

- event (reports)
- sense (immediate state/status)
- statistic

REPORTS

EVENT, event-type, event-inf.

By return the fields contain the information

- event-type   identification of the causing event

- event-inf. further information concerning the event.

The following events will trigger the report function:

HDLC: (1) line connected
(2) line disconnected
(3) connection failed

X.25 DTE: (1) lack of internal resources
(2) virtual call established
(3) virtual call cleared
(4) unsuccessful VC request
(5) diagnostic received
(6) virtual call reset
(7) DTE restarted

SENSE

The SENSE operations are used to get either an immediate state or an immediate status for the whole module or for a data path through the module.

The following operations are supported:

HDLC: SENSE HDLC
SENSE LINE
SENSE LINE SPEED
GET LAST FRMR

X.25-DTE: SENSE DTE
SENSE CHANNEL

STATISTIC

The STATISTIC operations are used, as the name indicate, to retrieve statistical information for either the whole module or for a data path through the module.

The following operations are supported:

HDLC: GET HDLC STATISTICS

This operation returns statistical information concerning one HDLC line.

X.25-DTE: GET DTE STATISTICS
GET CHANNEL STATISTICS
GET BUF STATISTICS

The last operation returns statistical information concerning the internal buffer pools in the DTE module.

5.3.5 LOGICAL STRUCTURE of X.25-DTE

For a description of the logical structure of the HDLC-module please refer to ref. (80).

A number of individual blocks constitute the DTE module. Figure 5.3.3 illustrates the selected structure. In this section a short functional description of each block is given.

DTE INTERFACE:

Primitives destined to the DTE module must be validated before the demanded actions are performed. The administration of individual streams to the associated blocks are also implemented by this block.

MONITOR/SUPERVISOR:

The correlation between streams and logical channels are provided herein. When a stream, and thereby a virtual call, is requested the associated protocol block incarnation is created by the supervisor block.

The service offered to the NCP module, see sec. 5.3.4, is treated by this block. This means that the necessary statistical collection is performed, and state monitoring done when requested.

BUFFER ADMINISTRATOR:

Buffer management of the resources owned by the DTE module is performed by this block.

X.25 CHANNEL/LCNZERO

These blocks implement the DTE part of an X.25 interface.

LINE INTERFACE IN:

Messages from the HDLC module concerning input must pass this block. The demultiplexing of packets belonging to different logical channels and thereby treated by different protocol blocks, is therefore necessary.

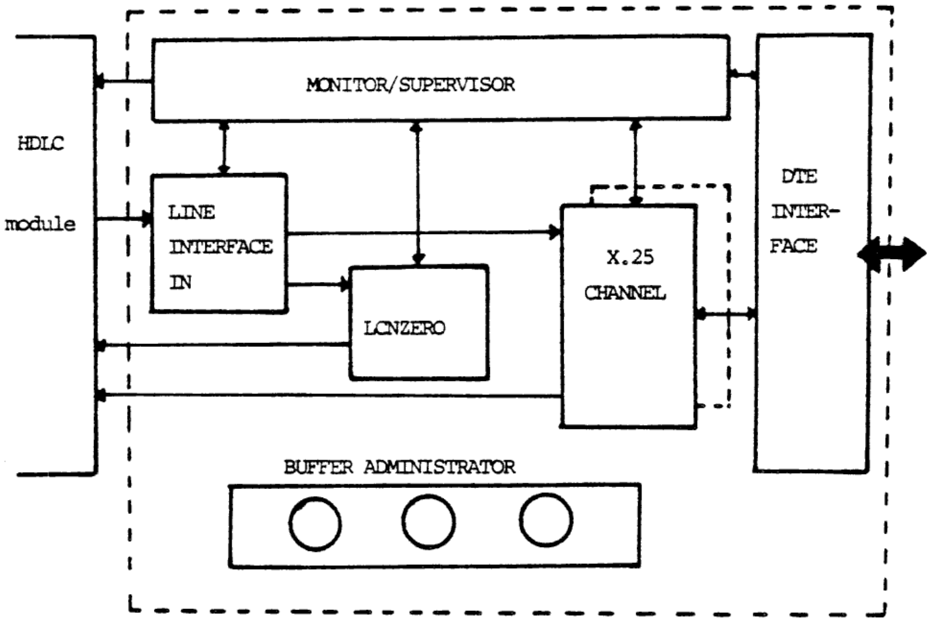


Figure 5.3.4 Logical Structure of X.25-DTE Module.

5.4. CENTERNET TRANSPORT STATION

The most general definition which can be made of the transport station (TS), is that it is an access-method which interposes logically between a network and application programs by providing them with distant communication service.

5.4.1 TS Functions

The basic service provided by the TS is copying a buffer at the senders end into a buffer at the receivers end. Sender and receiver can be attached to the same TS, for which reason traffic is looped inside the TS.

The TS defines a common namespace, ports (PT),for addresable enties, i.e. ressources, applications, etc., and means to establish and terminate full duplex associations, liaisons between pairs of ports.

Reliability of the transport service implies end-to-end errorcontrol.

Independence of associations requires individual flow control for each conversation.

There are two modes of operation for an association: liaison mode and lettergram mode. In lettergram mode, letters are sent independently of each other, with a predefined priority. The users are responsible for having their ports activated prior to exchanging letters.

In liaison mode, initialization commands are exchanged prior to transmitting any letters. This initialization is intended:

1. To make sure both ends of the liaison are active
2. To agree on the set of services to be put into operation.
3. To initialize parameters consistently.

In liaison mode transmission/reception of lettergrams is also possible.

The transport protocol, formally defined in ref. (1) provides for transfer of letters (LT) and telegrams (TG) from one part to another within the context of an association.

A letter is a variable-length unit of information with a maximum size. The idea is that almost any physical record can be placed in a letter, thus avoiding fragmentation of meaningful data above the transport level. The letter is given as a whole by the sending proces to its TS, and then delivered as a whole to the receiving process. Thus in the TS, buffer management is handled at the letter level. Since error control and flow control are tied to buffer management they will also be introduced at the letter level.

The LI-TG command is used to pass an "interrupt" signal from one transport user to another over a liaison. The exact interpretation of an "interrupt" signal will be dependent upon the receiving transport user.

The sending TS must transmit "interrupt" information even if the flow control option would prohibit transmission of letters. Similarly, the receiver must accept (and possibly acknowledge) an LI-TG immediately,even if there are previous unreceived letters.

Telegrams differ from lettergrams in that their size is limited to two octets of data which allow them be transmitted and received more easily (e.g. minimal resources required).

If acknowledgement of a telegram is required (R-bit set), the sending TS is responsible for giving to it a MY-REF which distinguishes it from all other telegrams within the association. The MY-REF name space for telegrams on an association is independent of the MY-REF name spaces for letters on the same association. In order to simplify handling of telegrams requesting acknowledgement, and since telegrams are not intended for high bandwith, the number of unacknowledged telegrams requesting acknowledgment and outstanding at any one time, is limited to one.

If acknowledgment of a telegram is requested (R-bit set) it must be acknowledged i.e. an LI-TAK must be sent back indicating in the YR-REF field the reference of the telegram received.

The acknowledgment means that the telegram has been made available to the receiving process. It does not mean that the process has read it or that the process agrees with its contents. It just means that the process agreed to receive a telegram on that liaison, and that the LI-TG was correctly received by the TS and made available to the receiving process.

The sender of the telegram expects to receive the acknowledgement within a maximum delay after the telegram has been sent. If the acknowledgment is not received, the unacknowledged telegram will be assumed lost and will be sent again. Acknowledgement will again be expected. If acknowledgment is not received, then this process will be repeated. If a telegram has been sent "N" times without success, the sending TS will report this condition to the transport user, and terminate the liaison.

When using X.25 to access networks,the size of transport commands is limited by the size of datapackets crossing the X.25 interface. Therefore, the TS may need to break letters into pieces that will fit into an X.25 data packet.

For this purpose the mechanism fragmentation, for breaking letters into fragments (FR) is the responsability of the TS.

All fragments belonging to the same letter, except the last one which is marked by an End of Letter flag, have the same length. Fragments within a letter are numbered to allow proper reordering, reassembly, at the destination TS. Since each letter has a unique sequence number, there can be no confusion between fragments of different letters.

The abovementioned acknowledge mechanism for telegrams also applies for letters.

5.4.2 TS Access

The information exchange between two associated transport stations is based on X.25 access to a packet switched network. The service offered by the Data Terminal Equipment (DTE) is not part of recommendation X.25. For this reason a variety of services exist, depending on the interconnecting network, e.g. traffic with priority.

Taking the needs of CENTERNET users into account, the service described below should fullfill all requirements.

When two transport stations wish to communicate, a connection to the data network must be established. As described in section 5.3.3 the individual flows through the level 3 module are denoted streams. A stream to the data network is achieved when the TS issues a

CALL REQUEST

parameters: stream no, called DTE addr,
priority, call user data
Q-bit indication

The selection of stream numbers is done by the TS to distinguish between different flows. Called DTE address is computed from the receiving TS address, while the priority is a function of the throughput-class indicated in the OPEN-LI primitive to the calling TS. User data is used when the calling TS wants additional information transferred during the establishment phase.

For the called TS to accept the call and administer the allocation of stream numbers, it will have indicated its willingness to communicate on more streams by the use of

RECEIVE INCOMING CALL

parameters: buffer, timer.

The second parameter will, if present, indicate a period in which a distant call will be taken under consideration. When this occurs the TS is able to inspect e.g. the identity of the calling TS and characteristics of the stream, as these are returned in the buffer.

If the call is acceptable the

ACCEPT INCOMING CALL

parameters: streamno, reference

will prove the DTE to end the call set-up phase by accepting the call while

REJECT INCOMING CALL

parameters: reference

will cause a clearing of the call.

For both primitives the parameter reference is an internal response value, returned with the answer to RECEIVE INCOMING CALL, identifying one particular, of maybe many, pending incoming calls.

When the stream has been established the transfer of TS commands can commence. Fragments of letters and lettergrams are transmitted using.

SEND DATA

parameters:

stream no	:	the actual stream to which the fragment belongs
data	:	address indicating start of TS command
data length	:	length (in bytes) of the command.
M-bit	:	set to one in all fragments, except the last one, together constituting the letter/lettergram.
Q-bit	:	set according to use

If the receiving TS is able to receive a command it must have given an input buffer to the DTE module using the primitive

RECEIVE DATA

parameters: streamno, buffer, buffer length

Control commands, such as LI-ACK, LI-NACK etc. are sent and received using the abovementioned primitives following the rules laid out in ref. (1).

During the transfer it is vital for the TS to monitor the actual state and status of the stream. This is done through the primitive

IMMEDIATE STREAM STATUS

parameters: streamno, buffer, buffer length

The retrieved information contained an 16 bit status word, the last received clear cause, clear diagnostic, reset cause, reset diagnostic, restart cause, and restart diagnostic in case any of these events have occured since last read.

If certain events occur the TS must be able to bring the stream to a welldefined state. This is done by using

RESET REQUEST

parameters: streamno, diagnostic

When the transfer has been completed the TS will clear the stream to the data network by the use of

CLEAR REQUEST

parameters: streamno, diagnostic

and the stream is considered removed when a possitive reply returns.

5.4.3 TS Service

In this section the available access primitives accepted is described. A short functional explanation follows the mnemonic and the necessary parameters conclude each primitive.

OPEN-PT: Activates a TS address (PT) for liaison establishment. If followed by the CNTR-LI primitive a liaison initialisation is possible on this port.

parameters: local portnumber

CLSE-PT : Deactivates a TS address

parameters: local portnumber

OPEN-LI : Initialises a liaison to a distant TS

parameters: local portnumber, remote portnumber, remote TS address, throughput class.

CNTR-LI : Allows the TS to report to the user when a liaison has been initiated/terminated.

parameters: local portnumber, controlbuffer.

RECV-LG : Allows the TS to receive information from a distant partner carried in a lettergram.

parameters: local portnumber, receive buffer, buffer length

SEND-LG : sends a letter in lettergram mode

parameters: local portnumber, remote portnumber, remote TS address, acknowledgement request indication, data buffer, data length

RECV-LI : Allows the TS to receive a letter on an established liaison.

parameters: local portnumber, receive buffer

SEND-LI : Sends a letter on an established liaison

parameters: local portnumber, data buffer, data length

SEND-TG : Sends a telegram (16 bits) on an established liaison

parameters: local portnumber, acknowledgement request indication, data buffer

RECV-TG : Allows the TS to receive a telegram on an established liaison.

parameters: local portnumber, receive buffer.

CLSE-LI : Terminates a liaison

parameters: local portnumber, termination cause.

5.4.4 Network Control Service

The TS module supports two types of NC functions as specified in 6.3:

control functions (CONTROL)
monitoring functions (EVENTS, SENSE and GET STATISTIC)

For each type of function, one or more specific functions are defined. The next two sections will give a general description, while the exact contents and formats are described in ref. (12).

5.4.4.1 Control Functions

When the NC wants to control the TS module, if may utilize one of the below listed operations

SET NC MASK
MAX NO PT
SET STREAM RANGE

The last two is used to set the maximum number of ports/-streams, respectively.

SET NC MASK is used to activate/deactivate the monitoring functions. I.e, the NC can indicate to the module whether or not statistical information shall be gathered, and which events that shall be logged.

5.4.4.2 Monitoring Functions

The TS module supports three different monitoring functions.

- event (reports)
- sense (immediate state/status)
- statistic

REPORTS

EVENT, event-type, event-inf.

By return the fields contain the information

- event-type identification of the causing event.
- event-inf. further information concerning the event.

The following events will trigger the report function:

(1) lack of internal resources
(2) liaison init
(3) liaison term

(4) port opened
(5) port closed
(6) unsuccessful liaison init
(7) unsuccessful lettergram deliver.

SENSE

The SENSE-operations are used to get either an immediate state or an immediate status for the whole module or for a data path through the module.

The following operation is supported:

PORT STATE

STATISTIC

The STATISTIC-operations are used to retrieve statistical information for either the whole module or for a data path through the module.

The following operations are supported:

GET TS STATISTIC
LIAISON STATISTICS
PORT STATISTICS

5.4.5 Logical structure of TS

The implementation of the TS module is utilizing the facilities offered by the Real-Time PASCAL language concerning process communication. The module consists of a number of blocks as outlined in figure 5.4.1.

The functional description of each block (group of blocks) are given.

User Interface-process:

(processname ts_ihp)

This process checks that all user messages are valid and legal. The process forwards the message to the relevant protocol process. Except from this is open port (open_pt) and close port (clse_pt), these messages are forwarded to the supervisor process.

Supervisor-process:

(processname ts_sup)

This process is father to all other process in the TS module. The process creates and removes protocol processes when it receives respectively open port (open_pt) and close port (clse_pt).

The process answers or forwards mmessages from the NCP to the relevant processes.

From the number of liaisons the process delivers or draw back buffers to and from the pool handler.

Protocol-process:

(processname ts_pp_versyx, y is a letter and x is numeric). This process is handling all the transport protocol issues. The transport protocol is described in ref. (1).

DTE Interface-process:

(processname ts_sip)

This process is handling all the communication to the DTE module.

Pool Handler-process:

(processname ts_p_handler)

This process is detributing a number of resources according to the needs of the protocol processes and the DTE interface process.

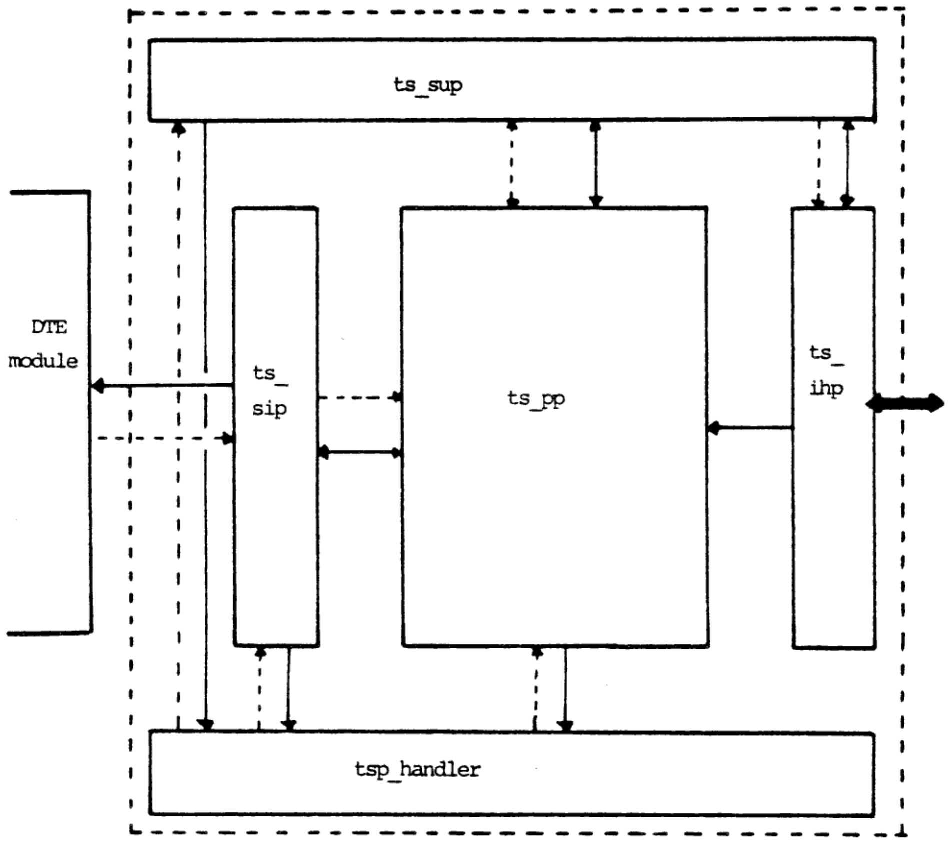


Figure 5.4.1 Logical Structure of the Transport Station

5.5 CENTERNET SESSION CONTROL

According to the ISO reference model (chapter 3) the Session Control is the module between the Transport Station and the Presentation Module. In CENTERNET the Session Control is active at establishment and removal of sessions and ports. In the data exchange phase the Session Control is total transparent in lettergram mode and partly in session mode. Figure 5.5.1 shows the environment of the CENTERNET Session Control.

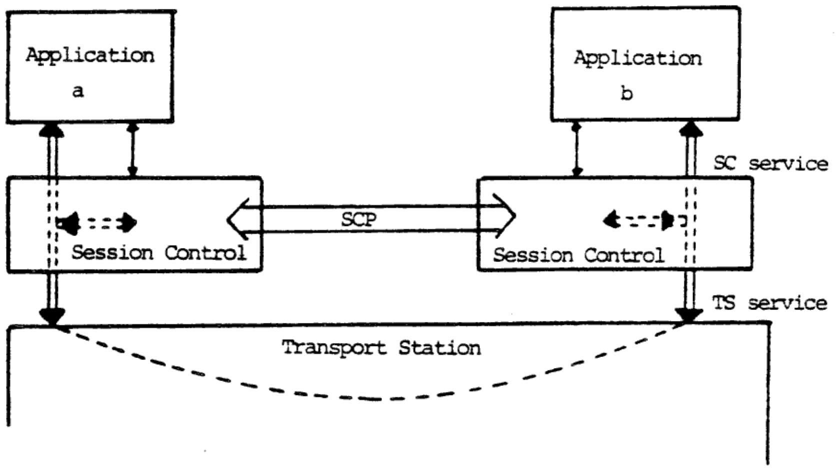


Figure 5.5.1 Session Control Environment.

The basic for the Session Control Protocol (SCP) used in CENTERNET is the Session Protocol proposed by ECMA (ref. (123)). The changes are caused by the structure and functions defined for CENTERNET - Session Control. In ref. (7) the protocol is defined.

5.5.1 Session Control Functions

The Session Control as mentioned above has as main purpose establishment and removal of sessions. The connection and disconnection protocol of the ECMA standard are utilized for this purpose and an addressing mechanism is defined to support the addressing functions supported by the Session Control.

The descriptive model of the Session Layer follows the one proposed by ECMA. A session entity consists of two modules (figure 5.5.2):

a) Session Protocol Machine (SPM)
b) other functions = addressing in CENTERNET

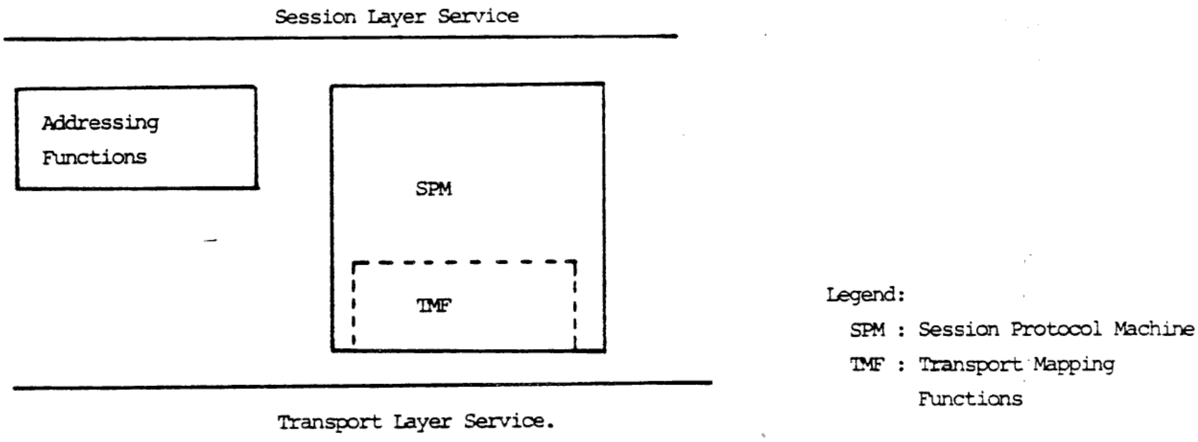


Figure 5.5.2 Descriptive Model of the Session Layer.

In the next four sections the individual functions are shortly described.

Each Session Control module maintains a table (PT-table) of active applications, i.e. known network entities, placed in the same Network Unit as itself. The table is updated each time the "state" of a port is changed or as a result of a Network Control request.

5.5.1.1 Session Establishment

The connection protocol supported by the Session Control guarantees a safe and collision free session establishment.

The decision whether to accept or reject an establish request is placed at the called application. The connection protocol supports exchange of a limit amount of data in the establishment phase.

At the connection operation the application delivers a symbolic or absolute network application address. If the address is symbolic the SC translates it to an absolute and returns the absolute address in the answer.

Furthermore in the connection phase a 32 bits mask is exchange between the two SC. The mask is used for validation purpose. Together with opening a port, an application delivers a receivemask indicating from whom it will accept connection requests.The mask is stored in the PT-table. The calling application delivers a transmitmask together with the connection request.

Each of the above mentioned masks consists of 32 bits. Each of the 32 bits indicates a closed user group. A one in a bit position means that the application belongs to this user group.

The Session Control will check the received transmitmask against the local receivemask. If any bits position are set in both masks the connect request is valid and the session establishment will proceed.

Figure 5.5.3 shows a normally session connection.

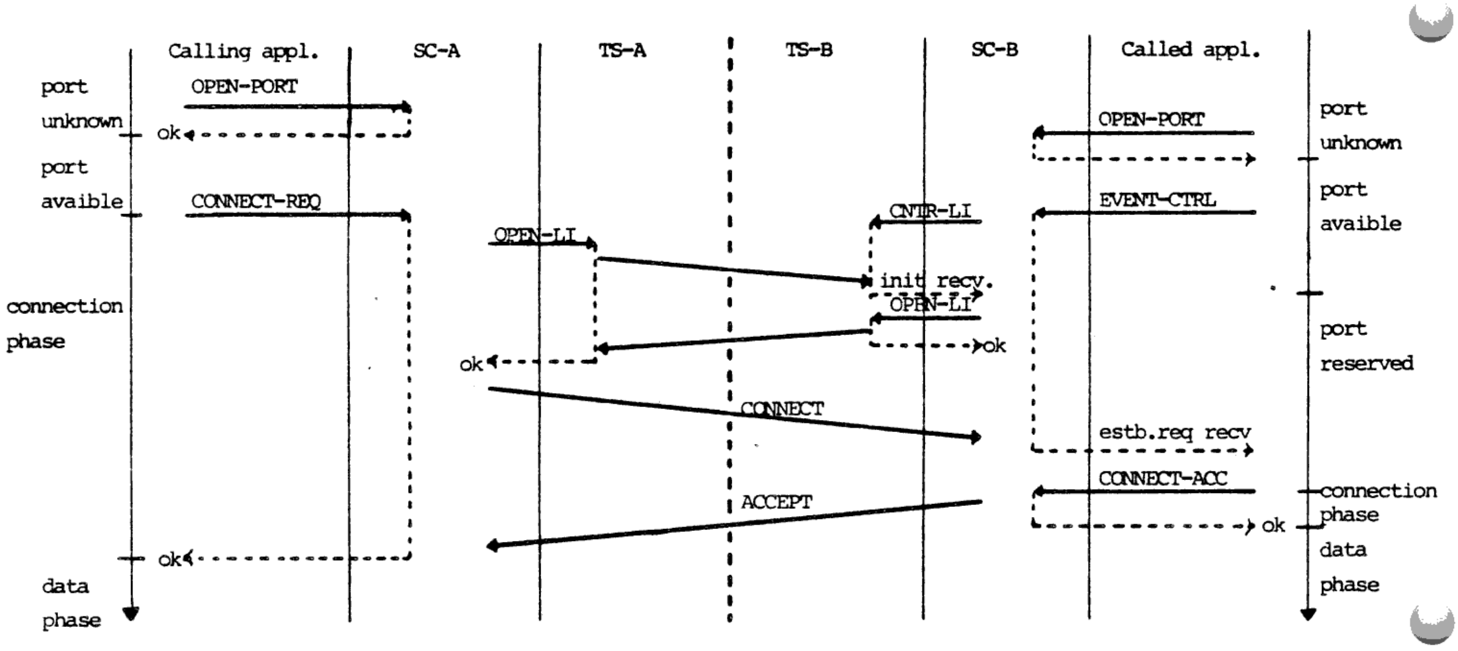


Figure 5.5.3 Normal session connection.

5.5.1.2 Session Removal

The disconnection protocol supported by the Session Control guarantees a safe removal of sessions, i.e. no data is lost if the orderly disconnection mechanism is used.

The disconnection protocol offers two types of session removal, an orderly where both parts are active, the one initiates the removal and the other accepts it, and an abnormal where the session is removed without application accept.

In the first case the application receiving the removal request has the opportunity to reject the request, which cause the session to remain in the data transfer state.

Figure 5.5.4 shows an orderly removal and an abnormal.

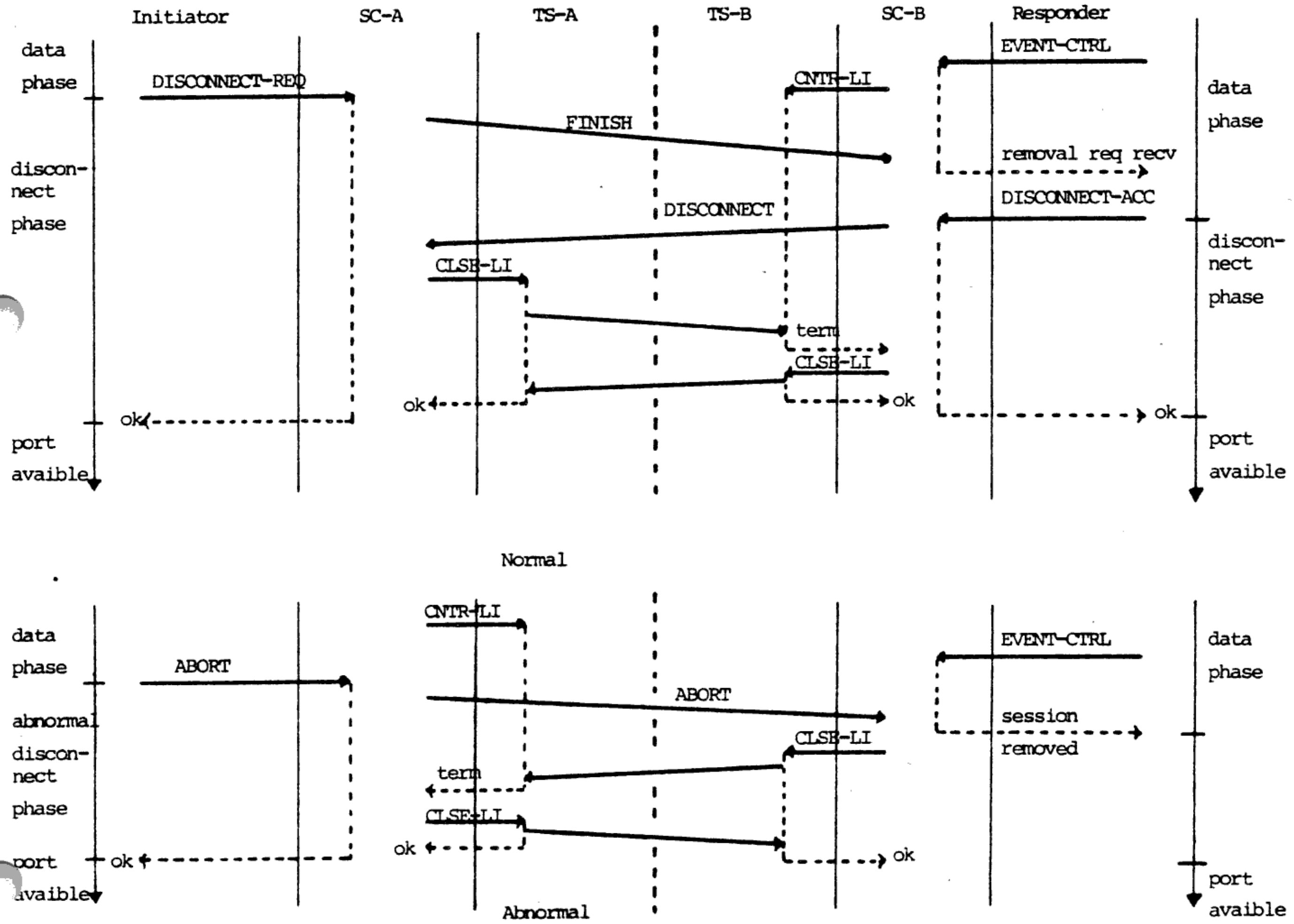


Figure 5.5.4 Normal and abnormal session removal

5.5.1.3 Data Transfer

No function concerning data transfer are supported. The service offered by the Session Control is the service offered by the Transport Station. The Session Control will only monitor the transfer to pick up session removal requests and abnormal terminations.

Two modes of data transfer exists, lettergram mode, where the data is transferred without any knowledge about the state of the receiver, and session mode, where the parts are bound together by a session before data transfer.

5.5.1.4 Addressing

The Session Control is the module managing the addressing in the CENTERNET network.

Applications intending to use the network opens a port identifying themselves by a symbolic name and getting an absolute address.

In the establishment of sessions the initiator may use either symbolic names or absolute addesses of the called application. The names/addresses used must follow the conventions given in section 2.5.2.

In each SC module a configuration table defining all possible network users exists.

This table is included in the SC module at compilation time. When a user connect to the SC module, i.e. issues an open port primitive, the SC module checks that the accociated parameters are identical. If not the open request is rejected.

Furthermore the SC module utilizes the table at session connection time to transform a possible symbolic name to an absolute address and to check that the receiver is a possible session partner.

If the possibility for a session exists, the SC module will make an attempt to establish the session without any prior knowledge about the status of the partner.

5.5.2 Session Control Access

All exchange of SCP-control primitives between the two envolved SC module is made utilizing the liaison established for user data transfer.

Opening/Closure of Ports

The open/close port requests from an application will provoke the following two actions respectively.

OPEN-PT

parameters:	appl. portno	: the calculated/received port number identifying the application.

CLSE-PT

parameters:	appl. portno	: identification of the application requesting the closure of the port.

Establishment/Removal of Sessions

As indicated in figure 5.5.3 - 5.5.4 several TS accesses are used to establish or remove a session.

A normal session establishment will consists of the following sequense:

calling part:
----
(1) table search for identification of the receiver.

(2) initiation of a liaison using the TS service primitive

OPEN-LI

parameters	:	l-ptno, r-ptno, TS address, class
where		
l-ptno	:	port no. of initiating application
r-ptno	:	port no. of receiving application
TS address	:	network address of remote TS
class	:	throughput class used for the liaison, equals the session priority delivered in the connect request.

(3) exchange of Connection Protocol Primitives (liaison service on user port).

called part:
----
(1) waiting liaison initiation using the primitive

CNTR-LI

parameters: l-ptno, buffer

(2) accepting liaison initiation

OPEN-LI

(3) exchange of Connection Protocol Primitives.

A normal removal will consists of the following sequence:

initiating part:
----
(1) exchange of Disconnection Protocol Primitives

(2) termination of liaison using the TS access primitive

CLSE-LI

parameters:	l-ptno	: port no of the requesting application
	cause	: termination cause delivered by the requesting application and which will be delivered to the remote application

accepting part:

(1) exchange of Disconnection Protocol Primitives.

(2) waiting liaison termination using the TS access primitive

CNTR-LI

Caused by network troubles or errors, a liaison may be terminated abnormaly without any preceding SC communication. In order to get this information the SC will always have a CNTR-LI primitive waiting at the TS on each port.

As mentioned above the Session Protocol Primitives are exchanged using the liaison service. Table 5.1 gives the relation between a protocol primitive and the TS service used.

protocol primitive	TS service utilized
CONNECT	letter service
ACCEPT	- -
FINISH	- -
DISCONNECT	- -
NOT FINISH	- -
ABORT	telegram service

Table 5.5.1 Relationship of Session Protocol Primitive transferred and the TS service used.

Data Exchange

The Session Control uses the three set data exchange primitives (lettergram, letter, telegram) offered by the TS service, for exchange of application data.

LETTERGRAM	SEND-LG
	RECV-LG
LETTER	SEND-LI
	RECV-LI
TELEGRAM	SEND-TG
	RECV-TG

5.5.3 Session Control Service

The Session Control module (SCM) supports opening/closure of ports (application entities), establishment/removal of sessions, network addresses maintenance and transformation from symbolic to absolute. In the data exchange phase the SCM is transparent offering the lower layer's data transfer services. The functions of the Session Control Module can be summarized to SESSION MONITORING.

The service interface (needed for these functions) consists of a set of primitives for session supervision (OPEN-PORT, CLOSE-PORT, CONNECT, DISCONNECT, ABORT, EVENT-CONTROL) and a set for data exchange (SEND-LG, RECV-LG, SEND-LT, RECV-LT, SEND-TG, RECV-TG). The last set equals the service primitives offered by the Transport Station (Section 5.4.3) for data exchange.

5.5.3.1 Session Supervision Primitives

OPEN-PORT

parameters : port, func, receivemask, port id

An application uses the primitive in order to become a known network entity, identifying itself to the network. Furthermore a port is opened. An SC port is used as absolute network address of the application and as the application's identification of itself at the SC interface. The application may ask for either the first free port or a specific port. The absolute port number is returned.

CLOSE-PORT

parameters : port, func, port id

If an application wants to stop being a known network entity it must close the port using this primitive.

CONNECT

parameters : port, transmitmask, class, receiver id, data

The CONNECT primitive has two functions, it is used both as a establish request and as a establish response, accepting an incoming establish request.

Request:
----

the application requests the Session Control to establish a session to the specified application, either remote or local. The primitive contains the priority (class) of the data transfer and the application's transmitmask. Further more a small amount of user data can be transferred to the called application. The answer contains the result of the connection attempt and if ok the data delivered by the called application.

Request accept:

The Session Control will continue and finish the session establishment requested from the distant end. The data delivered will be transferred to the calling application.

The primitives are respectively:

CONN-REQ

CONN-ACC

DISCONNECT

parameters : port, cause

The DISCONNECT primitive exists in three variations, a request, a request accept, and a request reject.

Request:

The application requests the Session Control to remove the session and transfer a cause to the receiver.

Request accept:

The application accepts a session removal request and the SC will remove the session.

Request reject:

The application reject a session removal request. The SC will transfer the reject cause to the initiator of the request and perserve the session.

The primitives are respectively:

DISC-REQ

DISC-ACC

DISC-REJ

ABORT

parameters : port, cause

Used to remove a session without the acceptance of the remote application. The Session Control will remove the session and inform the other part.

EVENT-CONTROL

parameters : port, buffer

The primitive is queued at the Session Control and returned when a session event occurs. These are defined to be: establish request received, removal request received, abnormal removal received, session error indication.

At return the primitive contains the data the Session Control has received with the request and if "establish request" the name/address of the requesting application.

If no event primitive is pending the action of the SC depends upon the event according to the following table:

event	SC action if no EVENT-CONTROL is pending
establish request	rejected
removal request	rejected
abnormal removal	accepted
session error indication	accepted

Table 5.5.2 SC action on different events.

5.5.3.2 Data Transfer Primitives

The primitives for data transfer support equals the TS set for the same purpose in order to make the SC transparent in the data phase. Six primitives are defined for 3 kind of data transfer:

Lettergram mode	:	LETTERGRAM	SEND-LG, RECV-LG
Session mode	:	LETTER	SEND-LI, RECV-LI
		TELEGRAM	SEND-TG, RECV-TG

For parameter specification please refer to section 5.4.3.

Further details about the usage and the parameters please refer to section 5.4.3.

5.5.3.3 Receiver Identifications

The identification consists of two parts of which one always shall be present. Both absolute and symbolic, abbreviated addresses are allowed.
The identification is delivered to the Session Control as an ASCII text string.

The format used is specified in section 2.5.2 and further details may be find in that section.

The addressing service supported in lettergram mode, will equals the one offered by the Transport Station.

When the Session Control returns an identification/address it will always be an absolute address consisting of two parts, a port number in binary code and a network unit address as an ASCII textstring.

5.5.4 Network Control Service

In this section the service offered the NCP is described. Two types (control and monitoring) of network control functions are supported:

control functions (CONTROL)
monitoring functions (EVENT, SENSE, and GET STATISTIC)

The next two subsections will give a general description, while the exact contents and formats are described in ref. (13).

5.5.4.1 Control Functions

The NC may activate/deactivate the monitoring functions. I.e. the NC can indicate to the module whether or not statistical information shall be gathered, and which reports that shall be logged. A mask indicates this and the NC can set this mask utilizing the command

SET NC MASK

The NC can close a port utilizing the command

CLOSE PORT

The intention of the operation is that the NC should have the possibilities to close a port, in case the user is incapable of closing the port.

The relation between an symbolic TC address and the absolute may be changed using the operation.

SET TC TABLE ENTRY

5.5.4.2 Monitoring Functions

The SC module supports three different monitoring functions

- event (report)
- sense (immediate state/status)
- statistic

REPORTS

EVENT, event-type, event-inf.

By return the fields contain the information.

- event-type identification of the causing event.
- event-inf. further information concerning the event.

The following events will trigger the report function:

(1) session connected
(2) session disconnected
(3) port opened
(4) port closed
(5) unsuccessful connection

SENSE

The SENSE operations are used to get either an immediate state or an immediate status for the whole module or for a data path through the module.

The following operations are supported:

SENSE SC
SENSE PORT
GET OPEN SC PORTS

As specified in section 5.5, the Session Control uses a network configuration table for addressing purpose. The NC can get information about this table using the operation

GET NET CONF ID

The Session Control maintains a PT-table also for addressing purpose. The Network Control can get entry information from this table using one of the two operations

GET PTAB ENTRY ABS
GET PTAB ENTRY SYMB

STATISTIC

The STATISTIC-operations are used to retrieve statistical information concerning either a data path or an access address.

The following operations are supported

GET SESSION STATISTIC
GET PORT STATISTIC

5.5.5 Logical Structure of the SC Module

The logical structure of the Session Control Module reflects the division of functions into the addressing mechanism and the protocol support. The module consists of three blocks:

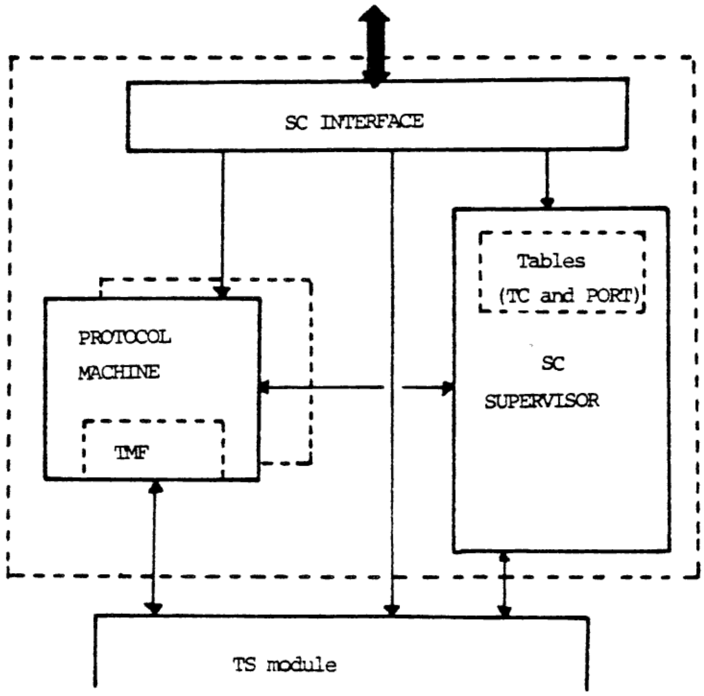


Figure 5.5.5 Logical Structure of the SC Module.

SC INTERFACE (switch):

This block interfaces the applications to the SC module. It checks the primitives and routes them to the right block for further processing.

PROTOCOL MACHINE:

The block is the implementation of the Session Control Protocol. Furthermore it include a set of access routines for TS access.

SC SUPERVISOR

The block implements the addressing mechanism of the Session Control Module. The network configuration table, the TC-table and the PORT-table (telephone directory) are located in this block. Furthermore it interfaces to the NCP module to support NC operations.

5.6 CENTERNET VIRTUAL TERMINAL PROTOCOL (VTP)

The Virtual Terminal (VT) defines a model of a terminal, i.e. a concept used networkwide to describe it. The Virtual Terminal Protocol (VTP) defines the set of primitives being exchanged through the network together with the rules for using them. The VT software exchanges the primitives by means of the Session Control Service Interface. This means that the VT lies on top of the Session Control (figure 5.6.1).

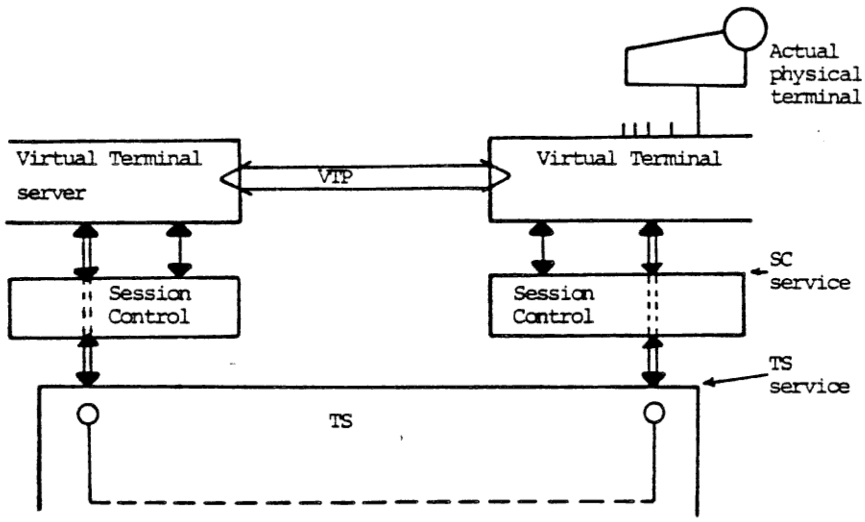


Figure 5.6.1 VT Environment.

One Virtual Terminal will occupy one SC port, as it is a CENTERNET entity.

The VT software converts VTP primitives to the actual terminal representation and visa versa and may handle several terminals of the same physical type.

5.6.1 Virtual Terminal Functions

The Virtual Terminal Protocol includes an abstract terminal model (defined in ref. (2)). The model consists of the following components (figure 5.6.2):

- a presentation unit, used to visualize the data structure
- a data structure
- an alarm device

- a keyboard, used to enter data. The keyboard may include attention and function keys

- a control unit

- auxilary devices, used for hardcopy or as alternative input/output. Only the hardcopy device is supported at present in CENTERNET VT Protocol.

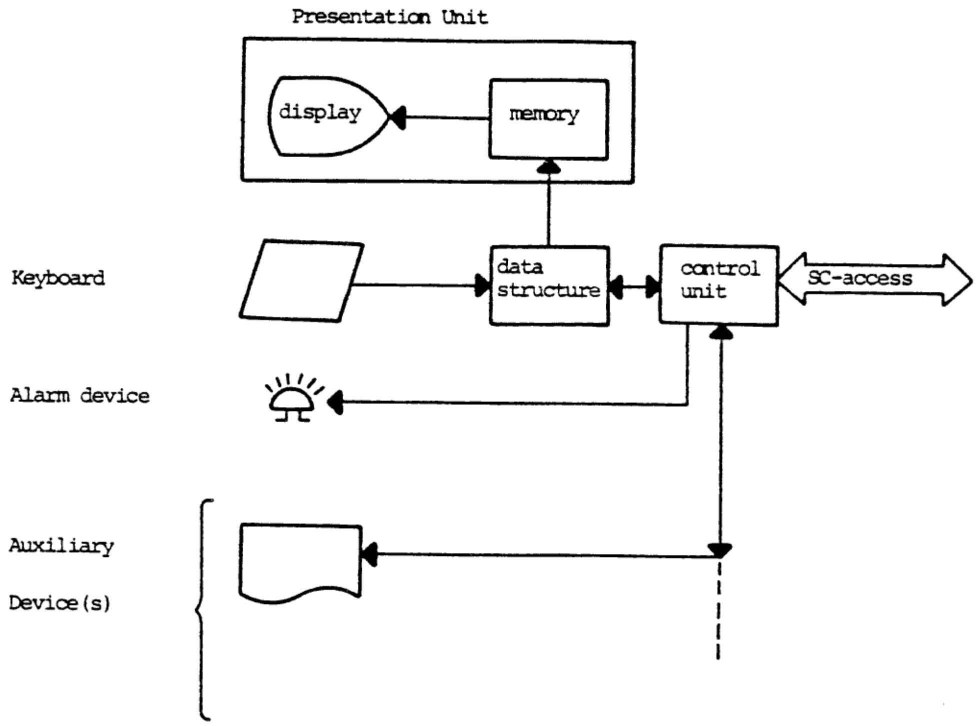


Figure 5.6.2 VT Components.

The components are described in details in ref. (2).

The functional characteristics of the components depend on the VT parameters (please refer to section 5.6.1.3 and ref. (2).

The abstract model of the terminal, in particular the data structure, may be structured in different ways:

- SCROLL mode (line oriented terminals)

- DATA ENTRY mode (format oriented terminals)
- NATIVE mode (non standard terminals)

The mode determines the precise effect of the VTP primitives and reflects the abilities of the physical terminal.

5.6.1.1 Session establishment and removal

In principle a VT may access any port supporting VTP. These ports fall in one of two categories:

1) A host timesharing interface module, i.e. a VT server at a host.

2) Another VT.

Whenever the VT accesses a port the Session Control module validates the access right as described in section 5.5.

Network access is divided into the following phases:

1) establishment of a local connection from terminal to VT handler

2) establishment of a session through the network

3) normal data transfer by means of VTP primitives

4) removal of the session

5) removal of the local connection.

Phases 1 and 5 are only significant when it is possible to distinguish between a terminal-idle condition and a terminal-disconnected condition as is the case for dial-up services. These phases includes steps concerning opening and closure of a SC port, which reflects the availability of the terminal to other network entities.

Phases 2 and 4 will influence allocation of network resources, according to the needs for a session to have buffers etc. allocated.

When the physical terminal is ready for work, the VT software will set the default profile for the terminal in question (including values and range masks) and open a port. The terminal is now present as a network entity. The address is fixed for permanently attached terminals, to allow a VT Server to establish the session.

The connection between a VT and a VT Server may in principle be initiated from both ends.

The connection is performed in two steps

1) SC session establishment.
   User data is exchanged to establish which party is to play the VT role and which is to play the VT server role.

2) Connection phase described in the VTP ref. (2).

The user of a terminal gives a command requesting a connection to a certain port supporting the VTP. The command allows as well absolute as symbolic addressing, with abbreviations for the services most frequently used (please refer to section 5.5.3.3).

A session may be removed due to one of the following reasons:

- The user or application has finished its activities.

- Components of the system are closing as a part of the daily routine.

- Parts of the network fail.

The VTP disconnection procedure is replaced by the SC disconnection procedure.

The port will be closed if the terminal is physically disconnected.

5.6.1.2 Data exchange

In the data phase primitives are exchanged according to the 'elements of procedures' described in ref. (2).

The primitives belongs to one of five classes:

1)	TEXT:	TEXT-SEG, NL, CR, POS, NEXT-U-FIELD, ATTRIBUTE, HIDE, DEL-ATT, ERASE-UN, DEL-ALL.
2)	CONTROL:	PLEASE, CLEAR-MARK, ASSIGN.
3)	INTERRUPT:	ATT, CLEAR.
4)	PARAM:	R/I/S/A/D-PARAM.
5)	CON/PARAM:	CONN, CACC, DISC, MARK.

The individual primitives, the usage, and the representation are described in CENTERNET Virtual Terminal Protocol, (ref.

(2)). Furthermore it describes the subset for each terminal mode, native, scroll, and data entry.

5.6.1.3 Virtual Terminal Parameters

The parameter settings for a VT have influence on the functional characteristics of the components. The parameters defined are described in details in ref. (2) and are listed here:

terminal mode	: structure of the components
addressing capability	: addressing possibilities in addition to new line
line-discrete	: the with of the data structure
page-discrete	: the length of the data structure
character-set	: the character representation used in text segments
printer-line-discrete	: the with of the attached hard-copy device
attribute-capability	: the possibility of implemented attributes

The VTP supports two mechanisms of parameter setting.

1) As part of the connection phase using the commands CONN, CACC, and DISC.

2) During a dialogue on the servers initiative using the param-block and the primitives R/I/S/A/D-PARAM.

If a VT wants to start a parameter setting phase, it has to send an attention (ATT(15)) to the server, indicating the wish of parameter setting.

5.6.2 Virtual Terminal Access

The Virtual Terminal uses the services offered by the Session Control to establish a session and to exchange data.

A connection between a terminal and a server is implemented as a session. Text blocks, control blocks, param blocks, and connection commands are exchanged using letters, interrupts by using telegrams.

When a terminal is physically connected, a port is opened using the SC-command

OPEN-PORT

parameters:

term-id	: terminal identification	
func	: 0 = first free	please see sec. 5.5.3, too
port	: not used	
receivemask	: please refer to section 5.5.1.1	

If the open function is ok the absolute port number is returned in the answer otherwise an error status is returned.

The terminal is now a known network entity and can either initiate or receive a session establishment request.

A session is established using the SC-command

CONNECT

parameters:

port	:	own identification
transmitmask	:	please refer to section 5.5.1.1
class	:	term class
receiver	:	symbolic or absolute identification of receiver (see section 5.5.3.3).
user data	:	indicates whether a VT or an application (VT server) is calling.

In case the terminal intends to be the passive part of the connection, it uses the command EVENT-CONTROL. When the EVENT-CONTROL is received indicating a session establishment request, the following SC command is used to accept the session.

CONNECT-ACCEPT

parameters:

port	:	own identification
user data	:	VT if the caller is a VT server, other wise VT server.

The EVENT-CONTROL command may also be issued after session establishment and the answer will, if so, be received at session removal.

After the establishment of a session the terminal uses:

RECV-LI (for normal data input)

SEND-LI (for normal data output)

RECV-TG (interrupts)

SEND-TG (interrupts)

To remove the session the next three commands as can be used

DISCONNECT-REQUEST

parameters:

port: own identification

cause: VTP disc reason X'01' (normal termination)

will remove the session but keep the port open, so the terminal has the same possibilities as after an OPEN command.

DISCONNECT-ACCEPT

parameters:

port: own identification
cause : VTP disc reason X'00' (confirmation of disc)

The command is used when a session removal request is received in an EVENT-CONTROL answer.

ABORT

parameters:

port : own identification
cause : VTP disc reason (except X'00' and X'01')

When the terminal is physically disconnected the port will be closed using the SC command

CLOSE-PORT

parameters:

port : own identification
func.: 3 = close absolute port

5.6.3 Virtual Terminal Service

In this section the service interface of the Virtual Terminal should have been described. This is not possible because the software supporting the actual physical terminal and the Virtual Terminal software are very integrated.

This is caused by the fact that the VT is responsible for the changing from a specific terminal language to the network standard language and visa versa (figure 5.6.3).

So if a general interface to the Virtual Terminal Protocol exists the adaption/changing functions would be placed outside the Virtual Terminal Protocol Module, and this module would only contain very rudimentary functions as primitive blocking and connection/disconnection mechanism.

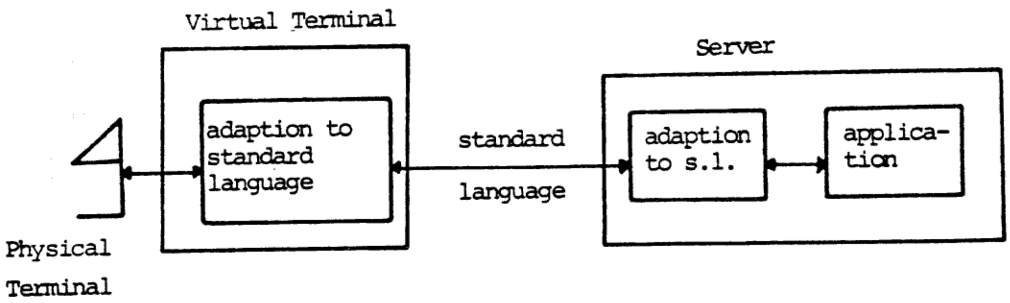


Figure 5.6.3 Adaption to standard language.

5.6.4 Network Control Service

In this section the interface to the NCP is described. As specified in section 6.3 three types of network control functions exist:

1) control
2) monitoring
3) message broadcast

To support these functions five primitives are defined:

1) CONTROL
2) EVENT, SENSE, GET STATISTIC
3) BROADCAST

The individual functions and primitives are described in the following sections.

5.6.4.1 Control Functions

The control function are performed by the NC using the primitive SET. The interpretation of the information in the CONTROL operation, i.e. the individual control function, depends on the actual Virtual Terminal module, and will be described in the relevant sections. E.g. an X.28-Scoll Mode function is to set a terminal profile and is explained in section 7.2.4.

A general control function is the possibility to set the receive- and transmitmask. This is done utilizing the command

SET USER MASK, physical port, receivemask, transmit-mask

For details about the two masks please refer to section 5.5.1.1

The NC may activate/deactivate the monitoring functions. I.e. the NC can indicate to the module whether or not statistics shall be performed, which reports that shall be logged. A mask indicates this and the NC can set this mask utilizing the command.

SET NC MASK

5.6.4.2 Monitoring Functions

Three type of monitoring functions exist:

1) an event generated report.
2) an immediate status/state.
3) statistical information.

The event generated status (reports) will inform the NC every time a specified event occurs.

The structure of the information in the buffer depends upon the actual VT module and is described together with the module. The primitive used for this type of status is EVENT.

The other type, immediate response, is obtained using the primitive SENSE or GET STATISTIC. The receiving of these primitives by the VT module will trigger a reading of the state or a statistical record, which will be returned. Again the details are described together with the actual VT module.

5.6.4.3 Broadcast

The broadcast function is located in each relevant VT module. The broadcast message is transferred from the NC to the VT module using the primitive BROADCAST.

5.6.5 Logical Structure of the VT Module

The logical design of the Virtual Terminal Module is elaborated to support enhancements. Figure 5.6.4 shows the basic structure of the module.

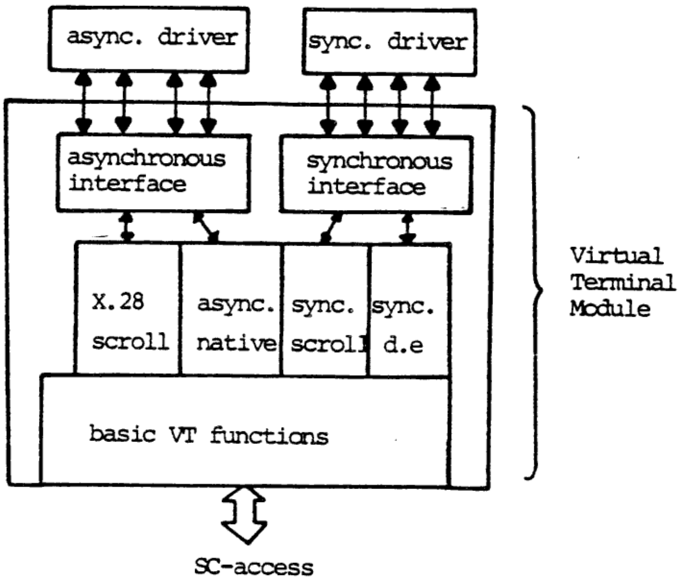


Figure 5.6.4 Logical Structure of VT.

The individual logical blocks are

1) asynchronous/synchronous interface:
   interface to the actual terminal driver

2) X.28-scroll, async-native, etc.:
   logical module connecting the actual terminal protocol with the virtual terminal protocol

3) basic VT functions, SC access:
   function for establishment and disconnection of data paths and access to the transport network represented by the Session Control.

An example of an implementation specification is given in section 7.2 and ref. (3).

5.7 CENTERNET REMOTE PRINTING PROTOCOL (RPP)

This section may appear later.

5.8 CENTERNET FILE TRANSFER PROTOCOL (FTP)

5.8.1 FTP Functions

The FTP provides a method for the transfer of data, in the form of complete files, across a network, both to and from a remote computer.

The FTP is built on the premise that a single standardised representation can be set up for a File, or for an organised set of files (a Filestore). This representation can be used in the protocol to express the transactions which are to be performed, mappings being made by each participant to relate the standard descriptions to local resources. The description of this standard conceptual filestore is resolved into a set of distinct characteristics called attributes, the values of these attributes identify or describe the files to be transferred.

The file transfer takes place in a number of stages. The initial exchanges take the form of commands, each with a number of optional parameters. Each parameter specifies one of the Attributes by using a code number, and gives a suitable value for that Attribute. Once the file has been correctly identified, and the conditions for the data transfer established and agreed, the file data can be transferred. Firm control of the flow of data is achieved by using a second set of commands. The exact details of the different levels of information exchange are given in ref. (4).

One of the most important of the transient attribute values to be considered during the actual file transfer is that for the 'Mode of Access'. This can take one of several values which fall into two general groups:

a) the transfer of data into the filestore, and
b) the transfer of data out of the filestore.

A Start File Transfer (SFT) command specifying a transfer of data into the filestore (such as Make or Append) is termed a 'take' (i.e. the filestore takes the data). An SFT command specifying a transfer of data out of the filestore (such as Read) is termed a 'give' (i.e. the filestore gives the data). This convenient terminology will be used in the following sections.

The diagram below (figure 1) illustrates the general exchange of commands and data in the FTP. In this example a file is transferred from Q's filestore across the network to the process P:

INITIATOR (P)	RESPONDER (Q)	Commentary
SFT + give and other parameters →		Process Q receives request, finds file referred to, then accepts transfer
←	RPOS + parameters	Process P checks that required file has been found and then enters data transfer phase
Go →		
⇐	data of file	The data records of the file are now sent from Q to P, followed by an end of file exchange
STOP →		Process P finally terminates transfer

Figure 5.8.1 The General Mechanism of the FTP.

5.8.2 FT Access

In the CENTERNET implementation the process that initiates a transport association must operate on the SC interface. When establishing the association (the session) the necessary functions are performed by the SC module. After a successful establishment the actual file transfer is carried out by the SC module.

If the initiating process is placed in a host machine, connected to a RC3502C through an FDLC/HDLC connection,this connection must be set-up in order to communicate with the respective modules. The services offered by the NETWORK INTERFACE MODULE (NPM) are the same as those offered by the abovementioned RC3502C modules but the exact format of primitives is different.

Before starting the transfer, the FTP PROCESS issues an OPEN primitive to the SC module/actual service interface. When a positive response returns the session is established utilizing the function

CONNECT-REQUEST

possible parameters :

portnumber	: answer parameter in OPEN
transmitmask	: please refer to section 5.5.1.1
class	: file transfer class
receiver	: Ident of receiving filemanagement system.

The establishment phase is considered finished when the CONNECT REQUEST primitive is received with a positive result.

The partner of the transfer will have used the function

EVENT-CONTROL

to get hold of an incoming connection request. If the request is acceptet the filemanagement system will indicate this to the SC module utilizing the function

CONNECT-ACCEPT

Commands and data are exchanged between the FTP-processes through the use of the SC-primitives

RECV-LI/SEND-LI

and

SEND-TG/RECV-TG

according to the type and size of the information to be sent.

For all primitives the parameters are

local portno	: the portnumber received as responseparameter to the OPEN primitive
data-buffer	: address indicating start of buffer
data length	: number of bytes to be transmitted.

For the SEND-TG primitive the acknowledgment parameter can be used if the situation requires end-to-end control.

When data are to be received flow control can be controlled by the responder, through the

RECV-LI

while data can be sent through

SEND-LI

When the file transfer has been completed, the responder can remove the session by issuing a DISCONNECT primitive to the SC module. The initiator will receive this request in the answer of an

EVENT-CONTROL

and will accept the request using the function

DISCONNECT-ACCEPT

The initiator may at any time during data transfer (in error situations) request the session disconnected or aborted utilizing the functions

DISCONNECT-REQUEST

ABORT

respectively.

After the termination phase the ports will remains open for file transfer to other processes.

5.8.3. FT Service

The CENTERNET file transport protocol, formally defined in ref. (4) is a subset of

A Network Independent File Transfer Protocol

prepared by High Level Protocol Group. The original document is denoted INWG Protocol 86 ref. (124).

As the original protocol is intended in future to be used as a Job Transfer Protocol and Remote Printing Protocol as well, and these are no part of CENTERNET facilities at this moment, the subset has been defined by extracting the necessary commands from the original protocol. This should make future additions easily incorporated.

Transfer of a file, between two computers, demands that the two processes performing file operations in either systems are associated. The protocol allows information to be exchanged, concerning the attributes of the transfer in an establishment phase.

Such attributes are

Direction of transfer
Mode of access

Data compression
Temporary transmission stop
Retransmission possibility.

Both processes can indicate their capabilities and reject the transfer if a mismatch of attributes occur. If the initialization is successfully performed, data will be transferred as binary octets. The body of the file is carried in records, limited by the SC service to 2000 bytes, that can be structured into subrecords.

If errors are detected by either process, an error reporting/recovery mechanism is available.

The actual primitives offered will depend on the surrounding file system. As an example section 9.2 describes an RC8000 implementation.

This page is intentionally left blank.

6. NETWORK CONTROL

This chapter describes the functions and outlines the structure of the Network Control System in CENTERNET.

Section 6.1 introduces the reader to the basic ideas and the conceptual approach of the NC system.

Section 6.2 describes the topological framework under which all NC facilities operate.

Section 6.3 lists the basic services which are available for the NC management, and section 6.4 defines the program modules which are put into operation in the initial phase.

It should be noted that this chapter only gives a functional description. Exact definitions of formats and information contents are part of the program specifications and may be found in the references given in Appendix A.

Note:

some of the subsection in this chapter will not be available until January 1983.

6.1 Introduction to CENTERNET Network Control

Realizing that the control, monitoring and maintenance of a larger network system involves many complex interactions between network components and between the network and the operating staff, emphasis is put into defining a strong framework within which the facilities may expand in a smooth and continous manner.

This framework contains many levels of control mechanisms. In the context of CENTERNET 3 categories of NC functions are defined::

Monitoring functions

Facilities for retrieving information from each network component. Incorporates the automatic failure detection of transitions into the "down" state of any network component (hardware/software), the collection and presentation of information relative to the "on/off" status of individual components and collection of information stating the general activity in the network. The information may be timerelated (statistics, counters) or event driven (state, errors). Selected parts of the information should be reported automatically according to specific rules.

Control functions

Primitives which enables the control of specific modules (hardware and software) within the network.

Enables the execution of command functions, which consist of a set of capabilities to interpret and execute commands which might be generated either manually or automatically. These commands are intended to initialize and modify the status of network components. Some typical command primitives are:

- loading and initialization of software modules
- start/stop of software/hardware modules
- change of operating parameters
- activation of special test or trace routines

Management functions

Functions or utilities which give the network administration suitable user-oriented tools for network control/monitoring, test, maintenance and management of the entire network. The NC administration utilizes the monitoring and control primitives previously mentioned.

The basic function is the recording function, which ensures the logging of all relevant information on a permanent medium, preferably host based. Thereby a selected set of network events such as important status transitions and critical incidents can be printed. The recording function provides the network management with real-time information to be used in decision-taking.

The network management include further functions such as:

- utilities for maintenance and manipulation of log files (statistics, overviews, alarms etc)
- utilities for maintenance of module configuration files
- file transport utility
- artificial traffic tools
- diagnostic aids.

Described topologically the monitoring/control primitives are related to the network modules involved with the data transports whereas the management functions relate to one or more network control centers (fig. 6.1).

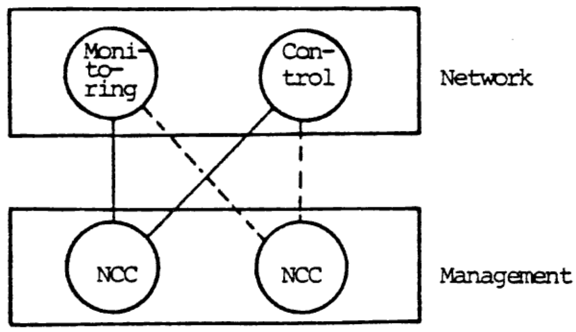


Figure 6.1 Trisection of NC.

The interaction between management, control and monitoring primitives is carried out by using the network as transport system.

In order to establish a common structure for this information exchange a network control protocol - the supervisor protocol - is defined.

The supervisor protocol system establishes a framework for addressing and accessing each individual NC facility. It relies on the correct functioning of the transportation service, but does not interact with normal network operations.

Distributed Control Versus Centralized Control

Within the network community, there is a continuing debate whether, when designing a control system, to use the distributed capabilities of the network itself by choosing a decentralized scheme, or whether to prefer a centralized approach, or a mixture of both. The debate is particulary active concerning recovery functions such as packet routing or even worse: updating of routing tables.

In a centralized approach, all nodes and terminal concentrators requiring services must have access to the centre, which in turn needs to be able to obtain information about the network as a whole. In a degrated network this becomes difficult if not impossible.

In a distributed approach, each part of the system is individually responsible for performing the services required.

The proposal for CENTERNET is to adopt a combination of both approaches, with the balance between them dictated by the level of service required.

Control Centre Constraints

Since management decisions may be based on external factors, no control system can be entirely automatic. For this reason, an important consideration is the ease and efficiency with wich an operator can communicate with the system.

As most of the CENTERNET nodes and concentrators will be located within the computer centres, most likely in the same rooms as the large mainframes, actual network operation may be carried out by the local computer operation staff, for whom CENTERNET operation forms part of their standard duties.

Consequently, operational constraints are:

1. A simple and flexible operator interface, both for command and information presentation.
2. Remote command capabilities, since operators need not be physically present at remote nodes.
3. Overall network status and activity monitoring, since operational descisions depend on an aggregated picture of the whole network.
4. Powerful local control capabilities, to allow hardware and software interventions to be performed by specialists directly at the relevant equipment and during production.

Important to realize is, that in order to provide a sufficiently high availability for CENTERNET, essential control functions - such as automatic retries after component failures - can not be dependant on the accessibility of a single critical element, such as a control centre.

6.2 Network Control Architectural Structure

6.2.1 NC Topology

The CENTERNET NC system is designed according to a distributed approach, i.e. the NC facilities including monitoring, control and management primitives are distributed on several geographical locations. Thus the Network Control Centre functions are distributed on a number of Control Centres, each Control Center being able to manage the entire network. Each Control Centre relies on a number of subcenters Network Control Probes (NCP)- located in each network subsystem (terminal concentrator, network node). Again the NCP relies on internal control points Local Control Probes (LCP), associated to each program module.

Summarizing, the NC topology consists of the following:

- Network Control Centres:

  Located in a RC8000 Host Computer. A Control Center consists of a nucleus and a number of NC utilities, each offering a specific network management service. Each Control Centre in CENTERNET has an identical nucleus but may vary in respect to the set of utilities.

  The Control Centres may communicate with each other and with NCP's by means of the network supervisor protocol (cf. section 6.2.3).

- Network Control Terminal (NCT):

  The software constituting the NCT is located in an RC8000 Host Computer, whereas the display unit may by located anywhere in the network (directly connected to the RC8000 or connected as a normal CENTERNET terminal). The NCT communicates with the nucleus (NCC) as a special utility.

- Network Control Probes (NCP):

  Located in each subsystem of the network. A subsystem is defined as a logical network entity, e.g. terminal concentrator (TC) or network node (NN) and is typically related to a specific network processor.

  The NCP can be accessed by a Control Centre via the supervisor protocol. The major task of an NCP is to execute network monitoring and control functions on request from a Control Centre. The NCP again invokes or accesses local control probes.

- Local Control Probes (LCP):

  Each subsystem consists of a number of program modules each containing a LCP (module control and supervision, hardware control etc.) A LCP is accessed from the NCP utilizing a set of standard procedures.

NCC and NCP modules are accessed through the Session Layer (dedicated ports), which implies that two NCP's may communicate with each other (as well as two NCC's).

Fig. 6.2 illustrates the topological relations.

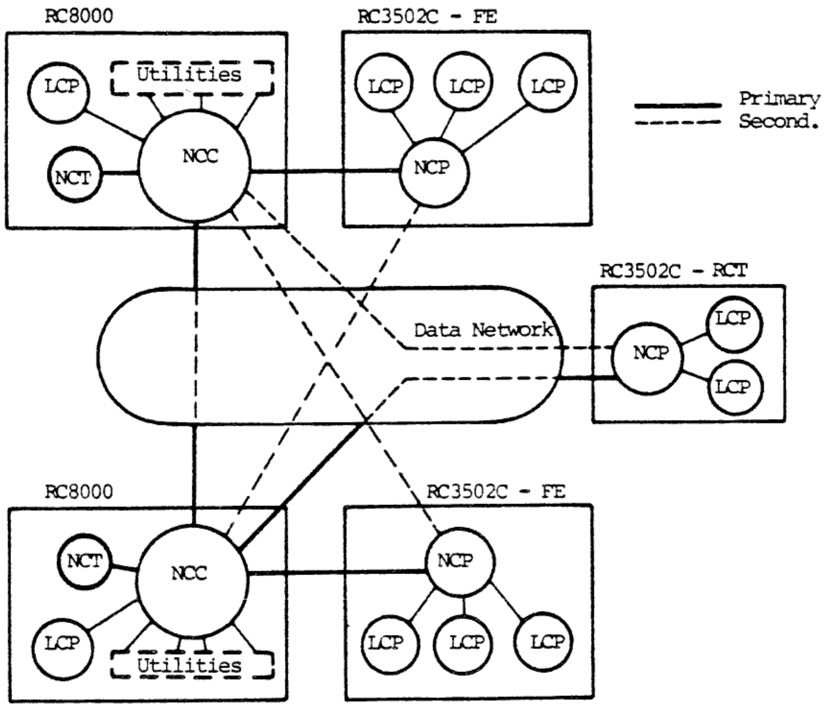


Figure 6.2 NC Topological Relations.

6.2.2 NC Addressing

As illustrated by fig. 6.2 the intercommunication between the NC modules requires an access and address structure to be defined.

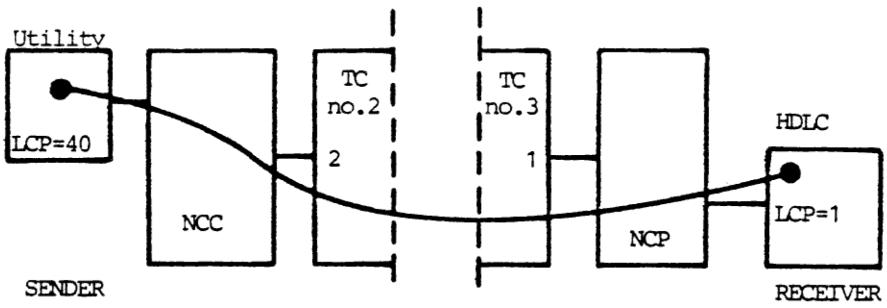
The NC information exchange between NCC's and NCP's relies on the transportation service of CENTERNET, i.e. the Session Layer - SC. The LCP's and the NCC utilities represent a sub-address scheme indicating the ultimate receiver/originator of the supervisor transaction. Because of the one-to-one corre-spondance of NCC/NCP modules and a TC the access between NCP and NCC can be permanently allocated to specific port numbers, e.g. port no. 1 and 2 respectively.

In this way a NCP function (represented by a LCP) is address-ed by:

RECEIVER ::= TC, PORT, LCP.

The LCP subaddress is assigned to each utility module (management) and to each software module which is subject to network control.

Fig. 6.3 gives an example of NC addressing.



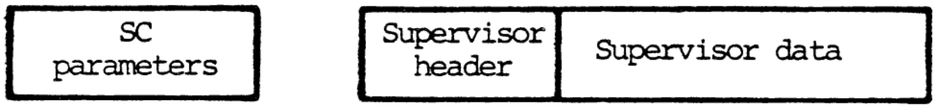
RECEIVER: TC, PORT, LCP = 3,1,1 SENDER: TC, PORT, LCP= 2,2,40

Figure 6.3 Example of NC addressing.

6.2.3 Supervisor Protocol

The purpose of the supervisor protocol is to offer a standard method for exchanging supervisor information between any pair of NCC or NCP. The basic transport service is offered by the SC operated in lettergram mode with acknowledgment.

The datapart consists of a supervisor header and the supervisor data field.



The SC parameters are set as follows:

- Local portnumber = { 1: NCP originator; 2: NCC originator }

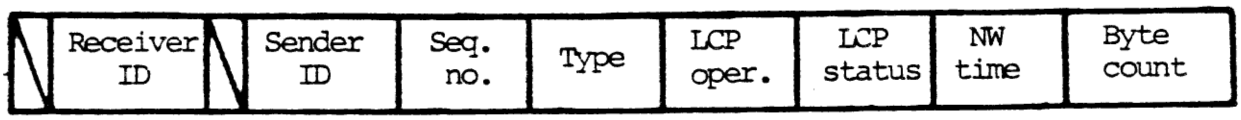
- Remote portnumber = { 1: NCP destination; 2: NCC destination }

- Remote TC address

- Acknowledgment Request = 1

The above mentioned parameters are, except the last, also available from the SC when a lettergram in received.

The supervisor header follows the format as shown:



RECEIVER ID : Receiver of the transaction (LCP address)
(15 bits)

SENDER ID : Originator of the transaction, normally
(15 bits) receiver of answers and events (LCP address).

SEQ.NO: (8 bits)	Used to identify message/answer/events correlations (each producer of supervisor messages/-events applies sequence numbers, the response initiated by the message use the same sequence number).
TYPE: (8 bits)	Message Answer Event
LCP OPER: (8 bits)	Specification (mode) of message, answer or event.
LCP STATUS: (16 bits)	Optional status field used to indicate current status of module addressed (answer, events).
NW TIME: (12 BCD digits)	Network time, time of generation of transaction Format: second, minute, hour, day, month, year.
BYTECOUNT: (16 bits)	Length of supervisor data

The use of message, answer, events is defined as follows:

- Supervisor Message:

  Used as a <u>command</u> from a NCC/NCP to a NCP/NCC, initiates typically some kind of action, which may produce a result or which may cause a state or status change. An answer containing result of the action may be returned to the sender (solicited message). The action may also incorporate the initiation of an automatic reporting mechanism, where results are transmitted to a defined receiver.

- Supervisor Answer:

  Carries the <u>result</u> of a specific supervisor message. The answer is normally directed to the sender of the message, but may also be directed to another receiver. This receiver is called the <u>exception</u> receiver and in the NCP the address can be set utilizing the LCP operation 'set exception return address'.

- Supervisor Event:

  Produced by a LCP as a result of an event. The receiver of the event is determined by the event address in the NCP. This address can be set/changed by the LCP operations 'set init event address' and 'set event return address'. The generation of an event in a LCP may be locked/unlocked utilizing the LCP operation 'set NC mask' to the specific LCP.

The contents of the supervisor user field varies according to the NC function being executed.

Some typical contents are:

- statistical information
- text strings
- parameters for control of program modules
- program load information/contents
- status and event information
- remote addresses.

Concludingly the NC supervisor protocol offers a transparent service which can be used between any two NCC utilities and NCP functions (=LCP). This enables a flexible development of new NC functions without affecting the existing.

6.3 NC Facilities

This section describes the required NC facilities which are available in the initial state. New facilities can be included either as a consequence of new LCP functions within the network or due to enhancements of the utilities within the network management. Thus the NC facilities may be described both as a service offered by LCP's and as the utilization of this service as it is carried out by the network management. The realization of the NC facilities (as a number of program modules) is described in section 6.4.

6.3.1 Configuration Files

The entire network is build up by a number of hardware and software modules.

At any moment the system can be described by a certain set of these modules. The current information is held in a

HARDWARE CONFIGURATION FILE (HCF)

and a

SOFTWARE CONFIGURATION FILE (SCF)

the contents and updating of which are at the disposal to the network management.

Updating of configuration files

To appear January 1983.

Contents of configuration files

To appear January 1983.

6.3.2 Remote Hardware Control

The normal operation of each hardware module may be switched on/off controlled by supervisor transactions.

The hardware modules include processing units (PU's), communication links, controllers and physical ports. The on/off switching of a PU implies program load and initiation whereas the activation/deactivation of links and ports are performed by issuing control operations to the driver in question or the module handling the driver.

The remote hardware control overrules any software control and may thus interrupt current operations.

The following hardware elements are subject to remote hardware control.
- RC3502C Sync.ports/multiplexers
- RC3502C Async.ports/multiplexers
- RC3502C Channel Link (FPA)
- RC3502C Processing unit (PU)
- RC8000 Channel Link (FPA).

6.3.3 Remote Hardware Monitoring

The operation of each hardware element which has been activated by the hardware control is supervised by the LCP located in the software module (driver) which operates on the hardware. The on-line hardware monitoring includes:

a) A report is generated if the status of a hardware element changes from on-line to off-line or if the status changes from off-line to on-line.

b) Remote hardware sense can be executed in the associated LCP.

The following hardware elements are subject to remote monitoring:

- RC3502C Sync.ports/multiplexers
- RC3502C Async.ports/multiplexers
- RC3502C Channel Link (FPA)
- RC3502C Processing Unit.
- RC8000 Channel Link (FPA).

6.3.4 Remote Test

The remote tests are LCP facilities which can be executed parallel to the normal network operations.

Two types of remote tests are specified:

a) Artificial traffic tests which can be executed in addition to the normal dataflow in the network.

b) Reliability tests which are carried out on selected hardware parts of the network. These parts must be taken out of normal operation prior to the test execution. The actual test is performed outside the NC system.

The results of a remote test are returned to the Control Centre which initiated the test. The following remote test facilities are incorporated in the RC3502C terminal concentrator:

- Low level reliability tests (network level 2) The execution of these tests require that the components in question are taken out of normal oepration.

  The following reliability programs are defined:

1. RC3502C HDLC reliability test.
   A loop between two HDLC lines on the same RC3502C must be established.

2. RC8000 FPA reliability test.
   A loop facility is activated in the RC3502C, enabling an RC8000 reliability test to exercise the channel connection.

- Network SC artificial traffic facilities (network level 5). For this purpose a SC traffic generator is defined being able to generate/repeat test patterns or to receive artificial traffic by an echo or drop facility. The module connects to the SC as a normal network user.

- Network VT server traffic generator (network level 6/7).This facility is incorporated in the X.28-SMT module and may be activated at specified ports instead of network users.

The VT traffic generator operates in one of the following modes:

a. Generates and repeats a test pattern which is sent to a SMT through the network (network test).

b. Generates and repeats a test pattern which is sent to a terminal connected to the port (access test).

c. Echoes or drops received traffic from the network (network test).

d. Echoes or drops received traffic from the port (access test).

6.3.5 Remote Software Control

Apart from the above mentioned control and monitoring facilities, which are primarily oriented towards hardware modules, a number of control functions can be executed on the software modules.

The RC3502C terminal concentrator includes:

- update of X.28-SMT terminal profiles
- update of receive-/transmitmasks in X.28-SMT module
- update of TC table in the SC module
- instructions to CNADAMM in RC3502C
- setting of TS port range
- restart of the DTE module
- allocate/deallocate HDLC lines
- activation/deactivation of test routines instructions to the artificial traffic modules
- setting of HPM port range.

The total set is outlined in Tabel 6.1, 6.2, 6.3, and 6.4.

The RC8000 Host Computer offers the following software control facilities:

- update of NCC routing table
- update of NCC address table

The total set is outlined in Table 6.3.

Apart from the above mentioned control possibilities each NC Utility can be individually controlled via supervisor transactions.

6.3.6 Remote Software Monitoring

The software monitoring is somewhat more complex than the hardware monitoring. The monitoring of a software element (module) includes 2 types of supervisor transaction exchanges:

1. The unlocking of the software monitoring is carried out by issuing a supervisor command (Set NC mask). The receiver address is set as specified in section 6.2.3.

2. The monitoring information is either a supervisor answer or a sequence of supervisor events (triggered by an internal event). The typical contents of a monitoring information are error messages, reports of statistical records, state changes or table/module information.

If the Control Centre to which a report must be sent is not accessible the 'exception' Control Centre is chosen by the NCP.

The following types of software monitoring information are defined in the RC3502C Terminal Concentrator:

- reports from HDLC, DTE, TS, SC, X.28-SMT, HPM, CNADAM, NCP, and the SCAT modules.
- statistics from HDLC, FDLC, DTE, TS, SC, X.28-SMT, HPM, and the SCAT modules.
- table information from SC, NCP.

The RC8000 modules offer the following reports/statistics

- reports/statistics from the network interface modules (NPM).

6.3.7 Reports and Statistical Records

The contents of a statistic record should at least incorporate:

- counters for accumulated number of logical and physical connections
- accumulated traffic volumes
- accumulated error rates

A report should be generated in the following situation:

- logical and physical connections/disconnections
- unsuccessful connection
- lack of resources
- error detection including hardware monitoring
- system error detection.

The exact contents and formats are described individually for each software module in ref. (10), (11), (12), (13), (14), (15), (16), (17), (18), (19) and (38).

6.3.8 Message Broadcast

The broadcast facility is located in each X.28-SMT module and can be activated/deactivated by a NC supervisor command. The data field of this command must contain the contents of the message (in textform) and the length is max 194 bytes.

Each time a user logs on or types the STATE command the broadcast message will be displayed on the terminal.

6.3.9 Alarm Handling

The NC nucleus (NCC) maintains a routing table containing information about receivers of transactions (events and answers) and other information. One field indicates that the received transaction should generate an alarm. This alarm is handled by the NCT, which print a message on the main console of the RC8000 Host Computer.

6.3.10 Control and Monitoring Functions, Overview

Based on the previous general description and the NC services outlined in chapter 5 a survey of all basic control and monitoring facilities is given in the following tables.

	CONTROL FUNCTIONS - COMMANDS -	MONITORING FUNCTIONS - SENSE /STATISTICS -	 - EVENTS -
RC3502C SMT/AMX	SET NC MASK SET INIT SET TERM PROFILE SET USER MASK OPEN/CLOSE AMX PORT START/STOP ARTIFICIAL TRAFFIC BROADCAST	SENSE SMT SENSE PORT GET SMT STATISTICS PORT STATISTICS	NET CONNECTION FAILED TERMINAL CONNECTED TERMINAL DISCONNECTED NET CONNECTION NET DISCONNECTION EVENTS LOST
RC3502C HI	SET NC MASK MAX NUMBER OF PORTS	SENSE HPLC GET HPM STATISTICS HPLC STATISTICS	LACK OF RESOURCES HPLC CONNECTED HPLC DISCONNECTED ERROR DETECTED EVENTS LOST
RC3502C CNADAM	SET NC MASK CREATE/REMOVE CHILD LINK/UNLINK PROCESS REINIT TC	SENSE CPU LOAD	AUTO CREATE/REMOVE AUTO LINK/UNLINK CONSOLE MESSAGE EVENTS LOST
RC3502C HDLC	SET NC MASK SET LINE TABLE SET HDLC PARAM CONNECT/DISCONNECT LINE SET MODEM SIGNAL START/STOP TESTOUTPUT PRINT TESTOUTPUT	SENSE HDLC SENSE LINE SENSE LINE SPEED GET LAST FRMR GET HDLC STATISTICS	LINE CONNECTED LINE DISCONNECTED CONNECTION FAILED TESTOUTPUT MODE CHANGED EVENTS LOST
RC3502C DTE	SET NC MASK RESTART DTE	SENSE DTE SENSE CHANNEL GET DTE ID GET DTE STATISTICS GET CHANNEL STATISTICS GET BUF STATISTICS	LACK OF RESOURCES VC ESTABLISHED/CLEARED UNSUCCESSFUL VC REQUEST DIAGNOSTIC RECEIVED VIRTUAL CALL RESET DTE RESTARTED EVENTS LOST

Table 6.1. LCP operations.

	CONTROL FUNCTIONS - COMMANDS -	MONITORING FUNCTIONS - SENSE /STATISTICS -	- EVENTS -
RC3502C TS	SET NC MASK MAX NUMBER OF PORTS SET STREAM RANGE	PORT STATE GET TS STATISTICS LIAISON STATISTICS PORT STATISTICS	LACK OF RESOURCES LIAISON INITIATED/- TERMINATED PORT OPENED/CLOSED UNSUCCESSFUL LIAISON INIT UNSUCCESSFUL LG DELIVERY EVENTS LOST
RC3502C SC	SET NC MASK SET TC TABLE ENTRY CLOSE PORT	SENSE SC SENSE PORT GET SC ID GET OPEN PORTS GET NET CONFIGURATION ID GET PORTTABLE ENTRY ABS/- SYMB GET SESSION STATISTICS GET PORT STATISTICS	SESSION CONNECTED SESISON DISCONNECTED PORT OPENED PORT CLOSED UNSUCCESSFUL CONNECTION EVENTS LOST
RC8000 NPM	SET NC MASK	SENSE HPLC SENSE FDLC HPLC STATISTICS FDLC STATISTICS	LACK OF RESOURCES HPLC INITIATED/TERMINATED FDLC ONLINE/OFFLINE ERROR DETECTION EVENTS LOST

Table 6.2. LCP operations.

	CONTROL FUNCTIONS - COMMANDS -	MONITORING FUNCTIONS - SENSE /STATISTICS -	MONITORING FUNCTIONS - EVENTS -
RC3502C NCP	SET NC MASK SET INIT EVENT ADDRESS SET NETWORK TIME SET EVENT ANSWER ADDRESS SET EXCEPTION RETURN ADDRESS	GET CONNECTED LCP GET EVENT ANSWER ADDRESS GET EXCEPTION RETURN ADDRESS	LACK OF RESOURCES LCP CONNECTED LCP CONNECTED LCP DISCONNECTED EVENTS LOST
RC8000 NCC	SET NCC TABLE SET TS ADDRESS CLOSE NCC DELETE TRIM		
RC8000 NCT			OPERATOR CONNECTED OPERATOR DISCONNECTED NCT STARTED NCT CLOSED
RC8000 EVENTCOL	CLOSE EVENTCOL		
RC8000 OVERVIEW	SET OVERVIEW INTERVAL STOP OVERVIEW		

Table 6.3. LCP operations.

	CONTROL FUNCTIONS - COMMANDS -	MONITORING FUNCTIONS - SENSE /STATISTICS -	MONITORING FUNCTIONS - EVENTS -
RC3502C SCAT	CONNECTION REQUEST	GET TG RECEIVAL STATISTICS	REMOTE CONNECTION
	CONNECTION ACCEPT	GET TG GENERATION STATISTICS	
			REMOTE TERMINATION REQUEST
	DISCONNECTION REQUEST		REMOTE ABORT
	DISCONNECTION ACCEPT	GET LT RECEIVAL STATISTICS	
	DISCONNECTION REJECT	GET LT GENERATION STATISTICS	TG/LT/LG DUPLICATION
			TG/LT/LG LOST
	ABORT		
	OPEN/CLOSE PORT	GET LG RECEIVAL STATISTICS	TG/LT OUT OF SEQUENCE
		GET LG GENERATION STATISTICS	TG/LT/LG RECEIVAL TERMINATION
	TG ECHO TRAFFIC		TG/LT/LG GENERATION TERMINATION
	TG RECEIVAL		
	TG GENERATION		TG/LT/LG WINDOW ERROR
			TG/LT/LG RECEIVAL ERROR
	LT ECHO TRAFFIC		TG/LT/LG GENERATION ERROR
	LT RECEIVAL		
	LT GENERATION		
			EVENT CONTROL RESULT
	LG ECHO TRAFFIC		LT/LG PATTERN ERROR
	LG RECEIVAL		
	LG GENERATION		LG SENDER ID
			LG ILLEGAL SENDER
RC3502C SCECHO	SET NC MASK		REMOTE CONNECTION
	OPEN PORT		REMOTE TERMINATION
	CLOSE PORT		REMOTE ABORT
			TG/LT/LG IN RESULT
			TG/LT/LG OUT RESULT

Table 6.4. LCP operations.

6.4 NC Modules

This section gives an overview on each individual NC module put into operation at the initial state. NCC, NCP, NCT, and NC Utilities are described.

6.4.1 Network Control Centre

As previously indicated, the Network Control Center is composed of a number of more or less independent function modules. These modules fall into three categories, the nuceleus (NCC), NCT and NC Utility modules.

- the nucleus module (NCC):
  Is defined as being the basic set of tools which is required by all utility modules and which is necessary for a 'basic' version of the Network Control Centre.

- NCT module:
  Is the human interface to the Network Control System and is necessary for a 'basic' version of the control system. The display unit (terminal) may be physical connected either directly to the RC8000 Host Computer or as a normal CENTERNET terminal. The terminal is not absolutely necessary for the function of the NCT.

- NC Utility modules:
  Upon request a utility module is loaded and/or activated and remains active as long as the module itself demands. Several utility modules may be active simultaneously. Each module occupies a specific subaddress (LCP no) in the supervisor address format (see section 6.2.2).

According to the above mentioned rules, an internal structure of a Control Centre is outlined in fig. 6.4.

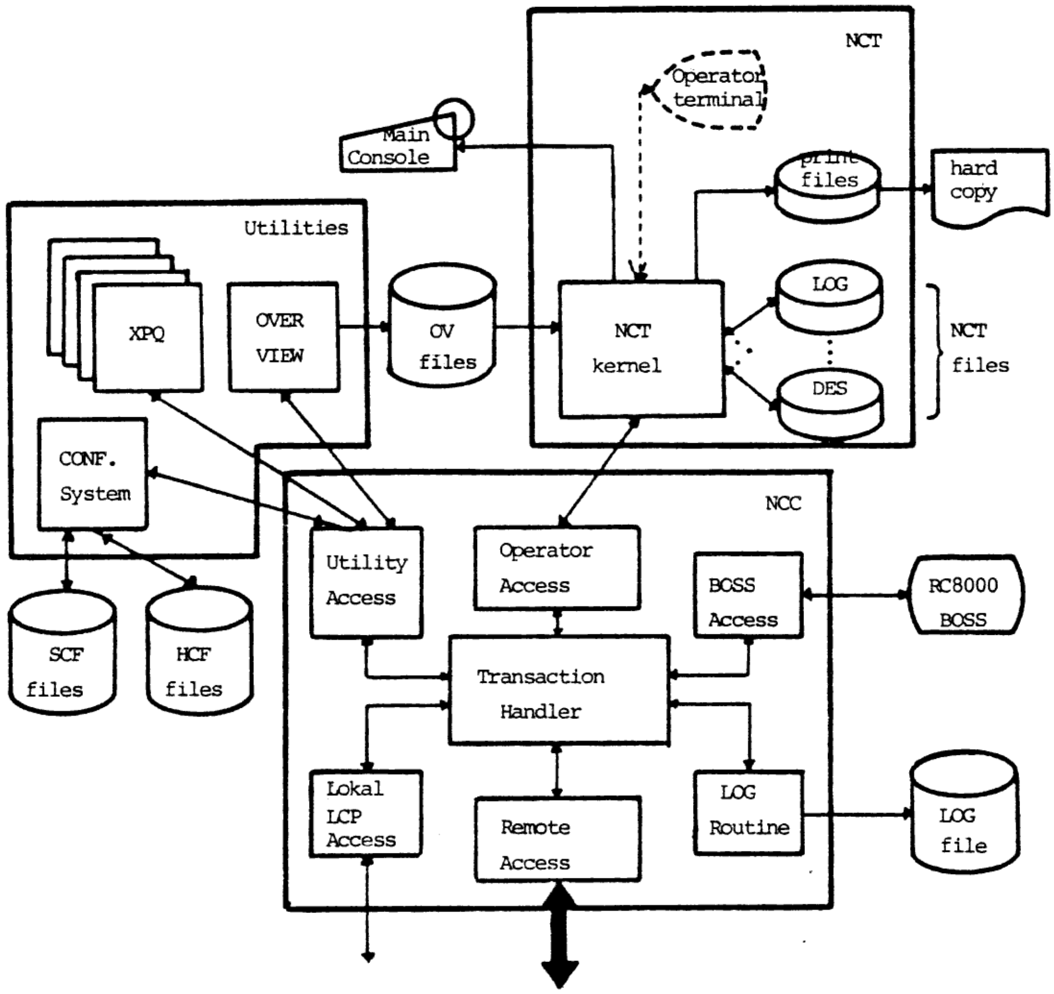


Figure 6.4 Internal Structure of a Control Centre.

The following modules are considered as a part of the NCC:

- BOSS Access.
  Interface, which enables a supervisor transaction to be transformed into an internal command for the BOSS operating system. Is used, when an NC utility must be loaded or removed.

- Utility/Operator Access.
  Presents a standard interface to the NCC. Any program may use this interface to send and/or receive supervisor transactions.

  All utilities must in the connection phase identify themselves as an NC utility (subaddress). The interface logic does not transform the contents of the transactions.

- LOG.
  Every command/answer/event transaction is logged on an RC8000 disc file.

- Transaction handler.
  Incoming transactions are inspected for
  - validity
  - alarm
  - addressing

  and are forwarded to the addressed ultimate receiver.

  For each valid subaddress belonging to own Control Centre a description record exists. It defines the actual destination of the transaction, and it also specifies special actions e.g. if the transaction should generate an alarm or not, if an utility program should be loaded or not.

  The actual destination may be one of the following modules:

  - BOSS Access Module
  - Operator Access Module
  - Utility Program Access
  - Remote Access Module
  - LCP Access Module

  If the receiver is a utility program which is not loaded and connected to the NCC, a directive is forwarded to the BOSS access module, which initiates a start of the utility with the transaction contents as parameters.

- Remote Access Module.
  Accesses the Session Control (via the NPM) and utilizes the lettergram service with acknowlegdment. The content of the SC address field in the transaction is used as parameter information to the SC. The remaining content is not used. All incoming transactions are forwarded to the transaction handler.

- LCP Access Module.
  Accesses the NI (NPM) for control and state/statistics retrieval.

The control, whether a supervisor message must be answered or not, is purely an agreement between the originator of the transaction (in this case a utility program or the NCT) and the receiver of the transaction (in most cases an LCP module).

Apart from the logging, the NCC keeps no track of outstanding answers or event.

6.4.2 Network Control Probes (NCP)

The network control probe - NCP - maintains similar tasks as the NCC, it communicates with a number of LCP's via standard access procedures and forwards the information to a Control Centre via the supervisor protocol. Likewise it receives commands from a Control Centre.

The RC3502C Terminal Concentrator NCP follows a structure as indicated in fig. 6.5.

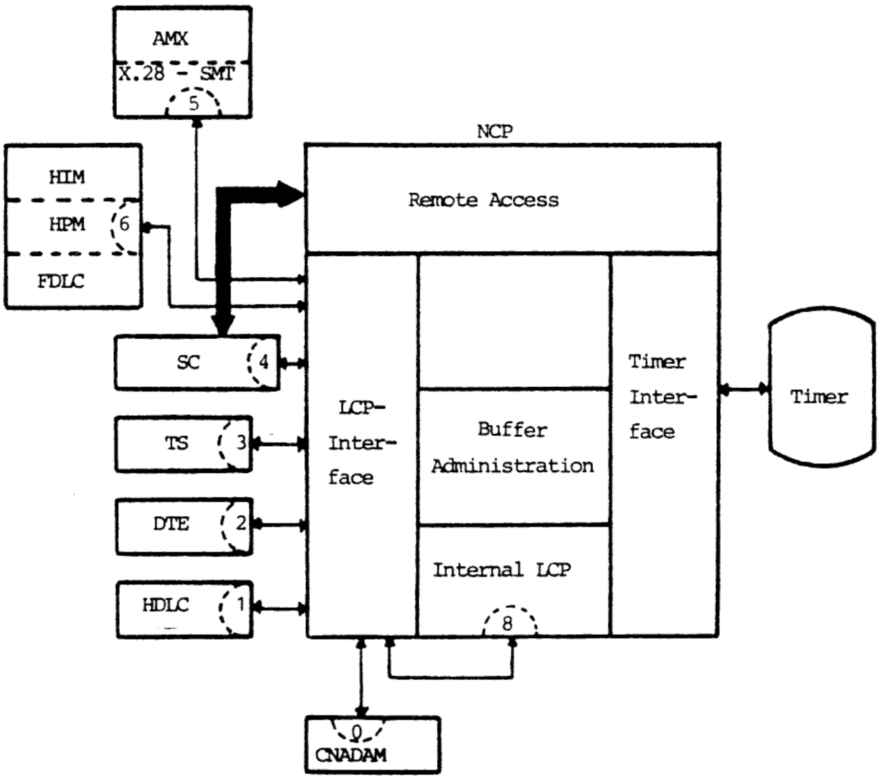


Figure 6.5. Logical structure and environment of RC3502C NCP.

Each LCP intercommunicates with the NCP utilizing the primitives:

- CONNECT LCP
- DISCONNECT LCP
- WAIT MESSAGE
- REQUEST EVENT BUFFER
- WAIT EVENT BUFFER

Each supervisor transaction which passes the NCP and which originate from one of the LCP's is time-stamped before submission to the Control Centre.

If the NC transaction specifies an automatic reporting to the Control Centre (e.g. statistics each 5 minutes) the NCP is responsible for repeating the operation.

6.4.3 NCT Handler (NCT)

This module forms the network operator access to the NC system.

The NCT handler is able to function correctly, i.e. receive and log all incoming supervisor transactions, without requiring a network operator.

The facilities being offered by the NCT handler reflect the functions available at each NCP/LCP location in the network, i.e. the NCT handler is able to create and send any supervisor transaction and is likewise able to receive any supervisor transaction.

Furthermore the NCT maintains the following files:

- COMFILE: Supervisor transactions with predefined contents. The transactions may be collected in groups and the NCT can, in one command, be requested to submit a group.
- DESFILE: field descriptor file for every defined supervisor transaction.
- LCPFILE: is organized a three logical files. An addressfile containing the relation between LCP names and absolute LCP addresses. A namefile containing the relation between transaction names and transaction codes. And a statfile containing a status text per status bit per LCP.
- NCTLOGFILE: used to log all submitted and received transactions.
- OPEFILE: user catalog containing usernames with passwords and each users access rights.

The generation of supervisor transactions can be done in three ways:

a) All information required for a transaction is keyed in/modified by the network operator. The generation is controlled by the NCT Handler. After the creation of the transaction it can be either sent directly to the NCP/LCP location indicated, or it may be stored in the NCT COMFILE.

b) The transaction is retrieved from the NCT COMFILE. The operator may specify two actions:

   1. The transaction is sent immediately to the NCC for processing.

   2. The transaction contents are displayed on the network control terminal, and further control is carried out by the NC operator (as specified in a).

c) The NC operator specifies a transaction command group to be executed. This implies that a sequence of supervisor transactions will be retrieved from the NCT COMFILE and sent to the NCC for execution without NC operator interventions.

   During initialization of the NCC system the NCT Handler automatically executes a command group.

The receival of supervisor transactions will provoke the following actions to be performed:

a) if the alarm indicator is set, an alarm is printed on the main console.

b) log of the transaction in the LOGFILE.

c) display of the transaction, if the display unit is connected and according to the actual filter.

The display of supervisor transactions invokes an interpretation according to the following rules:

a) With the values of LCP address/OPER/TYPE as index a message text is read from a text file.

b) If the transaction user field contains data, these will be supplied to the operator output text according to specific rules.

The NCT operator output is always written (in text form) on the operator log, whether a display unit (terminal) is connected or not. All received supervisor transactions are time stamped. The NCT operator may specify a filter, which reduces the information displayed:

NCT filter:

1. Alarm messages only.
2. All messages.
3. Nothing (NCT disconnect).
4. Only communication with a chosen LCP address.

5. A specified transaction type / transaction.

In case the NCT has not been connected or if a review on the NCT messages is required, the NCT operator may request a specified lookup in the operator log LOGFILE.

The NCT handler contains a log-on and password mechanism which enables the enforcement of competence areas of each of the NCT operators and may also prevent unauthorized access.

The NCT is also used in connection with the Overview System. The logical overview files generated by the NC Utility Overview can be accessed from the NCT as specified in section 6.4.6.

6.4.4 Remote Load Utility (RLU)

To appear January 1983.

6.4.5 Event Collector (EC)

The Event Collector is an NC Utility which is able to receive all hardware/software monitoring information reported by the LCP's. The start of the reporting mechanism at a given LCP is initiated by the NCT operator or by default, and the operator communication, which is required by the Event Collector itself, is also performed via the NCT.

The Event Collector maintains and updates the event file:
- EVENTFIL

The above mentioned file is not identical to the NCC log file, but can be considered as filtered/processed information retrieved from the reports produced by the LCP's.

The reports contain events information and originate from the following RC3502C terminal concentrator locations:

- HDLC driver
- FDLC driver
- Async driver (IMS)
- Host Interface
- DTE
- TS
- SC
- X.28-SMT
- CNADAM
- NCP
- SCAT modules

and the RC8000 module:

- NPM

The reports received from the network components are normally all processed.

6.4.6 Overview System (OVIEW)

The Overview System is an NC Utility, which is able to at regular time intervals to request retrieval of information and to receive reports from the LCP's.

Based on this information a number of overview files are generated. The intention of the overview files is to present extracts and processed information appropriate for presentation to the network management.

The updating of these files must be performed at regular time intervals (e.g. every 3 minutes). The logical files is saved in one RC8000 disc file.

The display of the logical files is handled by the NCT utilizing the overview supercommand. For all files it is possible to request a lookup, which either displays the last records or displays all records in a specified time interval.

Alternatively the line printer can be choosen as output medium.

The overview files include:

- Terminal Concentrator overview.
  Number of active TC's, current number of users, current number of X.25 active VC's, number of fatal errors, CPU utilization

- Transport Station overview.
  Current number of X.25 VC's and users active. Current number of liaisons. Number of fatal error. Average throughput.

- Host Interface overview.
  Current number of users. Number of fatal errors. Average throughput.

- Session Control overview.
  Current contents of SC port table.
  Accumulated number of connections and disconnections.
  Number of abnormal disconnections.

- Scroll Mode Terminal overview.
  Current number of opened ports.
  Current number of terminals.

Accumulated number of successful/unsuccessful connections and disconnections.
Average throughput.

Figure 6.6 illustrates the information flow to and from the Overview system.

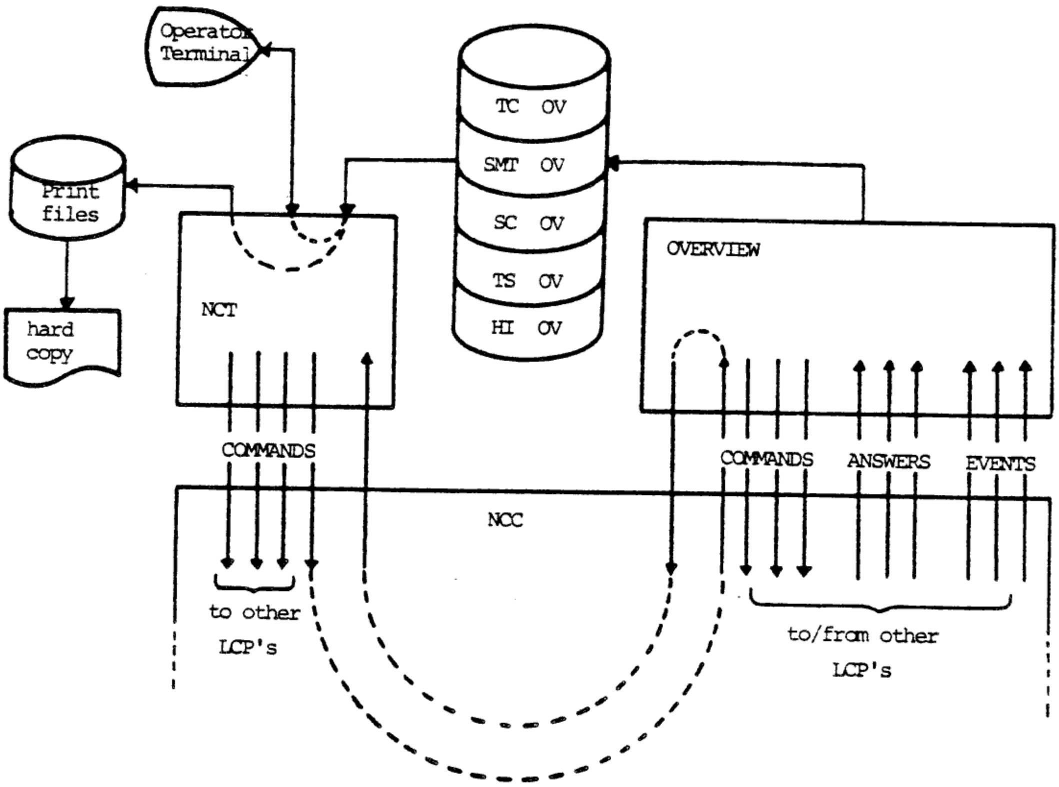


Figure 6.6. Transaction flows to/from the Overview system.

7. CENTERNET TERMINAL SUPPORT

7.1 Basic Structure

The individual terminal modules have access to the Session Control, and are located together with SC in the RC3502C.

Each module makes use of the Session Control's network connection mechanism and data transfer service, even if the connection is to an attached host, so the connection is "local". I.e. the modules do not directly use the Host Interface.

For each type of attached devices (different terminal types, remote printing station etc.) or functions (file transport, remote job entry) a dedicated module exists as indicated on figure 7.1. These modules may share the necessary drivers.

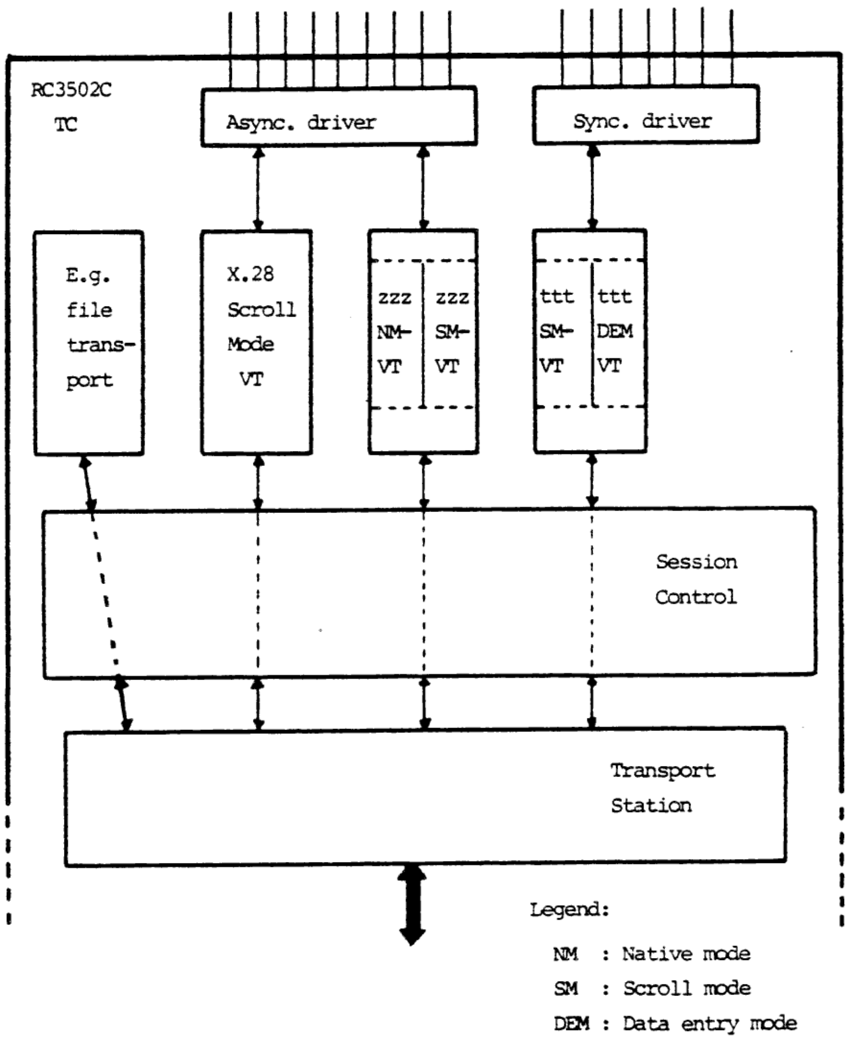


Figure 7.1 Logical structure of CENTERNET Terminal Concentrator

7.2 X.28 - Scroll Mode Terminal

The X.28 - Scroll Mode Terminal supports connection of physical terminals, being start-stop (or asynchronous) terminals, to CENTERNET. In ref. (3) the control characters the terminal shall be able to support are described. The terminals are connected by an X.28-like interface and the communication through the network is based upon the Scroll Mode part of the CENTERNET Virtual Terminal Protocol, as specified in section 5.6 and ref. (2).

Each physical terminal will occupy one port and this port is used to identify the terminal throughout the total network. The module uses the Session Control (section 5.5) for establishment of a session through the network and for exchange of data.

The next five sections give a general overview of the functions, the access and service interfaces, network control service, and a logical structure of the module. A detailed description and implementation specification are given in ref. (3).

7.2.1 Functions

As explained in section 5.6.1 a Virtual Terminal consists of a set of components. The following components constitute the X.28 - Scroll Mode Virtual Terminal.

presentation unit	:	hard copy unit or screen operated in scroll mode,
data structure	:	an infinite number of lines with character positions (scroll mode as defined in VTP, ref. (2))
keyboard	:	as defined in VTP ref. (2) with added control functions for local communication
alarm device	:	bell
connection control	:	implemented by the Session Control
auxiliary devices	:	none

A detailed description can be found in ref. (3), but some remarks will be made here.

The keyboard is intended for entering of data and commands, either local commands or network commands. The commands take form of texts or function keys. A command belongs to one of these four groups:

- data path control
- parameter setting
- virtual terminal control
- information request

The virtual terminal keys are implemented either as a command or as a function key. The commands and keys are described in section 7.2.3.

No auxiliary devices are supported by the X.28-Scroll Mode Terminal. I.e. it is not possible to assign other devices than the presentation unit in the VTP, but an other input device (e.g. a paper tape reader) can be used if the switching is done manually. It is possible because the X.28-like interface supports flow control on input using the ASCII characters DC1 and DC3.

The above described X.28-Scroll Mode Terminal supports the following facilities:

- automatic speed detection
- establishment and removal of network connection (sessions)
- selection of a parameter profile within a limited set
- setting of individual parameters range
- changing of parameter values in the VTP during a session
- setting of a initial terminal profile by network control interaction
- and naturally data and attention exchange between a terminal and a server. A server is either an other terminal or an application.

7.2.1.1 Session establishment and removal

A terminal is logged on raising the carrier and issuing a number of carriage returns (CR=0/13) allowing for speed recognition in case this is indicated by the parameter X.28-SPEED, ref. (3). When the speed of the communication line is determined, the system will respond with a local text message.

```
CENTERNET
PORT <own port id>
<broadcast if any>
<date>   <time>
```

The terminal is now in a local connection state.

An initial terminal profile is set as part of this local connection. The initial terminal profile may be changed by the network control.

Whether or not a text shall be output to the terminal is controlled by the parameter X.28-SERVICE SIGNAL SUPPRESSION.

Normal session establishment

In the local connection state, the operator can set parameters ranges and read parameters values (section 7.2.1.4 and 7.2.3) Furthermore a Selection Command may be issued to establish a session between the terminal and a server. The format of the Selection Command is given in section 7.2.1.2.

When a session has been established, the following text is output to the terminal

PORT <own SC port no> AT <TC-adr> CONNECTED TO
PORT <SC port no> AT <destination>
<date> <time>

The terminal is now in the data transfer state. All inputs are considered as data, commands has to be preceded by an escape character (DLE=1/0).

Session removal

To terminate a session the 'clear request' command (CLR) must be issued. When the session is removed a text is output to the terminal

TERMINATION ACCEPTED
<date> <time>

or

ABNORMAL TERMINATION <cause>
<date> <time>

The latter is used if an error has occurred during termination.

Now, the terminal is again in a local connection state, and it is possible for the operator to initiate a new session.

The local connection is removed when the carrier is turned off.

Auto establishment of a session

Being in the local connection state a terminal may receive a request for connection, either from an other terminal or from a host.

The action taken in the two cases is different and the terminal will end in two different states.

Terminal request

If the request is accepted, it is answered and the text

```
AUTO CONNECTED
PORT <own SC port> AT <TC adr>
CONNECTED TO
PORT  <SC port> AT <destination>
<date>  <time>
READY
```

is output to the terminal. The auto connected terminal will act as a simple server, and is at start in send mode (the other in receive mode), so the operator of the auto connected terminal has the possibility to refuse the connection manually by issuing a 'clear request' command.

Application request

If the request is accepted it is answered and the text

```
AUTO CONNECTED
PORT <own SC port> AT  <TC adr>
CONNECTED TO
PORT  <SC port>  AT <destination>
<date>  <time>
```

is output to the terminal. The auto connected terminal is in receive mode.

7.2.1.2 Format of Selection Command

The Selection Command consists of two parts, of which one always shall be present. It is possible to use abbreviated addresses, indicated with a point. The two parts are separated by a comma.

```
                                                             1
                          ┌ <destination>                    ┐
Receiver ident        ::= { <destination>, <network application }
                          └  , <network application>         ┘
                                                             1

                                         1
                          ┌ .            ┐   (1.1)
<destination>         ::= { . <name>     }   (1.2)
                          └ d-address    ┘   (1.3)
                                         1
```

```
                                    1
                          ┌.         ┐  (2.1)
network application ::=  { . <name>   }  (2.2)
                          └n-address ┘  (2.3)
                                    1
```

(1.1) default destination
(1.2) abbreviation for a destination known by keyword. <name> is at most 10 graphical characters of the IA5 alphabet except ',' (2/12) and 'SP' (0/12) (ref. (3)).
(1.3) full destination identification.1-11 digits
(2.1) default server
(2.2) abbreviation for an application known by keyword At most 10 characters as (1.2)
(2.3) full application identification. 1-5 digits.

The following three Selection Commands will give the same result (are equal), default destination and default server '.' , '.,.' , ',.'.

7.2.1.3 Data exchange

Input from terminal

In the data transfer state inputs belonging to the character-set (VTP parameter character-set) are considered as data except the "data forwarding" signal. A number of other control characters may be excepted as they can have special functions depending on the parameter values.

The escape character indicates, that the succeeding text is a command (section 7.2.3).

The physical terminal sends the two characters, X-ON, X-OFF, to the TC to control the output to the physical terminal (output flow control, ref. (3)).

All data input characters are entered into the data structure, which is sent in blocks to the server every time a 'data forwarding' signal is received from the terminal. Four other events may cause transmission of the data structure. These are described in ref. (3), section 2.3.3.

The operator may select the character he intends to use as 'data forwarding' signal. Two possibilities exist, either CR (0/ 13), or any character belonging to column 0 or 1 or DEL (7/15). The parameter X.28-DATA-FORWARDING (section 7.2.1.4) indicates the chosen representation.

Ouput to terminal

All characters received from the server in text blocks are output to the terminal without any changes. The control primitives CR and NL are changed to the characters CR and CR,LF, respectively. Every time CR or CR,LF is output to the terminal, a number of padding characters are output too. The parameter X.28-PADDING indicates the number.

The two characters DC1 and DC3 are output to the terminal for input flow control purpose (ref. (3)).

Furthermore a number of service signals are output to the terminal, either as a response to a network event or as a response to an operator command. The output of the service signals is suppressed if the parameter X.28 - SERVICE SIGNAL SUPPRESSION indicates so. The individual service signals are described in ref. (3), section 2.6

Interrupts

The two types of interrupts (CLEAR, ATT) are both supported by the X.28-like interface.

- ATT(1) is ignored when received and sent as part of the break-action or as a result of the INT command.
- ATT(2) is output to the terminal as a 'BELL' (0/7) when received and sent as part of the break-action or as a result of the INTD command.
- ATT(15) is ignored when received and sent as a result of the XPARAM command.
- CLEAR will cause the text 'REMOTE RESET' to be out put to the terminal when received and is sent as part of the break-action or as a result of the RESET command.

The break-action is one action among a set of defined actions, performed when a break-signal is received. The operator is able to select which action to be performed. Five possibilities exist:

(1) take no action
(2) send an ATT(1) to the server
(3) reset the dialogue, i.e. enter the clear phase by sending a CLEAR to the server and clear the virtual data structure
(4) send an ATT(2) to the server

(5) leave the data transfer state and enter a state in which commands may be issued (perform the escape function).

Different kinds of dialogue

If the communication is between two terminals only minor changes are needed. The 'server' terminal will have to begin and end every text message output to the terminal with LF (0/10) and place a NEW LINE in front of and after every text message sent to the other terminal.

7.2.1.4 Terminal parameters/profile

Each individual terminal is characterized by a profile. The profile is defined by a value for each of the parameters defined in the VTP and the X.28-like interface.

The protocol and the interface provide means for both the operator and the server to set and change the parameters according to a specified set of rules as defined in ref. (3) in details. The server can only change the VTP-parameters, whereas the operator can change both the VTP-parameters and the X.28 parameters according to the following class definition.

The parameters can be divided into three classes:

(1) parameters that may be changed before and during the session without involving the server. These parameters are only significant for the X.28-like interface, and is only known by the operator, who can set the parameter to a specific value.

(2) parameters that may be changed before and during the session. During the session the server is involved in the changing. These parameters are significant for both the X.28-like interface and the VTP. The operator can specify a range for parameters and the server chooses the value inside this range.

(3) parameters that can never be changed by the operator. Parameters only significant for the VTP and a few X.28-like interface control parameters, such as speed etc.

In the VTP the following parameters are defined (section 5.6.1.3):

- terminal mode	class 2
- addressing capability	class 3
- line - discrete	class 2

- character - set class 2
- attribute capability class 3

and the following is added:

- X.28-PAD RECALL	class 1
- X.28-ECHO	class 1
- X.28-DATA FORWARDING	class 1
- X.28-IDLE TIMER PERIOD	class 1
- X.28-INPUT FLOW CONTROL	class 1
- X.28-SERVICE SIGNAL SUPPRESSION	class 1
- X.28-BREAK - ACTION	class 1
- X.28-DISCARD OUTPUT	class 3
- X.28-PADDING	class 1
- X.28-SPEED	class 3
- X.28-OUTPUT FLOW CONTROL	class 1
- LINE WIDTH	class 1
- LINE FOLDING	class 1

The parameters and their possible ranges and values are specified in details in ref. (3).

7.2.2 Access to Session Control

The X.28-Scroll Mode Terminal (X.28-SMT) uses the service offered by the Session Control to establish and remove sessions and to exchange data to and from the server.The access equals the one described in section 5.6.2, so only a short overview will be given in this section.

Connection/disconnection to network (local)

OPEN,

parameters :
term-id : symbolic port id (includes physical port no)
func : 0 } { details please
port : not used } { see section 5.5.4
receivemask : please refer to section 5.5 and 7.2.4.1

Return : result, actual SC port no.

CLOSE

parameters :
port : own port
func : 3

Return : result

Establishment/removal of a session

CONNECT-REQUEST

parameters	:
port	: own port
transmitmask	: please refer to section 5.5 and 7.2.4.1
class	: terminal class
receiver	: symbolic or absolute address of receiver
user data	: indicates call from a VT
Return	: result, receiver absolute address, received user data

DISCONNECT- REQUEST

parameters	:
port	: own port
cause	: VTP disconnect reason transferred to the other part
Return	: result, responded cause

EVENT-CONTROL

parameters	:
port	: own port
Return	: result, session establishment/removal indication, user data. If session establishment, user data must indicate whether a VT or a VT server is calling.

CONNECT-ACCEPT

parameters	:
port	: own port
user data	: VT if the caller is a VT server, otherwise VT server.
Return	: result.

DISCONNECT-ACCEPT

parameters	:
port	: own port
cause	: VTP disconnect reason.
Return	: result

ABORT

parameters	:
port	: own port

cause : VTP aborting reason

Return : result.

Data transfer

RECV-LI

parameters :
port : own port

Return : result, data

SEND-LI

parameters :
port : own port
data : VTP data

Return : result

These two are used to transfer VTP connection commands, VTP text, control, and param blocks. The size of the buffers depends upon the size of the data structure.

RECV-TG

parameters :
port : own port

Return : result, 1 octet data

SEND-TG

parameters :
port : own port
interrupt
data : 1 octet data

Return : result

These two are used to transfer interrupts (CLEAR, ATT(n)).

7.2.3 X.28 - Scroll Mode Terminal Service

The X.28-Scroll Mode Terminal Service interface consists of a set of commands and function keys offered to the operator of the terminal. An overview of the commands and the keys are given in this section and a detailed description may be found in ref. (3).
The command set and the keys cover the needs for establishment/ termination and control of a dialogue, for setting the terminal profile parameters range/value, for getting status information and for operating on the virtual data structure.

A total of thirteen commands is defined and these commands can be divided into groups using two different classification criteria:

(1) whether the command requests a response or not
(2) the type of function the command performs:

- dialogue control (type 1)
- parameter setting (type 2)
- virtual terminal control (type 3)
- information request (type 4)

Two types of function keys are defined, a break-action and a data forwarding action.

The different commands and function keys are acknowlegded by a service signal, ref. (3).

7.2.3.1 Commands of Type 1

This group consists of commands used to control the dialogue, including the commands used to initiate and terminate the session. A total of six commands are defined:

Selection Command

to initiate a session. For further details, please refer to section 7.2.1.2.

CLR

to terminate the session.

INT

to transmit an interrupt (ATT(1)) to the server. This interrupt will not interfere with the normal data flow.

INTD

to transmit an interrupt (ATT(2)) to the server. This interrupt will not interfere with the normal data flow.

PLEASE

to request the server to give over the turn.

RESET

to clear the data structure and interrupt the dialogue. This interrupt is a VTP-CLEAR function and works as described in ref. (2).

7.2.3.2 Commands of Type 2

In this group all commands used to set and change parameters are gathered. The different parameters are described in section 7.2.1.4.

In the two commands, SET and SET? a list of parameter references and ranges/values are needed. The formal definition of list is:

$$\langle \text{list} \rangle \ ::= \left\{ \langle \text{param spec} \rangle \right\} \left\{ , \ \langle \text{param spec} \rangle \right\}_0^*$$

$$\langle \text{param spec} \rangle \ ::= \langle \text{reference} \rangle : \left\{ \langle \text{value} \rangle \mid \langle \text{range} \rangle \right\}$$

<value> is a decimal representation of the value
<range> is the decimal representation of a byte, where a bit set to 1 indicates that the corresponding value is included in the range.

PROF <profile id>

to select a profile within a limited set. The <profile id> is a number identifying the selected profile.

SET <list>

to set the range/value of the specified parameters. <list> is defined above. Whether it is a value or range depend upon the type of parameter (please refer to section 7.2.1.4).

Example: SET 3 : 1, 24 : 5

sets parameter 3 to the value 1 and the range of parameter 24 is set to include two possible character sets.

SET? <list>

to set the range/value of the specified parameters and to list the value and range of the parameters. <list> is defined above. See also SET.

Example : suppose parameter 3 has the value 1 and parameter 24 the value 4 and the range 4.

SET? 3 : 2, 24 : 5

will set parameter 3 to the value 2, the range for parameter 24 to 5, and output the text

PAR 003 : 002, 024 : 004 (005)

XPARAM

causes that ATT(15) is sent to the server. As a consequence of receiving ATT(15) the server may initiate a parameter negotiation phase, but it depends on the server.

7.2.3.3 Commands of Type 3

This group defines commands used to operate directly on the virtual data structure. At present only one command is defined.

NL

to issue a virtual new line.

7.2.3.4 Commands of Type 4

This group consists of commands used to retrieve information.

PAR? <list>

to list the current value and range of the specified parameters in the profile. <list> is either empty or consists of parameter references. If empty, the value and range of all parameters are listed.

STAT

to request the status information regarding the session, and to retrieve any pending broadcast message.

7.2.3.5 Function keys

As mentioned above two types of function keys are supported, break and data forwarding. Data forwarding is explained in section 7.2.1.3.

Only one break key exist, but may activate different functions. The function actually performed is specified by the 'break-action' parameter (section 7.2.1.3).

7.2.4 Network Control Service

The X.28-Scroll Mode Terminal communicates with the NCP for control and monitoring purpose. The X.28-SMT provides a service interface for the NCP to support the needed functions.

As explained in section 5.6.4 five main primitives are defined.
They are used to support the control-,monitoring-, and broadcast functions of the X.28-SMT.

The exact format and the detailed content for each individual function are described in ref. (14). The next three sections give a general description of the functions.

7.2.4.1 Control Functions

The service interface provides means for the NCP to set the initial terminal profile, to set the initial value of SMT parameters including the default VT server, to set the receive/transmitmask, and to open/close AMX ports.

Receive/transmitmask:

SET USER MASK

sets the receive- and transmitmask as explained in section 5.5.1

Terminal profile setting

SET TERM PROF

sets the initial terminal profile. A parameter may either be set to a value or to a range . For parameters of class 2 and 3 a range is set and for class 1 a value.

SET INIT

is used to set the initial value of various parameters of the X.28-SMT module including the TC address and symbolic port name of the default VT server.

OPEN AMX PT

used to open a specific port (physical port) on the AMX.
A port can be closed again using the command

CLOSE AMX PT

The NC may activate/deactivate the monitoring functions. I.e. the NC can indicate to the module whether statistics shall be performed or not and which reports that shall be logged. A mask indicates this and the NC can set this mask utilizing the command

SET NC MASK

7.2.4.2 Monitoring Functions

The X.28 Scroll Mode Terminal supports three different monitoring functions.

- event (reports)
- sense (immediate state/status)
- statistic

REPORTS

EVENT, event-type, event-inf.

By return the fields contain the information

- event-type — identification of the causing event.
- event-inf — further information concerning the event.

The following events will trigger the report function:

(1) network connection failure
(2) terminal connected
(3) terminal disconnected
(4) network session established
(5) network session disconnected.

SENSE

The SENSE operations are used to get either an immediate state or an immediate status for the whole module or for a connection of a terminal.

The following operations are supported:

SENSE SMT
SENSE PT

STATISTIC

The STATISTIC operations are used to retrieve statistical information concerning either the whole module or a terminal connection.

The following operations are supported.

GET SMT STATIST
PT STATIST

7.2.4.3 Broadcast

Every time a terminal has established a local connection a broadcast message is included in the network identification output to the terminal. The broadcast message may also be retrieved using the STATE terminal operator command. The broadcast message is set by the NC. The total text length of a broadcast message can not exceed 194 octets.

BROADCAST, text length, broadcast text

the broadcast text replaces an earlier issued broadcast message.

7.2.4.4 Artificial Traffic Generator

In section 6.3.4, Remote Test, a VT server traffic generator is defined. The generator is incorporated in the X.28-SMT and is controlled from the NC. The NC can start and stop the generator and specify in which mode it should operate,and which traffic type it shall generate.

Two NC operations are provided for this purpose.

START GEN, mode, own-pt, own SC port id, dest-address.

starts the generator in the mode specified (mode see section 6.3.4). If local traffic is to be generated the X.28-SMT module awaits connection of a terminal at the port own-pt, whereas in the network traffic case the X.28-SMT module either connects to the local SC port, own SC port id, and establishes a session to the remote SC port, at dest-address or sets up a network port, own SC port id, and awaits at this a remote session establishment request.

STOP GEN

stops the generator. The network port - or the local port connection is removed.

7.2.5 Logical Structure

The X.28 - Scroll Mode Terminal Module is build as a number of submodules each managing a specific set of functions. Figure 7.2 shows the structure.

The 'asyn-driver' is a driver servicing an AMX asynchronous multiplexer.

The Connect-Control module is a kind of a watchdog all the time polling the physical ports not yet locally connected to check if the status of the carrier is changed. When the carrier is raised, the module will determine the actual speed of the communication line, requiring the terminal to go on sending the character CR (if indicated by the X.28-SPEED parameter, section 7.2.1.1) until recognized. When determined, a local text is written, indicating that the terminal is ready.

The X.28 Status Control and Terminal Command Handler takes action every time a Terminal Operator Command is received. The module interprets the command and either it informs the VTP handler Module or it itself performs the needed actions.

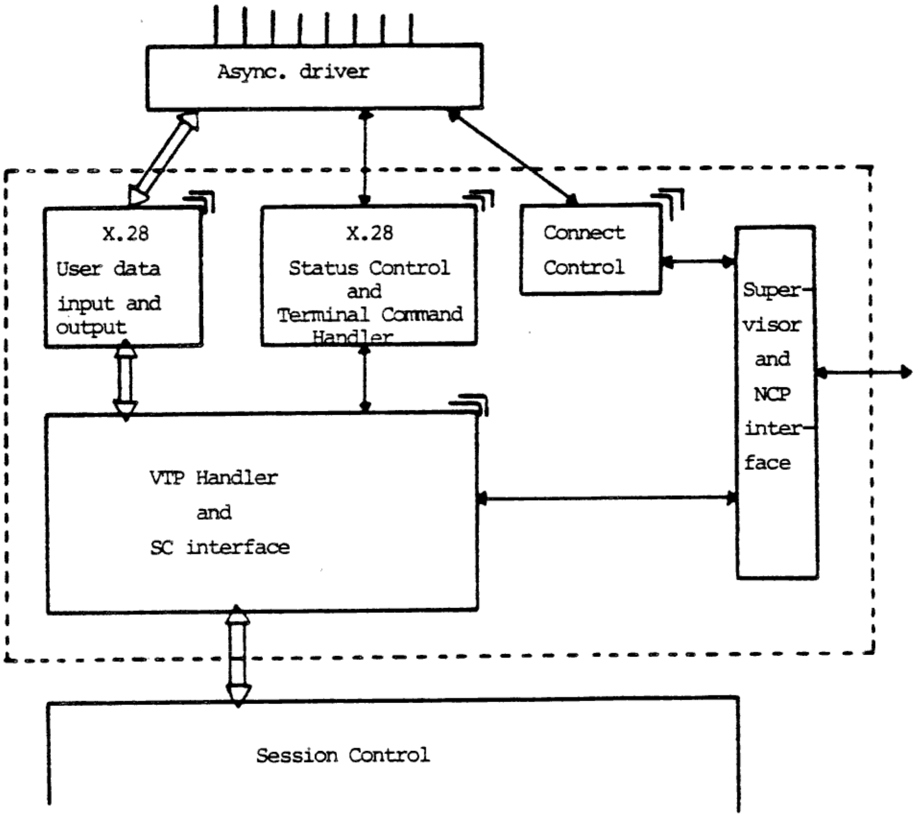


Fig. 7.2 Structure of the X.28 - Scroll Mode Terminal.

The VTP handler manages the communication to the server according to the "elements of procedures" in the VTP. Furthermore this module handles the SC access.

The X.28 User data input and output accesses the asyn-driver and routes the data between the driver and the VTP handler.

The interface to the NCP is supported by the Supervisor and NCP interface module. Furthermore this module controls the existence of all other modules forming the X.28 SMT module.

8. CENTERNET RC8000 HOST INTERFACE

8.1 Basic Structure

The CENTERNET transportation service is offered by the session layer as a user service. The Session Control Modules is located in the RC3502C TC.

Above this service level the different presentation and user levels are implemented. According to chapter 3 these levels are represented by a number of independent terminal and application interfaces, each with specific characteristics (file transport protocol, remote printing protocol, scroll-mode virtual terminal protocol, data-entry virtual terminal protocol, gateways to other networks, etc.).

All of these interface modules rely on the same SC service. Consequently a host interface offering all services of the transport system should be based on the same service level. This is illustrated on fig. 8.1.

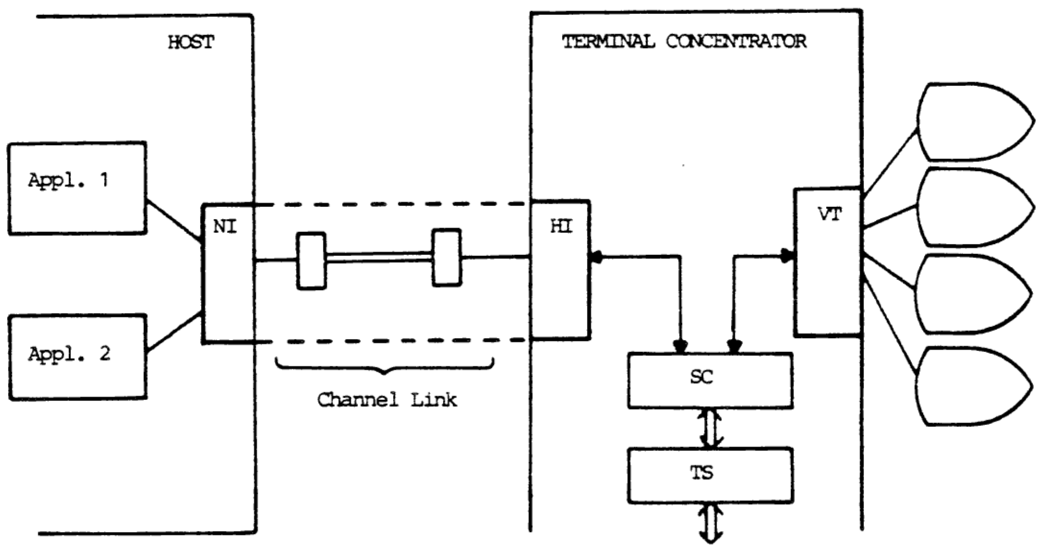


Figure 8.1 Principles of Host Interfacing.

The RC8000 HI in CENTERNET consists of the following parts:

- Channel Link (CL)
  The modules FDLC (in RC8000 and RC3502C) implements the channel protocol: Front-End Processor Adaptor Data Link Control.

- Host Port Logical Channels (HPLC)
  The modules HPM (TC) and NPM (RC8000) implement the host port protocol (HPP) which is applied for each logical channel.

- TC Host Interface Mapping (HIM)
  Interconnects the HPLC with SC ports.

The service offered to RC8000 users include:

- Access primitives from internal processes to Net Interface (NI) offering the service of SC

- Adaption to RC8000 operating systems enabling applications to communicate with X.28/SMT. This adaption is performed via the RC8000 Scroll Mode Mapping (SMM) Module, which converts the X.28-SMT terminal formats to RC822 standard formats.

8.2 Channel Link

The channel link and the associated link protocol - FDLC - constitute the low level transportation service between two computers connected via a high speed, full duplex, parallel datachannel (fig. 8.2).

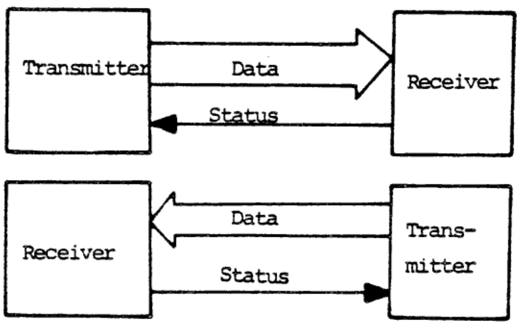


Figure 8.2. Logical structure of channel link.

Datablocks of any size can be transferred. The reverse statusindication consists of one byte. The following gives a short discription of the FDLC protocol elements (further information can be found in ref. (6).

8.2.1 Transmission Elements

The protocol is based on the exchange of Transmission Units (TU). A TU consists of a startbyte followed by a number (at least one) of data bytes.The receiver of a TU will return a statusbyte which specifies the result of the transfer and indicates that the receiver is ready for the next transfer.

The startbyte, databytes, and the statusbyte each occupie 8 bits.

8.2.2 Blocks

The protocol assumes that the users exchange information in form of blocks. A block may be transmitted as a number of TU's. Error detection and correction mechanisms refer to TU's.

It should be noted that though division of a block into a number of TU's resembles the functions of packet switching the mechanism in the FDLC protocol is more simple. Each TU (i.e. packet) must be acknowledged with a statusbyte before the next is transmitted.

8.2.3 Error Detection and Correction

Error detection is based on a parity checking mechanism performed by hardware. A parity bit is added to each transmitted byte and is checked by the receiver. Error correction is based on retransmission of the erroneous TU.

8.2.4 Transmission Unit Identification

Retransmission of a TU is detected by a TU number which is added to each TU. This number counts modulo 2, which means that only one TU may be outstanding waiting for acknowledge.

8.2.5 Flow Control

The flow control is based on the assumption that the users of the protocol should in general always assure that a buffer is ready in which to receive data. If a buffer is not always present, the time in which the FDLC handler is without a buffer should be short compared to the time which will cause retransmission of a block.

The flow control is implemented by means of the status byte in the following manner:

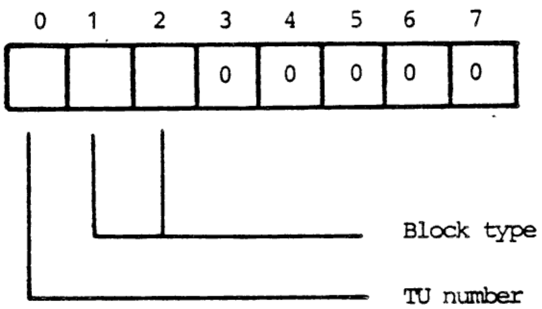
> A status byte is not returned until a buffer is ready for reception of a new (or retransmitted) TU.

This means that if the FDLC handler stores a received text-block in the last buffer from the user, it will not return the status byte until a new buffer is present.

In case this exceeds the timeout value of the transmitter, a retransmission, will take place. This will, until a buffer is ready, terminate immidiately because no input operation is present at the receiving end.

8.2.6 Formats

All Transmission Units are preceded by a startbyte with the following information:

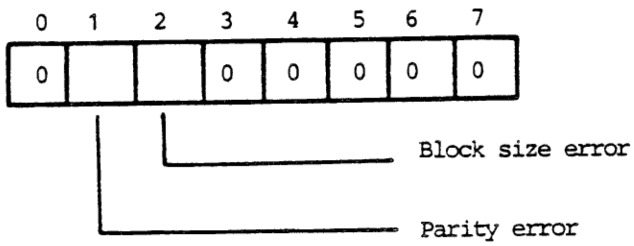


TU number is the number of this TU (modulo 2)

Blocktype
- 0: User data
- 1: not used
- 2: Master Clear information
- 3: Not used

A statusbyte is returned, when a TU has been received. It indicates whether the TU was received correctly or whether an error occured.

A statusbyte has the following format:



<u>Parity error</u> indicates that the parity checking mechanism indicated an error for at least one of the characters received.

<u>Block size error</u> indicates an overflow in the user buffer, in which the textblock should be stored.

<u>Master Clear Blocks</u>

A master clear block is used to synchronize the two ends of the transmission line. It is used after initial load and as a restart after an irrecoverable error.

Transmission of a master clear block shall cause the block-number counters to be reset and the receiver of a master clear block must return a zero statusbyte.

The master clear block carries no data.

The service offered by the FDLC modules is very similar to the service offered by the HDLC driver modules. In the RC3502C the FDLC can be replaced by a HDLC line, without any software changes.

Ref. (6) outlines further details regarding the FDLC protocol, and ref. (41) describes the RC3502C FDLC user interface.

8.3 Host Port Protocol (HPP)

The primary purpose of HPP is to enable different independent users, the Subscribers (SB), to share the same Channel Link. Thus a number of individual, bidirectional dataflows are multiplexed on one link offering both addressing, flow control and security primitives (Fig. 8.3).

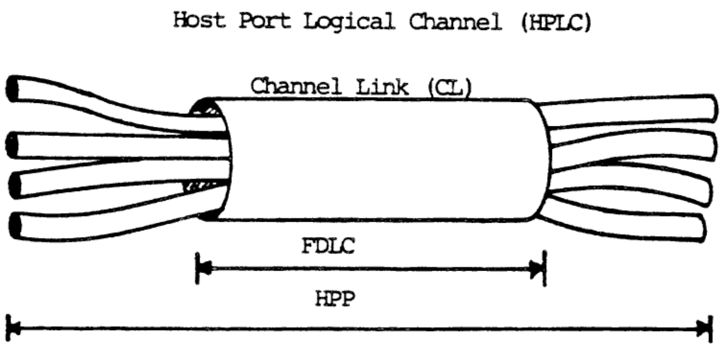


Figure 8.3. Multiplexing scheme of HPP.

As described earlier the FDLC offers a safe and sequential transportation service with retransmissions in case of transient channel errors.

Thus the HPP must offer addressing and flow control facilities in order to perform multiplexing on the FDLC.

The following gives an overview of the major HPP primitives. For a detailed description please refer to ref. (5).

8.3.1 Host Port Addressing

The HPP service modules, i.e. the RC3502C HPM and the RC8000 NPM,identifies independently each active port with an 16 bits integer, ADDRESS. During the establishment of a HPLC (OPEN) each module allocates a free port address and uses these afterwards as reference in the block transported until the HPLC is closed (CLOSE).

8.3.2 Flow Control

A credit and a sequence numbering scheme is applied for flow control.

The sequence numbering of block is performed modulo 256 and is used in order to identify blocks in different states in the transport across the host interface.
These states are:

- blocks not sent and for which no credit is given
- blocks not sent for which credit has been given

- blocks sent for which no acknowledge has been received (An acknowledge is returned by the receiver if the block has been processed, the term, 'Processed' being defined explicitly, see 8.4.1)
- bloks sent and acknowledged.

The receiver signals a credit value to the transmitter indicating the space of free buffers reserved for the port in question. A credit of 1 corresponds to a single buffered operation, a credit of 2 corresponds to double buffered operation, etc.

In order to accomplish the above mentioned functions following variables are used:

CREDIT:	Refers to permission to transmit blocks beyond the one last acknowledged. The receiver regulates the CREDIT value according to available resources.
YOUR REFERENCE:	This variable is signalled from receiver to transmitter and is the sequence number of the block being acknowledged, including all earlier blocks not yet acknowledged.
MY REFERENCE:	This variable is signalled from transmitter to receiver together with user data and indicates current reference. An upper limit for MY REFERENCE is the sum of the values of CREDIT and YOUR REFERENCE last returned from the receiver.

The receiver end (HPM or NPM) implements a strategy in using the flow control primitives.
This strategy should incorporate:

- The calculation of CREDIT values should only be based on safe resources, i.e. once a CREDIT value has been given it cannot later be reduced.

- The calculation of YOUR REFERENCE should be based on the condition that the block has been finally processed, i.e. transferred across the network without errors allowing the originator of the block to proceed (within the credit).

- If the sequence number (MY REFERENCE) of a received block is within the last acknowledged, or if it is beyond the receiver credit, it should be considered as a duplicate.

Other application of the HPP may chose different strategies.

8.3.3 Transport Commands

In order to separate between different types of information being transferred each block is marked with a command, OP-CODE, which identify actions to be performed by the receiver of the command.

OP-CODE contains 8 bits with the following meaning:

bit 0: R-bit:
= 1, immediate acknowledge, i.e. the acknowledge is given as soon as the letter has been passed to the subscriber.

= 0, acknowledge is given when credit is raised.

bits 1 and 2: reserved for future use.

bit 3: zero

bits4-7:

0000:	LT	- normal textblock with acknowledge
0001:	ACK	- acknowledgement only
0010:		not used
0011:	INIT	- host port initialization (OPEN)
0100:	TERM	- host port termination (CLOSE)
0101: 1111:		not used

8.3.4 Format of HPP blocks

Related to the previous definition the general format of a block being transferred by the Host Port Protocol is defined as follows.

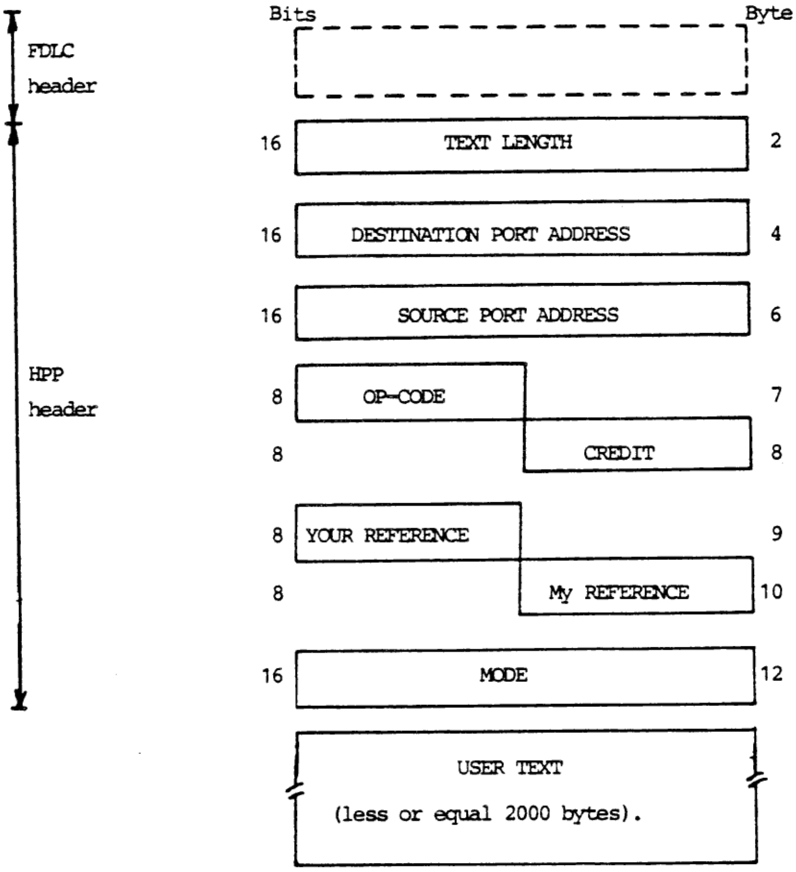


Figure 8.4 Format of HPP blocks.

8.3.5 Initialization (OPEN)

A creation of a Host Port Logical Channel is initiated by the NPM.

A normal HPP block is passed from the NPM to the opponent, identifying

- own port address
- initial credit value

Furthermore a number of parameters is passed, following the block:

- Name of originating subscriber (10 alphanumeric characters)
- Name of opposed subscriber (-)
- Parameter to SC port open (see 8.4).

The parameters of the OPEN primitive are passed to the opposed subscriber which processes/evaluates the parameters before accepting the OPEN command. Herafter a similar command is submitted back to the NPM confirming the initialization, and stating the address and resources of the opponent.

If, for some reason, an initialization cannot be accepted by the opponing end, a termination command will be returned (se 8.3.6).

Please refer to fig. 8.5.

8.3.6 Termination (CLOSE)

Both the NPM and HPM may terminate the HPLC. The parameters are transferred back and forth:

- termination code
- current values of Your Reference and My Reference.

Fig. 8.5 shows the initialization-termination scheme in the form of a state diagram.

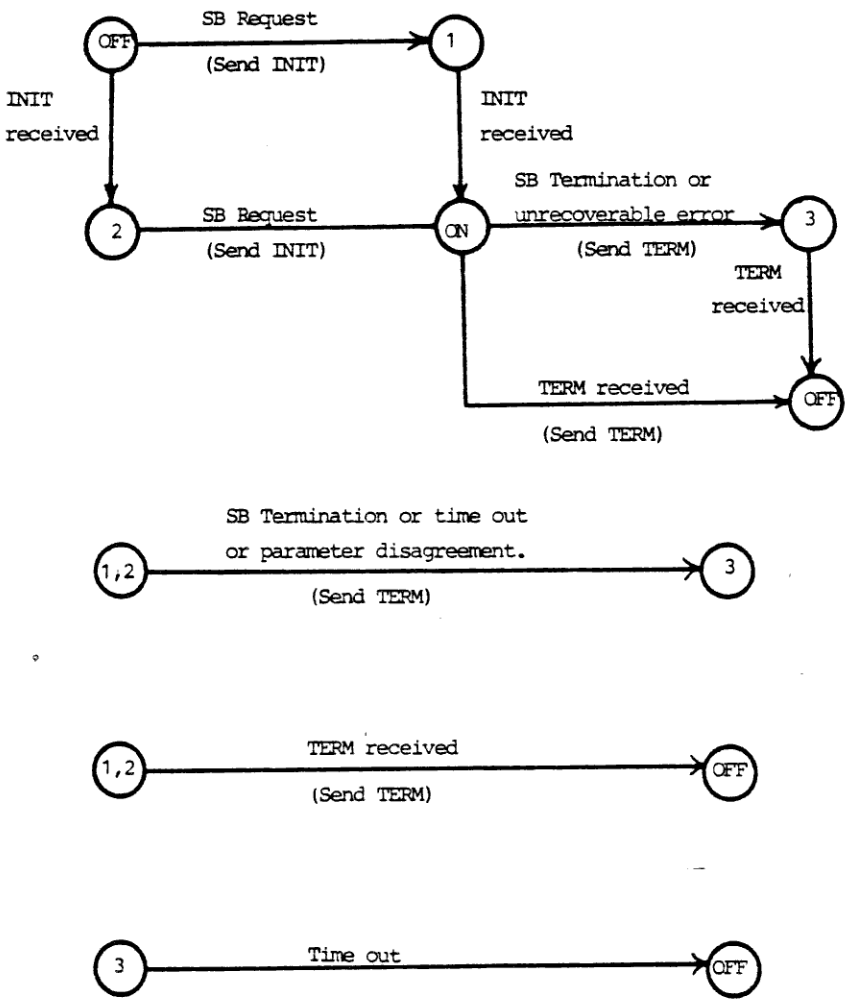


Figure 8.5 Initialization and termination of a Host Port Logical Channel.

8.3.7 Exchange of Supervisor Information

In respect to the HPP supervisor blocks are regarded as normal datablocks which may be intermixed with the normal dataflow.

They are normally used to identify special type of information being exchanged between the two HPP users (cf. section 8.4).

8.3.8 Access Primitives to HPM and NPM

Initialization/Termination:

OPEN HOST PORT (Opponent's name, Own name, Parameters)

Return: Result, Port Address
This primitive is issued by NPM subscripers only.

AWAIT OPEN (Own name)

Return: Result, Port Address, Opponent's name, Parameters.
This primitive is issued by HPM subscripers only.

CLOSE PORT (Port Adress)

Return: Result.

Information transfer:

RECEIVE (Port Address, mode).

Return: Result, Mode, Information block.

SEND (Port Address, Mode, Information block).

Return: Status, mode (only if immediate acknowledge is not requested).

Result: If zero the transport across the channel was succesful.

If non zero the transport did not succeed and the value of Result indicates the cause of the malfunctioning.

Mode: Specifies a command which is passed from sender to receiver together with the information.

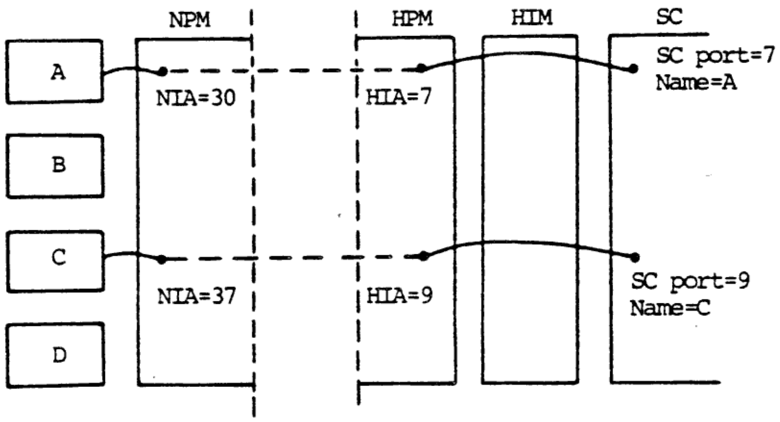
The MODE parameter is set by the user of the NPM and HPM modules.

8.4 Mapping HPP-SC

The initialization of an HPLC only affect the CENTERNET transport system in the sense that the SC tables are updated with the resulting values of the following parameters:

- SC port number opened (=HI address)
- Name of reserver
- Attributes

The established relation is illustrated by the example shown in fig. 8.6.



NIA: Network Interface Address
HIA: Host Interface Address

HPM: Host Port Module
NPM: Network Port Module

Figure 8.6. Example of Host Port relations to SC port.

The RC8000 application programs should be able to receive information from the network at any time, e.g. have issued a RECEIVE to the NPM. Alternatively the applications could initiate a network interaction.

The above example illustrates that the open host port primitive includes the opening of just one SC port.

8.4.1 Mapping of SC Primitives

SC supports a diversity of primitives on an opened port. The RC8000 Host Interface includes a mapping of these primitives, in order to make them available to the reserver. The mode field in the HPP block header is used for this purpose, thus:

- RECEIVE operations from the reserver include a MODE. This causes a receive operation to be set up for the SC, using the MODE as operation code. When the operation is returned from SC, a result (and possibly data) is delivered. This causes the RECEIVE operation to be answered, including result (added to MODE).

- SEND operations from the reserver include a MODE together with data. This causes a send operation to be set up for the SC, using the MODE as operation code. When the operation is returned from SC, a result is delivered. This causes the SEND operation to be answered, including the result (added to MODE). Note that if immediate acknowledge for SEND is requested, the SC result cannot be included in the answer to the SEND operation.

  For the specification of MODE (SC operation codes) please refer to ref. (31).

8.5 NC Service Interface

Both the RC3502C HI modules and the RC8000 NI modules offer services for the network control system (NCC in RC8000, and NCP in RC3502C).

As defined in chapter 5 the access from NCC/NCP is performed via the following primitives:

CONTROL : Control function, which enables the NCC/NCP to perform control on the NI/HI modules.

EVENT : Monitoring function, which enables the NI/HI modules to report important events to the NCC/NCP system.

SENSE : Monitoring function, which enables the NCC/NCP to read status from the NI/HI system.

GET STATISTIC: Monitoring function, which enables the NCC/NCP to get statistical information from the NI/HI system.

The following gives the control and monitoring facilities available on both the RC3502C and the RC8000.

8.5.1 CONTROL Functions

When the NCC/NCP wants to control the NI/HI it can issue one of the following functions:

SET NC MASK
MAX NUMBER OF PORTS

SET NC MASK is used (for both modules) to activate/deactivate (filtering) the monitoring functions. I.e. the NC can indicate to the module whether or not statistical information shall be gathered and which events that shall be reported.

8.5.2 REPORT Functions

The EVENT primitives are returned to the NCC/NCP after the following events:

(1) FDLC on-line.
(2) FDLC off-line.
(3) HPLC established.
(4) HPLC removed.
(5) FDLC restart.
(6) Error detection (system error or FPA error).
(7) Lack of resources.

8.5.3 SENSE Functions

The SENSE primitives are used to get either an immediate state or an immediate status for the module or for a data path through the module. The following primitives are supported:

NPM : SENSE HPLC
      SENSE FDLC
HPM : SENSE HPLC

8.5.4 GET STATISTIC Functions

The STATISTIC primitives are used to retrieve statistical information for either the whole module or for a data path through the module. The below listed primitives are supported:

NPM : HPLC STATISTICS
      FDLC STATISTICS
HPM : GET HPM STATISTICS
      HPLC STATISTICS

8.6 RC3502C HI Structure

The previously described function of the CENTERNET Host Interface is reflected on the module structure within RC3502C.

As shown in figure 8.7 the RC3502C Host Interface consists of the following functional modules:

- FDLC
  Channel Link protocol handler

- Host Port Module (HPM)
  Implements the Host Port Protocol and offers a general host interface for any RC3502C process.
- HI Mapping (HIM)
  Functions as the gateway between the HPP and the SC transport facilities of CENTERNET.

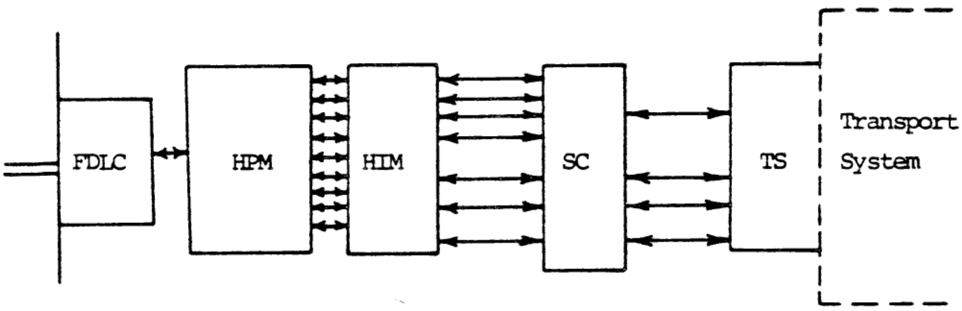


Figure 8.7 Logical Structure of RC3502C HI system.

8.7 RC8000 NI Structure

Like the RC3502C HI a corresponding structure is applicable to the RC8000 Network Interface System. In addition the NI must also cover terminal adaption to the RC8000 operating systems. Three modules (in two s-processes) constitutes the RC8000 CENTERNET Interface.

- NPM
  Network Port Module. Implements the Host Port Protocol and offers access to the service of SC. The same process includes

- FDLC
  Channel Link Protocol Handler.

- SMM
  Scroll Mode Mapping (SMM) Module. It uses the HPP/SC service in order to communicate with CENTERNET Virtual Terminals operated in scroll mode. The VT is converted into standard terminal formats well known to all RC8000 operating systems (i.e. RC822 VDU cf. ref. (103)). The interaction with RC8000 operating systems are performed via pseudo-processes (area-processes) which are created for each terminal user connected.

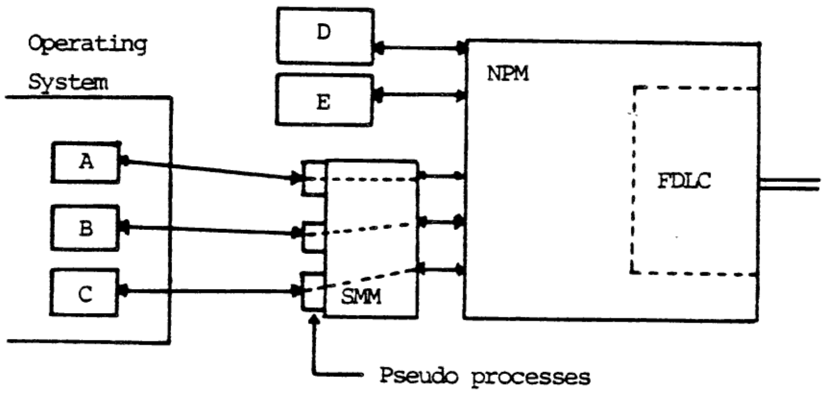


Figure 8.8 Logical structure of RC8000 Network Interface including the SMM Module.

This page is intentionally left blank.

9. CENTERNET RC8000 HOST UTILITIES

9.1 Basic Structure

The RC8000 Host utilities are performed by a number of RC8000 processes under the basic operating system s. They are acting as service modules to other RC8000 systems, supporting a varity of functions on the network.

They interact both with each other and with the Net Interface process (NPM). All the interaction is performed on the standard RC8000 message format.

9.2 File Transport Utility.

This function is performed by an s-process started either from the main console or automatically by operation system s after autoload.

The task to be performed is:

- create a session to a similar process in another machine using the SC primitives - or await session establishment from the outside.

- perform transportion of an RC8000 disk file to or from the local disk, following the File Transfer Protocol described in ref. (4).

9.2.1 FTP Functions and Operations

The File transport utility is operated from another RC8000 process by the normal send message/wait answer operations. The process will normally not make any messages on the main console.

The basic function is a transport request. The other functions are informative or aborting an active transport. An FTU accepts only to perform one file transport at a time. The relevant messages to send to the FTU are:

Transport request

```
Message:  5 <12+2
          first address
          last address.
```

The data block contains definition of the transport requested, i.e. file name and entry base (identifying both the involved files), identification of the responding machine, and mode of access. Mode of access may be make only, replace only or read only.

After an attempt to establish the transport, an answer is supplied to the sender, telling either the cause of failure, or OK (just after the GO command).

Answer: status word
halfwords
characters
status information.

Abort transport

Message: 5 <12+1
first address
last address.

The data block must describe a transport going on, in a format as in transport request. The transport is aborted if it exists, and answered after the STOP command.

Answer: status word
halfword
characters
status information

Status information will include

- transport aborted
- not current transport
- no transport going on

Sense

Message: 0< 12 + mode

The answer contains a status for the FTU. The answer is delayed, controlled by mode just as in get status message. (see below).

Get Status (and statistics)

Message: 3< 12 + mode
first address
last address

The answer contains status and statistical information, as much as the supplied buffer can hold. The status information is in the first part of the buffer. The information concerns the current or last executed transport. The answer is delayed, controlled by the bit pattern of mode, thus:

mode = 0 no delay.

mode = 2 delayed until transport stop.

mode = 4 delayed until transport start.

mode = 6 delayed until transport stop or start.

Start FTU

Message: 202 < 12 + reserve

The message starts up the operation of the FTU, i.e. a host port is opened and the initial receive operation is performed. The sender of the message is identified by name and process description address. reserve = 1 if the sender reserves the FTU for exclusive use (among the RC8000 processes), otherwise it is 0. The answer will include information about

- host port opening failure
- reservation rejected
- start rejected (reserved)
- started already.

Stop FTU

Message: 204 < 12 + 0.

The sender must priviously have started the FTU. The operation of the FTU is now stopped, and the host port is closed. The answer is returned, including information on the legality and success of the operation.

Use of primary output (console)

The FTU will write an output message on the console at process start up, at host port open and close. No other output is sent during normal operation.

In situations where a file transport is started, but no "get status" message is received, the FTU will as a default report the event on the console.

FTU Log File

The FTU maintains a logfile, containing information about all important events. The logfile may be inspected from other RC8000 processes.

9.2.2 Access to SC

The FTP-processes situated, in an RC8000, that wants to exchange information, in the form of files, across the data network, must operate on the network interface, utilizing the service offered by the RC8000 Network Interface process (NPM). These services include, the establishment/termination of connections from the RC8000 to the RC3502C as well as network sessions, and transmission/reception of blocks containing FTP-commands.

9.2.2.1 Initiator FTP-Process

Before any actual transfer can be made, the RC8000 process must establish a host port logical channel. This is accomplished when issuing a OPEN primitive with parameters set in accordance with the description in section 8.3.

When the session to the responding FTP-process is desired, this session is established when the SEND primitive is issued on the opened port with the following parameters.

MODE	:	4
TRANSMITMASK	:	please refer to section 5.5
CLASS	:	1
NAME of RECEIVER	:	Symbolic name of responding FTP-Proc
ADDRESS of RECEIVER	:	Symbolic/absolute addr.of respond. FTP process.
USER DATA	:	RC8000 - FTUP

Now the communication with the responder can commence, following the rules laid out in the FTP protocol, described in ref. (4). All commands created in the FTP-process are transmitted across the network using the SEND primitive on the allocated port with parameters set as follows.

MODE	:	2
buffer	:	Start of buffer containing FTP-command
buffer length	:	Number of bytes in FTP-command.

Controlling the flow on the session is done through the RECEIVE primitive. The result parameter indicates what type of information is received.

After the transfer has been completed the session is removed on responder request and accepted by initiator when the FTP process issues a SEND where

MODE	:	24
CAUSE	:	Normal Termination.

9.2.2.2 Responder FTP-Process

For the initiator to be able to start the transfer to/from the host where the responder is situated, the latter must have declared its existence in the network by issuing a OPEN to the NPM module succeded by a RECEIVE where the receive buffer will be returned, when the initiator's request arrives to the Terminal Concentrator in question. If the answer indicates the receival of a connect request, the FTP process will accept the request by issuing a send where

MODE : 20
USER DATA : RC8000 - FTUQ.

As for the initiator, the SEND/RECEIVE primitives will be used to transmit/receive commands on the session.

In certain situations the responder has to remove the session, and this is done as in initiators end through the SEND primitive with appropriate parameters, including a diversity of causes.

The resonder process in responsible for removing the session after the transfer has been completed. The FTP process will issue a SEND primitive where

MODE : 8
CAUSE : Normal Termination.

9.2.3 System Environments

The FTU process is a normal s-process co-existing with other RC8000 processes. As the FTU is using the NPM process, it must exist in advance. No other demands must be fulfilled in order to execute the file transfer.

The FTU is a service-module to the other RC8000 processes, who are requesting files to be transferred. The FTU is prepared to communicate with any other RC8000 process, but only one transport is active at a time. If the FTU is busy, a transport request message is answered "busy", and the message repertoire is designed to ease the user processes in "waiting for ready FTU".

The essential parameters in the transport request are:

File name and entry base at initiator.
File name and entry base at responder.
Mode of access.
Symbolic name and address of responder.

File name and entry base is a unique identification of a disc file in the RC8000 catalog system.

The file transfer now proceeds in the following main phases, in accordance with the file transfer protocol in ref. (4):

a) check the transport request parameters and reserve the file at initiators end.

b) create session to responder.

c) send SFT.

d) responder checks parameters and reserves the file at responders end. Send RPOS or RNEG.

c) initiator enters level 1 by the GO command (messages waiting for start are answered).

d) data transfer takes place, using the facilities negotiated at level 0.

e) initiator sends STOP.

f) responder disconnects the session. (Now messages waiting for stop are answered).

If anything unusual happends during these phases, the proper escape actions are taken, as stated in ref. (4).

9.2.4 Logical Structure

The FTU is implemented in ALGOL8.

The utility is able to perform the role of initiator or responder, and as sender or receiver. As only one transport is handled at a time, only one role is active.

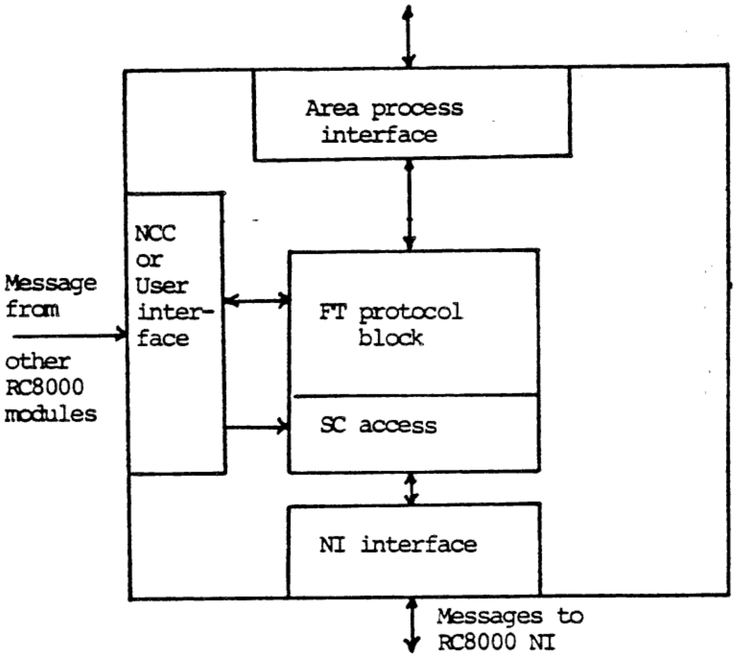


Figure 9.2.1 Logical structure of the RC8000 FTU.

The NCC/User interface handles message from other RC8000 processes and is the access to the environments.

Area process interface is the block accessing the disc files through an area process.

FT protocol block is the actual implementation of the FTP.

SC access is the block handling the access to the SC module located in a TC. This access is performed through the NPM module and NPM interface.

This page is intentionally left blank.

10. SYSTEM ORGANIZATION

The management, operation and maintenance of a larger network system with a constantly varying number of involved hardware and software components indeed requires a powerful set of descriptive and operational tools. These can be grouped into the following types:

1. Identification and description system for each hardware and software module.

2. Basic system operation.
   Dead start and basic network initialization.

3. System Supervision.
   Reports and statistics gathering.

4. Network component maintenance.

5. Development tools.
   Tools available for RC8000 programming.

Note: As the SC/HC files and the down line load are not yet defined some subsections in this chapter may be updated / will not be available until January 1983.

10.1 System Modules

The establishing and management of the CENTERNET network system requires the administration of a number of system modules. Two main categories are defined, hardware modules and software modules.

The hardware modules are arranged in physical installations (cabinets, racks and crates), which from the point of view of operations represents an entity, i.e. no insertion and removal of hardware modules can be performed without interrupting the functioning. In CENTERNET there are defined two such physical entities, the RC3502C Terminal Concentrator and the RC8000 Host Computer. The software modules are arranged in logical blocks which represents a typical function within CENTERNET. Interactions with other logical blocks are performed in order to realize the hierachical structure of the protocol and control system. In CENTERNET there are defined the following logical blocks: Basic TC (CNADAM, HDLC, DTE, TS, SC), X.28 SMT support, Host Interface, Network Interface, Network Control Probe, Network Control Center, File Transfer Utility and RC8000 Scroll Mode Mapping.

The following sections describes the basic structure of the hardware and software configurations as they are described in the HCF- and SCF files. These files are under control by the network control system, and may thus be read/updated only via the NC system.

10.1.1 Hardware Modules, TC

The basic unit consists of a RC3502C CPU with the following elements:

- 16 bit CPU with preprocessor

- 64/256 Kbytes RAM memory module

- Crate and power supply.

The crate has 14 free slots for additional memory and controller boards. All slots within one crate are interconnected via a common internal backplane bus.

Fig. 10.1. outlines the basic TC processing unit.

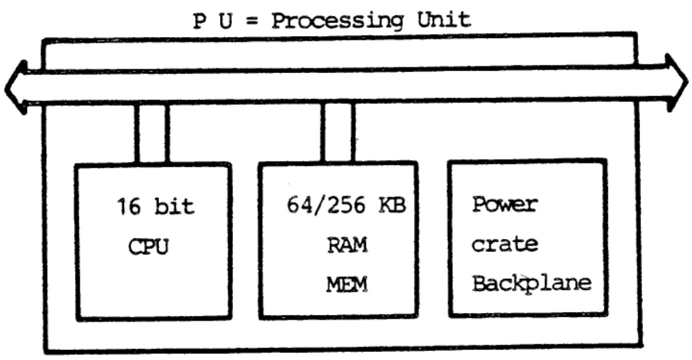


Figure 10.1. Processing Unit of a TC.

A common designation of the basic system is PU (Processing Unit).

A PU can be expended with PC boards up to a total of 14 additional boards. These PC boards encounter the following types:

1) MEM, a 64/256 Kbytes RAM memory module (1 PCB).

2) IMS, a 8 lines asynchronous multiplexor (2 PCB's).

3) COM, a 2/4 lines synchronous HDLC multiplexor (2 PCB's).

4) IOM, I/O board for 8 high-speed serial channels (1 PCB).

5) MBA, adaptor board for the INTEL MULTIBUS (1 PCB).

6) TES, programmable read only memory for image load (64KB-1 PCB).

The IOM board is used for connection of external equipment including the FPA100 channel attachment to the RC8000 Host Computer.

Fig. 10.2 illustrates an example of a single-PU terminal concentrator with RC8000 channel attachment.

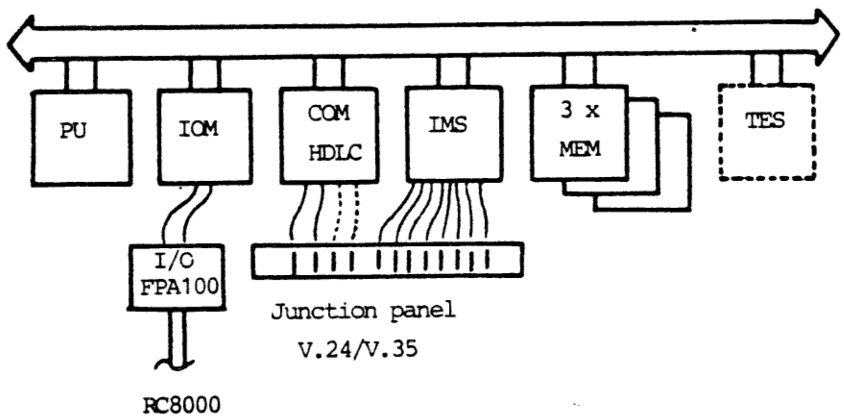


Figure 10.2 Hardware structure of a single-PU TC.

In order to manage the different TC configurations an adequate description and addressing scheme must be defined.

The individual hardware modules within the TC is addressed and described based on the address tree in fig. 10.3.

The hardware Configuration File (HCF) of a TC is thus build up around an element address and an element description (attribute).

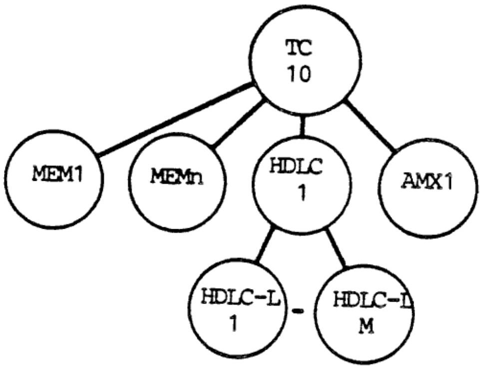


Figure 10.3 3-level identification structure of a TC.

Referring to fig. 10.3. an element e.g. HDLC line 2 is identified by

TC10.HDLC1.HDLC-L.2 - Attribute.

Each element being addressed within a TC has assigned an attribute (50 alphanumeric characters), e.g. hardware serial number, V35 modem number, crate slot number, spare module.

The exact layout of the hardware module attribute is defined with respect to operational and maintenance considerations.

10.1.2 Hardware Modules, RC8000 Host Processor.

The basic unit consists of a model 35S or 45S system which includes the following elements:

- 24 bits processer

- 64 KWord RAM memory

- I/O device processor.

Additional modules include:

1) MEM 64 KWord memory module

2) DSC Disc storage channel

3) DSD Disc storage drives (30 MB, 60 MB, 128 MB, 248 MB) (up to 4 drives per channel)

4) MTC PE Magnetic tape drive channel

5) MTU Magnetic tape unit (up to 4 units per channel)

6) CONS Main operator console

7) FPA Front end processor adaptor for attachment of RC3502C TC/NN (one TC or NN per FPA).

8) LPT Line printer.

Similar to the TC a description and addressing scheme is defined, the elements of which are contained in the hardware configuration files. The attribute field of each addressable unit contains information about sales and serial number, options and features and other informations applicaple for maintenance purpose.

Fig. 10.4. illustrates the address tree of an RC8000 host computer.

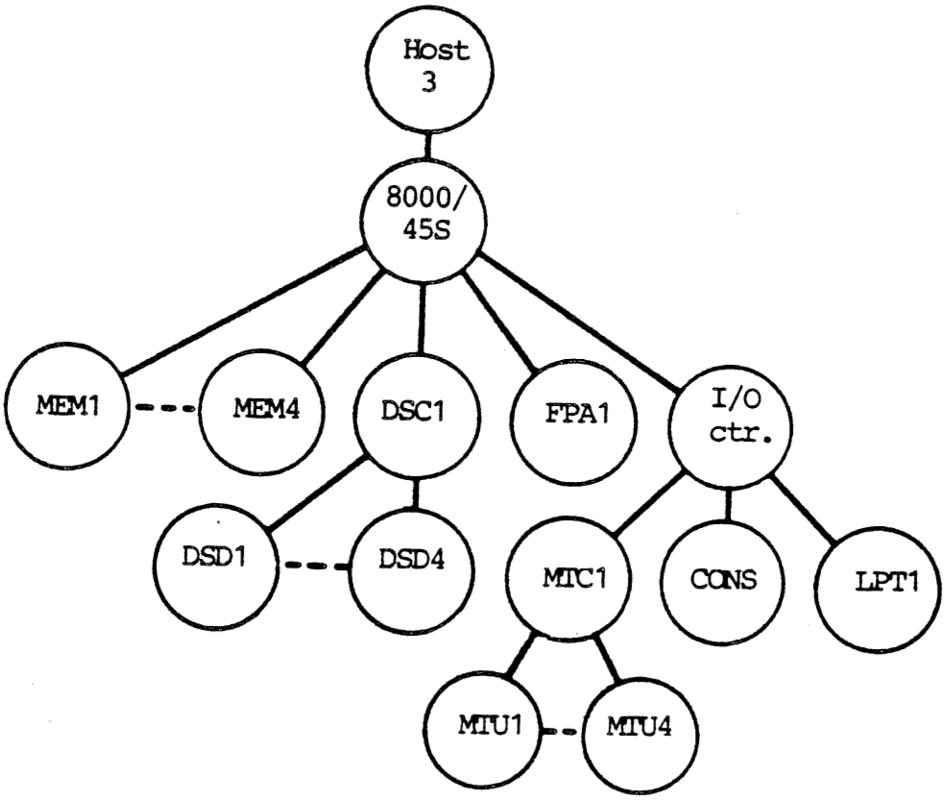


Figure 10.4 Address tree of an RC8000 host processor.

10.1.3 Software Modules, TC

The logical set-up of the software system within a TC is arranged in the following blocks:

- Basic TC process - CNADAM

- Basic transport elements - DTE, TS, SC - DTS.

- X.28 Scroll Mode Terminal support - SMT.

- RC8000 Host Interface - HI.

- Network Control Probe - NCP.

- Basic Software - BS.

An address and identification structure is defined (fig.10.5) where each addressable unit is described by a description record which contains:

- generation parameters

- program version number

- special facilities, options or registrated errors

- name of source code

- name of load file

- system requirements.

The contents of the description record is used each time a software module is changed and the updated record is contained in the software configuration file (SCF).

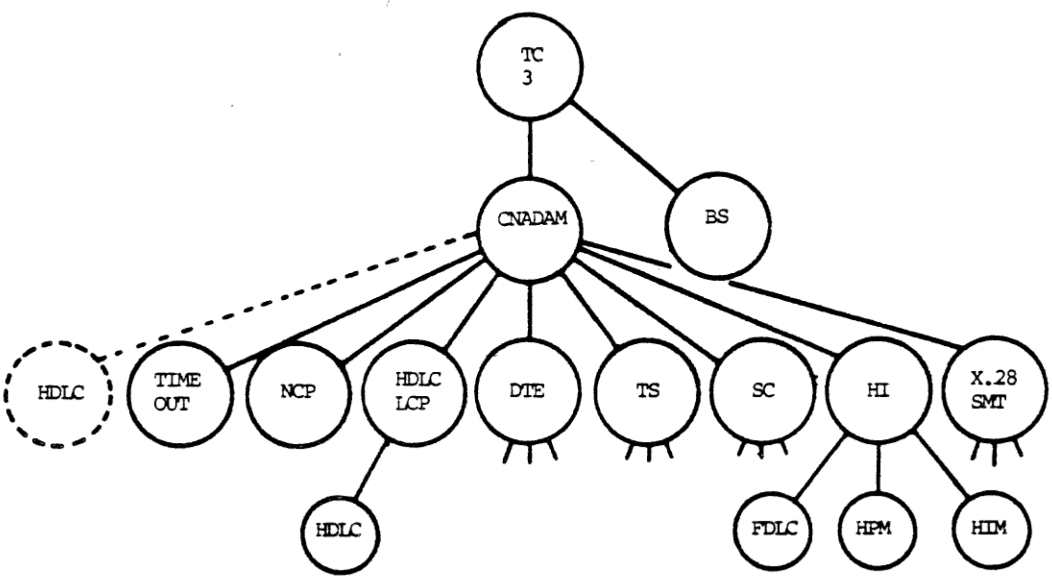


Figure 10.5 Software identification tree of a TC.

10.1.4 Software Modules, RC8000 Host

Following logical blocks of modules are defined.

- Network Interface - NI
- Operating systems - OS
- Network Control Centre - NC
- File Transfer Utility- FTU
- RC8000 Scroll Mode Mapping - SMM

Fig. 10.6. illustrates an example of the software modules in an RC8000 Host Computer. Only the modules required for CENTERNET operations are shown, additional facilities such as compilers, editors and other utilities are available as either BOSS, SOS/TEM, or s tasks.

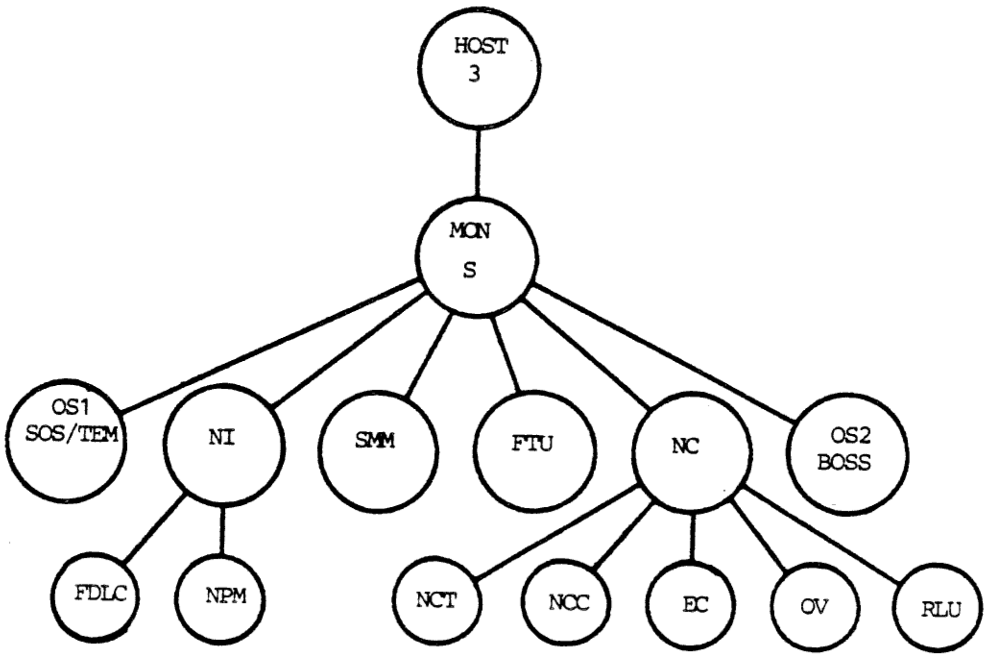


Figure 10.6 Software modules of an RC8000 Host Computer.

10.2 Basic System Operation

The basic operation of CENTERNET always requires an operator who uses either a local system terminal or the operator terminal.

The local terminal only enables basic control of the modules belonging to the machine to which the terminal is connected. The NC operator communicates with the network control system which gives him access to all controlling primitives defined within NC (cf. sections 6.3.2 and 6.3.5).

10.2.1 Local Operator Interactions

Only the RC8000 supports a local operator. The RC3502C basic software supports an operator console, which will be used to test, debugging, and as output medium for module messages.

The local operator of the RC8000 is needed when dead starting and closing down. The I/O device controller is considered as an intergral part of the RC8000 in contrast to the RC3502C TC.

The local operator control on RC8000 include:

- initiating and closing the CENTERNET software components residing as s-processes in the machine.

- logging information concerning local devices

- system restart in case of severe errors.

CENTERNET software residing in the RC8000 is divided into a number of multiprogrammed processes, each executing a program module. In addition to this, one or more operating systems offers the services needed to execute the programs. Local operator interaction is necessary to start these programs but may be done in an easy way and should occur infrequently.

10.2.2 Dead Start and Software Module Change

CENTERNET contains three types of components which may be deadstarted by means of an autoload and in which additional software modules can be loaded.

- RC8000 Host Computer

- RC3502C Local Terminal Concentrator, i.e. directly attached to an RC8000

- Remote RC3502C TC with no RC8000 attached.

<u>RC8000</u>

Deadstart is initiated manually by an operator on the local console and the autoload is performed from its normal auto- load device, e.g. disc.

Additional module load is normally carried out by the local operator too, but a method exists so that a module load can be initiated via the NC system. The method applied is that a special NC transaction addressed to the module in question causes the module to perform a reload from a predefined loadfile. In this way the s-process modules can be loaded via the NC system.

RC3502C Local TC

The RC3502C contains a watch-dog function which will invoke an autoload sequence performed in two steps:

1) The bootstrap loader is loaded from the 16 KB PROM in the first memory module (MEM 0).

2) This bootstrap loader activates an autoload via the FDLC channel. This autoload enables the RC8000 to load the TC with any appropriate software system.

RC3502C Remote TC

To appear.

10.3 Network Monitoring/Control

As described in chapter 6 the monitoring and control facilities enable a network operator, via the network control terminal (NCT), to supervise and control all basic functions in the network.

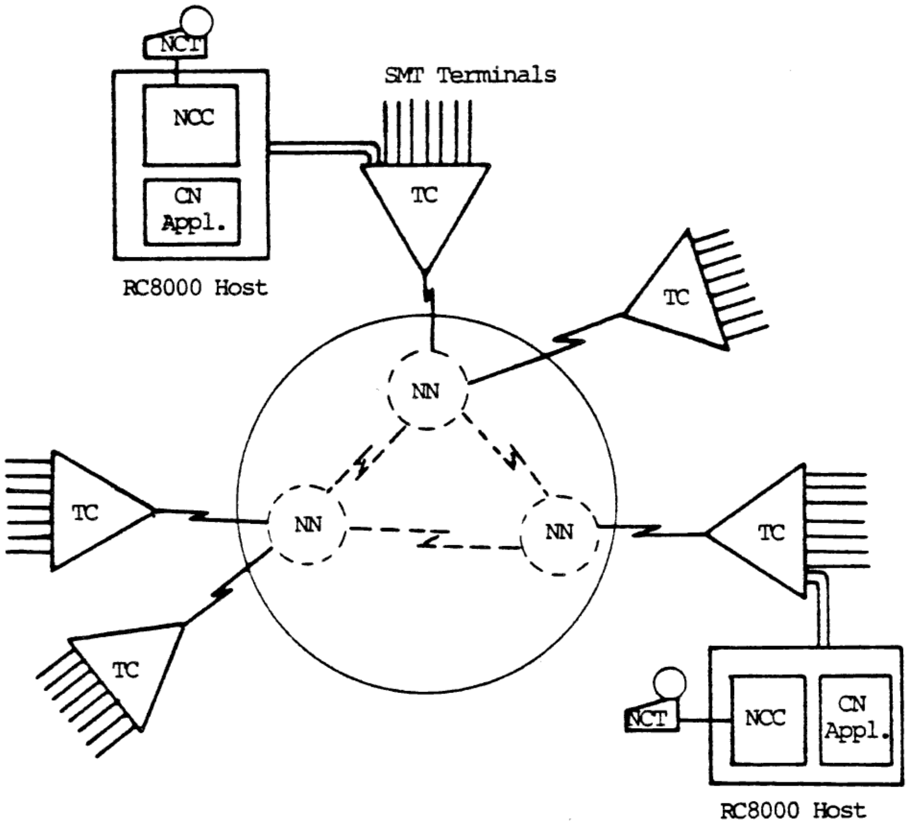


Figure 10.7. Network monitoring/control sites (example).

The NCT of an RC8000 may be any Scroll Mode Terminal (X.28-SMT). The NCT handler module contains a log-on and password mechanism which enables the enforcement of competence areas of each of the NCT operators and may also prevent unauthorized access.

The NC facilities available for a NCT operator enable supervision of the total network, supplying information as:

- display of number and location of active TC's
- alarm reports in case of hardware failures
- display of number of active SMT ports
- main traffic values
- average throughput on different levels

Furthermore the NCT operator has at his disposal a number of network control facilities and utilities:

- close of SC ports
- max number of HI ports
- max number of TS ports
- open/close of specified hardware paths (lines, channels, AMX ports)
- software control of different network and NC modules
- access to traffic and session control primitives
- load/autoload of specified hardware elements

Further details concerning monitoring and control facilities are found in chapter 6.

10.4 Network component maintenance

The maintenance task is normally three-fold:

1) Determination of weaknesses in the system, e.g. bottlenecks. This is done concurrently with normal network operation.

2) Reliability test of single components. The component is taken out of service for separate testing.

3) Diagnostics, to locate faulty equipment.

10.4.1 On-line test

The basic tool for on-line test is artificial traffic. Traffic may be generated between two network components on the same logical level. The following possibilities exist:

- SC traffic, using the SC and lower logical levels

- Traffic between two SMT's. Two types of tests (network-, access test) are possible, and the generator may operate in two modes. Either it generates and repeats a test pattern or it echoes/drops the received traffic.

10.4.2 Component reliability test

The component in question cannot be tested without taking it out of normal operation. Normally hardware communication channels are the only components tested for reliability. The procedure is to establish a loop including the components in question and generate traffic which is returned to the sender and compared with the original data.

1) An RC3502C HDLC line is tested by means of the standard HDLC Reliability. The loop is established by using two HDLC lines in the same RC3502C.

2) FDLC channel link. As in (1), but the reliability program resides in the RC8000 and a mirror is running in the RC3502C.

10.4.3 Diagnostic test

The faulty equipment is taken out of service, and the standard RC diagnostics used to locate the error. These diagnostics programs are stand-alone programs used by RC technicians.

10.5 Development Tools, RC8000

To maintain and further develop network software it is necessary to provide tools that permit editing, compiling, and execution of programs, written in Algol8 and Real-Time Pascal.

Any SMT terminal connected to CENTERNET can access the operating system of the RC8000 Host Computer.

The terminal connect via the SMM to the RC8000. When the connection is established, the terminal potentially has full access to the RC8000 system, acting as a normal RC8000 TTY terminal (RC822).

In the following a resume of the RC8000 system facilities is given. For an exhaustive description refer to references given in Appendix A.3.

10.5.1 Basic System Operation-FP

The basic communication kernel between user and system is the File Processor, FP.

A system command is principally a request for execution of a system program. The task of FP is then to interpret the request, seek the system program, load it into core, start its execution, and pass any parameters to it.

The system is provided with a host of system programs termed utilities. Ref. (98) gives a full description of utilities available. However, the structure of the system allows the user to write his own utilities easily, and thus expand and shape the system for his own purpose.

10.5.2 Backing Storage Organization

Backing storage or the file system is organized as set of files, that is accessed through the catalog. Access rights to a file are given by the concept of bases. This concept is thoroughly explained in ref. (93).

Generally speaking, the access to a file is defined by the scope with which a user views the file. An illustration is the comparison to the lexical levels of an Algol60 program.

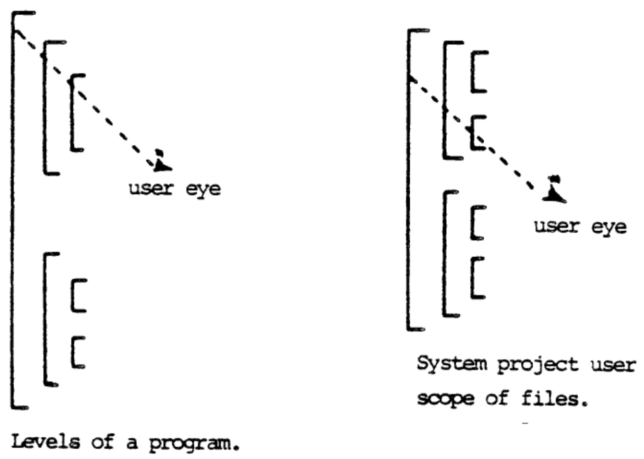


Figure 10.8. The Scope Concept.

A user can 'see', i.e. read, his own files (with scope user), his projects files (with scope project) and system files. He is able to write in user and project files but not system files.

Organized appropriately, users and projects can be protected against each other. The system contains a host of utilities to manipulate and maintain the File System and provides simple procedures for back-up on magnetic tape or discette.

10.5.3 Editing

Editing of text files is performed applying the Utility Edit.

Edit is a general context oriented editor, containing the classic commands for searching, replacing, deleting, and inserting text strings in a file. Details are given in ref. (98).

10.5.4 Compiling

The system will provide compilers for the Algol8 and Real-Time Pascal. The operation of the compilers is like a utility and is as follows:

> The compiler translates the input text file into an object file ready for load, if the input was error free. Otherwise a comprehensive list of error messages is given to the terminal.

10.5.5 Program Execution

The object program is viewed as an utility - thus an execution of the program is a FP call of the object program's name followed by any parameters. It is the responsibility of the program to read and interpret the parameters passed by FP.

10.5.6 Interface to the NC

The terminal connected to the RC8000 system has no direct contact with the Network Control situated in the RC8000. Files, programs, etc. are passed to the Network Control by altering their scope so as to allow the Network Control Centre to access the files.

The terminal can then connect to the Network Control and inform it to do what ever necessary with the file.

This page is intentionally left blank.

A. REFERENCES

A.1 CENTERNET Documentation

A.1.1 Report and Logical Specifications

(1) RCSL No. 43-GL11411
CENTERNET, Transport End-to-End Protocol
Report

Peder Thisted, August 1981.

(2) RCSL No. 43-GL11428
CENTERNET, Virtual Terminal Protocol - VTP
Report

Per Høgh, Inger Marie Toft Hansen, December 1981.

(3) RCSL No. 43-GL11412
CENTERNET, X.28 Scroll Mode Terminal Support
Report

Inger Marie Toft Hansen, March 1982.

(4) RCSL No. 43-GL11401
CENTERNET, File Transfer Protocol - FTP
Report

Carl Henrik Dreyer, February 1981.

(5) RCSL No. 43-GL11209
CENTERNET, Host Port Protocol
Report

Karsten Kynde, May 1981.

(6) RCSL No. 43-GL11415
CENTERNET, FDLC - FPA Data Link Control
Report

Niels Carsten Jensen, August 1981.

(7) RCSL No. 43-GL11278
CENTERNET, Session Control Protocol - SC
Report

Viggo Lomborg, May 1982.

(8) RCSL No. 43-GL11418
CENTERNET, Network Control
LCP Specification Sheets

Peter Bjerregård Lauridsen, July 1982.

(9) RCSL No. 43-GL11424
NCP Data Structures, Reference Manual
Revision 1.02

Claus Houlberg Hansen, August 1981.

(10) RCSL No. 43-GL11825
CENTERNET, Network Control
HDLC LCP Specification Sheets

Per Høgh, July 1982.

(11) RCSL No. 43-GL11826
CENTERNET, Network Control
DTE LCP Specification Sheets

Per Høgh, July 1982.

(12) RCSL No. 43-GL11827
CENTERNET, Network Control
TS LCP Specification Sheets

Per Holager, July 1982.

(13) RCSL No. 43-GL11828
CENTERNET, Network Control
SC LCP Specification Sheets

Per Høgh, Viggo Lomborg, July 1982.

(14) RCSL No. 43-GL11829
CENTERNET, Network Control
SMT LCP Specification Sheets

Inger Marie Toft Hansen, July 1982.

(15) RCSL No. 43-GL11830
CENTERNET, Network Control
HPM LCP Specification Sheets

Viggo Lomborg, July 1982.

(16) RCSL No. 43-GL11831
CENTERNET, Network Control
NPM LCP Specification Sheets

Carl Henrik Dreyer, July 1982.

(17) RCSL No. 43-GL11832
CENTERNET, Network Control
NCP LCP Specification Sheets

Claus Houlberg Hansen, July 1982.

(18) RCSL No. 43-GL11833
CENTERNET, Network Control
NCC, NCT, EVENTCOL and OVERVIEW
LCP Specification Sheets

Peter Bjerregård Lauridsen, Jørgen Christensen, July 1982.

(19) RCSL No. 43-GL11835
CENTERNET, Network Control
CNADAM LCP Specification Sheets

Viggo Lomborg, July 1982.

(20) RCSL No. 43-GL
CENTERNET,
Capacity and performance considerations

(Not yet printed).

A.1.2 User's and Programming Guides

(30) RCSL No. 43-GL11341
CENTERNET, RC8000 Network Access
Programming Guide

Karsten Kynde, August 1981.

(31) RCSL No. 43-GL11277
CENTERNET, Session Control
Programming Guide

Per Høgh, August 1982.

(32) RCSL No. 43-GL11282
CENTERNET, Transport Station
Programming Guide

Peder Thisted, August 1981.

(33) RCSL No. 43-GL10963
CENTERNET, DTE Module
Programming Guide

Per Holager, Per Høgh, August 1982.

(34) RCSL No. 43-GL11421
CENTERNET, Network Control
Network Control Terminal, User's Guide

Jørgen Christensen, July 1982.

(35) RCSL No. 43-GL11420
CENTERNT, Network Control
NC-Utilities, Programming Guide

Peter Bjerregård Lauridsen, August 1982.

(36) RCSL No. 43-GL11419
CENTERNET, Network Control
Overview Files, User's Guide

Peter Bjerregård Lauridsen, August 1982.

(37) RCSL No. 43-GL11294
CENTERNET, RC8000 File Transport Utility - FTU
User's Guide

Carl Henrik Dreyer, August 1981.

(38) RCSL No. 43-GL11761
CENTERNET, Session Control Artificial Traffic Module (SCAT)
User's Guide

Viggo Lomborg, March 1982.

(39) RCLS No. 43-GL11149
CENTERNET/PAXNET
Network Control Terminal Handler - NCTH
User's Guide

Uffe Harksen, March 1981.

(40) RCSL No. 43-GL11414
CENTERNET, Host Port Module - HPM
Programming Guide

Karsten Kynde, August 1981.

(41) RCSL No. 43-GL11413
CENTERNET, RC3502 FPA Data Link Control
Programming Guide

Karsten Kynde, August 1981.

(42) RCSL No. 43-GL
CENTERNET, Network Control
Remote Load Utility, User's Guide

(Not yet printed).

(43) RCLS No. 43-GL11696
CENTERNET, RC8000 Scroll Mode Mapping Module (SMM)
Operating Guide / User's Guide

Lis Clement, November 1981.

(44) RCLS No. 43GL
CENTERNET, System Software Configuration,
Generation and Installation Guide

(Not yet printed).

A.1.3. Reference and Maintenance Documentation

(50) RCSL No. 43-GL11422
CENTERNET, Network Control
Network Control Center Nucleous, Reference Manual

Peter Bjerregård Lauridsen, August 1982.

(51) RCSL No. 43-GL11423
CENTERNET, Network Control
Event Collector, Reference Manual

Peter Bjerregård Lauridsen, August 1982.

(52) RCSL No. 43-GL11697
CENTERNET, RC8000 Scroll Mode Mapping Module (SMM)
Reference Manual

Lis Clement, November 1981.

(53) RCSL No. 43-GL
CENTERNET, CNADAM
Reference Manual

(Not yet printed).

(54) RCSL No. 43-GL
CENTERNET, Network Control
Remote Load, Reference Manual

(Not yet printed).

(55) RCSL No. 43-GL11738
CENTERNET, DTE Module
Reference Manual

(Not yet printed).

(56) RCSL No. 43-GL11747
CENTERNET, Transport Station (TS)
Reference Manual

Per Holager, April 1982.

(57) RCSL No. 43-GL11739
CENTERNET, Session Control Module (SC)
Reference Manual

(Not yet printed).

(58) RCSL No. 43-GL11762
CENTERNET, X.28 Scroll Mode Terminal Module (X.28-SMT)
Reference Manual

Inger Marie Toft Hansen, August 1982.

(59) RCSL No. 43-GL11426
CENTERNET, Host Port Module - HPM
Reference Manual

Karsten Kynde, October 1981.

(60) RCSL No. 43-GL
CENTERNET, Session Control Artificial Traffic Module (SCAT)
Reference Manual

(Not yet printed).

(61) RCSL No. 43-GL11417
CENTERNET, Network Port Module - NPM
Reference Manual

Niels Carsten Jensen, August 1981.

(62) RCSL No. 43-GL11293
CENTERNET, RC8000 File Transport Utility - FTU
Reference Manual

Carl Henrik Dreyer, August 1981.

(63) RCSL No. 43-GL
CENTERNET, Network Control
Network Control Terminal Module, Reference Manual

(Not yet printed).

(64) RCSL No. 43-GL11763
CENTERNET, Network Port Module - NPM
Installation/Operating Guide

Carl Henrik Dreyer, March 1982.

(65) RCSL No. 43-GL11580
CENTERNET, Dokumentation af fejlsituation i CN-programmel

Per Høgh, Juni 1982.

A.2 General RC3502 and RC3502C Documenation

Note: Documentation reference for the RC3502C system will be supplied later.

(80) RCSL No. 52-AA1074
RC3502 COM201 HDLC-Driver
Reference Manual

Per Mondrup, November 1981.

(81) (To appear later).

(82) RCSL No. 30-M285
RC3541/43, High Level Communication Controller
General Information

Lars Myrup Jacobsen, March 1981.

(83) RCSL No. 30-M283
RC3541/43, Reference Manual

Lars Myrup Jacobsen, March 1981.

A.3 General RC8000 Documentation

(90) RCSL No. 42-i1221
RC8000 System Architecture

Inge Boch, Henning Christensen, January 1979.

(91) RCSL No. 42-i0846
RC8000 Hardware, System Software, Survey

Inge Boch et al., August 1978.

(92) RCSL No. 31-D476
RC8000 Monitor, Part 1, Design Philosophy

Henrik Sierslev, Pierce C. Hazelton, November 1979.

(93) RCSL No. 31-D477
RC8000 Monitor, Part 2, Reference Manual

Tove Ann Aris, Bo Tveden-Jørgensen, January 1978.

(94) RCSL No. 31-D478
RC8000 Monitor, Part 3, Definition of External Processes

Palle Andersson, January 1979.

(95) RCSL No. 31-D584
Corrections to RCSL No.: 31-D477, RC8000 Monitor, Part 2

Niels Carsten Jensen, October 1979.

(96) RCSL No. 31-D643
OPERATING SYSTEM s, Reference Manual

Henrik Sierslev, May 1981.

(97) RCSL No. 31-D364
System 3 Utility Programs, Part One

Hans Rischel, Tove Ann Aris, April 1975.

(98) RCSL No. 31-D590
System 3 Utility Programs, Part Two

Finn G. Strøbech, January 1980.

(99) RCSL No. 31-D379
System 3 Utility Programs, Part Three

Tove Ann Aris (ed.), November 1975.

(100) RCSL No. 31-D581
Algol8

Jørgen Zachariassen, November 1979.

(101) RCSL No. 42-i1278
Algol8, User's Guide Part 2

Edith Rosenberg, October 1980.

(102) RCSL No. 52-AA964
Real-Time Pascal, Report

Jørgen Staunstrup, January 1980.

(103) RCSL No. 31-D580
RC8000 Terminal Process

Palle Andersson, September 1979.

.4 Miscellaneous

(120) Recommendation X.25

CCITT, Yellow Book
Data Communication Networks
Services and Facilities, Terminal Equipment and Interfaces
Vol. VIII - Fascicle VIII.2.
Geneva 1981.

(121) Recommendation X.121

CCITT, Yellow Book
Data Communication Networks
Transmission, Signalling and Switching, Network aspects,
Maintenance, Administrative arrangements
Vol. VIII - Fascicle VIII.3
Geneva 1981.

(122) Draft International Standard ISO/DIS 7498

Information Processing Systems - Open systems interconnection
- Basic reference model
ISO/TC97, April 1982.

(123) Standard ECMA-75, Session Protocol

European Computer Manufacturers Association
ECMA, January 1982.

(124) INWG Protocol 86, HLP/CP(78)
A Network Independent File Transfer Protocol

High Level Protocol Group, December 1977.

(125) INWG Protocol 96.1, ISO/TC97/SC6 N1557
ISO/TC97/SC16 N24
Proposal for an Internetwork end-to-end Transport Protocol

V. Cerf. et al, March 78.

This page is intentionally left blank.

B. Basic Terminology

In this appendix different terms used in the System Specification and other CENTERNET documentation will be explained. The list is not complete but the most commen are explained. The appendix is divided into 2 parts

- basic terms

- abbreviations for software and hardware components.

B.1 Basic Vocabulary

Access

The class of procedures defined for utilizing the service offered by the lower layer.

Artificial Traffic Generator

Software module or procedures incorporated in a module generating artificial traffic. Different types of traffic may be specified. Used to test network performance, data connections or terminal connections.

Association

An association constitutes the support for a conversation between a pair of network entities. A network entity can participate in several associations simultaneously. (Source INWG 96.1, edited).

Channel Link

The channel link is a data-link between the RC8000 Host Computer and a Terminal Concentrator.

CENTERNET Entity

CENTERNET user connected to the Session Layer and having a unique network address.

Class

As used in the Session Control Service and the Transport Station Service class means priority of data transfer.

Command Field

Name of level 0 data-field of the File Transfer Protocol (FTP).

Controlling (NC)

The direct interaction in network performance.

Data Entry Mode

One of the image classes of Virtual Terminals. The characteristics of the class is the division of the presentation unit into fields with assigned attributes.

Data Field

Name of the field holding user data in the File Transfer Information Unit and in the X.25 data packet.

Data Link

The assembly of two stations that are controlled by a linkprotocol and that together with the interconnecting datacircuit enables data to be transferred from one node to another node, whereby nodes are acting as a data-source and a data-sink. A data link is the equivalent term for linkconnection. (Source ISO/TC 97/SC16/N227).

Data Network

The assembly of functional units that establishes data circuits between data terminal equipments. (Source ISO/DIS2382/IX).

File

A general term, used for a backing store area holding information/data. (Source RCSL 31-D609, edited).

File Transfer Information Unit (FTIU)

All information transmitted by the FTP is divided into record (FTIU) and a record is delivered as a UIU to the session service.

Fragment

The Transport Station breaks the User Information Unit (UIU) into fragments adds a TS-header to each fragment and sends the fragments to the receiving Transport Station, where the fragments is reassembled into the UIU.

Hardware Configuration File (HCF)

A file located at the Network Control computer describing the hardware configuration of the network.

Host Interface Address (HIA)

HIA is the access-address of the endpoint, the host port of an Host Port Logical Channel (HPLC) in an TC.

Host Port

Host ports constitutes the logical name space in a Terminal Concentrator of applications in the host computer independently of the actual addressing method.

Host Port Logical Channel

A two-way simultaneous transmission path across a channel link, comprising associated send and receive paths.

Initiator

The user/application/module initiating an action. E.g. the application initiating a connect message to the session layer is called the initiator of the session establishment.

Letter

A message (User Information Unit) exchanged between two TS ports utilizing the liaison mode of the transport service.

Lettergram

A self contained, independent messages (UIU) exchanged between two TS ports using the lettergram mode of the transport service.

Lettergram Mode

A data transfer service offered by the Transport Station. The characteristic is independent messages exchange, no previous synchronization between both ends, no error recovery, i.e. a transaction oriented service.

Lettergram Service/Mode

See Lettergram mode but offered by the Session Control.

Liaison

A two-way simultaneous transmission path between two transport-service-access-points (TS ports) which is offered as liaison mode by the Transport Station for sequenced and synchronized conversations.

Liaison Mode

A data transfer service offered by the Transport Station. The characteristic is sequenced and synchronized exchange of messages, full error recovery.

Link/Line

The access address of the endpoint of a data link or a channel link.

Local Connection

The establishment of a data path between the Physical Terminal and the Terminal Concentrator.

Management (NC)

The administration/manipulation of the monitoring- and control functions, and the retrieved information.

Monitoring (NC)

Monitoring comprices the functions of report/statistical information retrieval.

Native Mode

One of the image classes of Virtual Terminals. The characteristic is transparently data exchange.

Network Application Identification (NAID)

A NIAD uniquely identifies an application at the Session Control Interface. Furthermore the NIAD is a part of the Network User Identification (NUI).

Network Control Center - Nucleus (NCC)

The NCC comprises the basic set of tools required by all NCC-utilities.

Network Control Center - Utility

A module/program performing a dedicated network control function.

Network Control Information Unit

A supervisor transaction. I.e. a selfcontained Network Control function exchange between two NC-units utilizing the lettergram mode of the transport service.

Network Inteface Address (NIA)

NIA is the access address of the endpoint, the port, of an Host Port Logical Channel (HPLC) in the host computer.

Network Unit Identification (NUID)

The identification/address of a unit (Terminal Concentrator, Host Computer) connected to the data network by the SC/TS interface. A NUID may be perceived as an SC/TS network address. Furthermore the NUID is a part of the Network User Identification (NUI).

Network User Identification (NUI)

A NUI uniquely identifies a network user in the network. The NUI is used to access a network user. The NUI consists of the two parts, NUID and NAID.

Responder

The user/application/module answering a request. E.g. the application answering a connect messages from an initiator.

Session

A cooperative relationship between two application-entities characterizing the communication of data between them. (Source, ISO/TC97/SC16 N227).

Session Control Port

SC Ports is the addressing space in the Session Control Interface. I.e. a SC port is the access address of the session at the SC interface.

Session Entity

See CENTERNET entity.

Session Service/Mode

A data transfer service offered by the Session Control. The characteristic are safe establishment and disconnection of the data path (session), sequenced and synchronized exchange of data and full error recovery.

Stream

A stream in the access address of a Virtual Call at the X.25-DTE interface.

Transport Station Port

TS ports is the adressing spare in the Transport Station Interface. I.e. a TS port is the access address of the endpoint of a transport-connection.

Virtual Call (VC)

A user facility in which a call set up procedure and a call clearing procedure will determine a period of communication between two DTE's in which users data will be transferred in the network in the packet mode operation. All the user' data is delivered from the network in the same order in which it is received by the network. (Source, RC.PAXNET.CHH.6).

X.25 Logical Channel

In packet mode operation, a means of two-way simultaneous transmission across a data link, comprising associated send and receive channels. (Source, RC.PAXNET.CHH.6).

X.25 Logical Channel Number

Each X.25 Logical Channel is identified by an logical channel number (LCN) in the DTE/DCE interface.

X.25 LAP B Link

See Data link.

B.2 Abbreviations for Software and Hardware Modules

CNADAM

Father process (Real-Time Pascal term) for all CENTERNET processes in a Terminal Concentrator (TC).

DCE (X.25-DCE)

Data Circuit-terminating Equipment.
Module, placed in a Network Node (NN), handling the X.25 level 3 interface of the Data Network.

DTE (X.25-DTE)

Data Terminal Equipment.
Module, placed in a Terminal Concentrator, handling the X.25 level 3 interface to the Data Network.

EC

Event Collector.
Network Control Utility collecting all events in a disc file.

FDLC

FPA Data Link Control.
Module handling the FDLC protocol used between an RC8000 Host Computer and the Terminal Concentrator. Is located both places.

FTU

File Transfer Utility located in an RC8000 Host Computer and handling disc file transfers.

HDLC

High level Data Link Control.
Driver located in a Terminal Concentrator or Network Node and handling the X.25 level 2 protocol LAPB.

HIM

Host Interface Mapping.
Module located in the Terminal Concentrator and performing the mapping function between the Session Control Interface and the Host Interface.

HPM

Host Port Module.
Is located in the Terminal Concentrator and handles the access to the host utilizing the Host Port Protocol.

LCP

Local Control Probe.
Integrated part of different software modules. It handles the access to the Network Control System.

NCC

Network Control Centre nucleous located in the RC8000 Host Computer. Is the kernel (mailbox) in the Control Centre performing the access to the network and the communication between different NC Utilities in a RC8000 Host Computer.

NCP

Network Control Probe.
Module located in every network unit performing different Network Control function and handles the communication between the individual LCP's and the Control Centre.

NCT

Network Control Terminal.
Module located in the RC8000 Host Computer and is the human interface to the Network Control System.

NPM

Network Port Module located in the RC8000 Host Computer and is the Network Interface for modules in the RC8000 Host Computer.

OVIEW

OverVIEW.
Network Control Utility located in the RC8000 Host Computer, gathering information from the different CENTERNET units and processing the data, so an overview picture can be constructed by the NCT.

RLU

Remote Load Utility.
Network Control Utility active at down-line load of Terminal Concentrators.

SMM

Scroll Mode Mapping module.
Mapping module located in the RC8000 Host Computer. The mapping functions concerns Virtual Terminal Protocol formats to/from standard terminal formats well known to all RC8000 operating systems (i.e. RC822 VDU).

SC

Session Control.
The module handles the functions defined for layer 5 and the addressing functions of CENTERNET. It is located in the Terminal Concentrator.

TC

Terminal Concentrator. A TC is a RC3502C minicomputer system performing the access to the Data Network and the RC8000 Host Computer. Different terminals may connect to the concentrator.

TS

Transport Station.
The module handles the functions defined for layer 4 and is located in the Terminal Concentrator.

VT

Virtual Terminal.
Is a framework for defininy a network wide standard description of a terminal.

X.28-SMT

X.28-Scroll Mode Terminal.
This module handles the physical terminal connection to the Terminal Concentrator and performes the mapping from physical terminal to the Virtual Terminal.

FINLAND
RC SCANIPS OY
Espoo, 0 51 35 22

FRANCE
RC COMPUTER S.A.R.L.
Paris, 12 33 53 63

HOLLAND
REGNECENTRALEN (NEDERLAND) B.V.
Gouda, 1820-29455

KUWAIT
KUWAITI DANISH COMPUTER CO. S.A.K.
Safat, 83 01 60

NORWAY
A/S RC DATA
Jessheim, 29 70 220

PHILIPPINES
CARDINAL ELECTRONICS CORPORATION
Metro Manila, 882478

SWEDEN
SCANIPS DATA AB
Stockholm, 8 34 91 55

SWITZERLAND
RC COMPUTER AG
Basel, 61 22 90 71

UNITED KINGDOM
REGNECENTRALEN (UK) LTD.
London, 1 606 3252

UNITED STATES
LOCKHEED ELECTRONICS COMPANY, Inc.
New Jersey, 201 757 1600

WEST GERMANY
RC COMPUTER G.m.b.H.
Frankfurt, 611 66 4006

RC COMPUTER
A/S REGNECENTRALEN af 1979

HEADQUARTERS: LAUTRUPBJERG 1 - DK 2750 BALLERUP - DENMARK
Phone: + 45 2 65 80 00 - Cables: rcbalrc - Telex: 35 214 rcbaldk