T E C H N I S C H E   U N I V E R S I T E I T   D E L F T

Faculteit der Elektrotechniek

Aard : Afstudeerverslag
Omvang : 124 bladzijden (deel I)
Datum : oktober 1987

Lab/Afd. : Laboratorium voor AUTOMATISCHE VERKEERSSYSTEMEN

Opdrachtnr. : Go/1986/2

Auteur : R.A.F.J. LIE

Titel : The MINEX's layer 3 specification.

Opdrachtnr. : Go/1986/2

Korte inhoud : Dit verslag beschrijft de MINEX's (Mini Exchange) laag 3 processen en signaleringssamenwerking. MINEX is een ISDN-huiscentrale tussen de ISDN-centrale en de abonnee-eindapparatuur.

T E C H N I S C H E   U N I V E R S I T E I T   D E L F T

FACULTEIT DER ELEKTROTECHNIEK

Vakgroep Telecommunicatie- en Verkeersbegeleidingssystemen
Laboratorium voor AUTOMATISCHE VERKEERSSYSTEMEN
Mekelweg 4 - 2628 CD   Delft

A F S T U D E E R O P D R A C H T

Te verrichten door : R. LIE

De opdracht zal worden uitgevoerd bij APT Hilversum.

Mentoren: ir. J.E.W. Winkelman (APT)
ir. R.A. Beukers (TU Delft)

Onderwerp: Interne verbindingen op de ISDN S-interface.

Omschrijving van de opdracht:

De door de CCITT ten behoeve van ISDN gespecificeerde S-interface biedt de mogelijkheid om maximaal 8 terminals aan te sluiten. Deze terminals kunnen alleen onderling communiceren via verbindingen met de ISDN-centrale, dit in tegenstelling met de mogelijkheden van het huidige telefoonnet, waar sommige abonnee-apparatuur onderlinge communicatie mogelijk maakt zonder tussenkomst van de centrale.
De opdracht omvat het volgende:
Uitgaande van de ISDN Basic Rate Interface specificatie van de 5ESS-PRX, die gebaseerd is op het door CCITT aanbevolen ISDN acces protocol (Q.920, Q.921, Q.930 en Q.931) moet:

* een NT-functie gespecificeerd worden die zowel interne als externe verbindingen tussen terminals mogelijk maakt. In het bijzonder houdt dit in het vaststellen van:
  - de productspecificatie;
  - de systeemarchitectuur (gebaseerd op commercieel beschikbare 'breadboards' en componenten;
  - de signalerings interworking (berichten inhoud en protocollen) en functionele specificatie van de NT software op laag 3;

* zo mogelijk een werkend proefsysteem gebouwd worden uit aanwezige hardware en software componenten en nieuw te ontwikkelen (laag 3) software.

De NT implementatie mag de wijze waarop de S-bus door de openbare centrale geïmplementeerd wordt niet beperken.

Aanvang opdracht:
december 1986

Opdrachtnummer: Go/1986/2
Delft, 9-2-1987
De Hoogleraar,

Prof.ir. H.J. Goebertus

The MINEX's layer 3 specification

R.A.F.J Lie

Delft University of Technology

ABSTRACT

This report specifies the MINEX's (MINi EXchange) layer 3 processes and signal interworking.

The MINEX is a ISDN network terminating equipment, it is situated between the exchange and the user's terminal equipments (e.g digital telephone set).

The layer 3 specification should satisfy the following requirements:

- An internal connection can be established between the terminal equipment which is connected to the MINEX, without the involvement of the exchange.

- the external connection facility between the terminal equipment and the exchange shall not be effected by the introduction of the internal connection.

August
D R A F T

CONTENTS

DRAFT

Samenvatting

Dit rapport beschrijft de specificatie van de MINEX (MINi EXchange) laag 3 processen, en de signalerings samenwerking tussen de MINEX en de centrale enerzijds en tussen de MINEX en de terminals anderzijds.

De MINEX is een ISDN netwerk eindpunt eenheid, die zich tussen de centrale en de gebruikers terminals (bijv. digitale telefoon) bevindt.

De specificatie moet een interne verbinding tussen de terminals die aan de MINEX zijn aangesloten mogelijk maken zonder de centrale te betrekken; en verder met behoud van zijn (de MINEX) externe verbinding faciliteit.

Preface

The subject matter of this report is the MINEX (an implementation of the ISDN functional groups NT1 and NT2) layer 3 (L3) specification, with special regard to the internal connection matter.

This report is divided into five chapters, the first chapter gives an introduction to the MINEX (MINi EXchange) its product specification and its functional requirements.

Chapter two outlines the MINEX system and hardware architectures, considerations concerning the used components, and some implementation alternatives are discussed.

Chapter three describes the software architecture; besides the introduction of the ISDN LAPD's environment, assumptions shall be stated with regards to the MINEX's L3 environment; further preview of the assigned task (i.e MINEX's L3 specifications).

Chapter four describes the MINEX's L3 call control processes (signaling interworking protocol) by means of reference to the appendices (i.e the time sequence diagrams, state diagrams and the SDL diagrams of the L3 processes), this also reflects the phase which has been passed through during the development. Also the necessary message definitions are shown.

Theoretically, according to the SDL diagrams for the MINEX application, it should be possible to implement the MINEX's L3, however the environment of MINEX (i.e management entity and L2) was not available.

To observe that the specified model does not have not-deterministic behaviours, the L3's processes of MINEX (for the internal call control) was implemented in a simulated environment, chapter five describes briefly the used test environment and the test approach.

Along with this report, a summary report has been made; details concerned with the considerations during the design are left out in the summary report, only the main features are outlined.

I would especially like to thank Prof.Ir. H.J Goebertus, Prof.Ir. J.L De Kroes, Ir. J.E Winkelman, Ir. R.A Beukers, Ir. H.J Steltenpool and Ir. P.H Venemans for their support and criticism; also the members of the APT, in particular Ir. G Franken, Ing. J.H Velthof, Ing. J.C Buys and Ing. W.A Romijn for their suggestion and attention to detail. I would also like to thank Drs. R.M De Jong and Mr. A Forsyth who read the draft and gave me their comments.

October 5, 1987
D R A F T

1. *Introduction of the NT12*

ISDN (Integrated Service Digital Network) will offer its services to the customers in the near future. The end-users can have access to this network via old telephone lines. However, network terminated equipment is needed at the customer's end. This physical entity has been referred to as Network Termination (NT) in CCITT I-series.

Two main functional groups are discernable in the NT as recommended in the CCITT's Rec. I.411, i.e NT1 and NT2.

Group NT1, which supports transparent information transfer, performs functions such as:

- line transmission termination,
- interface termination,
- layer1 multiplexing,
- layer1 maintenance function,
- providing operating power etc.

The NT2 functional group which mainly support the information distribution function, may performs functions such as:

- switching,
- data concentration,
- maintenance,
- multiplexing/ protocol handling at data-link and network level etc.

The product (NT) we intend to offer will combine both NT1 and NT2 functions, which will be further referred to in this paper as MINEX (i.e MINi EXchange, since it performs some exchange network functions). The figure below shows us in which environment the MINEX should be placed.

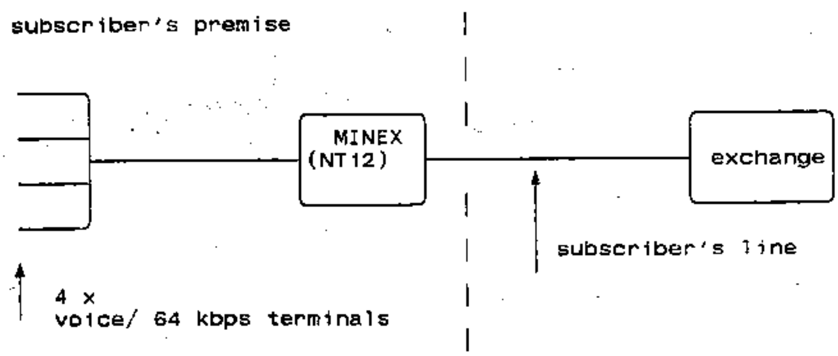


Figure 1. environment of MINEX

1.1 *Product_specification_of_the_MINEX_functions_*

The main goal is to create an in-house network using an MINEX which supports data services as well as voice; therefore the MINEX must have both information transmission termination and distribution characteristics. On the other hand the MINEX is not required to be able to support a great number of terminals (or telephones), because in that case a small PABX is preferred.

The phase 1 MINEX we intend to implement should have at least the following functions :

- Connecting of four terminals on one subscriber's line.
- Supporting of maximal two simultaneous external calls.
- Supporting of maximal two simultaneous internal calls.
- Simultaneously supporting an internal and an external call.

1.2 *Types_of_call_situations*

1.2.1 *External_call_*

With an "external call", we considered a not in house call situation where calling and called party can only be connected through an exchange network (further referred to as ET).

The requirements for the MINEX for an external call are:

1. A external call can be set up from any terminal connected on the MINEX;
2. At most two external calls can be set up simultaneously;
3. Accepting of a call can be done by any terminal which satisfies the required services mentioned in the call-set-up message.

1.2.2 *Internal_call_*

With an "internal call", we refer to an in house call connection. At the moment, an internal call can only be achieved by switching in the ET (since only NT1 is available at the market, but NT1 doesn't perform the switching function).

The minimum requirements for an internal call are:

1. Maximal two internal conversations can take place simultaneously;
2. An internal call can be set up from any terminal connected on the MINEX;
3. It must be possible for a terminal to offer a call to any other terminal.

1.2.3 *Support_of_simultaneous_internal_and_external_call_*

If internal call(s) and external call(s) are present simultaneously, the following combinations are allowed (with some restrictions):

1. one internal call and one external call at the same time;

2. one internal call and two external calls are simultaneous active.

3. During internal conversation, an external call can be offered even if maximum capacity (i.e. two internal calls) has been used.

1.3 *The_functions_required_for_the_MINEX_*

The main task of the MINEX is the distribution/transmission of signals (information,message) between the network exchange (ET) and terminals (TE's); and in case of internal communication of a subscriber: between the user's terminals. To realize this task, MINEX should perform at least the following functions:

1. power transfer / timer function;

2. Line transmission termination (i.e at the U interface);

3. Interface termination user-side (i.e at the S interface).

4. Multiplexing function for layer 1,2,3 .

5. Maintenance function (backup of MINEX in case MINEX fails).

6. Switching function.

7. Protocol handling function for layer 2 and layer 3.

The assignment is to specify the MINEX's functions which are necessary for building up internal and/ or external connections, and the specification of the needed protocol on layer 3 level; this implies that the accent of the work will be:

- Studying the characteristic of the protocol handling functions;

- Realization of the switching function.

The following list gives the items which the protocol handling functions should support:

- The capability to distinguish an internal or an external call.
- To support the set up of a circuit switch path.
- Generation of (simulated network response) messages to the user in case of internal communication.
- Transmitting user/network messages transparent in case of external communication.
- Supporting multipoint communication.
- Link layer protocol handling for the D channel.

1.4 *Environment_of_the_MINEX*

In this paragraph, the following aspects of the MINEX will be described :

1. physical configuration of the MINEX;
2. description of the main functions of the MINEX.

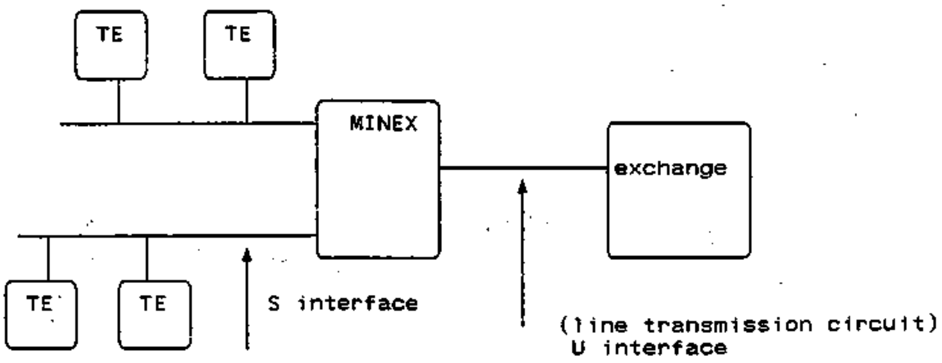
1.4.1 *The_physical_configuration_*

In figure 2, a scheme is drawn to show the environment in which the MINEX is situated; Components of this drawing are the functional block and reference points as mentioned in CCITT recommendation I.411.

In fact this configuration is a variant of the physical configuration mentioned in (I.411 figure 2/I.411 c); MINEX is a combination of NT1 and NT2 in one physical entity.

The user-side interface of the MINEX shall contain two S interfaces; the assumption has been made that on each S interface two Terminal Equipments (TE) are connected. TE's used here are TE1 type as defined in the CCITT Rec. I.411, which is an ISDN terminal that supports the ISDN interface; It could be a digital telephone set or an integrated voice/ data terminal.

On the network side of the MINEX, a line transmission circuit shall be provided (further referred to in this report as U interface, since there is no reference point assigned to transmission line in I-series of the CCITT Recommendations).



TE : Terminal equipment type 1
MINEX : Intelligent network termination
exchange: (ISDN) Network exchange

Figure 2. The physical interfaces of tne MINEX

1.4.2 *The_interfaces_of_the_MINEX_*

As shown in figure 2, MINEX's interfaces can be split into two parts:

1. a TE - MINEX part (two S interfaces);

2. a MINEX - exchange part (U interface).

The interface structure used in both parts shall be the basic rate structure as defined in Recommendation I.412 (i.e 2B+D, wherein B channel is 64 kb/s and D channel 16kb/s). Switching mode shall be circuit-switch on B channel and packet switch on the D channel.

Although it is defined that at most eight TE's can be connected to each S interface, for phase 1 MINEX we have made the restriction to have only two voice/ 64 kb/s TE's on each S interface. This decision was made to avoid conflict situations which could occur due to simultaneously activating three terminals on a S interface.

On the TE - MINEX part, protocols for internal and external call shall be provided. Protocol handlings which manage terminal key related actions (for extended features such as call hold, and call transfer etc.) are also assumed to be present.

On the MINEX - exchange part, only protocol for external communication shall be provided.

1.5 *Restrictions_&_assumptions_*

As the physical configuration of the MINEX has been introduced in the previous paragraph, some assumptions are required for the phase 1 MINEX, also some restrictions concerning the physical limitations can be derived.

The CCITT has developed an architectural reference model, which describes ISDN interface functional groupings and location. Figure 3 is a simplified representation of this model, it shows a numbers of reference interface location (i.e R, S ,T), and the functional groups (i.e NT1, NT2, TE1, TE2, TA).[1]

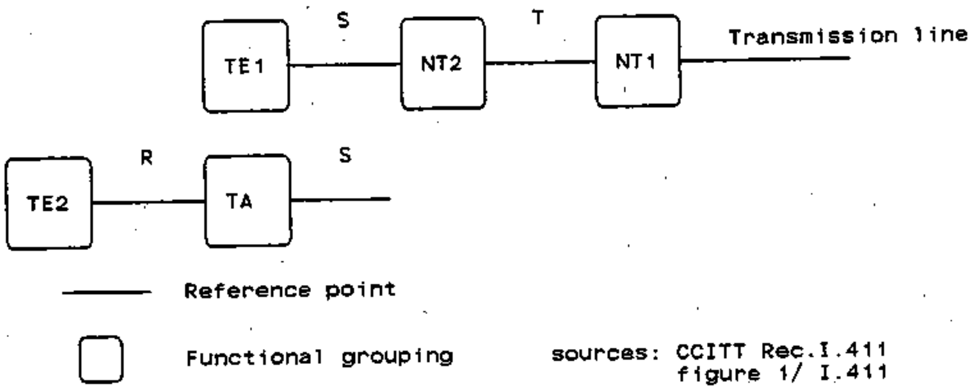
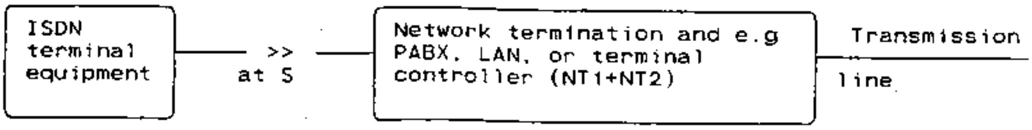


Figure 3. Reference configuration for the ISDN user-network interface

If we compare figure 2 with figure 3, the conclusion can be made that the physical interface of the MINEX occurs at point S of the reference model ; the other physical interface is the U interface which does not correspondent to a reference point.

---

1. The definition for these functional groups can be found in the definition attachment appendix.



An implementation where an ISDN physical interface occurs at reference point S but not T.

sources: CCITT Rec.I.411
figure 3b/ I.411

Figure 4. Example of implementation of NT1 and NT2 functions

1.5.1 *The_assumptions_*

What kinds of interface structure (i.e information-carrying capacity) should be defined for these U and S interfaces ?

According to Rec. I.412 basic rate interface structure is composed of two B channels[2] and a D channel.[3]

- For the phase 1 MINEX, the assumption has been made that both S interface and the U interface shall use the basic rate interface structure.

From figure 3 one could see that via the S interface two types of TE's can be connected; TE1 which supports the ISDN interface, and TE2 that requires a TA.[4]

- For the concerned MINEX application, the assumption is made that TE will be type TE1, so no TA will be required.

Concerning the transport mechanism (i.e circuit and packet switching); B channel supports services as digital voice and circuit or packet switched data, while D channel primary function is to carry signaling information for the two B channels.

- For this specification, we assumed that digital voice (+ circuit switch data) will be supported in the B channel, and the out band signaling information in the D channel.

---

2. B channel: 64 kbps channel, carry user information streams.

3. D channel: 16 kbps channel in basic rate interface structure, and 64 kbps channel in primary rate interface structure. A D channel carry signaling information for circuit switching by the ISDN.

4. TA: Terminal Adaptor

- For internal call numbering, either abbreviated numbers will be considered or use of keypad information element.[5]

1.5.2 *The_restrictions_*

Generally, each S interface can have at most 2 simultaneous active terminals at any instant, therefore the restriction is made to install only two voice/64 kbps terminals on each S interface.

Regarding the types of calls (i.e internal and/ or external), some restrictions can be stated.

1.5.2.1 *Restrictions_on_the_external_call_*

- In case two external calls are spread over both S buses (S1, and S2), MINEX can not establish an third external call connection any more (since U interface's channels are completely utilized).

- At the call terminated side (i.e terminated network-user interface), ET's setup for a call uses the broadcasting facility at the data link layer (as defined in Rec I.451); if destination is a specific TE, SETUP should also contain appropriate sub-address or DDI or a TE's identifier.

1.5.2.2 *Restrictions_on_the_internal_call_*

- An internal call is also private.

- An internal call should be offer to a specific TE, and not to a group of TE's.

- As in "External call", if more than one call exists on a MINEX, only two active terminals are allowed in each S interface at any moment.

1.5.2.3 *Restriction_on_simultaneous_internal/_external_calls*

- In case a TE (e.g ter1) is having an internal call, and an (incoming) external call also attempts to communicate with this ter1, then a warning signal should be given.

- If a TE (e.g ter1) is having an internal call, while another TE (e.g ter2) attempts to setup an internal call with ter1, a call reject message (REL_CPL) should be send to this ter2.

---

5. If TID (Terminal IDentifier) is available, selective information element is used (there are options for a broadcast or a selected call).

- If two external and one internal calls are active, no other calls are allowed (since there are no b-channels available anymore).

1.6 *Future_enhancements_of_the_MINEX_*

The first phase MINEX layer3 specification is mainly concerned with the basic call connection function; for the future MINEX's functions extension, some ideas can now be given to take into account in the future.

- The restriction of two TE's per S interface should be dropped, since according to the CCITT Recommendation, a S interface can be connected with at most eight TE's.

- Supporting external conference conversation (i.e after an external call connection has been established, a TE can join each of the parties). An external conference conversation should be possible with users on either the same S interface or on one of each of the S interfaces.

- An internal conference conversation should also be possible. (i.e. a third TE on the same user-network interface may join an internal conversation after an explicit handling by one of the original parties).

- One internal call (with conference) and one external call could be also one of MINEX's supporting features (MINEX's available channels fully utilized).

- Packet switch on D channel.

- Possibility of connecting the MINEX's in cascade (forming a small net work).

1.6.1 *Specific_future_functions_enhancement_on_external_call*

Some functions that are convenient and/ or necessary for an external call are for example:

- Supporting of "call transfer" facility.

- Supporting of "call hold", and "call forward" facilities.

- During an external conversation, the terminal should be able to put the call in a "hold state", and one of the following handlings can be performed:

  - After an internal consultation (case without involving other extension) has taken place, external conversation can be continued.
  - Internal conversation with another extension first; then, either the original party or the latter extension could continue the external conversation (i.e call transfer).

1.6.2 *Specific_future_functions_enhancement_on_internal_call*

An extra function that could support internal call in the future is for example:

- Supporting of "call hold" facilities (i.e "hold" for internal call).

2. *The_architecture_of_the_MINEX_*

After the (functions and environment) external behaviour of MINEX has been determined, we can now try to define the architecture of the MINEX. This chapter is divided into five paragraphs, the first one outline the system architecture of the prototype MINEX.

The second paragraph describes briefly the decision that was made concerning the hardware architecture.

In the third paragraph, we take a closer look at the components which will be involved in our implementation.

The forth paragraph sums up the problems we have to take into account, but the possible solutions will also be stated.

The fifth paragraph describes the final MINEX hardware architecture to be used.

2.1 *The_MINEX's_system_architecture_*

The prototype MINEX hardware can be divided into the following modules:

- switching module;
- user interface module;
- exchange interface module;
- link layer module (perform LAPD layer2, separation of the B and D channels of the basic rate interface).
- processor module (for the hardware control, link with the software).

The Software for the prototype MINEX contains:

- protocol handling function (layer3; main point of this report).
- generation of hardware control message (to switching unit).

The hardware and software modules will be described in more details in other sub-paragraph or chapter; In this sub-paragraph the cooperation between the hardware and the software (concerning the establishment of a call connection) will be outlined.

- Generally, MINEX receives signaling messages from either the exchange (further referred to as ET) or the user (further referred to as TE) involving respectively, the exchange interface module the user interface module;
- since these are out of band signaling messages (they appear in the D channel), it will be handled by the link layer module,

and separated from the B channels; an interrupt will be generated to a microprocessor (further referred to as uP) 8051.

- After uP has been interrupted, this D channel signaling information (L2 frame) will be read by the uP and processed by the protocol handling function coded in the ROM (belongs to the processor module).
- The respond messages which are generated by the uP are layer3 messages; to be able to transfer on a data link, layer2 address information is provided by the processor module. The link layer module will pack these layer3 messages and layer2 addresses in an L2 frame, provide with header, tail and other link control elements.
- This message is now sent to either the user interface module or the exchange interface module and to be sent to the TE or the ET.
- Depending upon the decision of the uP to perform the switching function, switch control information shall be generated by the uP and sent to the switching module.

2.2 *Considerations of the hardware architecture*

The task is to develop/implement the layer 3 protocol of the D channel signaling in the MINEX, which manages the interworking between the TE and MINEX, and also between the MINEX and the exchange (especially that part which realized the internal call). Verification of this protocol shall be done by testing the MINEX unit.

Because the MINEX unit does not exist yet, there are two possibilities which can satisfy the testing purpose:

- One can design a MINEX, including hardware, new operating system, and new software.
- With a breadboard offered by some IC′ firm, which contains ISDN oriented ICs and facilities for different configurations of the connection between subscribers and exchange. One can modify the main board to create a MINEX-like system.

The first possibility will take too much time, especially if one considers that layer 1 and layer 2 will have to be developed. For the second alternative, some adaptations are inevitable.

As this task has to be completed in a limited time, the second choice has been made. In consultation with the hardware department, two sets of ISDN userboard-kits have been considered:

1. Intel ISDN userboard-kit (contains LEK[6] and TEK[7]).

2. Siemens ISDN userboard-kit.

We have chosen the Siemens version, because it provides a more complete layer 2 (LAPD[8]) support.

2.2.1 *The_ISDN_parts*

With the Siemens userboard-kit and ISDN-ICs, an environment can be created, where ISDN-acces for various configurations between user and exchange is possible. The ISDN userboard-kit consists of the following types of boards:

- the main board.
- the layer 1 module.
- the terminal equipment board.

With these boards an ISDN-system can be set up, an example of a PABX-system including extensions was given in the manual [Siemens ISDN Userboard-kit STU 2000]. The ISDN-userboard-kit concept is shown on the next page.

According to the manual, the following functions are provided by these boards:

- The main board can perform the group control and/ or line control functions.
- Three different module-cards are designed for the various applications. these layer 1 module-cards are:

  -SBC-module for a PABX-system without U interface.
  -IBC-module as U interface, using burst procedure.
  -IBC-module as U interface, using the echo canceling procedure realizing a central office system.

  Depending on the layer 1 module, this userboard-kit can act either as a PABX system or as a local exchange network.

- The terminal equipment board is connected to the main board via the S-bus. Voice and data communication between the

---

6. LEK: Line card Evaluation Kit.

7. TEK: Terminal Evaluation Kit.

8. LAPD: Link Access Protocol D-channel.

terminal and the switching system is now possible (because conversion of analog- digital signals, correct filtering/sampling, and connection of the board to the S bus etc. has taken place).

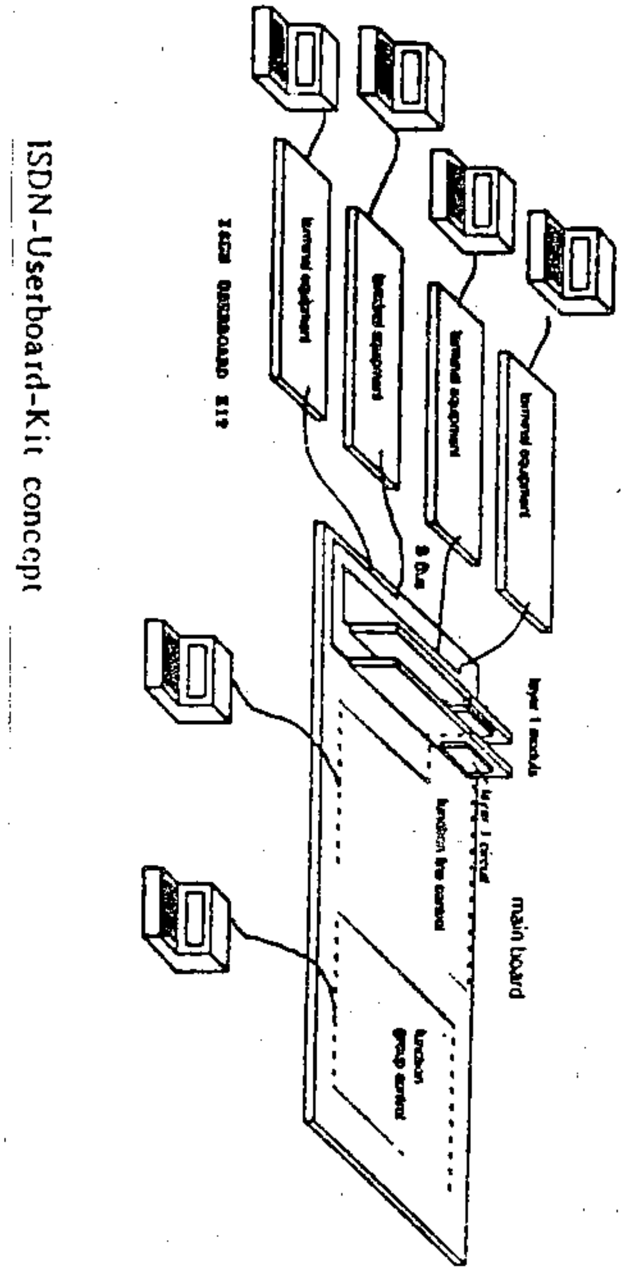
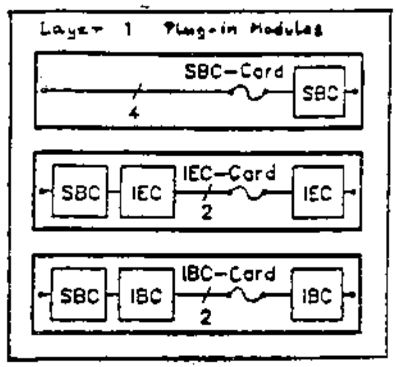


Figure 5. userkit figure.

Figures in this page shows the three module-cards mentioned in the previous list and the terminal equipment board. The term SBC[9] ,IEC[10] , IBC[11] and SICOFI[12] refers to the ICs used in these boards.



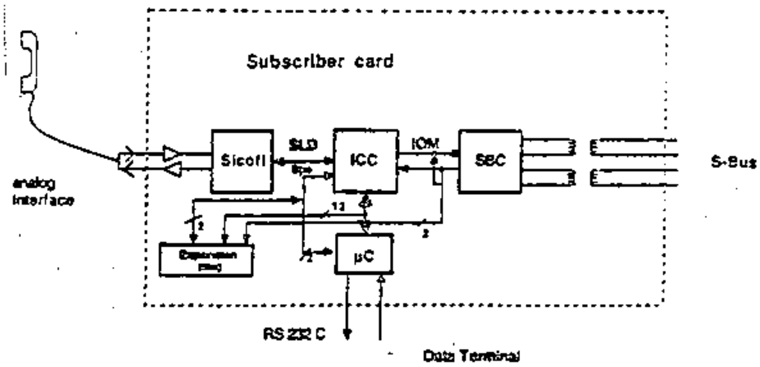


Figure Terminal equipment board

Figure 6. figure layer1 module

The main board (see the figure in the next page) can be discerned in two functional blocks and two interface-slots. In the figure of the main board we can see:

- Group controller part.
- Line card part.

---

9. SBC: S-Bus interface Circuit.

10. IEC: ISDN-Echo Cancellation Circuit.

11. IBC: ISDN Burst controller.

12. SICOFI: Signal processing Codec Filter.

- Layer 1 module connection slots part.

The main components used in line/ group control parts are HSCC[13], PBC[14], and ICC[15]. Another important component which is not shown in the figure but already mentioned in the layer 1 module is the SBC[16] IC.

The functional characteristics of all these components will be described in the next few subparagraphs.

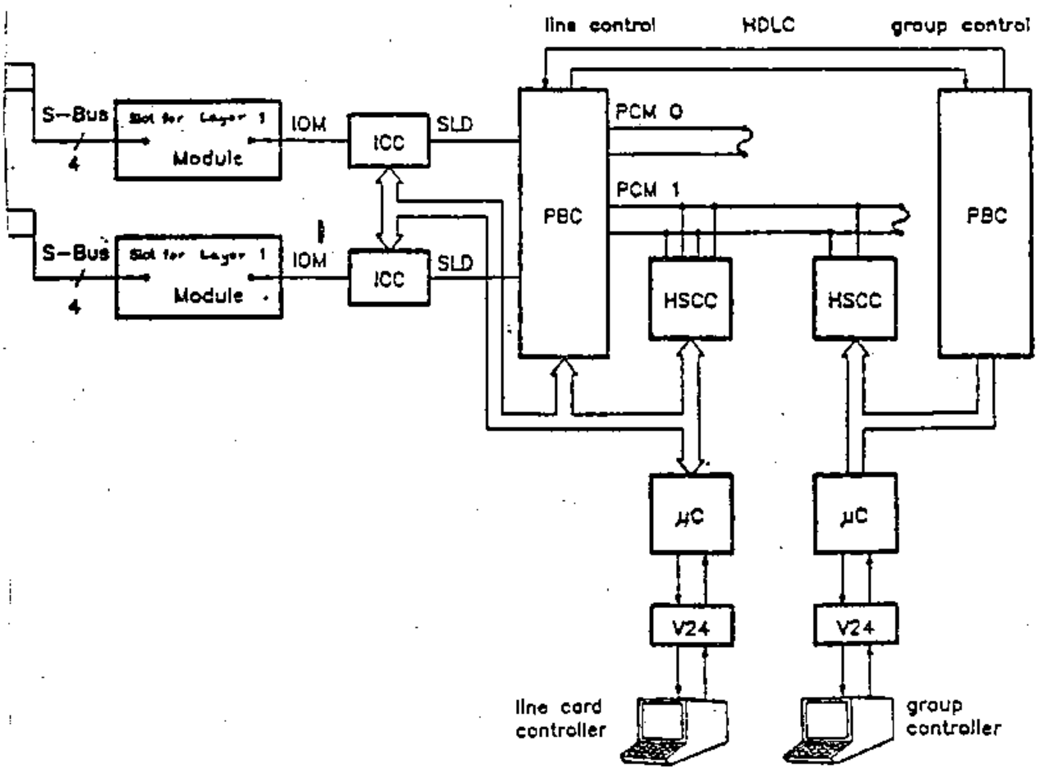


Figure 7. figure of the main board

---

13. HSCC: High-level Serial Communication Controller.

14. PBC: Peripheral Board Controller.

15. ICC: ISDN communication controller.

16. SBC: S-Bus interface Circuit.

2.2.2 *General_description_of_used_ICs*

ICs used in the userboard-kit are: PBC, ICC, SBC, SICOFI, HSCC, IEC, and IBC (reference to these abbreviations can be found in previous paragraph). For PBC, ICC, and SBC a brief functional description will be given, IEC's datasheet and IBC are not available , while HSCC and SICOFI are not used in our planned MINEX.

1. PBC:

   - Control space and time switching functions between terminals and time division multiplex highway.
   - Control the flow of information between the subscriber interface port and a processor (e.g. a line card processor).
   - Perform all protocol control functions, using HDLC protocol format for the information passing between the line card and the central processor (of switching network) via a HDLC line or via interleaved time slots on the PCM lines.
   - Time slot assignment freely programmable for all subscribers (up to 16) connected.
   - Provides two full duplex PCM highways for the system interface.
   - Standard microprocessor interface.
   - Two DMA channels for expansion of internal buffer capacity (16 bytes per direction).
   - Microprocessor access to all internal data stream (including time-slot oriented data streams).
   - Interfaces:
     - 8 serial, bidirectional I/O ports for transfer of data, voice and signaling information (also known as Subscriber Interface Port, i.e SIP).
     - Double constructed PCM (2 Mb/s) interface.
     - Bit-parallel interface for the connection of 8 bit standard microcomputer.
     - Fast serial communication link to the central processor.

2. ICC:

- -Act as the D-channel link access protocol controller. (support for LAPD protocol). ICC performs the protocol functions for the D channel and transfers the B-channels to the line side.

- -Buffer memories for the temporary storage of data packets (64 bytes per direction). Transfer of data packets.

- -Switching of the B channels.

- -Access to the 64kbit channels.

- -Control of layer 1 component.

- -Built-up of the ISDN frame (2B+D).

- -Interfaces:

  - -SPb (serial port b): serves IOM (ISDN Oriented Modular) and/ or HDLC interface.

  - -SPa (serial port a) and SIP (SLD interface port): interface for the B-channels.

  - -parallel processor interface.

3. SBC:

- -Providing layer 1 function with a 4 wire transmission for digital signals at a data rate of 192 kbps.

- -Transmitting and receiving of signals from the S interface according to CCITT I.430.

- -Activation and deactivation procedure according to CCITT I.430.

- -Frame alignment in the TDM application mode.

- -D channel acces control.

- -SBC can be programmed for the following application: TE (i.e terminal equipment), NT (i.e network terminator), TMD (i.e trunk module, connection of a PABX to the public exchange), and SLMD (i.e subscriber line module).

- -Recovery of clock and frame signal adjusted to the particular application mode.

2.3 *Problems_expected_and_possible_remedies_*

After studying the possibility of (utilizing) modifying the userkit breadboard such that it functions like the specified MINEX, some questions concerning the implementation have arisen. The next few chapters consider these subjects and in case the (possible) answers have been found for the problems, the solutions and/ or solution's arguments will be given.

2.3.1 *The_U_interface_problem_*

As stated in the previous paragraph about the MINEX's system configuration, the U interface is the network-side interface of the MINEX; it provides the line transmission circuit (function) over the subscriber loop. In the ISDN userkit manual, mention has been made that one of the layer 1 modules (i.e. either IBC-module or IEC-module) can function as a U-interface, But since these two chips (IBC and IEC) are not available, we will have to look for some other alternative for this part of interface.

Our main problem is: how to realize the layer 1 function at the network-side of the MINEX.

The problem with the U-interface was solved by using the other available layer 1 module (i.e. SBC module). The SBC IC has as main function: providing the layer 1 function. Although this module involves a 4 wire transmission instead of a 2 wire (as in U-interface), functionally there is no difference with regard to the information transmission.

Using SBC-module as U-interface equivalent (as regards to the layer 1 function), information can be exchanged between the MINEX and the assuming network exchange side, therefore our problem concerning the U-interface seems to be solved.

2.3.2 *The_PBC_as_switching_unit_?_*

In the previous chapter the PBC's functions have been described. The main function of the PBC can be characterized as a data-path controller for PCM, control and signaling data. In applications where switching function is desired, PBC can achieve this function in combination with another device (e.g. a time switch).

According to the specification of the MINEX, switching function is essential for the functioning of the MINEX; how should this function be implemented ? With a time switch or is there some other alternative?

After a closer look, specially to the PBC's abilities of time-slot related actions, we noticed the following feature of the PBC:

- Time slot assignment programmable for all subscribers connected to the PBC (maximum 16).

- There are two (other version four) full duplex PCM highways provided.

Keeping these facts in mind, let first take a look for an initial outline of a possible implementation of the MINEX, using the Siemens ISDN breadboard (see figure).

Use of the U-interface in figure is only possible if conversion (PCM - U) has taken place between the switching device (i.e time switch) and the U-interface.

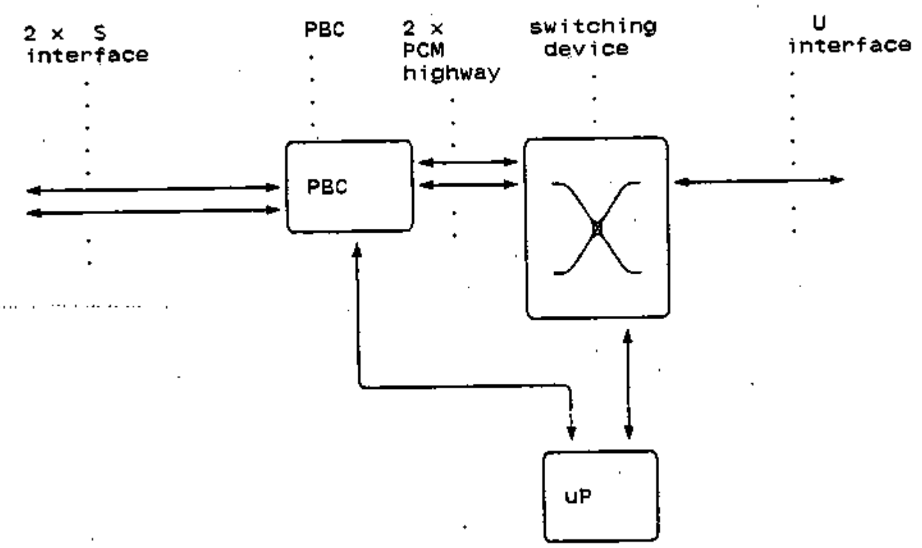


Figure 8. Possible implementation of MINEX

The switching device shown in the figure is not necessary however, if we consider the following facts:

- U-interface is supposed to provide the 2B+D channels (i.e. with transmission capacity of 2*64 + 1*16 kbps).

- Each S interface provides also 2B+D channels.

- The PBC can control up to 16 subscribers lines, this implies a channel capacity of (2 x 16 x 64 kbps) 2 Mbps at the PCM highways.

- Since there will be capacity left in case the PBC is placed in a configuration which performs the MINEX's function, and because the PBC also have time-slot assignment function, we can use the PBC as a switching unit (see the modified figure).

- Concerning the PBC configuration:

  one of the PCM high ways is looped;

  of the eight available SIP lines: two are used to connect the S interfaces, and one for the connection with the U interface.

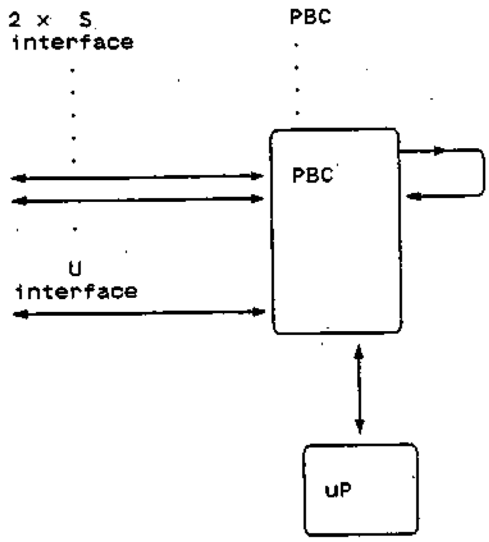


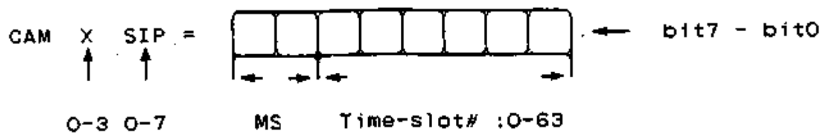
Figure 9. Alternative implementation of MINEX

2.3.3 *Argumentation_for_the_PBC_as_switching_unit_*

At first, there were some doubts as to whether the PBC can function like a switch or not; After studying the data sheet, we concluded that it is possible to let the PBC act as a switching unit. A detailed explanation of the time-slot assignment of the PBC is given in this sub-paragraph.

To prove that the PBC can support the switching function, and especially with regard to the MINEX application, we have to show that PBC is capable of setting up a call (by means of time- slot assignment) either on the same S interface (i.e. SIP in Siemens term), or between different S interfaces.

At first, let us take a look at the command used in the PBC for setting up a channel:



MS	
00	uP transfer
01	transfer highway 1
10	transfer highway 0
11	no transfer

Figure 10. CAM command format

The abbreviation CAM stands for Contents Addressable Memory, this memory in PBC is divided in four blocks (i.e. x: 0-3), each of these blocks consists of eight bytes (correspondent with a SIP: 0-7).

Each of these blocks implies the characteristics of the time-slots (transmit or receiving) and which channel is involved in the operation. The following table shows the concerning time-slot specification:

TS: time-slot

CAM 0	receiving TS, chan A
CAM 1	receiving TS, chan B
CAM 2	transmit. TS, chan A
CAM 3	transmit. TS, chan B

Figure 11. Relation CAM blocks and items specified

In combination with the SIP lines (Subscriber interface port), the physical interpretation is as follows:

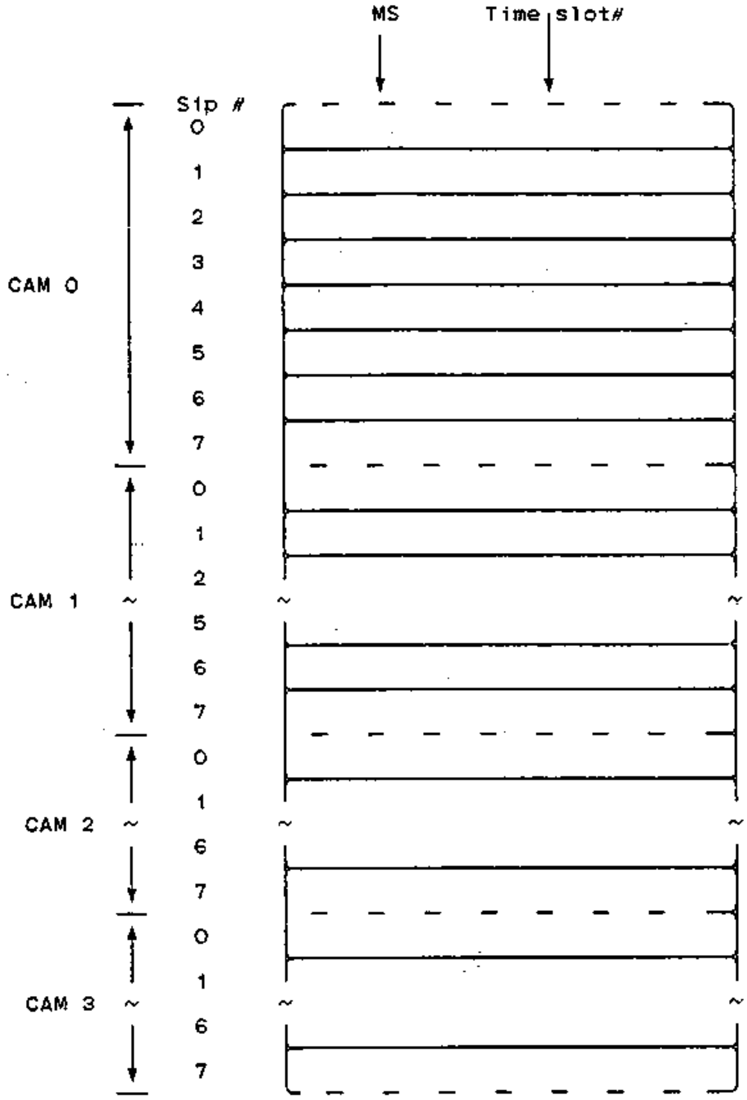


Figure 12. Physical relation of CAM, SIP, and TS#

After this brief introduction, we now examine how a call connection set up take place. Assume one of the highways (HW) is looped.

First, we investigate the case: call on the same S interface (i.e both TE on the same SIP, see figure).

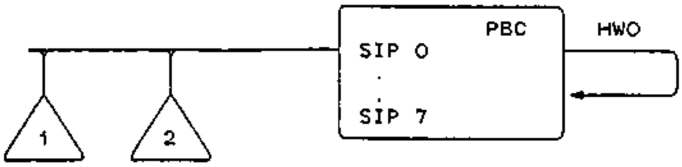


Figure 13. Call on the same S interface (example 1)

Supposed TE 1 uses channel A (TS=3), and TE 2 uses channel B (TS=7) for the communication, and suppose the looping highway is HW0, then we can draw up the following table.

In this table all information which determined the call set up can be found.

CAM X	SIP#	HW	TS#
2	0	0	3
1	0	0	3
3	0	0	7
0	0	0	7

Figure 14. Connection information for example 1 (same S Bus)

Next, we shall investigate the case in which a call occurred and two S interfaces are involved (i.e TE 1 on SIP 0 and TE 4 on SIP 1).

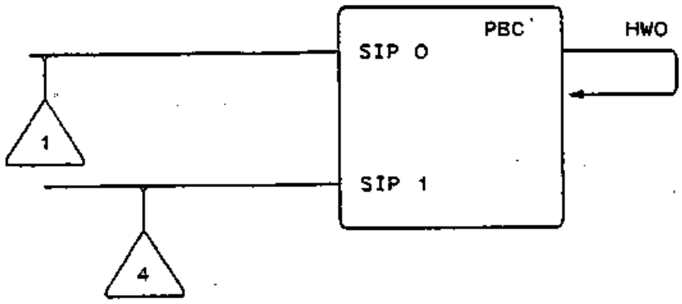


Figure 15. Call using two S interfaces (example 2)

Supposed TE 1 uses channel A (SIP 0, TS=3), and TE 4 also uses channel A (on SIP 1, TS=7) for the communication, and suppose the looping highway is HW0.

The following table provides all information that the PBC needs to set up a call.

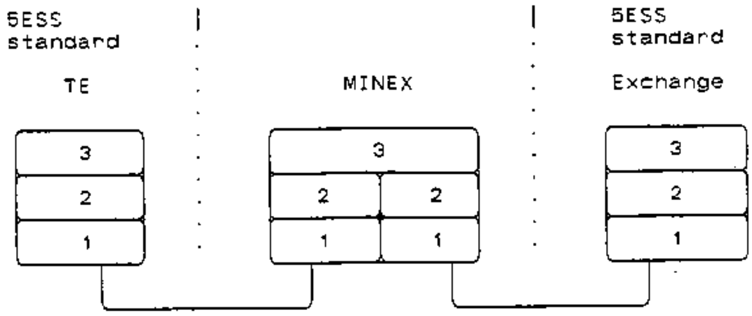
CAM X.	SIP#	HW	TS#
2	0	0	3
0	1	0	3
2	1	0	7
0	0	0	7

Figure 16. Connection information for call using two S interfaces (example 2)

In fact, the information shown in these two tables implied that the PBC is capable of setting up (switch) communication links which proved that the PBC can function as a switching unit properly.

2.3.4 *Compatibility_of_used_units_*

Another possible problem came up related to the use of devices of different companies. A scheme shall provide more clearness (see figure 17).



	layer 1	layer 2	layer 3
TE	I.430	I.441	I.451
EXCHANGE	I.430	I.441	I.451
MINEX	I.430	I.441	Task

Figure 17. Interworking environment of MINEX (layer 1,2,3)

In this figure we see that at the exchange side and at the terminal equipment (TE) side the 5ESS protocol is being used, while for the layer1 and layer2 of the MINEX, the "other protocol" (Siemens) is supposed to be used, since Siemens components are used in the MINEX.

However, the assumption is made that the protocols on layer 1/ 2 level are implemented according to CCITT I.430/ 440 recommendation for both 5ESS and Siemens products, which normally results in compatibility of these products.

In case the used protocols create conflict situations, adaptations shall have to be made.

2.4 *The_hardware_architecture_of_the_MINEX_*

After the preparation works described in previous paragraphs have been done, and some consultations were made with the hardware department, a preliminary outline of the hardware architecture is determined.

The figure in next page shows the general block diagram.

2.4.1 *The_block_diagram_of_the_MINEX's_HW_architecture_*

The Network Termination as shown in the figure within the dotted line, is composed of five functional blocks. These functional blocks are:

- S block: Providing the S bus interface function as recommended in the CCITT.
- U block: Which will provide the transmission function over the subscriber loop.
- LAPD/ Channel separator block: This block provides the basic LAPD layer2 function, and it separates the B channels from the D channel.
- Switching block: Performs the time switching function.
- uP block: Handles the LAPD layer3 protocol, the control of the switching block.

The U block can not be realized due To the fact that the IC for this purpose is not available yet; however, we shall simulate the information transfer at this block with other available components (i.e. SBC combine with ICC which will be shown in next next sub-paragraph).

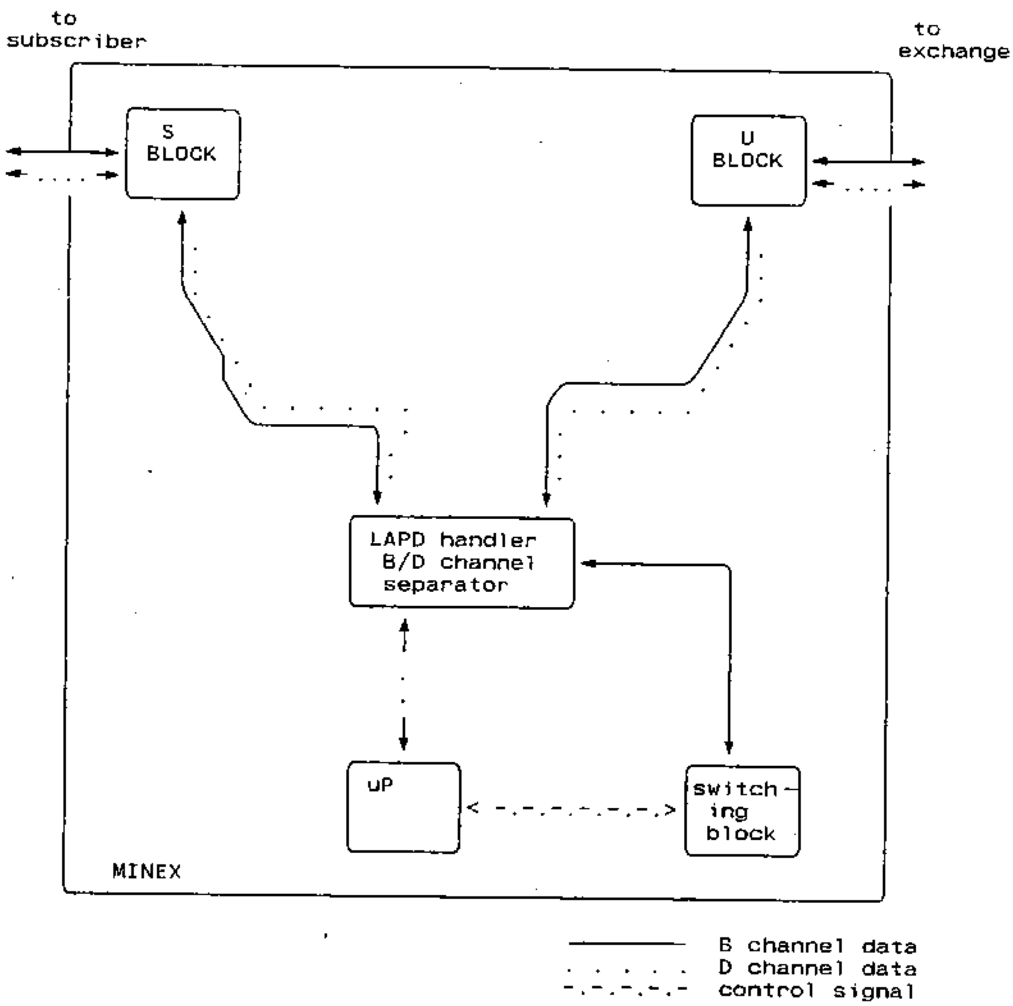


Figure 18. Block diagram of the MINEX

2.4.2 *The_block_diagram_on_IC_component_level*

After the functional blocks were discussed; choosing the components is the next phase. Since a Siemens breadboard is going to be the base, mostly those IC's which are elements of this userkit breadboard are used.

The figure in the next page shows a block diagram on IC component level.

Notice that at the U-interface part SBC is also being used, just as with the S interface part.

Also notice that on one of the two PCM highways on the PBC, a loop has been made.

Some information of the used components:

uP: Intel 8051.
ROM: 16 K.
RAM: 8 K.

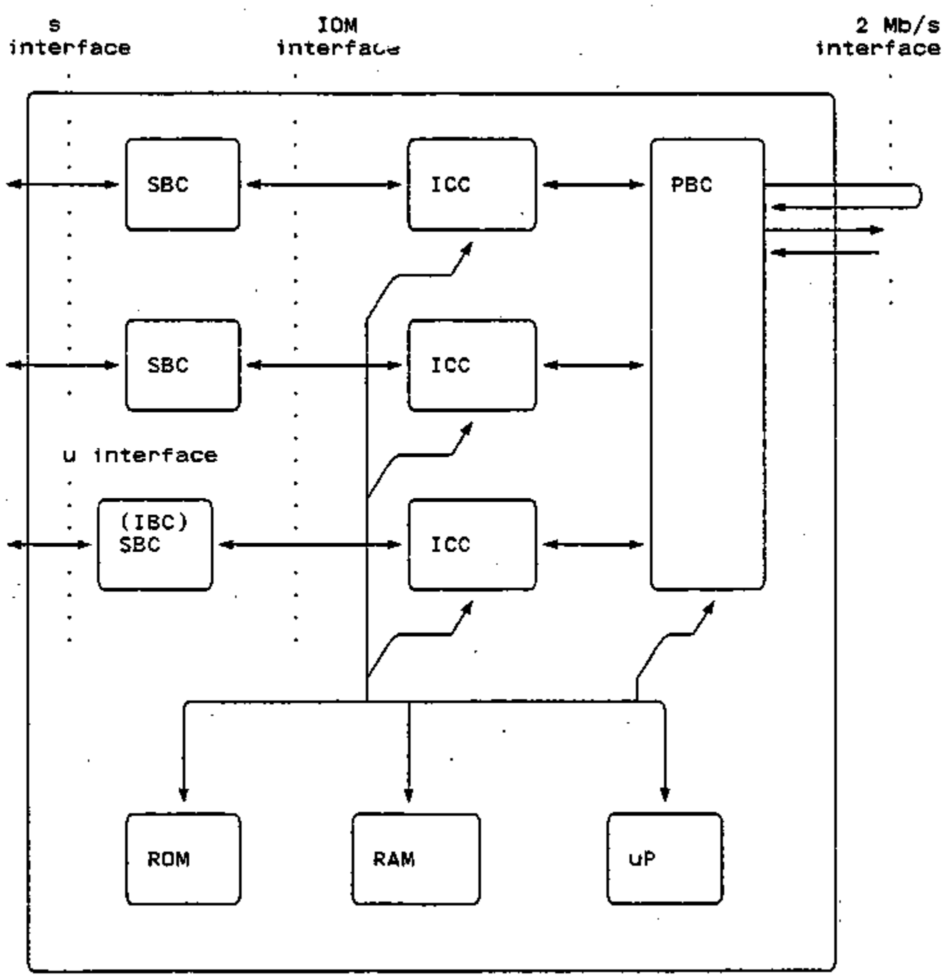


Figure 19. Detailed block diagram of MINEX

3. *The_software_architecture_*

In the previous chapter the global hardware architecture of the MINEX has been shown. The software architecture is the main object of this chapter and will be considered in more detail in the following paragraphs and subparagraphs.

This chapter is divided in four paragraphs, the first paragraph introduces the used referenced model and terminologies;

the second paragraph: assumption of the initial condition , and considerations concerning some implementation alternatives;

The third paragraph outlines the call handling approach within the network layer; including signal, call coordination function, relation to DL layer and management layer;

in paragraph four: the signaling lists and parameters used in the MINEX's L3 are stated.

3.1 *The_ISDN_protocol_reference_model_*

This paragraph is divided in two sub paragraphs:

- the first sub-paragraph gives a brief introduction to the CCITT's ISDN protocol reference model and it's relation to the 7 layers model (Rec.X.200) of the OSI, and the introduction of the management block.
- in the second sub-paragraph terminologies of the LAPD environment are given.

3.1.1 *Reference_model_*

As stated in Rec. I.320 concerning the concept for protocol modeling, the OSI model (seven layers, Rec. X.200) was formed for a datacommunication purpose, while the ISDN is supporting different kinds of communications (including data, voice and video application).

The following list give some examples of the ISDN needs which have not been taken into account in the OSI layers model:

- information flow for out-band call control process;
- information flow for multipoint connection;
- information flow for selection of connection characteristics;
- etc. (for more example see Rec. I.320).

The OSI model therefore needs to be "extended" in order to cover these ISDN specific features.

The information flows of previous list have been classified into two categories:

- user information - voice, data and information transmitted between the user.

- control information - e.g information for controlling a network connection.

A "protocol block" is used to described elements in the ISDN user premises and the network (e.g ET, TE, ET etc.).

In figure 20 a representation of a protocol block is given. Such a block contains:

1. user information and associated layered protocol;

2. control information and associated layered protocol;

3. local management aspects associated with the transfer of user and control informations.

The management aspects as mentioned in the previous list concerned purely local matter associated with a given entity, for both a ISDN TE and a network equipment (in our case: the MINEX).

Some of these local matters are for example:

- control of switching action within circuit switch ;

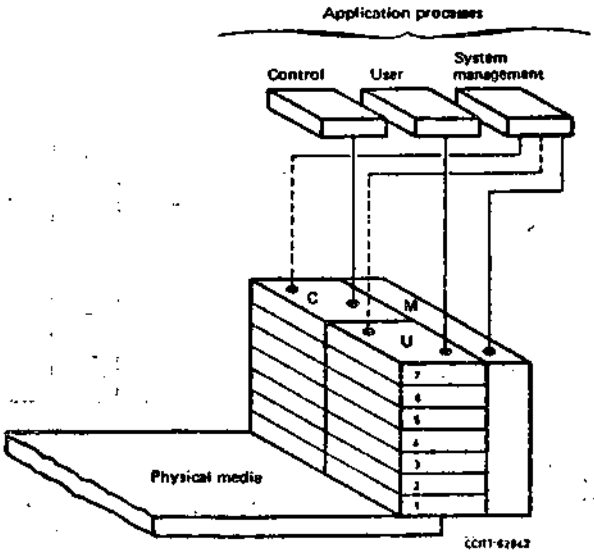
- selection of appropriate responses to connection failure;

- network management;

- traffic control (to optimize utilization of "network" resources).

In MINEX application, management functions are required for example for the circuit connection purpose.

Figure 21 is an attempt to draw a protocol block for MINEX (for application where circuit switch connection via B channel is taken into account).



Note — Peer-to-peer protocols associated with U and C are not shown.

source: CCITT Rec.I.320
figure 2/I.320

Figure 20. Interaction associated with a protocol block

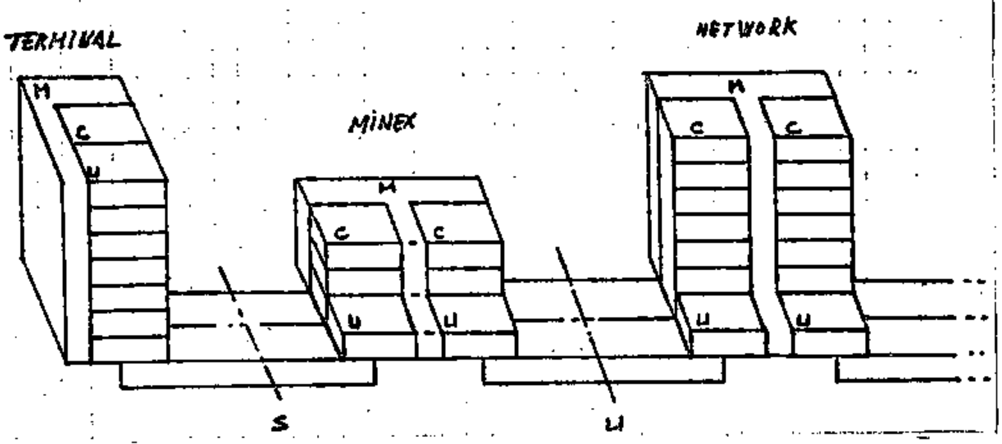


Figure 21. Interaction of MINEX's protocol block (for circuit-switched connection via B channel)

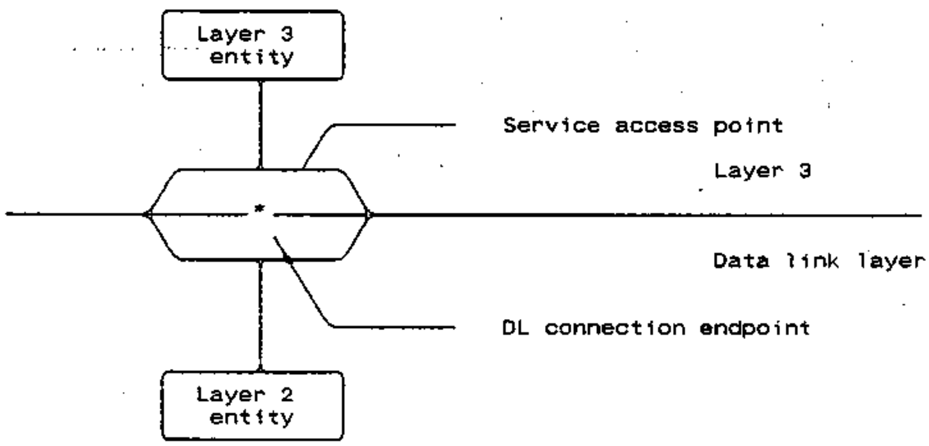
3.1.2 *A_LAPD's_environment*

In our implementation, we only considered the basic rate interface structure (i.e 2B+D channels). The access protocol on the D channel (i.e The channel which is mainly concerned with the transfer of signaling information to establish, to maintain, or to disconnect the communication link.)has been defined in the CCITT's Recommendations of the series I (I.430 - I.451).

LAPD (Link Access Protocol D-channel) are OSI oriented, the layer2 (further refer to as L2) protocol for signaling and packet data are the same (I.441), the layer3 (further refer to as L3) protocol for signaling is I.451, while L3 protocol for packet data is the packet level of X.25.

3.1.2.1 *Introduction*

Terms which are used in the L3 and L2 environment will be introduced by means of the figure 22. One can find more detailed explanation in Rec. I.440.



SOURCE: CCITT REC I.440
FIGURE 2/ I.440

Figure 22. Entities, service access points and endpoints

3.1.2.1.1 *The SAPI and TEI*

The information exchange between L3 and L2 takes place via "Service Access point", which can be identified by SAPI (Service Access Point Identifier). There are SAPIs defined for signaling data, packet data and management information.

The information exchange between two L3 entities is realized by a data link (further refer to as DL) connection on L2. Such a data link connection is identified by DLCI (Data Link Connection Identifier), which contains a SAPI and the TEI.

The TEI values are basically assigned by the network (assumption is made that the terminal is of the automatic category). However, the TEI value must be unique on the interface (i.e user-network), otherwise identification of a specific connection endpoint within a service access point will not be possible.

Although at least one TEI is used by the TE (i.e a DL layer entity which has been assigned a TEI value could use that value for all SAPs it supports); here the restriction of one TEI value per TE is assumed, since a voice communication application is involved.

Corresponding with this DLCI (on L2), we have the CEI (Connection Endpoint Identifier) on L3. CEI consists of SAPI and CES (Connection Endpoint Suffix). The CES value is provided by the layer management entity (At least in the AT & T 5ESS version). Procedures of TEI assignment and related matters will be discussed later.

The relations of some of these terms in DL layer are in summary:

A main characteristic of LAPD is that it provides a multiplex function that enables a number of network entities to operate over a single physical access.

Each of the link procedure acts independently of the others, the multiplex procedure distributes the DL channels according to the address information in the frame (i.e DLCI), each link is associated with a SAP, and within an DL procedure a specific endpoint TEI is further selected.

3.1.2.1.2 *The primitives*

Communication between different layers in the OSI reference model makes use of primitives. Via primitives information can be exchanged between the adjacent layers.

Primitives associated with the Data Link layer can be found in Rec. I.441 (pag. 215); the following list will only give a slight impression. Notice that M - L2 stands for the management entity to the Data Link layer interface.

	primitive name	message
L3 - L2	DL-ESTABLISH REQ/ IND	
	DL-RELEASE REQ/ IND	
	DL-DATA	L3 peer to peer message
	DL-UNIT DATA	,,
L2 - L1	PH-DATA REQ/ IND	L2 peer to peer message
	PH-ACTIVATE REQ/ IND	
	PH-DEACTIVATE REQ/IND	
M - L2	MDL-ASSIGN REQ/ IND	TEI value
	MDL-REMOVE REQ	,,
	MDL-ERROR IND/ RES	reason for error message
	MDL-UNIT DATA	management func. peer to peer message

Primitives associated with the DL layer

3.1.2.2 *The_Layer_2_frame_structure*

Having introduced the above terms, we now take a closer look at the L2 and L3 frame structure.

	flag	address	control	information	FCS	flag
octet nr.	1	2, 3	4		n-2,n-1	n

Figure 23. layer2 frame structure

The L2 frame (figure 23) contains the following elements:

- Flag: Mark the beginning of the frame. Bit combination is "01111110".
- Address: This address field contains the DLCI (Data Link Connection Identifier), DLCI is composed of the SAPI and the TEI value.
- Control: This field contains the sequence number of the frame.
- Information: The layer 3 message
- FCS: Frame check sequence, a number that performs the frame's error detection function.
- Flag: Mark the end of the frame, bit combination is the same as the begin flag.

Having introduced the relation between the SAPI and the TEI ., we shall look at the values defined to them (according to AT &T 5ESS).

SAPI value	Related entity
0	Call control procedure
16	Packet communication procedure
63	Management procedure
others	Reserved

Figure 24. SAPI's values

TEI value	User type
0 - 63	Nonautomatic TEI assign. user euipment
64 -126	Automatic TEI assign. user euipment
127	group TEI, for broadcast DL connection.

Figure 25. TEI's values

3.1.2.3 *The_Layer_3_frame_structure*

Specification of layer 3 protocol for signaling is the Rec. I.451. For D channel protocol layer 3 of packet data one should refer to X.25. Layer 3 frame structure is shown in the next figure:

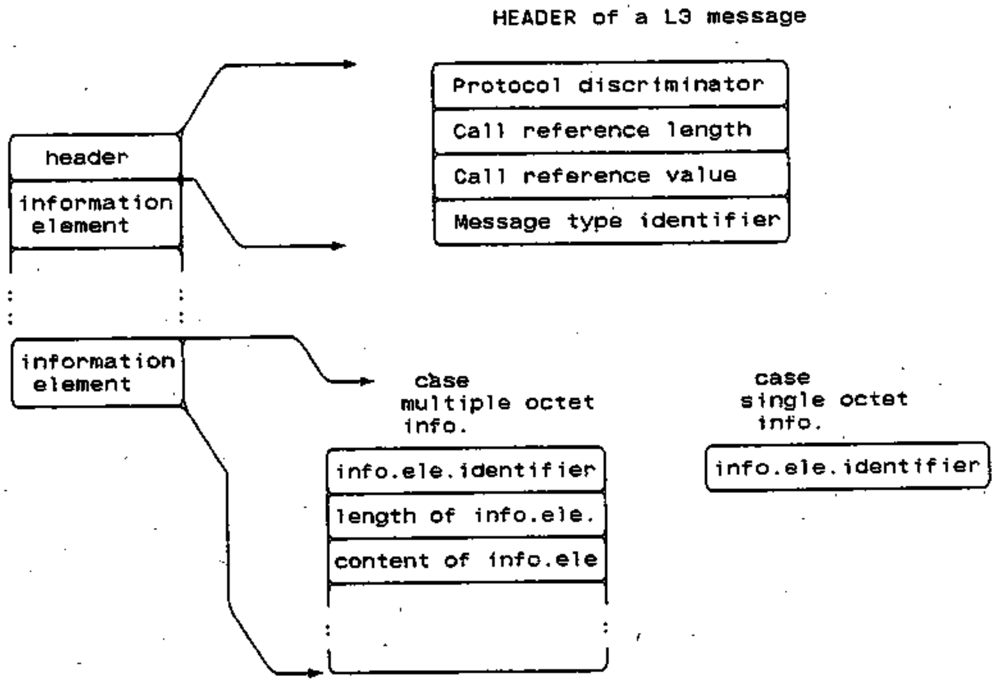


Figure 26. Layer 3 frame structure

Within this protocol, every message consists of the following parts:

- protocol discriminator.
- call reference.
- message type.
- information elements.

A brief description of these fields of interest shall be given:

- Protocol discriminator: This field identifies the L3 protocol to be used for the call control message.

- Call reference: The purpose of the CR is to identify the call at the local user-network interface. The CR is generated by the user in case of an originating call (user's view). In case of a terminating call, the CR is generated by the network.

- Message type: Message type ID identified the kind of signaling messages. This can be messages involved with call establishing, call clearing, miscellaneous (e.g. status), network specific message (e.g layer management related messages).

- Info. element ID: A signaling message may contain more than one information element (these can be bearer capability info., cause info., channel ID info., call state info., etc.) Notice that, if the receiving equipment is not interested in the info. message (i.e after analyzing the info. ele. ID, either not expecting it or can not interpret it), it can skip this info. ele. by using the info. ele. length field information (hereby avoiding the scanning of the entire message).

3.2 *Assumptions_and_implementation_alternatives_*

This paragraph contain three sub-paragraphs; the first sub-paragraph outlines the initialization procedure between TE-ET if MINEX is a transparent entity (i.e if only NT1 is implemented in the MINEX), the second sub-paragraph will outline the assumed initialization procedure between TE-MINEX and MINEX-ET. Also the link between HW and SW will be stated.

The third sub-paragraph deals with some implementation alternatives (Facts which we should take into account, possible solutions etc.) and choices which are made.

3.2.1 *The_initial_condition_of_a_transparent_NT*

In order to specify MINEX as an network termination with intelligence (i.e to be able to realize internal connections), we first study the interaction between the ET and the TE in case a transparent network termination (i.e NT1) is used; since the concerning MINEX must still be able to realize an external connection, and for the external case a NT1 is inevitable between the ET and the TE.

what are the requirements/ behaviours of the ET and the TE to be able to setup a B-channel circuit switched connection ?

First, at least a D channel DL connection must be established between the ET and the TE in order to convey L3 peer to peer messages (i.e for establishing a B channel circuit switched call), the TEI assignment procedure which is essential for setting up the DL connection will be outlined; secondly, the role of the L3 addressing parameter will be investigated, since such a L3 identifier will be used when broadcasting a call setup message to the TE's (i.e a particular TE can be selected by this L3 identifier).

3.2.1.1 *The_TEI_assignment_*

As outlined in the previous paragraph : distinction between DL connections on a D channel can be made by the DLCI, and the TEI element of the DLCI is assigned by the ET.

What are the procedures involved for the TEI assignment at both sides of the network-user interface ?

The following scheme is a simplified scenario for the procedures involved during the TEI assignment:

- TE's L3 send DL-ESTABLISH-REQUEST or DL-UNIT-REQUEST to TE's L2; (initiation of the TEI assignment, TEI unassigned yet)
- TE's L2 send MDL-ASSIGN-INDICATION to TE's layer management entity (further refer to as lMng);
- To cross the user-network interface using the L2 for the assignment purpose:

  TE's lMng send MDL-UNIT DATA-REQUEST to TE's L2;
  TE's L2 send MDL-UNIT DATA-INDICATION to TE's lMng (case ET respond);

  L2 shall transmit lMng message in UI command frames (Unnumbered Information), SAPI shall be 63, and TEI shall be 127.

  lMng message contains: ID request, Request reference number(Ri) and Action indicator (Ai).

- At the ET, after ET's lMng received ID request message, Ai is used to indicate a request at the Assignment Source Point (ASP).

- After ASP selected a TEI (or ignored the Identity request), it shall inform the ET's DL by MDL-ASSIGN-REQUEST, and a peer to peer (lMng) message (contains: ID assigned, Ri and TEI's value) is transmit to user side, with SAPI 63 and TEI 127.

- TE's lMng receives this ID assigned message, compares TEI, Ri to decide whether to accept this TEI or not.

- If TEI value is accepted, lMng entity will also notify a connection management (cMng) entity to initialize the associated link parameter.

- Upon completion of the parameter initialization procedure, the lMng entity shall inform TE's DL by means of the MDL-ASSIGN-REQ; this MDL-ASSIGN-REQ primitive shall contain the received TEI value and the Connection Endpoint Suffix (CES) value to which this TEI is mapped.

- At this time, TE's DL shall complete the link establishment procedure and a point to point L2 link is established between the user endpoint (TE) and the network (ET).

This assignment should take place in two situations:

1. as soon as a terminal powers up ;

2. whenever a terminal determines its identifier invalid and starts the L3 initialization procedure; before the L3 initialization, L2 initialization take place.

For more detailed TEI assignment procedure reference can be made to CCITT Rec.I.441 5.3 .

3.2.1.2 *The L3 addressing parameter*

Since an S interface allowes multiple terminals to be connected to it, a point to multipoint situation is formed in the user-network interface (i.e basic rate interface).

In case the ET is broadcasting a call setup message to the TE's without a specific terminal ID, only those TE's which satisfy the service requirements specified in the setup message will respond; among these TE's which are qualify to respond (i.e in this case four voice/ 64kbps TE's), only the one which first sent a CONNECT message will be connected.

Due to the basic rate interface definition:

- two calls (external) can exist at the same time (situation as:

one extension number for two TE´s which can each have its own call).

- several TE´s can competing for the same call.

3.2.1.2.1 *A proposal for a terminal ID*

The reason why a L3 terminal ID is needed can be summerized as follows:

- one must have the possibility to setup a call to a particular TE (i.e to specify it in a setup message). This also implies that the ET must be aware of such a terminal ID, it should even be it´s responsibility to issue this terminal ID;

- for L3 call control purpose (since CR[17] is not unique in L3 and some of the TE´s may respond to a call with the same CR, without terminal ID one can not discriminate between the different TE´s on L3 level);

- for testing and maintenance purposes.

We resume the relations of some used terms:

- a TE can have several TEI´s (although we assumed here one TEI per TE);

- a TEI is unique in the user-network interface;

- a CR is unique to the origination side within a particular D channel L2 logical link connection.

From the above list, one can derive that a TE connected to the MINEX, can be identified in L2 by the TEI value.

In L3, the CR identifier is not sufficient to indicate to which TE it belongs, since the same CR value can appear at different DL´s.

A combination of TEI and CR answer the requirement of indicating a TE, but TEI is an L2 element, thus we prefer not to use it; if we recall the TEI assignment scenario of the previous sub-paragraph and the introduction of the LAPD environment, we have:

- The DLCI of L2 has a counterpart in L3, i.e Connection Endpoint Identifier (CEI) which contains the SAPI and CES.

- The CES is generated by the layer management entity after connection management completes the link parameter initialization procedure; CES and TEI are fixed mapped.

---

17. CR: Call Reference

The problem is that CES is not perceptible in the L3, otherwise this can function as a terminal ID.

Is it not better to map the CES to a kind of TNR (i.e Terminal NumbeR, which is only significant in this basic rate interface) ?

CES is send by the lMng to L2; TNR[18] can also be sent by the lMng entity, but now to the TE's L3, and be downloaded in the read/write memory of the TE.

In this way a unique identifier for a TE is available in L3 at the user-network interface. Now, a TE is able to identify whether a call setup message is offered to him by recognizing this terminal number.

3.2.1.2.2 *The proposal in the AT & T spec.*

The necessity of the L3 identifier is realized mainly due to the point to multipoint situation. An L3 identifier initialization procedure is also covered in the AT & T "Maintenance and management section of the BRI[19] specification, Part VII"; a summary of the proposed procedures concerning the terminal identifier followed:

1. At subscription time, user gets a SPID[20] for a TE, the user must program this SPID into the terminal;

2. As a TE is plugged into the S interface and powers up, L2 initialization take place (TEI has been assigned and value is stored in the TE's memory);

3. To get L3 identifiers (including the TID), the TE will transmit a lMng (layer management) message which contains the SPID;

4. If SPID is valid, the ET will respond a lMng message to the TE that contains the USID[21] and the TID[22] element;

---

18. TNR: The reason why the term Terminal Number is used in stead of Terminal Identifier is to avoid confusion with an existed term TID (terminal identifier) used in the AT & T BRI spec.

19. BRI: Basic Rate Interface

20. SPID: Service Profile Identifier

21. USID: User Service Identifier, refers to a group of one or more functionally equivalent terminals on a BRI.

22. TID: Terminal Identifier, refers to one particular terminal in the group specified by the USID.

5. The USID and TID are loaded in the TE's memory, at this moment TE has identifiers which can unique identify itself in the basic rate interface.

L3 identifier initialization procedure take place in the following cases:

1. user goes off hook (or a feature button is pushed), while the TE's L3 identifier is not valid;

2. TE receives a broadcast message, but does not have valid L3 identifiers to determined if it should respond to the message or not;

3. ET (network) broadcasts a message (on the signaling link) to the users (not for call setup, but for endpoint initialization purpose).

A remark must be made to avoid misunderstanding, the TID as proposed in the spec. does not uniquely identify a TE on the basic rate interface; a unique identification of a TE in the basic rate interface is made by the pair USID and TID.

To be able to handdle the PTMP situation properly, a way must be found to specify a TE in the MINEX's L3; the assumption at this moment will be:

- A TE has a L3 identifier/ or number which is known and assigned by the ET.
- This L3 TE ID/ or NR is given at the initialization time (after the completion of the L2 initialization).
- Since a TE may have more than one TEI value assigned, it should also keep more than one L3 TID or TNR.

3.2.2 *The_initial_condition_of_the_MINEX_*

Having the scheme of previous subparagraph in mind, we can derive that:

In order to adapt the messages flows between the ET and the TE (concerning the results of the management procedures) and to be able to perform call switching function, the MINEX must at least be provided with (besides those NT1 functions):

1. a L2 for DL establishment and a L3 for call distinction;

2. a function that manages the TEI request and mapping;

3. a function that keeps administration for the L3 identifiers;

4. a layer management function;

5. a circuit management function;

6. a switching unit.

This sub-paragraph will mainly consider the requirements of list item two and list item three of above list, since no description on these subjects has been made yet.

3.2.2.1 *The administration of the TEI on the MINEX*

By using MINEX between the ET and the TE, the layers model for the user-network can be sketched as shown in figure 27.

If we look from the point of view of the MINEX's DL layer, this link is divided into two independent parts, i.e a TE-MINEX part and a MINEX-ET part.

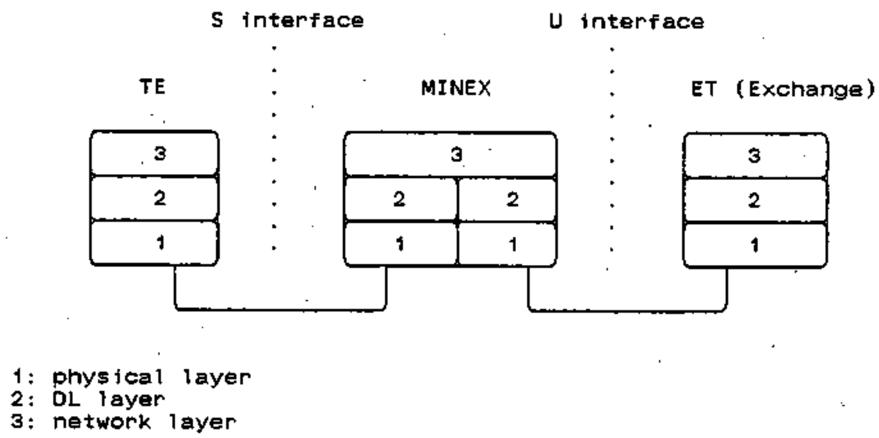


Figure 27. The environment of the MINEX's layers

For the establishment of a proper data link connection, the MINEX must keep a L2 administration.

Towards the TE, the MINEX will act as the ET; but towards the ET, the MINEX shall act as the TE.

The entities involved in this MINEX's L2 administration are those as in the transparent NT case, i.e L2, lMng and cMng entities.

The TEI assignment event in the MINEX is however triggered by the TE, a possible procedure is as follows (recall the simplified scenario for TEI assignment by the TE):

- As MINEX's L2 received a lMng message in UI command frames with SAPI 63 and TEI 127, and wherein the lMng message contained an ID request, Ri and Ai elements, origination of this message is determined by MINEX (from which of the two S interfaces it came from, Ri is related to this S interface).

- This lMng message will be further transmitted to the ET in the U interface, with unchanged data (i.e UI frame, SAPI 63, TEI 127 and same lMng message).

- When the ET selects a TEI, and transmits back via a lMng message to the user, MINEX will intercept this lMng message first, examine the Ri element to determine to which S interface this message should be sent.

- If the lMng is an ID assigned message, MINEX's lMng should keep record of the TEI value assigned by the ET. The lMng entity must be sure that the assigned TEI does not exist already in the MINEX-ET part L2.

- To retransmit this lMng message to the TE (which initialized the procedure) ,the MINEX's lMng entity has determined the destination (i.e to which S interface). A TEI value is also included in this message; but since the TE-MINEX's DL and the MINEX-ET's DL are independent of each other, the administration of these L2's are also separated.

  A TEI in the ET-MINEX's DL is mapped into a TEI in the TE-MINEX's DL (Alternatives concerning this TEI mapping will be discussed later).

  The lMng message is sent to the proper interface with: ID assigned, Ri and TEI element.

- At this moment, MINEX's lMng has kept the relation of the following items: Ri, destination S interface, TEI use in the ET-MINEX part and mapped TEI use in TE-MINEX part.

- The lMng entity shall notify the connection management (cMng) entity of both ET-MINEX's and TE-MINEX's DL to initialize the associated link parameter. Upon completion of the parameter initialization procedure, the lMng entity shall inform MINEX's DL about the TEI values and the mapped CES.

- Both DL's shall complete the link establishment procedure and a point to point L2 link is established between TE-MINEX and MINEX-ET.

3.2.2.2 *The_TE_L3's_identifier_*

As the L3 identifier initialization procedure is started, the L2 initialization procedure should be completed already. The TE may use its TEI (i.e TEI value gained from L2 initiation) and a lMng message (SAPI 63) to request a L3 identifier.

The L2 of the MINEX will transfer these lMng requests/ or assign messages transparently; however, a proper mapping of the TEI values will be performed (i.e TEI used in TE-MINEX's DL vs. TEI used in MINEX-ET's DL).

When the ET assignes the L3 identifier(s), the MINEX's lMng should add this identifier to the data group (by which their relations with each other are kept): the involved S interface, the TEI use in the ET-MINEX DL part and the mapped TEI use in TE-MINEX DL part.

By this administration, the relation of TNR (or USID+TID) and the S interface is established; in case a call is offered to a particular TE, the involved interface can be determined by the relation in the data group.

3.2.3 *Alternatives_concerning_MINEX's_functions_implementation*

Before the Time Sequence Diagram was drawn, uncertainties concerning the various implementation possibilities had arisen. On several points question marks were put, e.g. what will the impacts of these implementation alternatives be on the services provided by the present exchange ?

In this sub-paragraph considerations of the next list are pointed out:

- Concerning the call control functions performed by the MINEX, must all call control take place at layer 3 level, or are there other solutions possible ?
- Should CR's values be administered by using CR's generated internally or should these values (received from either the exchange or the user) be passed transparently ?
- Same question as previous item for the TEI's value.
- Should the four TE's on the two S interfaces be viewed as part of *ONE* interface (S bus), or should we view it according to the physical interfaces (i.e two S busses) ?
- Where is the place (level) that provides contention resolution. on s-Bus level, or on the NT level (see also Rec. I.440 page 199) ?
- An internal call - is it setup using broadcast TEI or by means of specific address information (e.g contained in a L3 message).

Some of the above mentioned items are related very closely to each other, some thoughts concerning these items will be outlined.

3.2.3.1 *Call_control_on_L3_level_?*

Speaking in terms of the CCITT's layers model, an ideal situation (concerning voice/ data communication) would be stated if after a call setup recognition, the MINEX can manage the signaling information flow in such a way that the internal call's signaling information flow passes through MINEX's layer 1 to 3, while the external call's signaling informations transfer takes place on layer 2 level without layer 3 "control".

The reason why this is desired, is the fact that if the call is external, the call control should be achieved by the exchange; so there is no need to bother the NT to unpack the message at L3 level, discover it should be sent to the exchange, thus packing the frame again, and sending it to the exchange.

Figure 28 is an attempt to visualize this ideal situation.

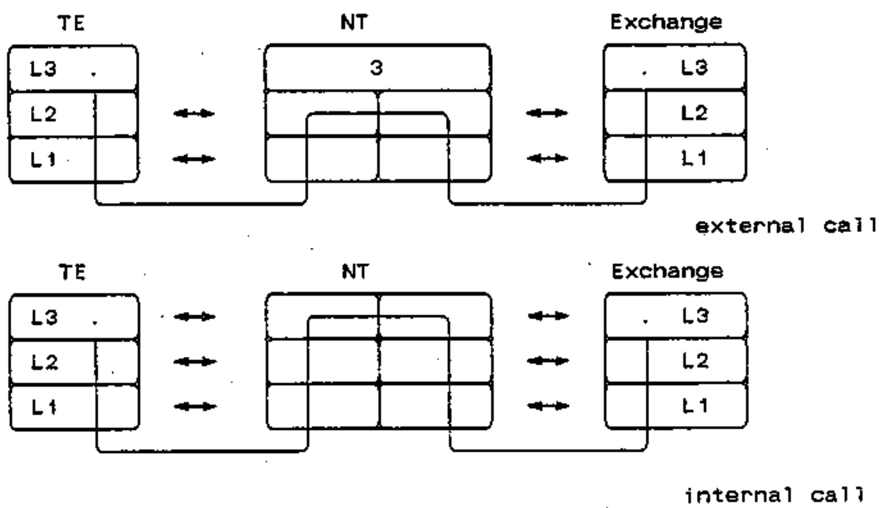


Figure 28. Scheme Of 'Ideal Handling' Of Int/Ext Call

After recognition of an "external" call setup by the MINEX (this take place in level 3, where destination address information can be found), the layer 2 address related to this setup information (L3 and with a unique CR) consists of the SAPI and TEI; but the DL with the same SAPI and TEI combination may also be the address of other L3 info (i.e with other CR's)[23] ; hence a call is identified by its CR in a DL, information for the distinction of a call type (internal or external) is not provided in L2, therefore the call control can't be performed properly in L2.

It seems that for a proper call handling, it is inevitable to let the MINEX's L3 scan all the messages; therefore, the call control function must take place at L3.

3.2.3.2 *Necessity_of_the_CR's_values_mapping_*

Before going into a detailed explanation of the call establishment procedure which is involving the MINEX, let us first take a look at the simple call setup scenario with a transparent NT.

From the time sequence diagram in CCITT Rec.I.451 (pag. 336), we see a setup message of layer 3, which is sent to request a call establishment. One of the essential elements in this message is the *call reference* value (CR). By the using CR, a call is identified at the local user-network interface. In REC. I.451 it was stated explicitly that CR does not have end-to-end significance across ISDNs (e.g figure 29).

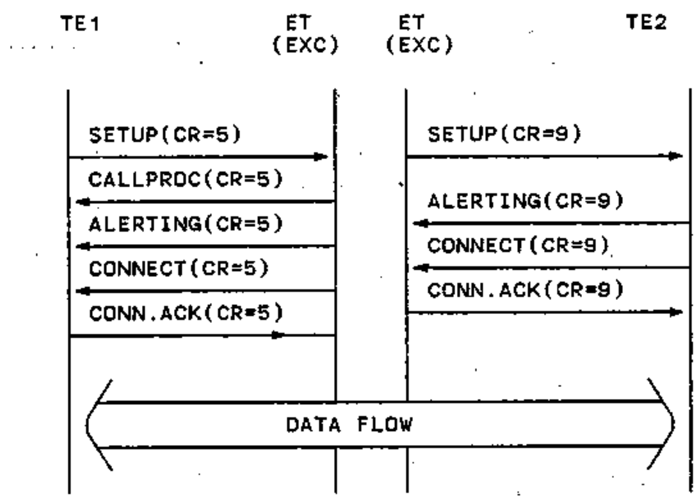


Figure 29. Time sequence diagram of a call setup (transparent NT)

23. Notice that TEI's value for a point to point DL connection may be associated with a single TE, however in the ISDN environment it is permitted to have several calls presented to a TE simultaneously.

Now for cases where the NT is not transparent. In these cases, a setup message shall be received by the MINEX. Three cases can occur:

- case1: an internal call setup is involved.
- case2: an external call setup initiated by the ET.
- case3: an external call setup initiated by the user.

In case2 and case3, MINEX can pass the CR used in the message to the user and to the exchange respectively; because towards an external call, the MINEX can pretend to be a transparent NT. This will not effect the further call request procedure.

In case1, the MINEX will have to act as an exchange. For the terminating side of the call (MINEX -> TE), the MINEX must generate a CR, as MINEX acts as the originator (i.e the one who is supposed to generate the CR).

Above cases will not come into problem if the calls on a user (i.e which are all handled by the MINEX) are all internal calls, or only external calls are involved.

Situation where things are going wrong can be illustrated as follow:

- There is an internal call going on, where at the calling part (TE_X) CR 5 is used, and at the called side (TE_Y) CR 8 is used.
- An external call setup coming from the exchange, using CR 8 , and the destination of this call is TE_Y. (This is possible, because the exchange is not aware of the issuing of CR 8 by the MINEX).
- If mapping of the CR's values is not provided, MINEX will not be able to distinguish the calls on TE_Y (i.e two different calls using the same CR at the same interface).

Figure below may clarify what is stated. For the correct functioning of call control, CR mapping is necessary.

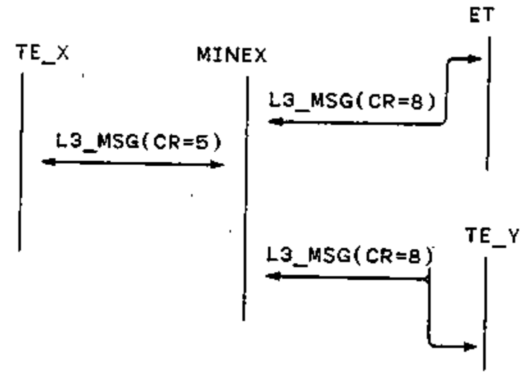


Figure 30. Situation where CR conflict occurred

3.2.3.3 *Views_concerning_the_involved_interfaces*

The MINEX is situated between the TE's and the ET. From the ET's point of view, user's (i.e TE's) behaviours can only be observed via the U interface. Thus in case an internal call is progressing, ET should not be aware (B channels on the U interface are not involved) of its existence.

For the internal interpretation of the MINEX (with respect to the TE side interface), which contains two S interfaces (i.e passive bus) and on each of these interfaces two TE's, two approaches can be discerned:

- All TE's are viewed as part of ONE interface.
- The TE's are viewed per physical interface.

These two approaches have their effects on the realization of the CR's mapping, and the TEI's mapping, and other matters concerning the implementation.

3.2.3.3.1 *A_virtual_interface_for_all_TE's_*

First, we shall investigate the case where all TE's are viewed as part of one bus (see figure 31).

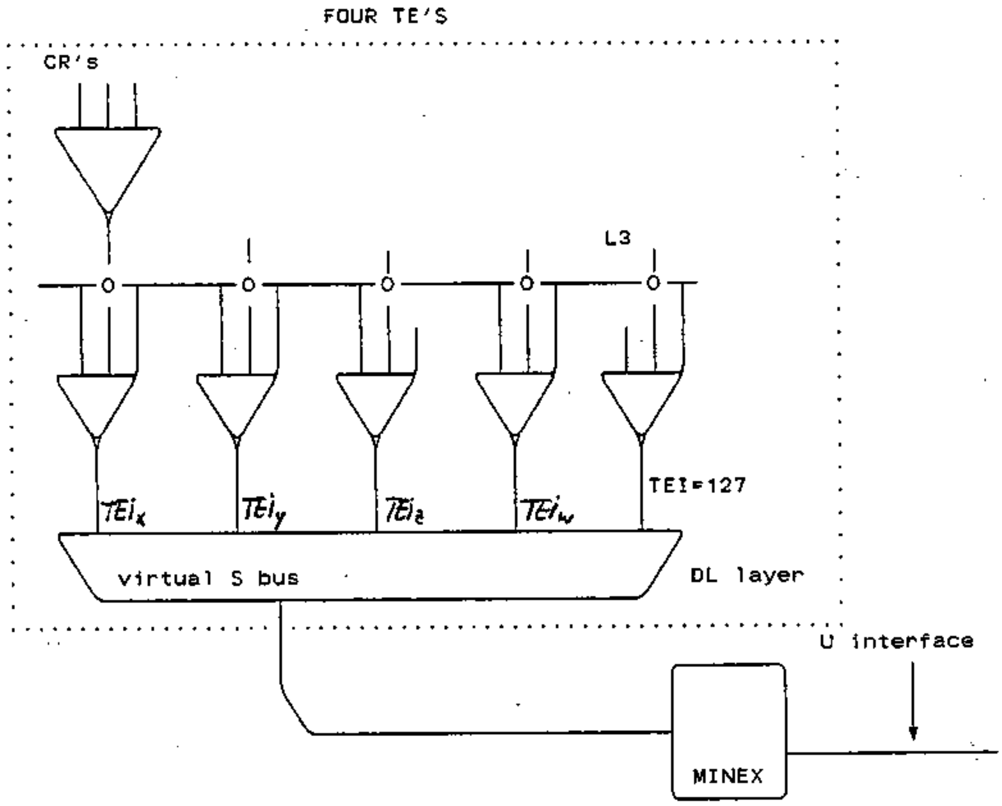


Figure 31. All TE's view as part of ONE bus

In spite of the fact that the four TE's are connected to the MINEX by two S busses, by its functional behaviour one can consider these TE's as being connected by *ONE S-bus*.

What are the effects of this point of view:

1. First of all, call control procedures (On L3) are effected, for example : The CR's value in an external incoming call setup message will be mapped to another CR; the MINEX shall generate a setup message containing this new CR (*ONLY ONE*) value to the four TE's .

A problem shall rise when one of the TE´s accepts the call using the CR. From this CR value, the MINEX can not derive to which of the two S interfaces the accepted call is related.

To determine this, MINEX would have to analyze the TEI value (on layer 2), and the TE associated with this TEI; by knowing which TE is involved, the involved interface identity shall also become clear.

2. An assumption must be made that the contention resolution of the S interfaces has been solved.

Other effects due to this approach concerning the assigned TEI´s values:

- Since we look at the four TE´s as if they were connected at one S-interface, there shall be at least *four different* TEI´s value at the "user"´s side (also four on the network side) of the MINEX. These four TEI´s values are all assigned (at lease approved) by the network; since MINEX is now connected between the exchange and the TE´s, an administration of the TEI´s values at "both sides" of the MINEX is necessary.

- In this alternative, "user side" TEI´s were assigned according to the "network side" TEI´s; in fact they are the same; because the MINEX shall act transparently in the TEI´s assignment matter, values are passed to the user. However, the relation of these TEI´s and the *circuit/ layer management entities* of the MINEX must be recorded somewhere.[24]

3.2.3.3.2 *An approach per physical interface*

The second way to look at the MINEX and its interfaces, is to treat it according to the physical boundary. Towards the user´s side *two S-bus interfaces* are discerned, while on the ET´s side a U interface will be taken into account.

Some significant effects due to this approach are:

1. First, concerning messages generation: After a call setup message is received by the MINEX (sent by the exchange), this message[25] with its CR will be regenerated into *TWO* setup messages, each containing a CR[26] , and each *ONE* of these

---

24. i.e the administration of the TEI values and the mapped CES values.

25. A call setup message from the exchange is usually using broadcasting TEI (127) sending to all TE´s on the subscriber´s line.

messages is retransmitted to *ONE* S-interface.

By the time responds are received by the MINEX for the acceptance (or rejection) of the offerred call , the MINEX can determine on L3 level which interface is involved.

2. Concerning the TEI value assignment.

In the former approach, we used one-one (TE-MINEX link's TEI vs. MINEX-ET link's TEI) TEI's value correspondence, where TEI's assignment is carried out by the ET. The MINEX's task in that case was :

- Transparently transmit the TEI request, and in case TEI has been assigned, transmit it back to the TE.

Because we have required that in the DL level, the TE-MINEX and the MINEX-ET parts have an end-to-end meaning, this implies that we can make the following assumptions:

- Per S-bus we have a couple of different fixed TEI's values. (these virtual values may be the same for both busses).

- There are N * TEI's values in the MINEX, which are assigned by the exchange.[27]

- When a TE is setting up a call, its needed DL is supposed to be existing already. The used TEI's in a DL connection (TE - ET) shall be managed by the MINEX.

- This means that the MINEX has a "distribution" function regarding to the assigned TEI's values. To be more specific, MINEX behaves towards the TE's as an exchange, by mapping the TEI's values which were gained from the network to some TEI's (on the S- interface's along with a BUS ID).

---

26. By issuing different CR's (one CR per S interface), call control procedure can be settled more efficiently. Association of these CR values is done by issuing a Call IDentifier, this CID is unique for a call in L3.

27. This as a result of a request from the TE, which activate a TEI's request procedure in the MINEX.

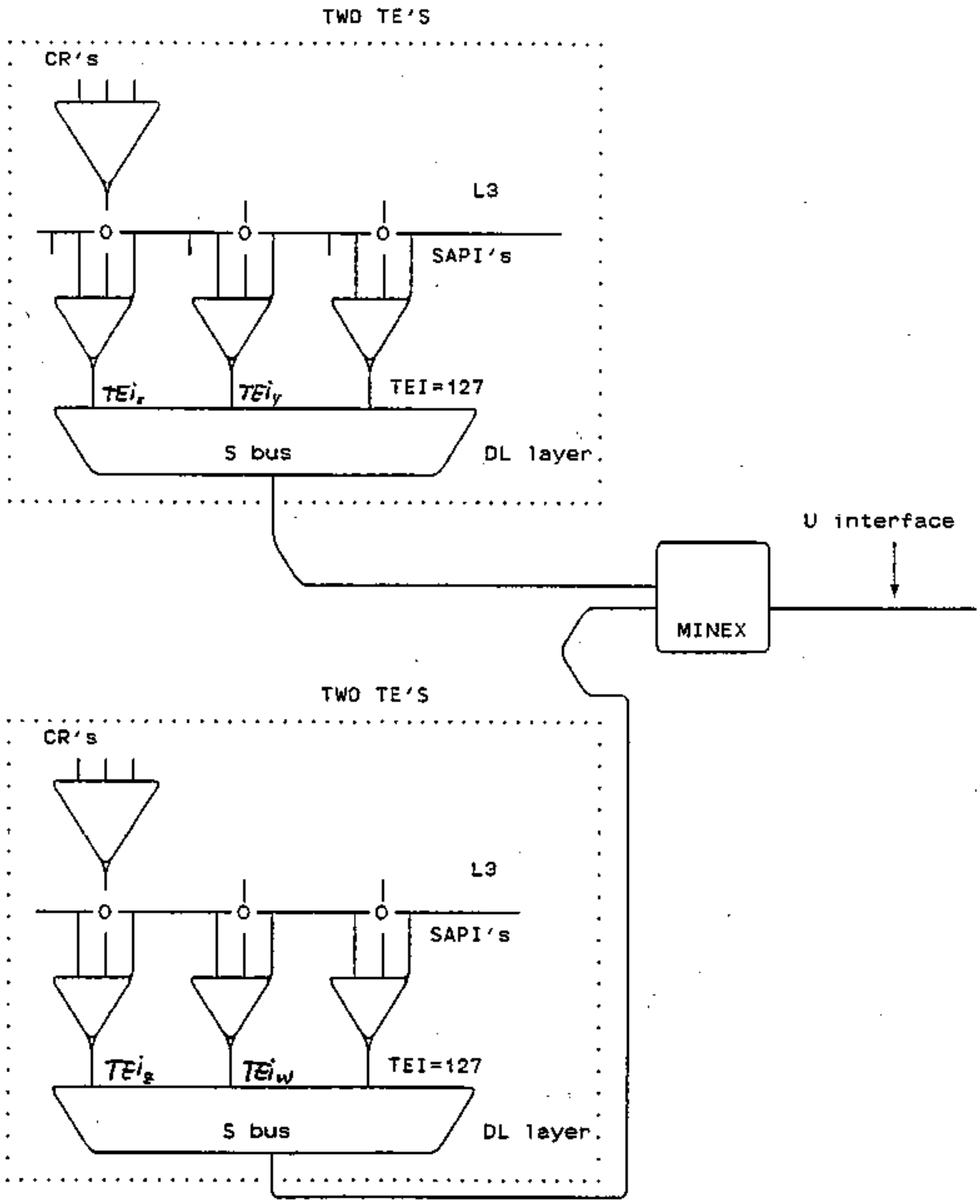


Figure 32. TE's viewed as parts of two S busses

What kind of advantages could we expected from such an approach:

- First of all, one can assume that the MINEX has become more flexible concerning TEI assignment (due to the use of a TEI's "resource pool").

* Since the DL connection on the user-network interface has been separated into two parts (i.e end-to-end significance on each part, and each part with its "own" TEI), more reliability can be created provided that MINEX manages the TEI's administration/ distribution properly.

* An example of reliability is as follow: Assume an internal call is progressing using two TEI's (of the S-interface); if the exchange wants to remove a TEI, it should not have consequences for the active internal call's TEI.

  The MINEX should act as a buffer, and regulate the new TEI assignment.

3.2.3.3.3 *The_TEI_value_administration_*

As TEI's values are assigned/ or approved by the exchange, there is no questions about to whom such a request should be placed; however, if the TEI's values have been assigned by the ET and are returned to the TE, the following options (concerning the TEI management in the MINEX) are up to us to choose:

1. The MINEX will not bother with the TEI's values reallocation, simply pass the values through to the TE (lMng and cMng functions still provided).

2. The MINEX shall issue it's internal TEI to the NT-TE link; this implies that the MINEX keeps records of the TEI's value assigned by the exchange, but issues on the NT-TE link it's own TEI's. The administration is separated into MINEX-TE part and MINEX-ET part, while the MINEX also performes the distribution and allocation function (regarding to the TEI's values).

These alternatives are related to the views mentioned in previous sub-paragraphs:

* TE's as connected to one bus: approach which TEI settlement conform to the first manner of above list.

* TE's viewed per physical interface: the second manner of above list belongs to this approach.

The following table may provide more clearness of the relation between the views and the TEI's assignment/ allocation problems.

We assumed that all TE's have requested for the TEI's value assignment, and the TEI's received from the network by the MINEX have the following values: 1, 2, 3, and 4.

For both alternatives, a table of the assigned values has been made (see table 1 and table 2).

TEI assignment : case virtual S bus					
	TE1	TE2	TE3	TE4	MINEX
BUS	1	2	3	4	
U interface					1,2,3,4

TABLE 1

TEI assignment per phys. interface					
	TE1	TE2	TE3	TE4	MINEX
BUS1	1	2	-	-	
BUS2	-	-	1	2	
U interface					1,2,3,4

TABLE 2

In case of viewing the busses as a virtual bus, TEI's value on both links (TE-MINEX and MINEX-ET) must be unique; and since the assumption is made that all TE's are "connected" to a *VIRTUAL BUS* , the same TEI's value may not be present on the **SAME BUS**.

The view per physical interface on the other hand (as shown in table1) allowes that the TEI's on the S interfaces could have the same values[28] , because the S bus interfaces are viewed apart; A unique TE 'address' is formed by TEI together with a *BUS IDENTIFICATION* (e.g BUS1, and BUS2). This gives the MINEX some freedom of allocating (issuing) it's TEI's values .

3.2.3.3.4 *The_choice_of_the_approach_*

Concerning the advantages which "view per physical interface" offer:

---

28. The TE3 of table 2 can also be assigned the values of 2, 3 or 4; the only restriction is: two same values may not appear at the same S interface.

- more efficient call control on L3;
- more flexibility (reliability) in the TEI's management.
- Contention resolution solved by interrupt.

And since the separation of the management of these values (administration of the TEI's value, bus ID), creates a more structural approach, which provide us more insight in handling these related data; the alternative of "view per physical interface" is chosen.

As a result of this decision, it implies the assumption that the TEI's values will be assigned/ allocated according to TABLE 2 (i.e along with a BUS ID).

3.2.3.4 *Treatment of the TEI during terminating internal call setup*

There are at least two ways of handling an internal call setup at the terminating side. These alternatives are:

1. As described in the CCITT Recommendation, if the ET is the originator of a call (i.e in external call case), this setup message will carry a TEI value of 127 (i.e for broadcasting); if this is also done by the MINEX for an internal call , all TE's will receive a setup request message.

2. Another possibility is:

   assume each TE has an terminal identifier (L3) assigned by the ET, and also known in the MINEX.

   An internal call setup request will not use the broadcast facility; the L2 address is determined with help of the terminal identifier of the L3 message.

What has been described in alternative 1), has the following adventages:

- The TE which accepts a call, responds with a message which is transferred by its data link; automatically, the destination TE notifies in this way its TEI.
- An L3 identifier is not always needed (depends on whether or not a specific TE is specified as destination).

The drawbacks of this alternative are:

- Due to the fact that more than one TE may respond, the MINEX must provide procedures to manage: the competing of the call acceptance and the release of the non-selected TE's after call acceptance.

- If a TE wants to communicate internally to a specific TE, an L3 identifier (of the destination TE) must be provided.

Considering the internal call specification, where an internal call must be able to offer to a particular TE, the restriction is made that the call setup of an internal call will not use the broadcast facility.

In this way, we have reduced this matter to a point to point situation; but this also implies that:

- an identifier (of the destination) must be provided in the call setup message;

- the MINEX must know of the relation between a L3 ID and the TEI which belongs to this L3 ID.

3.2.4 *Assumptions*

For the L3 specification, some assumptions concerning the environment wherein the specification take place shall be made.

Here follow some assumptions:

- One TEI per TE;

- TE of automatic category (i.e TEI assignment procedure involving peer to peer message transfer between TE and ET);

- Bus contention solved by TE through echo-D channel at S interface.

- Contention resolution of the S0, S1[29] and U interfaces is done by an interrupt mechanism of the micro processor.

- L2 and L3 initialization procedures are supposed to be completed before a call setup request sequence can take place.

- A call setup originated by the ET, is always broadcast at L2; on the L3 level choice can be made between a call to all compatible TE's or a call to a particular TE.

  In the former case, the L3 identifier element is omitted in the setup message.

---

29. S0, S1: i.e two S interfaces (S0 and S1).

3.3 *The_MINEX's_layer3_*

In this paragraph the MINEX functions which are essential for the call control on layer 3 level (and some on layer 2 level) shall be stated.

3.3.1 *The_software_environment_*

The environment where call control processes take place contains the interface with the L2 (i.e DL primitives[30] which are mentioned in chapter 2), the L3 itself, and the interface with the management entity.

3.3.1.1 *The_MINEX's_L2_and_management_entity*

First, those assumptions which are made in previous subparagraphs concerning the L2 entity, lMng and the cMng entities retained. These included:

- TEI_req(): As a TE is installed on a S interface of the MINEX, a TEI's value should be requested from the ET by the MINEX's L2 and lMng entities.

- TEI_assig(): As the ET assigned the TEI values and transferred it back to the MINEX, the MINEX shall first check whether this value has been used or not; if not, the MINEX may map this TEI to another value before it transmits it to the TE.

- TEI_relation(): The relation of the requested TEI's from ET and the assigned TEI's to TE's should be kept (TEI's relation of both ET-MINEX and MINEX-TE links), and further also which TE L3's IDs belongs to which TEI; also the mapping of the CES values with the TEI value in the cMng, lMng entities.

- Functions that build up L2 frame and also a function which strips off L2 header. Also L2 sequence and flow control functions (performed by the link layer module ICC component).

3.3.1.2 *The_L2-L3_interface_*

Regarding the signaling messages transfer, the following facts are stated:

- LAPD L2 is supported by the ICC;

- Primitives (L2-L3) specified in CCITT I-series should be present, for information convey from L2 to L3.

---

30. DL primitive: as stated in CCITT Rec.I.440, it represent the logical exchange of information and control between L2-L3.

Physically this corresponds with information transfer from the ICC to the uP.

- Three ICC chip's (two for S interfaces and one for the "U" interface) are used in this MINEX, which implies:

  if information of an ICC is read by the uP, the uP must be aware of the ICC address (i.e due to an interrupt from the ICC, or uP has polled all ICC's for information).

- Recall the paragraph where the hardware architecture is described:

  an S interface is connected to an ICC which performes the LAPD L2 protocol handling, L3's messages which are sent to the uP could retain their L2 address (i.e TEI) or not depending on the operation mode of the ICC.

  The above implies: although the processor module is assigned to manage the protocol handling function at L3, the received L3 message can be provided with a L2 address.[31]

- Derived from the above listed items, we can state:

  Along with the input/ output signaling L3 message of the uP, an interface address (i.e ICC) must be present, and the L2 address (i.e TEI) might be provided.

3.3.1.3 *The_MINEX's_L3_block_*

The call control functions on L3 and the switching functions are all in the L3_block, this block contains four Processes:

ARB, IN, OUT and CC processes.

The functions performed by these processes are:

- ARB process: The recognition of CR/CID[32] , determination of the operation mode[33] of the MINEX's L3, creation of the IN, CC

---

31. The reason why L2 and L3 elements are being mixed up instead of separated is: due to the passive bus (i.e S interface) and the broadcasting facility a point to multipoint situation is created, each endpoint identification is only possible after an endpoint has responded. Since an endpoint ID is not provided in L3, a L2 element is needed. More about this subject in next sub-paragraph .

32. CID: Call IDentifier

33. mode: mode1: for internal call; mode2: in case MINEX is involved at the side of a originating ET; mode3: The MINEX is involved at the interface of a terminating ET.

processes, L3 message transfer to the involving IN or OUT process, creation of a data structure which keeps all information related to a certain call.

- IN process: manage incoming L3 message to be transferred to the CC process; respond messages are transferred to the originated party of a call, IN process will determine the L2 address of the message, and send the message to the module which forms the L2 frame.

  In some cases the IN process also generates respond messages.

- OUT process: Functions as IN process, but now for the terminated (called) party of a call. In some cases, this process must perform a message sequences generation to support the 'release non-selected TE's procedure'.

- CC process: All functions needed for the call distribution, switching commands, status updating and respond message generation will be performed in this process; if a call can proceed, the CC process creates the OUT process.

The ARB process remains in the L3 permanently; while the IN, OUT and CC processes will be generated upon a call setup request, and these processes are killed if a call does not exist any more.

The IN, CC and OUT processes depend on the operation mode (refer to 4.1 of this report) of the MINEX; a suffix behind the process name indicates the operation mode. For example, if ARB determines that a message belongs to mode1, then those involved processes are IN1, CC1 and OUT1; if an other message has been classified as mode2 message, then IN2, CC2 and OUT2 processes are generated and involved.

3.3.2 *The_necessity_of_an_additional_L3_element*

At the beginning, an attempt is made to specify L3's processes; but without involving some L2 elements and the management entity, it seems impossible to realize an unambiguous call connection.

An example might clarify the above statement:

Assuming that the exchange has sent a SETUP (with CR=X) to the user. The MINEX receives this message first, regenerates two SETUP (each S interface get another CR, e.g CR=Y and CR=Z) messages to the four TE's on the two S interfaces.

As the four TE's each respond with their messages, it is insufficient to determine a call connection solely based on L3 information.

The case which we refer to can be described as follows:

Suppose a TE on the S interface will accept the call, and thus responds with a CONNECT message (for example with CR=Y); the other

TE on the same S interface is not able to accept the call, sending RELEASE_COM (CR value is also Y).

It is hard to determine at layer3 what this call result should be, because both messages have the same CR; in fact, it is impossible to assign the call to a TE without other information (such as the TEI).

One might ask: why not assign FOUR CR's to the two S interface at the beginning, each CR to a TE. The answer is: despite the fact that two TE's are on a S interface, only one call is offered to it (with a broadcasting TEI value of 127 as defined in the CCITT Recommendation). And with ONE call, ONE CR is involved on an S interface.

At this stage, one presupposition can be stated concerning the call control protocol:

An additional element is needed in L3 besides the CR; this might be: TEI or CES or the combination of USID and TID.

3.3.3 *L3's_functions_of_the_MINEX_*

In the previous chapter, four groups of processes have been identified as necessary in the MINEX. What kinds of functions should these processes perform ?

Here follows a list of functions which are essential at the first sight:

- det_mode(): When the CR value is not recognized, this function will scan the message, finding address information, and determine whether an internal or external call has been involved. After the decision is made, the relation of CR and type of call will be updated.
- CID_alloc(): For each new call, L3 must issue a CID in order to keep the administration of the involved call.
- rel_CID(): If a call does not exist anymore, its CID value should be released.
- det_CID(): Since CR mapping is necessary, CR's values used in the IN and OUT processes are not the same, although these values belongs to the same call. To be able to update data of the same call, determination of the CID must first be made.
- creat(process): creating IN, CC or OUT processes.
- kill (process): Killing IN, CC or OUT processes.
- recog(sender, tei, CR): Main goal of this function is the recognition of the sender (i.e S or U interface), L2 address and CR value; if CR is already in use, the CID belonging to this CR must be registered somewhere.

- det_Otei(): Determines the endpoint of the call originator.
- det_Ocr(): Determines the CR value used by the call originator.
- det_tei(): Determines the TEI used by the called party.
- det_crt(): Determines the CR value used by the called party.
- clear(db): Data involved with a call request is cleared.
- res_ts(): This function reserves the time-slots to be used.
- b_ch_alloc(): Selection/ allocation of available channels.
- rel_chts(): This function release the time-slots, channels of a call, making them available for other calls.
- chan_ok(): function decide whether a call setup is possible or not with knowledge of the utilization on the U/ S interfaces.
- crt_alloc(): CR mapping for use in the OUT process.
- rel_crt(): This function releases the mapped CR values.
- sel_tei(): Determination of TEI - in some cases, the relation of CR, and TEI is desired, this function should provide information concerning the association of CR, and TEI.
- sel_itf(): Determination of interface address information.
- rel_non-sel(): The release non selected TE's procedures (required in mode3).
- update_DB (): Update for example the call state information, the CCPID or the INPID of a certain call.
- updt st_ori(): Update the status on the calling side (This procedure is necessary in mode3, PTMP configuration).
- up_TEst(): Update responding TE's status.
- switch(): Active/ deactivate the switching function of the MINEX.
- status(): With this function, information of the channels occupation on the U and S interfaces are reported, thereby the MINEX can determine whether channel capacity is available for the call setup.
- concen(): From several responses received by the MINEX, reduction of these messages is sometimes desired (e.g it is not necessary to forward a messages which does not progress the call-state further.

- expand(): As a consequence of using concen(), it is sometime necessary to generate a few messages as a response to receiving one (e.g from a CONNECT message, CONNECT and RELEASE messages could be regenerated).

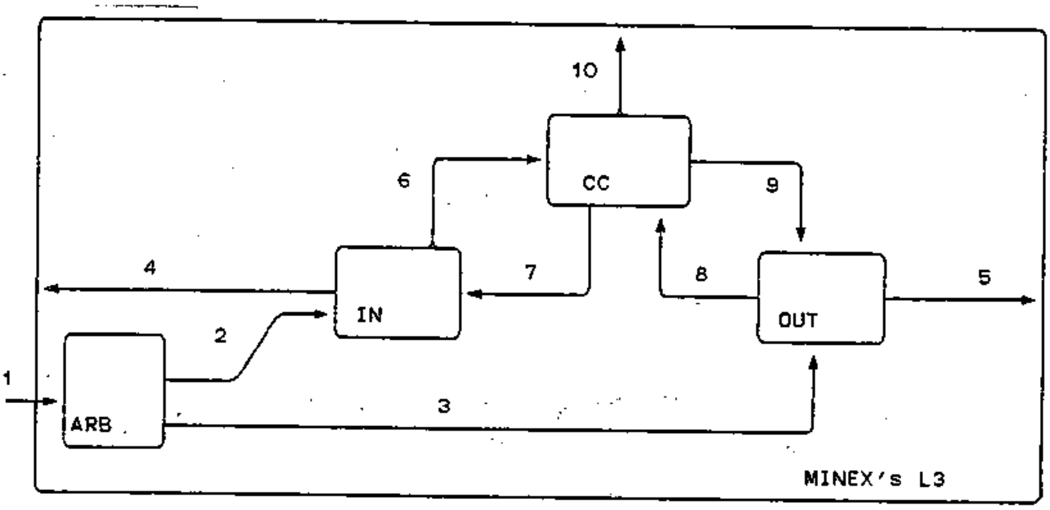
3.4 *The_primitives,_signaling_lists_and_parameters_*

In the OSI reference model, the term primitives refers to the logical exchange of information and control between adjacent layers. The primitives which are applicable to this specification are the DL_DATA_IND and the DL_DATA_REQ primitives (i.e between L2 and L3).

Within the L3's block several channels (in SDL term) are defined between the four processes (i.e ARB, IN ,OUT and CC) defined, these channels are uni-directional and contains a certain set of signals (i.e signal list).

Channels on L3 block are (refer to figure 33):

L2_ARB, ARB_IN, ARB_OUT, IN_L2, OUT_L2, IN_CC, CC_IN, OUT_CC, CC_OUT and CC_PBC channels



1: L2_ARB
2: ARB_IN
3: ARB_OUT
4: IN_L2
5: OUT_L2
6: IN_CC
7: CC_IN
8: OUT_CC
9: CC_OUT
10: CC_PBC

Figure 33. Channel names used in MINEX's L3 block

Here follows the signal lists of these channels:

Parameters which are required in a signal are also given; signals which are marked with "*" only appear in an external call message sequences.

signal list:	L2_ARB	parameter (as defined in the message content definition, Chap. 4.4.3.)
	ALERTING	
	CALL_PROCEED	
	CONNECT	
	CONNECT_ACK	
	INFO	
	SETUP	
	SETUP_ACK	
	DISC	
	RELEASE	
	REL_CPL	

signal list:	ARB_IN	parameter:
	iSETUP	CID, tid, ch_id;
	iCONNECT_ACK	CID;
*	iINFO	CID;
	iDISC	CID;
	iRELEASE	CID;
	iREL_CPL	CID;

signal list:	ARB_OUT	parameter:
	oALERTING	CID, TEI;
	oCALL_PROCEED	CID, TEI;
	oCONNECT	CID, TEI;
*	oSETUP_ACK	CID;
	oREL_CPL	CID, TEI;
	oRELEASE	CID, TEI;
	oDISC	CID, TEI;

```
signal list:          IN_L2          parameter:
                               (as defined in the message content
                                definition, Chap. 4.4.3.)

        CALL PROCEED
        CONNECT
        ALERTING
*       SETUP_ACK
*       INFO

        DISC
        RELEASE
        REL_CPL

signal list:          OUT_L2      parameter:
                               (as defined in the message content
                                definition in Chap. 4.4.3)

        SETUP
        CONNECT_ACK
*       INFO

        RELEASE
        REL_CPL
        DISC

signal list:          IN_CC            parameter

        SETUP_IND                    CID, itf, ch_id, tid;
        DISC_IND                     CID;
        CLEAR_IND                    CID;
        RELEASE_CNF                  CID;
        RELEASE_IND                  CID;
*       INFO_IND                     CID;

signal list:          CC_IN            parameter


        ALERTING_REQ                 CID;
        CALL_PROCEED_REQ             CID;

        DISC_REQ                     CID;
        RELEASE_REQ                  CID;
        REJ_REQ                      CID;
        CLEAR_REQ                    CID;
*       INFO_REQ                     CID;
*       SETUP_ACK_REQ                CID;

        RELEASE_RESP                 CID;
        SETUP_RESP                   CID;
```

```
signal list:            OUT_CC              parameter:

        ALERTING_IND                        CID, TEI;
        CALL_PROCEED_IND                    CID, TEI;
        CLEAR_IND                           CID;
        DISC_IND                            CID;
        RELEASE_IND                         CID;
        DISC_IND                            CID;
*       SETUP_ACK_IND                       CID;
*       INFO_IND                            CID;

        SETUP_CNF                           CID, TEI;
        RELEASE_CNF                         CID;

signal list:            CC_OUT              parameter


        SETUP_REQ                           CID, itf, ch_id;
        CLEAR_REQ                           CID;
        DISC_REQ                            CID;
        RELEASE_REQ                         CID;
*       INFO_REQ                            CID;

        RELEASE_RESP                        CID;


signal list:            CC_PBC       parameter

        SW_ON                        (CH_ID_ori, ts_ori, ITRF_ori,
                                      CH_ID_ter, ts_ter, ITRF_ter)

        SW_OFF                       (CH_ID_ori, ts_ori, ITRF_ori,
                                      CH_ID_ter, ts_ter, ITRF_ter)
```

4. *Call_control_processes*

In chapter 3 an introduction has been made to four processes, which are defined to perform required functions for the call control on L3 (i.e for the control of a connection establishment and generation/ interpretation of L3 messages).

This chapter's division corresponds with the development phases of the specification. The first chapter outlines the classification of call control processes into three modes after an initial study of the messages sequences behaviour of the MINEX.

The second paragraph introduces the Time Sequence Diagram (further referred to as TSD) which is used as the starting point of this specification.

After the TSD has been drawn, state diagrams of these processes are derived from the TSD; in the third paragraph, some remarks and or considerations are placed regarding these state diagrams.

From the state diagrams, SDL (Specification Description Language) diagrams for these call control processes can be drawn. The forth paragraph explains the SDL diagrams which are attached as appendix C.

The fifth paragraph presents the messages (structure) which are used and administration data (proposal) which are needed for the call control.

4.1 *The_mode_indication_*

This paragraph shall make clear what a mode means in this specification, and to what kind of modes we can refer to.

As known, MINEX is supposed to be installed between the ET and TE's; however, the MINEX's call control behaviour depends on the position of the MINEX in a call set up direction. This is shown in figure 34 , 35 and 36 for mode1, mode2 and mode3 respectively.

Figure 34 shows the case of an internal call, mode1 is defined for this situation. On both the originating side and the terminating side of the MINEX, TE's are involved.

With the originating side of the MINEX, reference is made to that part of the L3 where the IN and ARB processes are involved.

The MINEX's behaviours in mode1 can be characterized as "functioning as an exchange"; in this mode the MINEX's L3 must interpret and generate the L3 messages needed for call connection establishment.

Mode2 is defined for the call setup situation where the MINEX is on the user network interface of the originating exchange's side.

On the originating side of the MINEX a TE is involved, while the terminated side of the MINEX is interacting with the ET (figure 35).

The behaviour of the MINEX in this mode is limited. Since protocol handlings are performed by the ET and the TE, the MINEX just has to transfer the messages transparently (although CR mapping is still required, also preferred channel might be changed). Of course generation of the L3 reject message must be provided in case the channel resources are not available for a call setup.

Mode3 is defined for the call setup situation where the MINEX is placed on the user network interface of the terminating exchange's side.

On the originating side of the MINEX an ET is involved, while on the terminating side of the MINEX some TE's can be found (figure 36).

In this mode, MINEX's behaviours is more complicated than mode2; Due to the fact that in a point to multipoint configuration, each TE may respond (if it satisfies the requirement) to a call "SETUP" from the ET.

Only the TE which first responds with "CONNECT" gets the call, other TE's will be released. In this mode, the MINEX shall perform some messages concentration and/ or expansion to support the L3 protocol.

More details concerning the message sequences of each mode shall be given in those paragraphs where the states diagrams of each of these modes are presented.

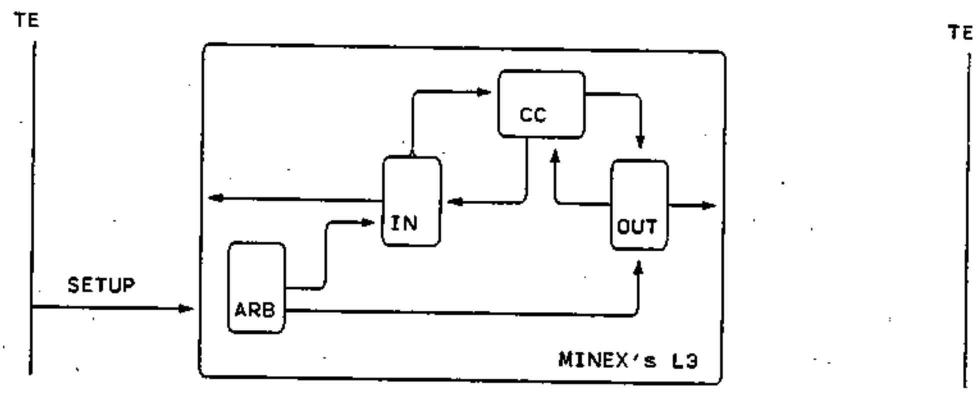


Figure 34. MINEX's L3 block in mode1

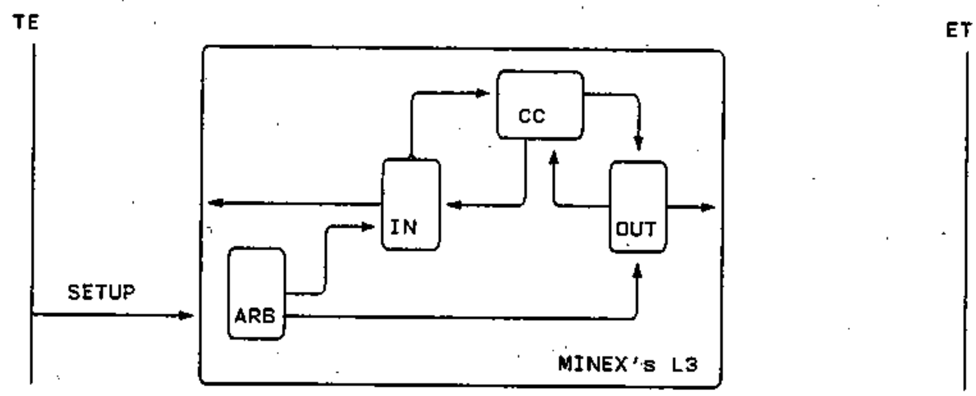


Figure 35. MINEX's L3 block in mode2

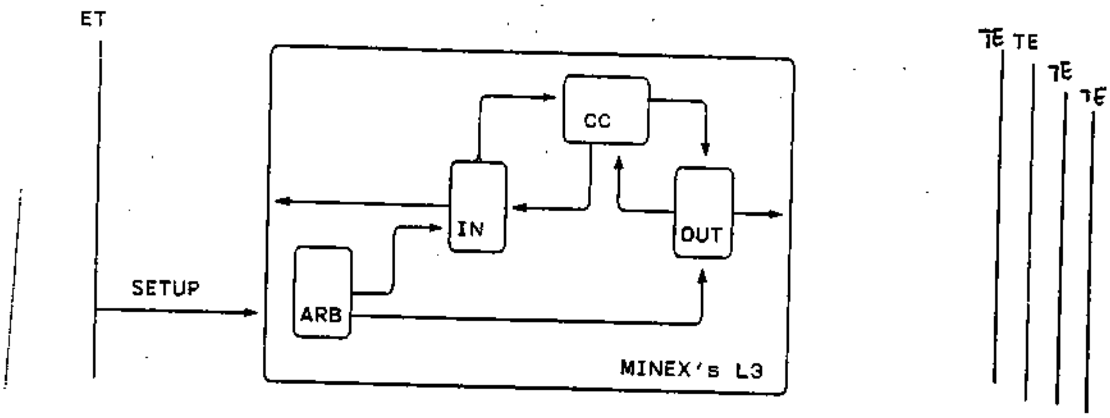


Figure 36. MINEX's L3 block in mode3

4.2 *Time_sequence_diagrams_of_the_MINEX's_L3_processes_*

In this paragraph, comments will be given on the Time Sequence Diagrams (TSD) presented in appendix A.

TSD's are provided to show the permitted temporal relationship between primitives (also between signaling messages and primitives), and the time sequence of these relations during the execution of the call control procedure.

Referring to appendix A, sequences of interaction are shown along the vertical line, which represents increasing time in the downward direction.

Along with a TSD, one of the three situations can occurred:

1. Two TE's are forming the endpoints (i.e related to mode1 internal call, appendix A figure.1 to figure.8).

2. A TE and a ET forming the two endpoints (i.e related to mode2 external call situation, appendix A figure.9 to figure.14).

3. A ET and a group of TE's (i.e external call mode3, a point to multipoint situation; figure.15 to figure.22 of appendix A).

For each of these three situations, several message sequences are shown; this includes scenarios for successful and unsuccessful call setup and for call clearing procedure.

Recall what has been stated in chapter 3 wherein the MINEX's L3 processes have been formed in four groups (i.e the ARB, IN, CC and OUT process); the ARB process which distributes signaling messages to the IN or OUT process is not shown in the TSD's.[34]

A deviation of the signal messages used in TSD with the presented signal list is as follows (refer to signals used in the TSD):

- Generally, all signals received by a IN process from the TE or the ET which are shown in the TSD are ambiguous; since these signals should actually transfer via the ARB process, which is not shown in the TSD's.

The signals names which are shown in the TSD's, are according to the point of view of the TE or the ET; when received by the ARB process, these signal names should all be prefixed with an "i" (i.e of input process) before being transmitted to the IN process.[35]

- The above stated is also valid for signals received by the OUT process from the TE or from the ET; only with the difference that these signals should be prefixed with an "o" (i.e of output process) instead of an "i".

An illustration of this deviation (refer to appendix A figure.1):

The signal "SETUP" from TE, if received by IN1 (i.e IN process mode1) should be "iSETUP", since the ARB process has transferred it through (after determined whether to send it to IN1 or OUT1 process).

The same applies for the "CONNECT_ACK" received by the IN1, this should be "iCONNECT_ACK" and not "CONNECT_ACK"; however, from the TE, "CONNECT_ACK" was sent.

The three signals received by OUT1 are actually "oCALL_PROCEED", "oALERTING" and "oCONNECT", and not "CALL_PROCEED", "ALERTING" and "CONNECT".

---

34. With the appearance of ARB's message sequences in the TSD's, the TSD will lose it's survey.

35. Although messages are received via the ARB process, in the TSD's notes or explanation, references are made to the TE or the ET (according to the TSD) in order to clarify the situation.

    For example, "CALL_PROCEED received by OUT1 from TE2" in note *2 of figure.1, should be interpreted as "oCALL_PROCEED received by OUT1 from ARB".

4.2.1 *Model_TSD's_*

All subjects in this sub-chapter are to be referred to in appendix A (figure.1 till figure.8). Along with most of the TSD's, additional information will be presented to explain one or the other.

Both figure.1 and figure.2 show scenario's for a successful call; figure.3 to figure.6 show some scenario's of call failures; and in the figure.7 and figure.8 the clearing procedures are given.

Some remarks concerning the presented model TSD's:

- It is assumed that the ARB has first performed the address analysis (recognized a call as an internal call, and create the IN1 and the CC1 processes), before "iSETUP" is sent to the IN1 process.

- IN1 determines the destination and passes this as parameter of the SETUP_IND to the CC1 process (preference of channel ID can also be given in the parameter list).

- Whenever a SETUP_IND message is received by the CC1, the MINEX's channel resources status will first be checked to determine whether a channel is available (or can become available) for the call setup. If it is possible, the call request is accepted, and CC1 returns a CALL_PROCEED_REQ to the IN1 (as in figure.1 to figure.5); if it is not accepted, a REJ_REQ is send to the IN1 which sends a REL_CPL to the calling party (as shown in figure.6).

  With the acceptance of the call, CC1 sends a SETUP_REQ to the OUT1, with the destination parameter (MINEX has knowledge of a terminal it's ID and it's TEI); with this destination information, OUT1 transmits a SETUP to the destination TE.

- If the called TE responds with CALL_PROCEED, or ALERTING or CONNECT, the CC1 shall activate the switching function of the MINEX (despite the fact that an active call state is only achieved with a CONNECT message; see figure.1 to figure.4).

- If a DISC follows the CALL_PROCEED, or ALERTING message (figure.3 and figure.4) from either the calling or the called TE then the release procedure shall be activated. By receiving the CLEAR_IND, the CC1 generates a command to switch off the

---

35. Notice that message transmission by the IN or OUT processes does not involve the ARB process; hence, there are no deviation for these messages. For example, "SETUP" transmitted by OUT1 is "SETUP" and not "oSETUP"; hence, these messages in the TSD's are unambiguous.

switching unit.

- If the called TE responds with REL_CPL, OUT1 sends a CLEAR_IND to CC1 (figure.5), and IN1 performs the release procedure after it receives the CLEAR_REQ from the CC1.

- The call clearing scenario's of figure.7 and figure.8 are all related to situations where the active call state has been achieved.[36] The figures shown are not exhausted for all clearing situations, since messages sequences after a DISC are ambiguous if related to the time.

  For example (refer to figure.7), the REL_CPL and RELEASE messages which are marked with a "*" in the first sequence, do not necessarily follow the time sequence; to be more specific, REL_CPL may appear earlier at the MINEX's L3 than RELEASE does.

  The call control process should persist with these sequence variations.

- An important item of the message sequences generation concerns the timer aspect (figure.9).

  The timers used in mode1 are as those specified in the ref[1] part.IV chapter 6.

  Timer T305 specifies the delay time interval of MINEX's sending DISC to the user's response with DISC or RELEASE; these are used in the clearing procedure (default value: 4 sec.).

  Timer T308 specify the delay time interval of MINEX's sending RELEASE to the user's response with REL_CPL (default value is also 4 sec.).

4.2.2 *Mode2_TSD's_*

This sub-paragraph is referred to in figure.9 to figure.14 of appendix A.

Generally, the message information on L3 will be transferred transparently in this mode; since in an external call case, the ET is in charge of the message generation matters.

However, a couple of the L3's message information elements are still to be controlled by the MINEX's L3; these are: the CR and the CH_ID.[37]

---

36. This implies a CONNECT has been received by the MINEX.

The necessity of mapping the CR's values has been stated in chapter three; the reason why CH_ID info.element should also be controlled by L3 can be illustrated as follows (refer to figure.11):

- Recall that on the user side of the MINEX two S interfaces are connected, implying that four B channels are available at this user side; on the ET side only two B channels are supported by the U interface.

- However, if the TE specifies a preferred channel in the SETUP's CH_ID element, it is not necessary that this channel is also free at the U interface for the realization of the call connection; but at the S bus where the TE is connected, this channel must be free to satisfy the demand.

- if one of the two B channels is free at the U interface, a SETUP_REQ can be sent to the OUT2 process; the OUT2 process then sends a SETUP to the ET with a "no preference" CH_ID element.

- ET determines which channel(s) is (are) still available at the U interface, notifies the user which channel is to be used in either the CALL_PROCEED, or ALERTING or CONNECT message (in the CH_ID information element).

- The modification of this CH_ID info. element is performed by the CC2 process (also the mapping of the time-slots related to these channels).

In general, the MINEX's L3 process will generate as few new messages as possible in mode2; but in case clearing procedures are involved, L3 message generation still takes place (for example figure.12, figure.14 and figure.15).

In scenario 12 case1, the disconnection of channels in the MINEX has taken place after the CC2 process receives CLEAR_IND; at the moment CLEAR_REQ is sent, the switching unit on the MINEX has already been switched off.

The main reason why IN2 and OUT2 handle the release procedure separately (by generating REL_CPL, or RELEASE) is to relieve some of the CC2's task.

Figure.14 and figure.15 are also release procedures, but in these cases active calls are involved; hence all message generation is controlled by the CC2 process. The switch off command is given after a RELEASE_IND is received by the CC2 or a RELEASE_REQ is sent by the CC2.

---

37. CH_ID: channel identification.

Less attention is paid to the timer aspects as in mode1, since messages due to time out are generated by the ET.

4.2.3 *Mode3's_TSD's*

A set of call scenario's which are related to mode3 are given in figure.16 to figure.23 of appendix A.

In this mode, the main characteristic is it's point to multipoint[38] configuration.

Concerning the call setup procedure, there is a difference between a call setup in the ISDN environment and a call setup in the present telephone network exchange environment; this difference occurs at the terminating exchange side.

In the conventional case, the ET can be imagined directing its SETUP message to a specific called TE (i.e a specific number), which is indicating a PTP communication.

The procedure described in I.451 (5.1.2.1) for ISDN specifies that a SETUP message sent from the terminating exchange must use the broadcasting facility at the interface (unless the ET has knowledge that a single-point configuration exists at the interface), which is applicable in this mode 3 situation; the decision of which TE on the user side is going to have the call is determined by the TE which will first send the "CONNECT" message.

The above described PTMP situation occurs before the connection establishment of the call has taken place; as the connection is completed, a PTP situation results.

Along with the PTMP configuration, the next problem shows up:

Since there is a competition for the call by several TE's, where should this TE selection take place ?

4.2.3.1 *TE_selection_by_the_MINEX_or_the_ET_?*

The selection of one of these TE's can take place: either in the OUT3 process, or the CC3 process or by the ET.

We have the following alternatives:

- A1: TE selection by the ET (MINEX transparent for messages).
- A2: TE selection by the MINEX, only messages which can progress the call farther will be transferred to the ET ("Partial" by CC or OUT). we have the options: either OUT3 is responsible for the selection, or CC3 is responsible for the

38. further referred to as PTMP.

selection.

To let ET perform the selection implies that all respond messages should be transferred transparently to the ET, whereby the ET is aware of the call states of every TE.

The advantage of alternative A1 is that MINEX's L3 is released from the TE selection task; but along with this contention, part of the message flows between the TE's and the ET is redundant.

The term "partial" (used in alternative A2) shall be interpreted as follows:

If (in mode3) the MINEX is performing the TE selection procedure, the respond messages from the TE's will first be "concentrated" (or "filtered"), and not all of them need be transferred to the ET, only those messages which can proceed the call state further will be passed through (In this case the ET could not keep all current call states of every responding TE).

Notice that in this way, there should be an indication to specify to which TE a message belongs.

The main drawback to allowing TE selection by the MINEX is: by doing it in this way, the user may not use those existing services offered by the network (i.e ET) any more (since the ET is not aware of the call states of every TE's in some cases).

With regard to the TE selection matter, the conclusions are:

- If TE selection is performed by OUT3, the tasks of MINEX are probably better divided among the four processes.
- It seems logical to keep the possibility of utilizing the existing services offered from the network.
- Alternative A2 (i.e partial TE selection by OUT process) seems better, because in this way we have the following advantages:

  1. Limitation of the redundant messages across the user network interface.
  2. The services supported by the network are still available (At least partly; because a "STATUS INQUIRY" from the ET can always be sent to the TE's, to regain the TE's status information).
  3. OUT3 process can lighten the task of the CC3 process.

- The IN3 process (might) also have to keep records of more than one datalink during one call setup (due to receiving those responds messages (i.e primitives) from the CC3 process that originated from different TE's).

4.2.3.2 *Comments_on_the_mode3_TSD's_*

As the processes approach concerning the TE selection has been determined in the former sub-paragraph, the involved mode3 TSD's will be discussed now (refer to figure.16 till figure.23 of appendix A).

With each one of these figures only one call is represented.

For the interpretation of these scenario's, it is important to notice the start point of an arrow at the user's side and the relation with the message which is forwarded to the ET.

For example, in scenario 16:

- TE2 sends CALL_PROCEED to the MINEX's L3 (via ARB to OUT3);

- The message is forwarded until the IN3 process sends CALL_PROCEED (i.e *3) to the ET (the TEI and CR of this message are all mapped values, and all administration is kept in the MINEX).

- As the ET receives ALERTING from IN3 (i.e *4), the TEI value of this message is different from that of the CALL_PROCEED, since TE3 has responded and not TE2 (TE2 and TE3 are both known by the ET at this time).

- These parameter mappings are performed by the MINEX's management entities, and IN3, CC3 and OUT3 processes must have access to this information to transfer it as parameter.

- The CONNECT message from TE1 determines the competition.

- OUT3 transfers a SETUP_CNF to grant the call request. The call state in the CC3 enters the active state; to TE1 a CONNECT_ACK is sent to confirm the connection establishment.

To TE's which are still contending for the call a RELEASE will be sent (i.e release non-selected TE's procedure). Notice that since TE2 has sent DISC, it does not belong to the call contender any more; RELEASE is only sent to TE3.

- IN3 transfers CONNECT with an appropriate TEI (which is fix mapped with the TEI which belongs to TE1 at the MINEX-TE link) to the ET.

As response the ET sends a CONNECT_ACK (with CR and TEI values as in the CONNECT message from IN3) to the MINEX's L3;

- Due to messages received earlier at this ET-MINEX interface, the ET is aware of the existence of the TE2 and TE3 (this information is also kept in the IN3 process); the ET starts the release non-selected TE procedure and sends RELEASE's (with destination TE2 and TE3) to the MINEX's L3.

The IN3 process will terminate the procedure by sending REL_CPL's back to the ET with the proper parameter.

The above stated makes clear that messages from or to the TE's and the ET must be able to distinguish each other by means of some link characteristic.

The message concentration function of the MINEX L3 can be seen in scenario 17 and 18; as OUT3 received an ALERTING from the TE's side, it transfers an ALERTING_IND to CC3 to update the call state. Until one of the TE's has sent CONNECT, OUT3 shall not transfer signal message to CC3 to proceed the call.

If all TE's have rejected the call request (as shown in scenario 19 ,20 and 21) either by DISC or REL_CPL, IN3 shall generate proper messages to the ET (a REL_CPL is permitted as the first respond message of a call refusal; if TE has first responded with other messages, then the refusal of a call afterward must start with a DISC).

For call clearing of a not active call (scenario 19 and 20), the amount of DISC's that are sent to the ET depends on the numbers of messages which reach the IN3 with different parameters.

In scenario 19 for example, CALL_PROCEED_REQ is sent by TE1, ALERTING_REQ is sent by TE3 and CLEAR_REQ also as result of a TE3; IN3 shall clear this call with two DISC's, one with a TEI value that is mapped from TE1, and one from TE3.

For the clearing of an active call, the procedures are the same as those of a PTP situation; because after a call enters the active state (and after the release non-selected TE procedure completed), the PTMP situation is reduced to the PTP configuration.

4.3 *The_processes_and_their_state_diagrams_*

After the TSD's of the MINEX L3 message sequences are stated, the state diagrams of the L3 processes can be derived.

The intention to specify the IN, OUT and CC processes of a call by aiming at a uniform description for each of these processes for all three modes has failed; since each process's behaviour is different at different mode. For example:

An IN process has (not including the CC process) interactions with:

- In mode1: with a TE. (PTP)

- In mode2: with a TE. (PTP)

- In mode3: with a ET.   (PTMP/ PTP)

An OUT process has (not including the CC process) interactions with:

- In mode1: with a TE   (PTP)
- in mode2: with a ET.   (PTP)
- In mode3: with a GROUP of TE's (PTMP/ PTP).

The ARB (arbiter) process on the other hand can be specified uniquely for all three modes; since the ARB has as task: to determine to which mode a message belongs to, and distribute this message to the correct process.

From this information, one can conclude that ONE (uniform) IN or OUT process description for all three modes can probably not be achieved (i.e OUT's behavior towards a ET can not be the same as towards to a group of TE's; In mode3 TE selection procedure is required, but not in mode1 and mode2).

The plan was made to specify all the L3's process in SDL (i.e Specification Description Languages) diagrams, this implied using 10 sets (one set for ARB, three sets for each of the IN, CC and OUT processes as shown in figure 37).

Noticed that in a call situation, not more than four processes can be involved (i.e ARB, IN@, CC@ and OUT@ processes; @ can be 1 or 2 or 3).

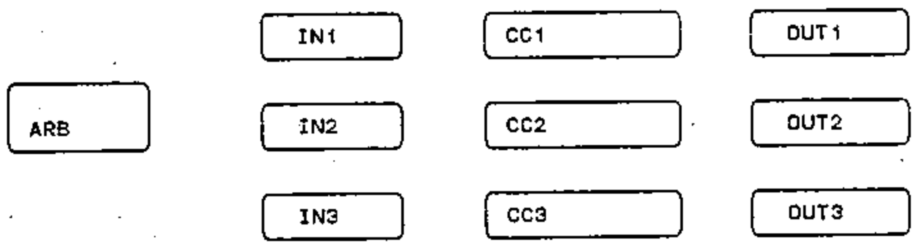


Figure 37. All possible MINEX L3's processes

Before the SDL diagrams for a process can be drawn, the involved TSD's for that mode are used to derive a process state diagram.

The state diagrams are attached in appendix B.

In a state diagram the relations of all states transitions (of a process) and the messages which cause these transitions are shown. This implies that only those messages which are received by a process and have an effect on the state are specified in the diagram.

4.3.1 *The_state_diagram_of_the_ARB_process*

For the ARB process, only one state can be derived.

After the creation of the ARB process (which occurred at the system initialization time), this process remains permanently, receiving messages; after message analysis, ARB transfer the message to the suitable process or generates some processes. Then, return to the state and wait for the next message to be received.

If the received message is a SETUP, ARB shall first determine which mode is involved; then it generates the IN and CC processes belonging to that mode. At the same time ARB issues a unique CID (Call ID) for this call, and a data structure is allocated to keep all information related to this call; then, the iSETUP message is sent to the IN process (can be IN1, IN2 or IN3) for further processing.

In case a message other than the SETUP has been received, ARB shall first try to recognize the sender, and whether this message is a continuation of an existing call; if it is, message shall be passed to the IN or OUT process with the CID and other parameters. If this message cannot be related to a current call, it will be ignored. In all cases, ARB returns to its old state waiting for the next message.

This behaviour of the ARB is the same for all operation modes, only one process description is needed. In the further explanation, the ARB process is assumed to present in each mode.

4.3.2 *The_state_diagrams_of_processes_in_mode1_*

The processes which can occur (besides ARB) in mode1 are IN1, CC1 and OUT1 (appendix B fig.2, fig.3 and fig.4).

4.3.2.1 *The_IN1_process_state_diagram_*

After the creation of the IN1 process by the ARB, IN1 enters the IN1.1 state as ARB sends the iSETUP primitive. At this moment, IN1 can have message interaction with the CC1 and the ARB processes. Also IN1 can output it's L3 signaling message to the environment (provided with address information, i.e to which ICC and which TEI; IN1 process is related to a TE).

In the IN1.1 state:

- with a CALL_PROCEED_REQ or a ALERTING_REQ, the IN1 remains in IN1.1 state;

- a REJ_REQ received in the IN1.1 state forces the IN1 process to be "killed" (scenario 6 of appendix A, i.e channel resources shortage);

- a SETUP_RESP indicates that the call enters the active state, IN1 process proceeds to the IN1.2 state;

- with CLEAR_REQ and iDISC, the clearing procedure for a not active call is started, IN1 process enters the IN1.3 state (refer to scenarios 6 and 7 of appendix A).

In the IN1.2 state (call active state):

- with an iCONNECT_ACK, the process remains in the IN1.2 state;

- the iDISC or DISC_REQ initiates the clearing for an active call, process enters IN1.3 state;

- iREL_CPL at this state is actually not according to the normal clearing protocol, but since a TE could send such an message, prevention is performed.

The IN1.3 state (disconnect state):

is involved with d both clearing for active and for non-active calls. This is the reason why there are two kinds of response for receiving iRELEASE.

For clearing of a non-active call, the IN1 process receives the following sequence of messages from the CC1 or the ARB process (refer to scenario 5 of appendix A):

- IN1 received CLEAR_REQ, sent DISC (enter IN1.3 state);
- IN1 received iRELEASE, sent REL_CPL (enter null state).

In case of the clearing of an active call (refer to scenario 8 , second part):

- IN1 received DISC_REQ, sent DISC (enter from IN1.2 to IN1.3 state);
- IN1 received iRELEASE, sent RELEASE_IND (remain in IN1.3 state);
- IN1 received RELEASE_RESP, sent REL_CPL (IN1 enter null state).

4.3.2.2 *The_CC1_process_state_diagram_*

The CC1 process is created by the ARB process, after the CC1 receives SETUP_IND from the IN1 and determines it is possible to establish a circuit connection (i.e channels available). The resources are reserved, then the CC1 process create the OUT1 process and enter the CC1.1 state; if no connection is possible, process sent REJ_REQ to IN1 and "kill" itself.

Notice that the CC1 process can only communicate with IN1 and OUT1 processes. Call state information is kept in this process, and further the mapping of the CR values, providing control information to the switching unit are also the tasks of this process.

In the CC1.1 state:

- if CALL_PROCEED_IND or ALERTING_IND has been received, a switching command is generated to the switching unit; CALL_PROCEED shall not be forward to the calling party, since this has been done already in the previous step, call enters to the CC1.2 state.
- if CLEAR_IND is received (scenario 5), the CC1 process "kills" itself
- it is possible that the user responds with CONNECT, which results in a SETUP_CNF from OUT1. A switching command is sent and the process enters the CC1.3 state (active state).

In the CC1.2. state:

- an ALERTING_IND can be received (only if CALL_PROCEED_IND preceded). ALERTING_REQ is sent to the IN1 and the CC1 remains in the CC1.2 state;
- by SETUP_CNF from OUT1, the process enters active state (CC1.3);
- if CLEAR_IND has been received, then clearing for not-active call is started (scenario 3, 4 and 5 of appendix A).

In the active state (CC1.3):

- the call clearing is initiated by a DISC_IND either from the IN1 or OUT1 process, call enters CC1.4 (disconnect request state).
- an REJ_IND (i.e a REL_CPL from the TE) in this state does not conform to the regular disconnect protocol; it is specified in order to cover the possibility of appearance.

In the CC1.4 state (disconnect state):

Two messages received in the disconnect state cause deviated treatment (concerning the RELEASE_CNF and the RELEASE_IND messages). References are made to a scenario to make clear the situation.

A possible message sequence (scenario 7, first part):

- CC1 receives DISC_IND, CC1 sends DISC_REQ (to OUT1) and RELEASE_REQ (to IN1); CC1 enters CC1.4 state.
- CC1 receives RELEASE_IND, sends RELEASE_RESP to OUT1, CC1 remains in CC1.4.
- CC1 process receives RELEASE_CNF, process enters null state.
- the received messages of steps 2 and 3 of the above sequence can be exchanged with each other (but CC1's state sequence remains).

Another message sequence (scenario 8, second part):

- CC1 receives DISC_IND, CC1 sends DISC_REQ (to IN1) and RELEASE_REQ (to OUT1); process enters CC1.4 state.

- a RELEASE_IND received from IN1, CC1 sends RELEASE_RESP to IN1, process remains in the CC1.4 state.

- after receiving a RELEASE_CNF process enters the NULL state.

- the reception of RELEASE_CNF and RELEASE_IND should be interchangeable; since the arrival time of these messages is arbitrary.

4.3.2.3 *The_OUT1_process_state_diagram_*

Like to the IN1 process, OUT1 communicates with the ARB and the CC1 processes; and also outputs L3 signaling message to the environment (information as in IN1, but now for called party side; OUT1 process is related with a TE).

With the SETUP_REQ received from the CC1, OUT1 determines the destination (address information should be kept in an administration database), provides the message with correct L3 (and L2) identifier and transmits to the destination.

The state transition in the OUT1.1 state is similar to those of CC1, which is not remarkable if we notice that after the received messages have been processed, these are sent to the CC1.

In OUT1.1 state:

- If oCALL_PROCEED or oALERTING has been received, CALL_PROCEED_IND or ALERTING_IND shall be sent to the CC1 in order to generate the switching, OUT1 enters the OUT1.2 state (stand by). Only ALERTING message is forwarded to IN1 process.

- On reception of oCONNECT from ARB, the OUT1 process sends SETUP_CNF to CC1 to activate the switching (if not occurred yet) and sends CONNECT_ACK to the called party to acknowledge the call acceptance (refer to scenario 1 and 2). Process proceeds to OUT1.3 (active state).

- When oREL_CPL is received, CLEAR_IND is sent to the CC1 to indicate the ending of the call. Process enters the null state; If CLEAR_REQ is received from the CC1, then OUT1 starts its clearing procedure and enters OUT1.4 state (disconnect state).

In OUT1.2 state:

- oALERTING can be received if the preceding received message was oCALL _PROCEED; OUT1 sent ALERTING_IND to CC1 and remained in OUT1.2 state;

- A oCONNECT causes the OUT1 process to enter the OUT1.3 state (active state);

- Receiving oDISC (from ARB) or CLEAR_REQ (from CC1) in this state indicates the clearing of a not active call (refer to appendix A, scenario 3 and 4). Responses after receiving oDISC: CLEAR_IND to CC1 and RELEASE to the TE. Response after receiving CLEAR_REQ: DISC to the TE. OUT1 process will enter the OUT1.4 state (disconnect state).

The OUT1.3 state (active state) has the following inputs:

- oDISC (from ARB, i.e disconnect request from the called party) or DISC (from CC1, result of disconnect request from the calling party) both caused the start of call clearing procedure for an active call (refer to appendix A, scenarios 7 and 8).
  By receiving oDISC, OUT1 process sends DISC_IND to CC1; in case DISC_REQ has been received, a DISC will be sent to the TE (called party). Process OUT1 enters the OUT1.4 state.

- An oREL_CPL is not a proper way to end the active call, the reason why it is specified is analogous to the case of an iREL_CPL in IN1.2 state (i.e in case TE sent it anyway, call control process must be able to manage it).

The transition of OUT1.4 can be derived from the scenarios 3, 4, 7 and 8 of appendix A; figure.4 of appendix B (OUT1 process state diagram) shows:

An oRELEASE in OUT1.4 state can cause two forms of transition.

The message sequence for these transitions can be:

In the case of clearing a not-active call (scenario 4):

- OUT1 process receives CLEAR_REQ from CC1, OUT1 enters the OUT1.4 state from the OUT1.2 state; a DISC is sent to the TE1 (called party).

- As TE1 sent back RELEASE (which is transferred by the ARB to the OUT1 as oRELEASE), OUT1 process check the call state of the call in the call database (update by the CC1 process), recognize it is in clear-req state and then, responds with a REL_CPL to the TE1 and enters the null state.

In the case of handling the clearing of an active call (scenario 7, first part):

- DISC_REQ received from CC1, DISC is sent to the TE2 (OUT1 determined the address information), process OUT1 enters the OUT1.4 state from OUT1.3.

- When OUT1 process receives oRELEASE from the ARB process (i.e TE2 sent RELEASE to the MINEX), OUT1 process also checks the call state of the call in the call database, recognizes it is in the disc-req state; then, respondes with a RELEASE_IND to the CC1. Here process remaines in OUT1.4 state.

- After OUT1 received a RELEASE_RESP from the CC1 process, the OUT1 shall send a REL_CPL to the TE2 and the OUT1 process enter the null state.

The process treatment for the reception of oRELEASE is clearly dependent on the call state of the call (which is updated by the CC1 process in a database, where IN1 and OUT1 processes can take note of it).

4.3.3 *The_state_diagrams_of_processes_in_mode2_*

In mode2, the processes are defined as IN2, CC2 and OUT2 (ARB process is also involved). In general, the handling of message sequences is similar to mode1, except that the environments of processes are different.

For example, the OUT2 process is now related to the ET (i.e with the U interface, and not the S interface as in mode1).

The IN2 process is related to a TE as in mode1 (IN1 process), yet there are some deviation in the state diagram due to the addressing forms.

IN this subparagraph, only those deviant situation with regards to the mode1 shall be outlined (for each process, comparison of states are still made between mode1 and mode2; refer to appendix B figure.2 to figure.6)

4.3.3.1 *The_state_diagram_of_the_IN2_process_*

If we compare IN2.1 state with IN1.1 :

- IN2.1 is added with iSETUP_REQ, INFO_REQ and iINFO messages, all as a result of the overlap sending mode where part of the address information is carried by the L3 INFO information element (refer also to scenario 10 of appendix A).

4.3.3.2 *The_state_diagram_of_the_CC2_process_*

The comparison of CC1 and CC2 has as results:

- The SETUP_IND and INFO_IND in CC2.1 state are as explained before: due to the overlap sending mode.

- REJ_IND can occur in CC2.1 in contrast with CLEAR_IND in IN1. This is because in mode2, until a CALL_PROCEED is received from the ET, no CALL_PROCEED is allowed to be sent to the calling party. (refer to scenario 1, 10, 11 and 13)

4.3.3.3 *The_state_diagram_of_the_OUT2_process_*

With the comparison of the OUT2 and the OUT1 processes (figure.7 and figure.4 of appendix B):

- Again , due to the overlap sending of address information oINFO, INFO_REQ and oSETUP_ACK can appear at the OUT2.1 state (scenario 10 of appendix A)

4.3.4 *The_state_diagrams_of_processes_in_mode3_*

As in the previous sub-paragraph, comparison of the mode3's processes and the mode1's processes will be made.

The IN3 process is related to the ET (U interface is involved, and a PTMP relation can occur).

The OUT2 process is related to a group ofTE's (i.e involved with two S interfaces, and also a PTMP situation).

In this sub-paragraph, references are made to appendix b figure.2 to figure.4 and figure.8 to figure.10 .

4.3.4.1 *The_state_diagram_of_the_IN3_process_*

All messages sent to the ET by the IN3 process must be provided by correct L2 and L3 address information (due to the PTMP situation); this information can be obtained by either consulting the call database or from the primitives sent by the CC3 process.

Some deviations between state diagrams of IN1 and IN3 (figures 2 and 8):

- iSETUP in IN3 is a result of the timeout in the ET; if a time interval is defined after a SETUP has been sent, and the ET does not get any response, then a second SETUP will be sent.

- iRELEASE in IN3.2 state is due to the release non-selected TE procedure of the ET. Recognition of the L2 address is necessary to determine which DL link (on ET-MINEX) should be released (scenario 16, 17 and 18 illustrate the situation).

- The reason IN3.3 state missed a iDISC is: if the ET received DISC, it shall respond with RELEASE (scenario 20); while in mode1 (no ET is involved), an TE may respond with either DISC or RELEASE (scenario 8 of appendix A).

4.3.4.2 *The_state_diagram_of_the_CC3_process_*

If comparing the CC1 and CC3 processes:

- In CC3.1 state, the reason for the appearance of the second SETUP_IND was explained; a second SETUP_REQ is only generated to the OUT3 if no response has been received yet by the OUT3.

- DISC_IND may appear in the CC3.4, this can be illustrated in appendix A (scenario 7, second part).

4.3.4.3 *The_state_diagram_of_the_OUT3_process_*

Comparison of OUT1 and OUT3 state diagrams resulted in:

- In mode3, call shall be offered to more than one TE, if a TE refuses to accept the call, OUT3 should remain at the OUT3.1 state; out3 shall have to update each TE's status. If all TE's have sent REL_CPL (i.e 4*), OUT3 sends REJ_IND to CC3 and enters null state (also refer to scenario 21 of appendix A).

- The oDISC, oREL_CPL and oCALL_PROCEED received in the OUT3.2 state belongs to the responses after the first response (i.e from other TE's; scenarios 16, 17 ,18 and 19 all can be used as reference).

- the OUT3.3 state is reached because one of the TE's has sent CONNECT (received as oCONNECT by the OUT3), to this TE CONNECT_ACK is sent; OUT3 shall also start the release non-selected TE procedure, since OUT3 is aware of the status of all responding TE's, to these TE's RELEASE's are sent. OUT3 remains at the active state (OUT3.3).

  If oREL_CPL is received in this state, it should be the completion of the release non-selected TE procedure for a certain TE, the TE's status shall be updated and OUT3 remains at the same state.

- The possibility that several TE's are concurrently completing their clearing procedure makes it possible that:

  The OUT3 process still remains in OUT3.4 state after an oREL_CPL has been received.

4.4 *Administration_data_&_messages*

This paragraph is divided into three sub-paragraphs, the first outlines the administration data needed to perform the call control processes; the second sub-paragraph outlines the involved status/resources data; and the third sub-paragraph specifies part of the message definitions.

4.4.1 *Administration_data_concerning_with_a_call*

Along with a call connection request, certain information must be administered by the MINEX for an outstanding call. The involved data is proposed in the next list:

data block of a call:

- mode : Mode is implicitly indicated by the kind of processes that have been created (i.e IN1, CC1, OUT1 are created for mode1 etc.).
- CID : Call identifier value; all information concerning a call can be accessed by this CID (each call has an unique CID value).
- SW : Indicates whether call is active or not.

  state : Current call state.
- CCPID : The process ID of the CC process.

calling side data group (for mode1, mode2):

- ts_ori : TS# used for transmitting at the calling side (IN process) of the MINEX.
- CR_ori : CR of a message from the calling side (user or ET).
- CH_ID_ori: Channel used at the calling side (IN process) of the MINEX.
- IN_PID : PID of the IN process.
- ITRF_ori: Interface involved at the calling side (i.e one of the S interfaces or the U interface; may assumed to be an originator address).
- TEI_ori : TEI value of a message from the calling side (user or ET).

In mode3, TEI_ori may obtain several values from the CC3 for a call (depending on the TEI_ori values passed by the CC3 process); along with each new TEI_ori values, a new TEI_ori field is needed in the calling side data group.

For each TEI_ori, the state information should be kept in a "state_ori" element. This implies that several pairs of "TEI_ori", "state_ori" may appear in the calling side data group, with state_ori:

- state_ori: Current call state of the involved TEI (i.e represent the call state of a virtual TE).

called side data group: (for mode1, mode2):

- ts_ter : TS# used for transmitting at the called side (OUT process) of the MINEX.
- CR_ter : CR generated by the MINEX for the called side (OUT process) of MINEX.
- tid : Proposed L3 identifier (i.e destination terminal ID).
- TEI_ter : TEI value of a message from the called side (TE or ET).
- CH_ID_ter: Channel used at the called side (OUT process) the MINEX .
- ITRF_ter: Interface involved at the called side (i.e one of the S interfaces or the U interface; may assumed to be an originator address).
- OUT_PID : PID of the OUT process.

In case mode3 is involved, the following items formed a subgroup: ITRF_ter, CR_ter and TEI_ter.

Some deviations of these elements with regard to mode1 are :

- TEI_ter should be substituted with TEI1_ter and TEI2_ter; TEI1_ter and TEI2_ter represent the current responding TE.
- Also to be added in this data group are the T1_state and T2_state items; which keep the call state of TEI1 and TEI2 respectively of the involved S interface.

Besides above mentioned differences, addition of an act_TEI field is also necessary (to indicate the active TE at the user). In case the active call state has not been achieved, act_TEI refer to the TE with the farthest progressed call state.

The reason why multiple data groups might appear is due to the existence of a point to multipoint situation; in such a case, the administration of all end points must be provided.

4.4.2 *Data_concerning_the_MINEX's_resources_and_status*

The call control procedure must have an overview of the resources and the status of the MINEX.

For example, the availability of the B channels on the S interfaces and on the U interface, and also those of the time slots must be known. Further, knowledge of some management information is also necessary, especially concerning the mapped TEI's values and the relations of TEI and L3's terminal ID.

Concerning the resources utilization, the following items are involved:

- B_CH [ITRF][ID] : availability of one of the B channels on the ITRF.
- CID [call_id] : availability of the call_id value.
- TS [time_slot] : availability of the time_slot value.
- CRT [cr_ter] : availability of the cr_ter (call reference terminated) issued by the CC process.

with regard to the MINEX's status information, the following information is inevitable:

- The association of the CID and the PID's (process ID) of the IN, CC and OUT processes.
- The relation of the TEI's values assigned by the ET and the TEI's assigned to the TE's by the MINEX's management entity (i.e mapping of a TEI from the ET to a BUS_ID (i.e S interface) and a TEI issued by the management entity).
- The relation of a TE's L3 terminal ID and the associated TEI's values.

Above listed information is assumed to be updated in a database, the call control processes may consult (or update) this status information.

Clearly, signaling messages and primitives used between the processes effect the status or resources information, some examples are shown in the following figure 38:

Message / Primitive	Items effected on data block	Resource's items effected	NT status
SETUP	CID; IN/ CC PID; ITRF_ori; TEI-ori; CR_ori;	CID;	IN/ CC PID;
iSETUP	ITRF_ter;		
SETUP_IND	CH_ID_ori; CH_ID_ter; ts_ori; ts_ter; ITRF_ter; CR_ter; call_state;	B_CH[][]; TS[]; CRT[];	update B_CH / TS used; mapped TEI/ BUS_ID;
CALL_PROCEED_IND	CH_ID_ter; ITRF_ter; call_state; TEI_ter; SW;		
SETUP_CNF	call_state; TEI_ter;		number of active call;

Figure 38. Effects of some used primitives/ messages on administration data

4.4.3 *The_message_content_definitions*

In this section the definitions of the L3 messages used in the MINEX application will be listed; these definitions are based on the message functional definition as defined in the CCITT Rec. I.451.

Each definition includes:

A. A brief description of the message direction and usage for both the internal and external call case.

B. A table listing the information element contained in the message. For each information element, the following items are indicated:

- the section where this information element is described in the CCITT's Rec. I.451.

- the direction in which it may be sent for the external call case; indicated as: user to network (u -> n), network to user (n -> u) or both.

- whether inclusion is mandatory ('M') or optional ('O') or dependent on the circumstances (explained by means of notes) of the external call.

- the direction in which it may be sent for the internal call; the direction in the internal call case can be user to MINEX (i.e u -> nt), MINEX -> user (i.e nt -> u) or both.

- whether inclusion is mandatory ('M') or optional ('O') or dependent on the circumstances (explained by means of notes) of the internal call.

- the length in octets (these values are according to 5ESS BRI basic call spec.).

  As defined in CCITT Rec.I.451. the information is listed in order of appearance in the message. For detailed explanation of these information elements refer to Rec. I.451.

  * signal and terminal capability information elements shall not be used, since functional TE is assumed to be used.

C. Further explanatory notes, as necessary.

A proposal shall be made for information element(s) which is (are) not defined in the Rec. I.451, but is (are) inevitable for this specific application.

ALERTing:

external: The called user sends this message to the ET, and the ET sends this to the calling user, to indicate that the called user alerting has been initiated.

internal: The called user sends this message to the MINEX, and the MINEX sends this to the calling user, to indicate that the called user alerting has been initiated.

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	both	M	1
Call reference	4.3	both	M	both	M	2
Message type	4.4	both	M	both	M	1
Channel Identi.	4.5.10	u→n	note1	u→nt	note1	3

note1: The user may include this only if this message is the first response to an incoming SETUP message.

CALL_PROCEEDing:

external: The ET sends this to the calling user, to indicate that the requested call establishment has been initiated and it will not accept any more establishment information. The called user may send this to verify the selected channel.

internal: Idem as for external case, only the term "ET" should be replaced by "MINEX".

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	both	M	1
Call reference	4.3	both	M	both	M	2
Message type	4.4	both	M	both	M	1
Channel Identi.	4.5.10	both	note1	both	note1	3

note1: This information element is present only if this message is the first response to a SETUP message.

CONNect:

external: The called user sends this message to the ET, and the ET sends this to the calling user, to indicate call acceptance by the called user. If this message is sent by the ET, it implies to the calling user that the ET has established an end to end circuit mode connection to the called user.

internal: Idem as for external case, only the term "ET" should be replaced by "MINEX".

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	both	M	1
Call reference	4.3	both	M	both	M	2
Message type	4.4	both	M	both	M	1
Channel Identi.	4.5.10	u→n	note1	u→nt	note1	3

note1: The user may include this information element only if this message is the first response to an incoming SETUP message.

CONNect ACKnowledge:

external: The ET sends this message to the called user to indicate the completion of the circuit switched connection; the calling user may send this message to the ET.

internal: Idem as for external case, only the term "ET" should be replaced by "MINEX".

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	both	M	1
Call reference	4.3	both	M	both	M	2
Message type	4.4	both	M	both	M	1

DISConnect:

Either the user,the MINEX (internal call case) or the ET (external call case) sends this message as an invitation to release the B channel and call reference value.

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	both	M	1
Call reference	4.3	both	M	both	M	2
Message type	4.4	both	M	both	M	1
Cause	4.5.8	both	M	both	M	4

INFOrmation:

external: Either the user or the ET sends this message to provide additional information (e.g addressing information in overlap sending).

internal: Not applicable.

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	-	-	1
Call reference	4.3	both	M	-	-	2
Message type	4.4	both	M	-	-	1
Display	4.5.14	n→u	O	-	-	3-22
Keypad	4.5.16	u→n	O	-	-	3-22

RELease:

Either the user,the MINEX (internal call case) or the ET (external call case) sends this message to indicate that the equipment sending this message has disconnected the B channels and intends to release the CR, and that the receiving equipment should also release the B channel and CR. Any call in the process of being set up shall be aborted.

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	both	M	1
Call reference	4.3	both	M	both	M	2
Message type	4.4	both	M	both	M	1
Cause	4.5.8	both	note1	both	note1	4

note1: This information element may be absent if this message is not the first clearing message (i.e it is preceded by DISC).

RELease ComPLete:

This message is sent from either the user,the MINEX (internal call case) or the ET (external call case) to indicate that the equipment sending this message has released the B channels and the CR (if any), and that the receiving equipment should also do the same. The channels will then be available for reuse.

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	both	M	1
Call reference	4.3	both	M	both	M	2
Message type	4.4	both	M	both	M	1
Cause	4.5.8	both	note1	both	note1	4

note1: This information element may be absent if this message is not the first clearing message (i.e it is preceded by DISC).

SETUP:

The user, the MINEX (in case internal call) and the ET (in case external call) sends this message to request call establishment.

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	both	M	both	M	1
Call reference	4.3	both	M	both	M	2
Message type	4.4	both	M	both	M	1
Bearer Capability	4.5.5	both	M	both	M	5
Channel Identi.	4.5.10	both	note1	both	note2	3
Keypad *note3	4.5.16	u→n	O	u→nt	M	3-22
TE L3 ID. *note4		both	O	both	M	3-?

note 1: For u→n it is optional, while for n→u it is mandatory.
If sent by the MINEX to the ET, it should be "any channel" option, if sent by the MINEX to the TE, channel ID determined by the MINEX.

note 2: For u→nt it is optional and nt→u it is mandatory.

note 3: The keypad information element is used to send address information (conveys IA5 characters); distinction of internal and external call can be determined by the appearance of a certain sequence of "*" and/ or "#" characters (e.g in internal case "#" and "*" are used. while for an external call, this information element contains the destination address digits)

note 4: In external case, this element can be used to select a particular TE in an address specified in the keypad information element.
For internal call case (i.e keypad contains "#*" sequences), this element indicates a certain TE on the user side.
? in length field indicate maximum length is undefined yet.
In fact this element could be the DDI or subaddress element which are mentioned in the REC. I.330.

SETUP ACKnowledge:

This message is only used in case of d external call; the ET sends this message to the calling user to signal that the network has begun call establishment but needs additional information to proceed.

information element	reference CCITT Rec. I.451	direction/ type external		direction/ type internal		length
Proto. Discriminator	4.2	n→u	M	–	–	1
Call reference	4.3	n→u	M	–	–	2
Message type	4.4	n→u	M	–	–	1
Channel Identi.	4.5.10	n→u	M	–	–	3
Display	4.5.14	n→u	O	–	–	3–22

4.5 *Additional_notions_to_the_presented_SDL_diagrams*

By means of the state diagrams, the TSD's, the administration data (described in this chapter) and the L3's functions (defined in Chapter 3.3), the SDL (Specification Description Languages) diagrams can be drawn for L3's processes for all three modes.

These SDL diagrams are attached in appendix C; with regards to the presentation of these diagrams, the following notions should be taken into account:

- Due to the limitation of the diagram generation tool, the presented symbols for SDL diagrams are partly incomplete in comparison with the standard SDL diagrams (recommented by the CCITT).

- Since the "process create request" symbol and the "procedure start", "procedure stop" symbols are not supported by the tool, alternatives are used. In stead of these missing symbols the task symbol has been used; expansion text which describes the task will indicate the perform functions.

  For example: the text for the creation of the IN1 process can be "create IN1"; text for a procedure call has the form of "procedure_name ()".

- The process stop symbol "X" is also not provided by the tool; a X state is used as alternative to indicate the ending of a process.

- Parameter of a signal or a procedure call are surrounded by parentheses (i.e "(" ")").

- Comment may appear within the text expansion symbol (i.e "-[").

- Data involved in a task (including "procedure call" described in previous items) are related to the data proposed in Chapter 4.4.

- Since different configurations (i.e PTP and PTMP) may appear, a different set of data structures is required in each case (refer to Chapter 4.4 for the proposed data structure).

- In an expansion text in the SDL digrams, the "To @@@" string occurred frequently (for example in a signal transmission); "@@@" indicates the "destination" of the signal.

  If it is a signal among L3's processes, "@@@" refers to a L3's PID (i.e Process IDentifer of the IN, the CC or the OUT process).

  If this signal is crossing the L3 block boundary (refer to channel 4 or channel 5 of figure 33), "@@@" refers to a L2's PID. Such a L2 process performs for example the "packing" of the L3's messages in a L2 frame and further other LAPD functions (e.g sequence control, detection of transmission error or provision of DL etc.). In implementation terms, "@@@" can be viewed as the address of one of the ICC components which support the basic LAPD L2 function.

The L3's functions which are required or needed described in Chap 3.3.3), have been assigned to L3's processes as follow:

- The ARB process : performs the det_mode(), the CID_alloc(), the det_CID(), the creat() and the recog() functions.

- The IN process : performs the det_Otei(), the det_Ocr(), the sel_itf() the expand() and the updt st_ori() functions.

- The CC process : performs the creat(), the res_ts(), the b_ch_alloc(); the update_DB(), the chan_ok(), the crt_alloc() and the switch() functions.

- The OUT process : performs the det_tei(), the det_crt(), the up_TEst(), the rel_non_sel(), the concen() and expand() functions.

5. *Test_Case*

After the SDL diagrams are drawn, model has been chosen to be implemented as a test case; the intention is to illustrate some aspects related to the message sequences and the connection establishment (e.g management aspect such as TEI and CR values mapping).

For this implementation purpose, an experimental operating system (developed at the Delft University of Technology, and designed to implement telecommunication software based on SDL) has been use.

In appendix D, the test program for model is listed; appendix E contains the tested message sequences and the results.

A brief description of the test environment and some remarks on the test sequences will be outlined in the next two paragraphs.

5.1 *The_test_environment*

The block diagram used for the model implementation is similar to figure 33 (i.e where channels and process blocks in L3 are shown); the edge of the MINEX L3's block represents the interface with the adjacent layer/ entity (refer to figure 33), the channels crossing the interface are:

- channel 1 (L2_ARB) : receives messages from the environment (i.e L2 entity of the calling and the called party).
- channel 4 (IN_L2) : transmit messages to the environment (i.e L2 of the calling party).
- channel 5 (OUT_L2) : also convey messages to the environment (i.e to the L2 of the called party).
- channel 10 (CC_PBC) : transmit control information to the switching unit.

For testing purpose all channels are assumed to communicate with a terminal (further refer to as monitor), which can visualized the message sequences through the "shell" and "logger" programs.

A tester can manipulate the message sequences (i.e arbitrary messages may be input from the monitor to the L3's processes), and also the simulation of call request message sequences from different TE's can be achieved (i.e by sending message with a certain address information, monitor can simulate different originations).

After a message is sent to the L3's processes, a response (or responses) from the L3 block is (are) expected; this can be observed on the monitor, the tester may input another message (either simulating the calling or the called party) and observe the L3's response(s) again etc.

In appendix D page 33, the data structures which are associated with this model are listed.

Structure "CINFO" is an implementation of the proposed data structure which is related to a call (refer also to Chapter 4.4).

The elements of the "resources" structure is concerned with the resources used or occupied by the MINEX.

The "status" structure obtains some relationships between data at the system initiation time (refer to the figure 39); during different call requests, "status information" are dynamically updated with each call.

Notice that the CINFO structure is associated with a call, while the other two structures are related to all calls controlled by the MINEX.

The assumed initial relations of some data (and values) in the test program are as follow:

interface	(S)0	(S)0	(S)1	(S)1
assigned TEI value	1	2	3	4
TE's L3 ID = TID	101	102	103	104

Figure 39. Assumed TID, TEI and interface relation/ values

Above table should be interpreted as follow:

each column represents information of a TE: to which interfaces it is connected (S0 or S1), which TEI value has been assigned to this TE (by the MINEX) on the S interface, and which TID value is associated with this TE.

5.2 *The_test_sequences*

By applying arbitrary message sequences to the MINEX's L3 (it is not necessary to only involving one call), and observing the responses of the MINEX's L3, we can prevent some non-determinable behaviour of the MINEX's L3 (because, if an error situation occurred, revision of the message handling specification should be considered.).

Each test sequence has been divided in two columns (refer to appendix E), the column at the left side represents the messages sent by the calling party, while messages in the right column represent the called party messages.

The numbers after each message represent the parameters along with that message (i.e generally it concerns the TEI, CR, flag and channel_id).

The test sequence 0, 1, 2 generates messages for the call request of a TE (e.g on S(0) interface) with another TE on the same S(0) interface, or with a TE on another S interface (i.e S(1) interface); the channel selection option is also tested in these sequences.

In test sequence 3, messages are generated for the release procedure of the active calls; while sequences 4 and 5 are concerned with the release procedure of the non-active calls.

The result of the test sequences are also attached in appendix E. The interpretation of the result is as follow:

- Each line which begins with a "send" is an input to the MINEX L3's processes (i.e might be the calling as well as the called party).

These "SENDER"'s lines are the simulated TE's messages (generated by the tester).

- A line which begins with env[0] represents the response of the MINEX's L3 after receiving a message.

Besides the response messages, also parameters are generated; these parameters are the mapped TEI, the mapped CR, the correct flag values and the selected channel ID (appears only in the messages when it is required).

6. *Conclusions*

Concerning the hardware architecture, the following conclusions have been stated:

- The PBC used is able to perform the switching functions.

- The MINEX can be extended to support more user interfaces (5 subscriber interfaces are still available in the PBC).

Through the introduction of the MINEX, the data link connection between the exchange and the user is separated in two parts. An important item concerning the information transfer is the correct management of the DL connection by the MINEX. With regards to this administration, the following conclusions can be stated:

- The TEI assignment procedure initiated by a TE, must be supported by the MINEX.

- The TEI values assigned by the exchange are intercepted by the MINEX; for the reassignment of the TEI value back to the TE, an approach of per physical interface has been chosen.

- The management of the corresponding DL's (i.e the TE-MINEX and the MINEX-ET part TEI mapping) is also the task of the MINEX's management entity.

- The CES mapping along with the TEI (re)assignment in the MINEX shall also be performed by the management entity.

For the MINEX's L3 call control the following conclusions are stated:

- The behaviour of the MINEX's layer3 has been identified to be depending on the position of the MINEX in a call setup request (three modes have been distinguished).

- For the L3 call control in each of these modes three processes are needed; a process which is related to the call originator, one which is related to the call terminator and a process which is performing the switching and other related management matters.

- Since the ISDN numbering used in a SETUP message is addressing an interface (T or S interface) in stead of a specific terminal; the subject of addressing a TE in the user network interface can only be solved if an extra L3 element which is related to a terminal will be added (besides the Call Reference value).

  (In this specification the combination of the USID and the TID has been used; the presummed condition was: the DDI and subaddress information are not available).

The MINEX's call control process of the mode1 has been implemented and the process behaviour has been observed through the simulation of the environment, the following conclusions can be stated:

- For the response messages generation and the channel resources management matters, the defined processes are function properly.

- The L3 processes are capable of handling ambiguous message sequences.

- The assumed necessity of an terminal related L3 ID (or address) has proved to be correct (otherwise a specific terminal can not be selected); the use of USID and TID as this L3 ID is however not elegant in the internal case.

- The PTP call setup request at the terminated side (internal call case) is obviously not necessary; a broadcast at L2 is also possible as long as a L3 ID element (related to the destination terminal) is contained in the SETUP message.

Annex: De relatie van de CES, TEI, SAPI, DLCI, CEI, USID, TID

- "SAPI"

  De informatie overdracht tussen L3 en L2 vindt plaatst via de SAP (Service Access Point). Voor verschillende services zijn er verschillende SAP's, hiervoor wordt de SAPI gebruikt om de verscheidenheid van de services aan te duiden; zo wordt voor de signalering doeleinde de 0 waarde aan de SAP toegekend (SAPI=0), voor pakket data services is de SAPI 16, en voor de management doeleinde is de SAPI 63.

- "DLCI"

  Voor de informatie overdracht tussen twee L3 entiteiten wordt gebruikt gemaakt van een data link connectie (op L2). Een data link connectie wordt gekenmerkt door de Data Link Connection Identifier (DLCI) die zich in de adres veld van een L2 bericht bevindt. De DLCI bestaat uit twee elementen: de SAPI en de TEI (Terminal Endpoint Identifier).

- "TEI"

  De TEI wordt uitgegeven door de centrale en is uniek in een centrale-gebruiker interface. Een TEI wordt gebruikt om een bepaalde connectie punt te identificeren binnen een SAP. De TEI geeft een "logische" terminal aan.

- "CEI, CES"

  Vanuit L3 bekeken, worden de berichten verkregen vanuit de data link layer, geidentificeerd via de CEI (Connection Endpoint Identifier). Dit CEI bestaat uit de SAPI en de CES (Connection Endpoint Suffix). De CES moet uniek zijn binnen een SAP, en heeft als doel connectie endpunt te identificeren. De CES is een een op een afbeelding van de TEI van L2 naar L3. Hoe de CES wordt bepaald en door wie is niet nader door de CCITT gedefineerd (voorlopig aanname is dat de management entiteit dit verzorgt).

- "Het verband tusssen CEI en DLCI"

  In feite duidt CEI en DLCI de zelfde connectie punt aan, alleen CEI is vanuit L3 naar de L3-L2 grens bekeken, en DLCI is vanuit L2 naar L2-L3 grens bekeken.

  Bij informatie overdracht tussen twee L3 entiteiten, zal de DLCI in beide systemen hetzelfde zijn (bijv. de centrale en de abonnee gebruiken beide SAPI 0 en de TEI van een bepaalde terminal); terwijl de CEI verschillend is in beide systemen (vanwege het feit dat aan een terminal maar een TEI waarde toegekend wordt, terwijl de centrale aan een centrale-gebruiker interface meerdere TEI waarden kan uitgeven).De vertaling van TEI naar CES in een systeem zal uniek moeten zijn.

- "USID, TID"

Met de huidige I.451 recommendatie zal bij een inkomende oproep (vanuit de abonnee gezien) het volgende gebeuren: met behulp van de informatie elementen in een SETUP, zullen alleen de terminals die compatibel zijn met de service vereisten reageren (dit houdt in dat niet een speciefieke terminal wordt geadresseerd), aangezien door de S interface een PTMP configuratie is ontstaan. Zonder een extra L3 kenmerk is men niet instaat een specifieke TE bereiken (met de ISDN nummering wordt de T of de S reference punt geidentificeerd, m.a.w een groep terminals wordt geadresseerd).

De USID (User Service ID) en TID (Terminal ID) combinatie wordt door de centrale aan de terminal toegekend bij het inpluggen van de TE, de waarde wordt bewaard in de TE geheugen. Dit L3 kenmerk is uniek in de centrale-gebruiker interface (BRI interface). Nu wordt bij het ontvangen van een SETUP door de TE behalve op de service compatibiliteit ook op de USID en TID gecontroleerd; aangeziend deze combinatie uniek is in de BRI, kan maar een TE hooguit reageren. De reactie van de TE gaat via PTP naar de centrale, de L3 van de centrale kan dan uit de CEI waarvandaan de reactie bericht afkomstig is weer respond bericht versturen.

References_list:

1. [CC 1] CCITT "ISDN protocol reference model", CCITT Rec. I.320
2. [CC 2] CCITT "ISDN user-network interfaces", CCITT Rec. I.410-I.412
3. [CC 3] CCITT "Basic user-network interface", CCITT Rec. I.420
4. [CC 4] CCITT "ISDN user-network interfaces: layer 2 Rec.", CCITT Rec. I.440-I.441
5. [CC 5] CCITT "ISDN user-network interfaces: layer 2 Rec.", CCITT Rec. I.450-I.451
6. [AT 6] AT&T "ISDN user-network interface network specification", AT&T Cross LOB BRI Part IV, september 86.
7. [AT 7] AT&T "Basic rate user-network management and maintenance" Part VII, september 86.
8. [SM 8] Siemens "ISDN userboard-kit STU2000", Preliminary Technical Description
9. [SM 9] Siemens "SBC, PEB2080 Technical Description"
10. [SM 10] Siemens "ICC, PEB2070 Technical Description" version A2, Edition 3/86.
11. [SM 11] Siemens "PBC, PEB2050 Technical Description", part I and part II.
12. [SM 12] Siemens "PBC userboard STU 2050 Technical Description"
13. [AR 13] G.Geiger, L.Lerach, M.Strafner "Peripheral Board Controller for Digital Exchange systems" IEEE journal of solid state circuit, VOL. sc-19 , NO. 4 August 1984.
14. [AR 14] E. Hoppitt "Integrated Digital Access - A step by step Description of the Network Terminating Equipment", Britisch Telecommunications Engineering, Vol 2, Oct.1983.
15. [AR 15] P. Debuysscher, H. Peeters "Intelligent Network termination", Electrical Communication VOL.60 Nr.1 1986.
16. [AR 16] A. Rockstrom "An introduction to the CCITT SDL"

Abbreviation

- 5ESS: (tm) #5 Electronic Switching System
- 64kbps: 64 kilo bits per second
- ARB: Arbiter (process name)
- Ai: Action indicator (used in the TEI assignment request procedure)
- BRI: Basic Rate Interface (i.e the 2B + D interface with bitrate of 144 kbps)
- CC1: Call Control mode1 (process)
- CC2: Call Control mode2 (process)
- CC3: Call Control mode3 (process)
- CCPID: Call Control Process IDentifier
- CEI: Connection Endpoint Identifier
- CES: Connection Endppoint suffix
- CID: Call IDentifier
- CINFO: Call INFOrmation
- CR: Call Reference value
- DB: DataBase
- DDI: Direct Dialling In
- DL: Data Link
- DLCI: Data Link Connection Identifier
- ET: Exchange Termination
- FCS: Frame Check Sequence
- HDLC: High level Data Link Control
- HSCC: High level Serial Communication Controller (tm Siemen's chip)
- HW: High Way
- IA5: International Alphabet Number 5
- IBC: ISDN Burst Controller (tm Siemen's chip)

- ID: IDentifer
- IEC: ISDN Echo Cancellation Circuit (tm Siemen's chip)
- IN1: IN process mode1
- IN2: IN prccess mode2
- IN3: IN process mode3
- IND: INDication (primitive suffix)
- INPID: IN Process IDentifier
- IOM: ISDN Oriented Modular
- ISDN: Intergrated services Digital Network
- L2: Layer2 (i.e Dala link layer)
- L3: layer3 (i.e network layer)
- LAPD: Link Access Protocol D channel
- MDL: Management Data Link (suffix of the service primitive between the management entity and the data link layer)
- MINEX: MINi EXchange
- Mbps: Mega bits per second
- NT: Network Termination
- NT1: Network Termination 1 (i.e functional group 1)
- NT2: Network Termination 2 (i.e functional group 2)
- OSI: Open System Interconnection
- OUT1: OUT process mode1
- OUT2: OUT process mode2
- OUT3: OUT process mode3
- Ocr: Originated call reference value
- Otei: Originated tei value
- PABX: Private Automated Branch eXchange
- PBC: Peripheral Board Controller (tm Siemen's chip)
- PH: PHysical layer (suffix of service primitive between the physical and the data link layer)

- PID: Process IDentifier
- PTMP: Point To MultiPoint
- PTP: Point To Point
- REQ: REQuest (suffix of primitives)
- RESP: RESPonse (suffix of primitives)
- Ri: Request reference number (used in TEI assignment procedure)
- SAPI: Service Access Point Identifier
- SBC: S Bus interface Circuit (tm Siemen's chip)
- SDL: Specification Description Language
- SICOFI: Signal Processing Codec Filter (tm Siemen's chip)
- SLD: Subscriber Line Device (Siemen's Term)
- SLMD: Subscriber Line MoDule (Siemen's Term)
- SPID: Service Profile IDentifier (APT's term)
- TDM: Time Division Multiplex; Trunk MoDule (Siemen's Term)
- TE: Terminal Equipment
- TEI: Terminal Endpoint Identifier
- TEst: TEminal state
- TID: Terminal IDentifier (APT's term)
- TNR: Terminal NumbeR
- TS: Time Slot
- TSD: Time sequences Diagram
- UI: Unnumbered Information
- USID: User Service IDentifier (APT's term)

Discussie_punten_van_de_examen_vergadering_17_sept_87

— Over het gebruiksaspekt en het plaatsing van de MINEX:

De MINEX is bedoeld voor "in house" gebruik, met de mogelijkheid een aantal MINEX eventueel onderling te koppelen, zodat een LAN omgeving wordt gecreeerd. De uitgangspunt was de MINEX te laten functioneren met minimaal de "klavervier" functies.

Verder ondersteunt de MINEX de openbare telefoon centrale diensten met de abonnee; de bedoeling is dat gebruiker de "centrale" diensten tot zijn beschikking heeft,de randvoorwaarde hierbij is dat de gebruiker van de MINEX leidend moet zijn (dus of hij een oproep wel of niet wil aannemen).

Bediening van de toestellen die aangesloten zijn op de MINEX en de noodzakelijke abonnee handelingen voor diverse functies (man-machine interface) zijn niet nader behandeld aangezien deze te terminal afhankelijk is.

De terminals die op de S interface van de MINEX zijn aangesloten dienen zich precies zo te gedragen als een S interface die "direct" (via NT1) met de ISDN centrale gekoppeld is. In alle daarvan in aanmerking komende gevallen zijn de CCITT ISDN specificatie gebruikt.

De MINEX is, indien nodig, in staat voor interne verbinding volledig zelfstandig te werken, in deze situatie zou de openbare ISDN centrale alleen nodig zijn voor de initializatie van de MINEX.

Voor de interne verbinding zal men eerst een bepaalde code moeten kiezen (bijv. door middel van een bepaalde sequence met "*" en "#"), ter aanduiding van de aard van de call; na de code zal de aan de destination TE gekoppelde ID (de Terminal ID) opgegeven moeten worden (Dit was het geval toen we de "subaddress" element nog niet tot ons beschikking hadden, zie ook noot1).

Voor de toepassing geldt: spraak service wordt verleend zolang er nog B kanaal beschikbaar zijn; data service wordt verleend via de D kanaal, maar kan ook plaats vinden via de B kanaal (in dit geval geldt ook de voorwaarde dat een B kanaal beschikbaar moet zijn).

— Over opschakelen op interne verinding:

Als voor de call offering van de interne call de PTMP is gekozen, dan kan de verkeerde (niet bedoelde) terminal kunnen beantwoorden.

Om dit te kunnen voorkomen kan de volgende plaats vinden:

- Van de ontvangen SETUP message, wordt de Terminal ID (een globale TE adres) omgezet in de bij de bestemming TE behorende TEI; dit houdt in dat een PTP configuratie is ontstaan (Dit is de visie toen de "subaddress" element nog niet beschikbaar was, zie ook noot2).

- Betreft het verband tussen TEI, CES, SAPI:

  Dit wordt in annex bijlage (bij dit verslag gevoegd) nader toegelicht.

- Over het aantal terminals per S interface:

  Er kunnen meer dan 2 terminals per interface aangesloten en geinitialiseerd worden, maar er kunnen er slecht 2 van tegelijkertijd gebruikt (m.b.t B kanaal) worden. Deze situatie zou zinvol kunnen zijn bij het gebruikt van verschillende type terminals (dit betreft uiteraard het gebruik van circuit switched diensten).

• noot1:

  * Het "subaddress" informatie element was nog niet opgenomen in de (Study period 1981-1984) CCITT REC. I.451, maar zal zo goed als zeker als een functionele element opgenomen worden in de volgende periode (1985-1988). De centrale zal het "subaddress" informatie element transparant doorgegeven aan de gebruiker, en zal dit element niet als routering informatie gebruiken. Dit "subaddress" is alleen bekend bij de gebruiker.

  Het is dan mogelijk om via de call SETUP een specifieke terminal te selecteren.

  * Alle terminals aangesloten op de MINEX hebben van buiten gezien een nummer (dit ISDN nummer identificeert dus een referentie punt "T"); voor het selecteren van een specifieke terminal zal gebruikt gemaakt moeten worden van de volgende informatie elementen in het SETUP bericht: "subaddress", "low layer compatibility" en "high layer compatibility".

• noot2:

  * In geval dat men de beschikking heeft over het "subaddress" L3 informatie element wordt het eenvoudiger; immers, op L2 kan men de broadcast faciliteit (PTMP) gebruiken, nu zal alleen die terminal die door de "subaddress" wordt aangeduid reageren met respond berichten (PTP).