Protocol Specification, Tes and Verification, VII
H. Rudin and C.H. West (E s)
Elsevier Science Publishers B.V. (North-Holland)

PROTOCOL TESTING FOR THE ISDN D-CHANNEL NETWORK LAYER

ERWIN P. RATHGEB

Institute of Communications Switching and Data Technics

University of Stuttgart, FRG

CHRISTIAN HOMANN, HONG LINH TRUONG, GERD WALDMANN

Standard Elektrik Lorenz AG, Stuttgart, FRG

In this paper the specification and documentation of a set of abstract, implementation independent test cases for the network layer of the user-network signalling protocol for an Integrated Services Digital Network (D-channel protocol) is presented. The tests cover the full range of the protocol as specified for the pilot service of the Deutsche Bundespost. In addition to the abstract test cases the major aspects of their implementation on a specific test system, the "D-Channel Simulator", are presented together with a trace resulting from the execution of an example case.

1. INTRODUCTION

In recent years the increasing demand for a wide variety of communication services (e.g. voice, data, text, facsimile, etc.) made it necessary to look for an economic and flexible way to satisfy these needs. So, as an alternative to various networks dedicated to one single type of service, the concept of an Integrated Services Digital Network has been developed.

The objective that ISDN communication in the future shall be possible all over the world has created an urgent need for international standardization, which during the last years has been done mainly by the CCITT [1].

Such an ISDN, providing two circuit switched B-channels at a speed of 64 kbps each plus a separate, packet oriented channel for signalling purposes, the D-channel (16 kbps), per subscriber access, is about to go into a pilot service in Germany by the beginning of 1987. From 1988 on, regular ISDN service will be started so that within a few years the ISDN will be available throughout Germany [13].

For the pilot service a series of technical specifications has been pul ed by the Deutsche Bundespost (German PTT) defining interface protocols, service fea...es, hardware equipment etc. One of these specifications, the FTZ standard 1R6 [3] defines the so called **"D-channel protocol"**, the user-network signalling protocol for ISDN.

This paper discusses the testing of the D-channel network layer protocol implemented in exchanges and user equipment developed for the pilot service.

After a short description of the D-channel protocol itself and the introduction of the common terminology for protocol testing, the specification of a **set of predefined, implementation independent test cases for the network layer** will be presented.

Then, the **"D-channel simulator"**, a test system for conformance testing of D-channel protocol implementations developed by the Standard Elektrik Lorenz AG, will be described. Some aspects of the implementation of the tests on the simulator and some example test traces will conclude the paper.

2. THE D-CHANNEL PROTOCOL

Subject of this section is a short description of the D-channel protocol used in the pilot service. Its specification, the FTZ standard 1R6, has been developed in close accordance with the CCITT results known in December 1983. Further information about the protocol and its differences to the CCITT protocol defined in the Red Books can be found in [6].

Figure 1 shows the reference configuration for the user basic access used in the ISDN pilot service.

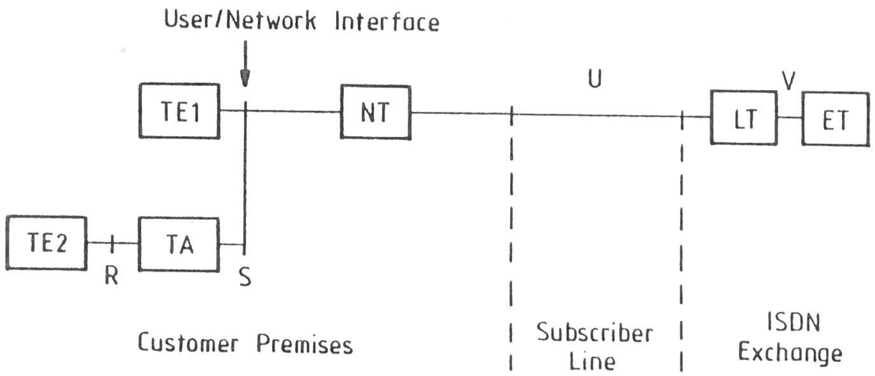

Figure 1 : The reference configuration for the ISDN pilot service

The ET (*exchange termination*) implements the network functions on the exchange side. The LT (*line termination*) and the NT (*network termination*) are responsible for sending and receiving signals with a net bit rate of 144 kbps on the two-wire copper subscriber line. In addition they are responsible for clock, synchronization and remote feeding.

The **S-interface**, which is a passive bus using four-wire copper cable, provides user access to the ISDN. Up to 8 TEs (*terminals*) can be connected to the S-bus. In detail, the S-interface offers to the connected terminals three duplex channels, namely two 64 kbps circuit switched B channels for the tr nission of user information and one 16 kbps packet oriented D-channel for the exchange of user-network signalling information. TEs with conventional interfaces (e.g. X.25, X.21) can be connected to the S-bus via a TA (*terminal adapter*) .

According to the OSI reference model, the signalling protocol used in the D-channel includes only the three low layers, namely the physical layer, the data link layer and the network layer.

In **layer 1**, besides the specification of the electrical and physical characteristics of the S-interface, a multi-access method with a contention resolution mechanism is defined, allowing the common use of the D-channel by all terminals connected to the bus.

The D-channel **layer 2**, also called LAP D (*Link Access Procedure on the D-channel*), can be seen as an adaption of LAP B, the link access procedure of X.25, to the special features of the ISDN basic access. Because of the bus structure of the S-interface, LAP D supports the parallel operation of more than one data link connection over a single D-channel by using the address multiplexing method, i.e. each data link connection is identified by a *data link connection identifier* carried in the address field of each layer 2 frame. The D-channel layer 2 also offers to the higher layers a broadcast data link, by means of which a message can be broadcasted to all terminals connected to the S-bus.

In **layer 3** of the D-channel protocol user-network signalling procedures are specified. By these procedures, the ISDN user is provided with the means to establish, maintain and terminate circuit switched connections on the two B-channels. Procedures for packet switched connections are not yet defined because this type of connection is not offered in the pilot service.

The call control procedures specified in the standard 1R6 are very close to those defined in the CCITT recommendation Q.931. However, a substantial part of the 1R6, which can not be found in the international recommendation, is the handling of user features such as conference calls, call diversion services, call waiting, barring services, change of service during a connection, etc. Not only these features, but also the signalling procedures for their use are fully specified.

This short description makes obvious, that the signalling protocol for the network layer is very complex. As a consequence, there is an urgent need for testing means allowing both, manufacturers and administrations, to test the signalling functions implemented in the various ISDN components prior to their use in the network.

3. CONFORMANCE TESTING

3.1 *Basic protocol mechanisms and test methods*

With the development of more and more complex communications protocols, the aspect of conformance testing became increasingly important and efforts were made to standardize conformance testing [4]. In this context generic, abstract test methods have been identified [10].

Figure 2 shows the model of a layered protocol architecture with a **protocol entity** existing in each layer. Protocol entities of the same layer, e.g. layer (N), communicate logically among each other according to the **peer-to-peer protocol** exchanging (N)-PDUs (*protocol data units*).

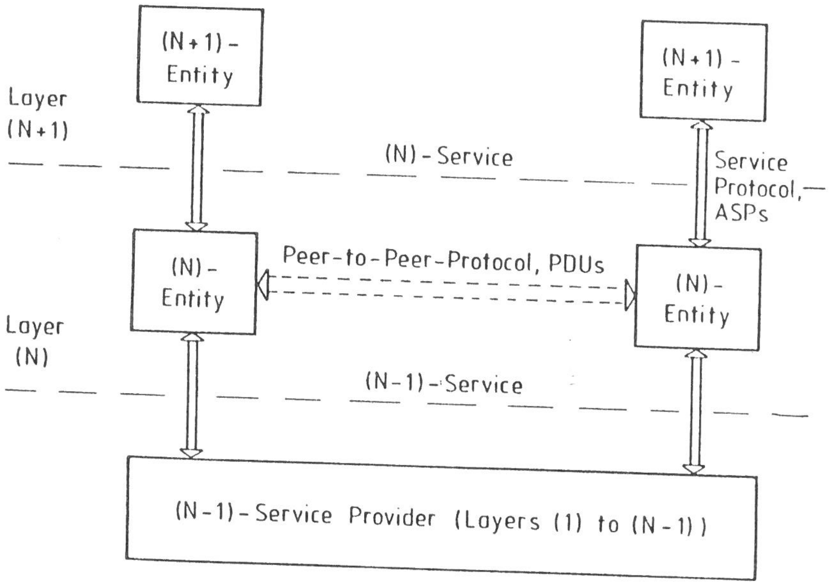

Figure 2 : Model of a layered protocol architecture

The transport of the (N)-PDUs is accomplished using the service provided by the underlying protocol layers from (1) to (N–1). The use of the (N–1)-service is controlled by the layer-to-layer **service protocol** and achieved by the exchange of (N–1)-ASPs (**abstract service primitives**) consisting of adjacent layer control information and (N–1)-SDUs (*service data units*). The (N–1)-SDUs contain the (N)-PDUs in their data section, whereby it is possible, that only a part of a PDU or more than one PDU is carried by one SDU.

To test the conformance of a (N)-entity, called EUT (*entity under test*), to the protocol specification, its behaviour at the upper and lower interface has to be observed and influenced (black box testing). Depending on the location of observation and control points, it is possible to identify the different test methods. For the DSE method (*distributed single layer embedded*), on the EUT side access for testing is only provided at the upper interface of the highest protocol layer. On the test system side, access is required to the ASPs of all layers. With this method an IUT (*implementation under test*) can be tested successively, layer by layer, beginning with the lowest layer.

Testing can be performed to various extent and therefore for conformance testing 4 categories have been defined. These are basic interconnection tests, functional range tests, full conformance tests and specific conformance resolution tests [10].

In addition to conformance testing it is possible to test the layer-to-layer service, the performance of an implementation and the robustness of an implementation against protocol violations. The latter three aspects, although not being subject of conformance testing as defined in [10] in accordance with an ISO "*working draft for OSI conformance testing methodology and framework*" [4], have nevertheless to be taken into account for the final assessment of an implementation.

3.2 *Realization of a test system*

A real test system generally consists of two parts called **upper tester** and **lower tester** according to the the interface of the EUT they control.

The lower tester is built of two parts:

- The **test driver** (*in layer (N+1) for a (N)-EUT*)
- An entity capable to handle the (N)-protocol.

The test driver does the test management and may in the simplest case only be a terminal for manual input of instructions and display of system reactions. For extensive, repeated tests, an automatic test driver is more suitable. This automation can be achieved using **scenario files**, where all actions during a test are predefined. A drawback of this approach is the lack of flexibility during test execution. This shows especially if nondeterminisms in the protocol have to be handled. An alternative approach is the use of an **extended finite state machine** as test driver, implementing e.g. the (N+1)-protocol.

The (N)-entity of the lower tester has to convert the instructions from the test driver into valid (N)-protocol. This entity could be a **reference implementation** of the layer (N) functionality. Obviously a reference implementation can only produce valid protocol sequences. Thus, to test the reaction of the EUT to erroneous messages or message sequences an **error generator** has to be added. An alternative providing full flexibility is the use of a simple **Encoder/Decoder** for (N–1)-ASPs. The drawback here is, that tests have to be specified down to the level of the (N–1)-ASPs, what leads to more voluminous test specifications.

As upper tester might well serve just the user interface of the tested system operated manually or a device similar to the test driver.

Upper tester and lower tester have to be synchronized to ensure proper operation. Synchronization may range from a very loose form applied during manual operation to highly sophisticated mechanisms involving special **test connections** for synchronization in addition to the **tested connection** or the **Ferry-concept** [15].

4. TEST SPECIFICATION

4.1 *Basic considerations*

The prime objective of the project was to design a full conformance test suite for the D-channel network layer as defined in the standard 1R6, which allows the assessment of an implementation used in an ISDN component. For this test suite in the first place an abstract, implementation independent test specification had to be developed.

The specification in the 1R6 is a verbal one, so it would not have been possible to use a method for the formal derivation of the tests even if such a method had been available, taking into account the complexity of the protocol and deadline and expense requirements.

Another consequence of the verbal specification is, that the tests had to be based on the functionality described there rather than upon defined states and transitions, because the latter approach requires a formal protocol specification that is a mandatory implementation rule as well.

Considering the specific features of the protocol and the fact, that the ISDN components aimed at were TEs as well as ETs and PABXs, four basic, abstract test configurations for the network layer have been identified, each of them requiring its own, specific conformance test suite.

The four basic test configurations are:

a) Tester acts as an ET, the IUT is a TE implementation.

b) Tester acts as one or several TEs, the IUT is an ET implementation.

c) Tester acts as an ET, the IUT is an implementation of the trunk side of an ISDN PABX.

d) Tester acts as an PABX, the IUT is an ET implementation.

It shall be emphasized here, that it is not possible to get e.g. a conformance test suite for configuration b) by simply exchanging tester and IUT in the test suite for configuration a) . This is due to the asymmetric structure of the protocol, having its origin in the point-to-multipoint configuration of the S-bus. In the case of the configurations c) and d) only a point-to-point configuration is used, but there is still enough asymmetry inherent to the protocol to make it impossible to define a useful test suite common to both configurations.

Another important aspect for our considerations has been, that the behaviour of the protocol in the different phases (mainly connection establishment phase, data transfer phase and connection release phase) does not depend on how the specific phase has been reached, as long as all relevant parameters, e.g. type of service, addresses etc., are known. This allows to test the phases independent of each other and to use a straight forward, standard sequence of test events to reach the initial, stable state of each phase. On the other hand, interdependency of the phases would have required the consideration of all combinations of all the alternatives in the phases.

Furthermore, we observed that the interdependence between the calling and the called party side of a connection can be neglected in most cases (exceptions will be discussed later) for the tests, allowing to split up the tests for both sides to make them less complex and also to reduce the number of combinations.

Attention had also to be devoted to the relations between the **test objectives** (i.e. the functions to be tested) and the **test cases** (sequences of test events) used to determine whether the EUT offers the functionality required. One possible approach is to try to pack as many test objectives into one test case as possible to reduce the number of test cases. The problem with this method is to determine what failure in such a multi-purpose test means. For the designer of an ISDN component it is more desirable to have a one-to-one relation between test objectives and test cases to be able to also use the tests for debugging purposes to a limited extent (although this is not the intention behind the design of conformance tests). The latter method makes it also possible to perform separate tests for independent functions, which otherwise might be combined in one test case, even if the test of one function fails.

To a certain extent tests to proof the ability of an implementation to recover from protocol violations should also be made available. These robustness tests are particularly important for an exchange in a public network.

Based on the results of these considerations, the specification of the test cases took place in two steps. First, working through the protocol specification, the test objectives have been identified systematically. In a second step, the sequence of test events corresponding to each test objective has been defined. In the following, major aspects of the resulting set of test cases and their representation will be presented.

4.2 *Classification of the test cases*

As mentioned previously, the first aspect for the classification of the tests is the test configuration they are aimed at. The further classification has been made as follows:

- **Tests for basic functions** These tests shall show the ability of an implementation to receive and interprete a correctly coded PDU. Furtheron, it is tested whether faulty PDUs are recognized and handled as specified in the protocol. These tests form the basis for all others and have to be executed prior to them.
- **Tests for the calling side** This section includes tests where the IUT is situated on the calling side of a connection. It is subdivided in successful call establishment, incomplete calls and call release and it provides test cases for detailed testing of the numerous alternatives during these phases.
- **Tests for the called side** This section provides the tests for an IUT situated on the called side of a connection. In addition to the sections mentioned for the calling side, the effects of more than one TE answering a call are included here.
- **Tests for the use of service facilities** This section includes testing for all features of the D-channel protocol exceeding basic call handling as known e.g. from normal telephone service. Interdependencies among different service facilities are also considered here.
- **Complex tests** These tests are only relevant for an ET implementation and focus on the communication between the ETs on both sides of a connection. These tests are called complex tests, because they require the ET network relay functions in order to be executed. They are suited for the loop-back as well as for the transverse testing approach [10] depending on the actual test configuration.

It is evident, that the tests for the basic functions have to be executed before all others and that the tests for the use of the service facilities and the complex tests depend on the successful completion of the tests for the basic call handling.

Furthermore, all sections, subsections and test cases are organized in such a manner, that the numerical order of the tests at the same time expresses a recommendation for the order in which the tests should be performed.

4.3 *The documentation for the abstract test specification*

The intention was, that the documentation for the tests should be as compact and as easy to understand as possible. On the other hand, it had to provide all relevant information for the implementation and execution of the tests.

In our case, the documentation for each test case consists of a verbal part and a precise description of the sequence of test events and EUT reactions expected.

The verbal part includes, in addition to the title of the test case and its serial number, a detailed description of the test objective and a list of special conditions and requirements for the test to be successfully executed. It shall be mentioned here, that the initial state of all tests is the idle state of the EUT, which means that a connection release test also starts with a connection establishment sequence. The reason for this redundancy is the great number of (interdependent) parameters in the protocol. This would make it very difficult to fully specify the starting conditions in any other state of the EUT. The verbal part also includes, where necessary, remarks and explanations concerning the description of the test event sequence.

The description of the sequence of test events, as shown for a simple example in figure 3, is expressed in terms of layer 3 PDUs including all mandatory elements in the order specified in the 1R6. Optional message elements are not considered as long as they have no influence on the progress of the test.

```
1.          TESTER AS EXCHANGE TERMINATION, IUT TERMINAL EQUIPMENT
1.2.        CALLING SIDE
1.2.1.      SUCCESSFUL CALL ESTABLISHMENT
1.2.1.3.    ADDRESS OF B-SUBSCRIBER IN SETUP AND INFO MESSAGES
-----------------------------------------------------------------------
SEQUENCE OF TEST EVENTS

TESTER AS EXCHANGE TERMINATION              IUT TERMINAL EQUIPMENT
-----------------------------------------------------------------------
                                    <--  PD =I, CR-1 = (0,cr1), MT = SETUP,
                                         OAD = (2, toa, npi, oaz),
                                         DAD-1,
                                         SIN
PD = I, CR-1, MT = SETUP ACK,        -->
CHI = (1, ics)
                                     |
                     max.  T302      |
                                     v
                                    <--  PD = I, CR-1, MT = INFO,
                                     |   DAD-2
                                     |
                     max.  T302      |
                                     v
PD = I, CR-1, MT = CONN,             -->
DTE

                                    <--  * PD = I, CR-1, MT = CONN ACK
                                         (B-channel ics must be assigned to TE)
```

Figure 3 : Example for a test case description

Parameters that do not change the sequence of test events if their value is altered are represented as variables, as e.g. parts of the origination address (OAD), the two parts of the destination address (DAD-1, DAD-2) and the service indicator (SIN) in the SETUP message in figure 3. To design a functional range test suite all these variable parameters may be varied over the full range specified in the 1R6 without influencing the sequence of the following test events.

In contrast, there are parameters that have to be set to a specific value in order to get exactly the series of events necessary for a particular test case. These parameters are represented by their mandatory value. For example, the protocol discriminator (PD) has to have the value "I" in all PDUs of the example to get a valid protocol sequence.

Timing requirements are represented by the timer name (defined in the 1R6) and an arrow between the messages that start and stop the timer, respectively. In the example, the EUT has to supply the second part of the destination address in an INFO message within the time given by T302 after the tester has sent the acknowledgement for the SETUP message (SETUP ACK) in order to meet the requirements and pass the test.

On the other hand, a real time tester must be able to send a connect message (CONN) within T302 after the reception of the INFO message.

In most protocols there exists a certain amount of nondeterminism, where reactions of IUTs may be different and implementation dependent, but well within the valid protocol range. An example is the connect acknowledgement message (CONN ACK) that a TE may or may not send. PDUs of this type are marked with an "*" and their absence must not result in a failure of the test. However, the presence or absence of an optional message may influence the timing conditions and has to be considered there, too.

The ET on the other hand must be able to handle both alternatives. Thus, for ET testing there have to be two test cases, one where the TE sends CONN ACK and one where the TE does not, and the ET has to pass both of them.

It is useful for the test staff if some informations not directly relevant for the protocol, but related to the function of the whole component are included. So in the example, after the test the B-channel with number "ics" (either B-1 or B-2, see fig. 3) has to be allocated to the terminal.

The number of test cases specified this way adds to a total amount of about 500 and provides, in our opinion, a sufficient test coverage for the full range of the network layer of the D-channel protocol for the pilot service as specified in the FTZ standard 1R6.

5. THE TEST SYSTEM

The test cases described above have already been implemented on a special test system, the so called "**D-Channel Simulator**" [5], which was designed to test all ISDN components that can be connected to the S-interface.

The D-Channel Simulator supports all four configurations described in section 4.1, which means it can be used to test terminals and terminal adapters as well as ETs and PABXs (on the trunk side). For ET tests, the simulator is able to simulate in real time the maximum configuration for one basic access with 8 terminals connected to the S-bus. This property allows loop-back testing for the ET as well.

The simulator has the capability to handle the full range of the layers 1 to 3 of the D-channel protocol and provides full control of all protocol parameters of these layers. Therefore, it allows to simulate both, valid and erroneous behaviour of an ISDN component. For testing of the layers above layer 1, the full service of the underlying protocol layers is provided by the simulator automatically.

5.1 *The hardware configuration of the D-Channel Simulator*

The main components of the D-Channel Simulator are a general purpose minicomputer with specific software and a **Simulator Probe** to connect the simulator to the S-interface as shown in figure 4.

The minicomputer is a MICRO PDP-11/73 with 1 MByte main memory, a Winchester disk, a streamer tape and standard peripherals. A PDP-11/23 coprocessor has also been added to do the layer 2 handling.

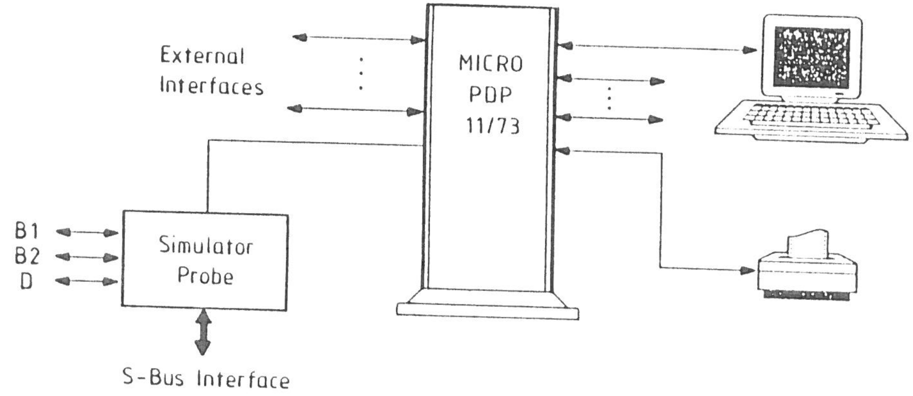

Figure 4 : The hardware configuration of the simulator system

Additional standard interfaces allow to connect and control external measuring equipment and to retrieve data from it. These interfaces could also be used to connect an appropriate test responder for automatic testing.

The Simulator Probe does not only provide the physical and electrical connection to the S-bus, but also offers the two B-channels and the D-channel at separate interfaces allowing e.g. to connect standard protocol analyzers for monitoring purposes.

5.2 *The software structure of the D-Channel Simulator*

On top of the standard operating system, RSX-11-M Plus, a fully menu driven software system has been built to support implementation, storage, execution and documentation of protocol tests.

The software is designed to handle simulation of a full basic access with 8 TEs, allowing 8 layer 2 connections per layer 1 connection. It is furthermore possible to activate up to 10 layer 3 connections per layer 2 connection.

Figure 5 shows the general software structure of the simulator application system.

For each layer there exists a **Virtual User** to execute predefined test cases. The Virtual User provides the user with the means to control the protocol parameters of the corresponding layer, to define the PDUs and to send and receive them. The test cases for the Virtual User are defined by means of **Scenario files**, but with inherent decision capabilities, achieved by the use of conditional branching, to provide more flexibility during test execution. Syntax and semantic checks are performed during the implementation of the test cases, which can be stored in a data base for repeated execution. The specification of the test cases is done in terms of mnemonics which are derived directly from the terminology of the standard 1R6.

The **Preevaluators** of each layer perform the decoding of received PDUs.

The **Control Processes** are reference implementations for the physical and data link layer

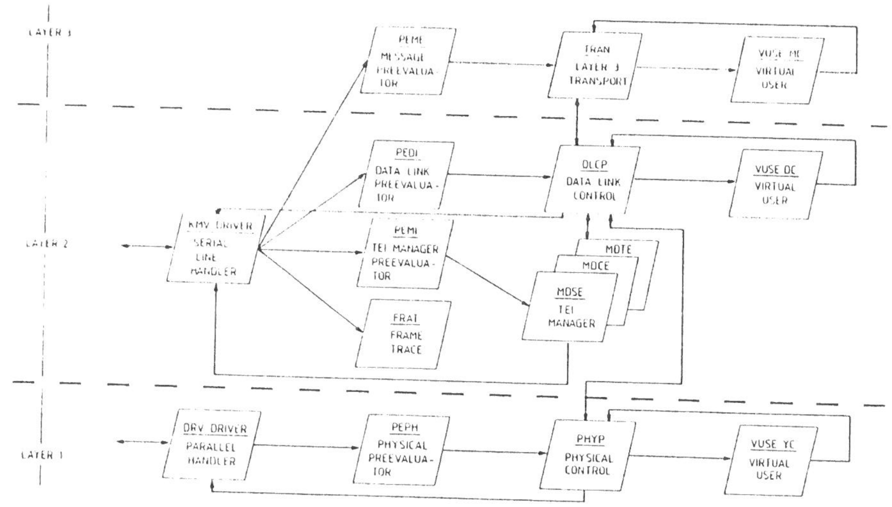

Figure 5 : The software structure of the simulator application system

and for the layer 3 message transport mechanism. On layer 2, TEIs (*terminal endpoint identifiers*) have to be assigned to the terminals by the ET upon request in order to be able to identify the TEs during a communication. As a consequence, in addition to the data link access procedure, a **TEI Management Procedure** with its own preevaluator has been implemented.

To keep track of all events and activities during a test session a **Report System** has been implemented, which is able to generate test traces at various levels of detail. The **Frame Trace Handler** shown in figure 5 is needed to support the Report System.

Simulator Specific Drivers are handling the interfaces to the standard operating system.

The **Application System** is embedded in a **Basis System** that includes the task scheduler and general service functions as e.g. menu handlers and data base access procedures.

This overview shows, that the simulator provides all capabilities needed to perform testing of the D-channel protocol and is especially well suited for the incremental embedded single layer approach described in section 3.1.

5.3 *The implementation of the test cases*

For the test case implementation several separate menus are used. Therefore e.g. separate parameter files are created, which are especially useful for functional range testing purposes.

The main scenario definition menu for the test case example from section 4.3 is shown in figure 6. A single test case is called "block" there and there is a block identification (BLID) assigned to it, which allows the concatenation of several test cases to a test suite.

The simulator remains in a "Previous State" (PST) until an event "Incondition" (INCOND) occurs. In the INCOND-column the PDUs expected from the EUT as well as the relevant timers can be identified. After the occurance of an input event the simulator creates an

	P	B	BLID	PST	INCOND	CO	OUTPAR	CO	NST	D E S C R I P T I O N
00100	0		0003			000	ATTENT	000		=====================
00101	0		0003			000	ATTENT	152		Scenarioname, BLID
00102	0		0003			000	ATTENT	100		<TEXT TO BLID>
00103	0		0003			000	ATTENT	101		<TEXT TO BLID>
00104	0		0003			000	ATTENT	102		<TEXT TO BLID>
00105	0		0003			000	ATTENT	103		<TEXT TO BLID>
00106	0		0003			000	ATTENT	104		<TEXT TO BLID>
00107	0		0003			000	ERRSTA	000	V58:	Set error state
00108	0		0003			000		000	V0:	
00109	0		0003	V0:	SETUP	006	ATTENT	001	V1:	SETUP received
00110	0		0003	V1:		000	TIMER	000	V7:	Delay for SETUP ACK
00111	0		0003	V7:	TIMER	000	SETACK	002	V3:	send SETUP ACK
00112	0	B	0003	V7:	TIMER	000	ALERT	000	V4:	send WRONG MESSAGE
00113	0	B	0003	V7:	TIMER	000		000	V5:	send NO MESSAGE
00114	0		0003	V4:		000	ATTENT	185	V6:	Wrong message sent
00115	0		0003	V5:		000	ATTENT	186	V6:	No message sent
00116	0		0003	V3:		000	ATTENT	055	V14:	SETUP ACK sent
00117	0		0003	V14:		000	TIMER	012	V16:	T302: OBSERVE INFO
00118	0		0003	V16:	TIMER	000	ATTENT	188	V58:	NO MESSAGE RECEIVED
00119	0		0003		INFO	007	ATTENT	011	V20:	INFO RECEIVED
00120	0		0003		ANY	000	ATTENT	187	V58:	WRONG MSG RECEIVED
00121	0		0003	V20:		000		000	V30:	JMP TO SEND CONN
00122	0		0003	V30:		000	TIMER	000	V32:	MAX. T302
00123	0		0003	V32:	TIMER	000	CONN	003	V34:	send CONN
00124	0	B	0003	V32:	TIMER	000	ALERT	000	V36:	send WRONG MESSAGE
00125	0	B	0003	V32:	TIMER	000		000	V38:	send NO MESSAGE
00126	0		0003	V36:		000	ATTENT	185	V59:	Wrong message sent
00127	0		0003	V38:		000	ATTENT	186	V59:	No message sent
00128	0		0003	V34:		000	ATTENT	053	V40:	CONN SENT
00129	0		0003	V40:		000	TIMER	030	V42:	OBSERVE CONN ACK
00130	0		0003	V42:	TIMER	000	ATTENT	182	V57:	NO MESSAGE RECEIVED
00131	0		0003		CONNACK	000	ATTENT	006	V57:	CONN ACK RECEIVED
00132	0		0003		ANY	000	ATTENT	187	V58:	WRONG MSG RECEIVED
00133	0		0003	V57:		000	ATTENT	179	V59:	TE KANAL ICS
00134	0		0003	V58:		000	ATTENT	198	V59:	Error occurred ?
00135	0		0003	V59:		000	TIMER	010	V60:	Delay
00136	0		0003	V60:	TIMER	000		000	V61:	
00137	0		0003	V61:		000	ERRSTA	000	V62:	Set error state
00138	0		0003	V61:		000	CLRCR	000	V62:	Clear CR
00139	0	B	0003	V61:		000		000	V62:	
00140	0		0003	V62:		000	ATTENT	199	V62:	End of scenario

Figure 6 : Implementation of a test case example

output event (OUTPAR) and goes to a "Next State" (NST) depending on the input event. It is possible to do conditional branching and to build loops or to use timers to delay the progress of the test. The "ATTENT" messages are used to insert specific comments into the test trace. Errors during test execution force the scenario to proceed to an error state (ERRSTA) and to terminate test execution. Since not all test cases end with the EUT in its idle state, a reset is made by clearing the call reference (CLRCR). The "Composers" following the input conditions and output events point to the corresponding menus, where e.g. the PDU elements and their parameter values or the actual texts for ATTENT-messages are defined in detail.

As an example, figure 7 shows the trace for a successful completion of the test case described in section 4.3. In this trace only the information relevant for layer 3 is included. However, all information about the underlying layers could have been included just by changing the trace parameters.

In addition to the mnemonics of the PDUs sent and received, their hexadecimal representation is being displayed. To communicate with layer 2 the SDUs beginning with "DL-" are


```
ETS    SUMMERY:MESSAGE BUFFER 0                26-JAN-87 10:51:50 PAGE:001

         123456789.123456789.123456789.123456789.123456789.123456789.123456789.1234
00495      DATA 10:48:45:88 PORT 0: INITIALIZE DATA
00496   1  VUAT 10:48:47:50 =========================================
00497   1  VUAT 10:48:47:50 Scenario = MCET01      : BLID =   3
00498   1  VUAT 10:48:47:50 1.          TESTER AS ET, IUT TE
00499   1  VUAT 10:48:47:50 1.2.        CALLING SIDE
00000   1  VUAT 10:48:47:50 1.2.1       SUCCESSFUL CALL ESTABLISHMENT
00001   1  VUAT 10:48:47:50 1.2.1.3.    ADDRESS OF B-SUBSCRIBER IN
00002   1  VUAT 10:48:47:50             SETUP AND INFO MESSAGES
00003   1  TRAN 10:48:55:84 DL-EST-RQ:   CR:---
00004      TRFI 10:48:55:96 0E000040 00190081 00080101 056C0281 31700581 34373131
00005                       9E010201 01
00006   1  PEME 10:48:55:96 DL-DAT-IN: MT = SETUP    PD = I    CR = 001
00007                       OAD: 1, ADRTYPE = 0, ADRPLAN = 1
00008                       DAD: 4711, ADRTYPE = 0, ADRPLAN = 1
00009                       SFT: SET = 6, LOCK = 1
00010                       SIN: ISDN-FERNSPRECHEN
00011   1  VUAT 10:48:56:24 SETUP received
00012   1  TRAN 10:48:57:04 DL-DAT-RQ: MT = SETACK   PD = I    CR = 001
00013                       CHI: ICS = 1, P/E = 1
00014                       DSP: Rufnummer:
00015      TRFQ 10:48:57:10 0E000040 00150801 010D1801 89280A52 75666E75 6D6D6572
00016                       3A
00017   1  VUAT 10:48:57:70 SETUP ACK sent
00018      TRFI 10:48:58:70 0E000040 000D0081 22080101 6D700281 31
00019   1  PEME 10:48:58:70 DL-DAT-IN: MT = INFO     PD = I    CR = 001
00020                       DAD: 1, ADRTYPE = 0, ADRPLAN = 1
00021   1  VUAT 10:48:58:84 INFO received
00022   1  TRAN 10:48:59:18 DL-DAT-RQ MT = CONN      PD = I    CR = 001
00023                       DSP: VERBINDUNG
00024                       SFT: SET = 6, LOCK = 1
00025                       DTE: 26.03.83-09:42
00026      TRFQ 10:48:59:24 0E000040 00230801 0107280A 56657262 696E6475 6E679E03
00027                       0E32362E 30332E38 362D3039 3A3432
00028      TRFI 10:48:59:90 0E000040 00090081 44080101 0F
00029   1  PEME 10:48:59:90 DL-DAT-IN: MT = CONACK   PD = I    CR = 001
00030   1  VUAT 10:49:00:10 CONNECT sent
00031   1  VUAT 10:49:00:10 CONN ACK received
00032   1  VUAT 10:49:00:10 B-channel ics must be assigned to TE
00033   1  VUAT 10:49:11:06 END OF SCENARIO
00034   1  TRAN 10:49:41:06 DL-REL-RQ:   CR:---

     INFO 10:52:06:06 SUMMARY LOG COMPLETED


SUMMARY LOG COMPLETED     26-JAN-87   10:52:06
```

Figure 7 : Test trace for the test case example

used. With the "DL-EST-RQ" and the "DL-REL-REQ" primitives the layer 2 connection is established and deactivated, respectively. The "VUAT" identifier in the second column marks the texts created by the ATTENT-messages.

6. CONCLUSION AND OUTLOOK

In this paper the development and specification of a set of abstract, implementation independent test cases for the D-channel network layer has been presented together with the major aspects of their implementation on a specific test system, the D-Channel Simulator.

The test cases described here have been delivered to the Deutsche Bundespost and will be used during the ISDN pilot service. If they prove to be useful, they might build a basis for the assessment of ISDN protocol implementations in the future.

With the practical experience gained in the pilot service it should be possible to improve and extend the tests where needed and to define appropriate subsets to fulfil the specific testing needs of administrations and manufacturers.

REFERENCES

[1] CCITT Red Book, Vol. III, Fascicle III.5, "Integrated Services Digital Network (ISDN), Recommendations of the Series I", Geneva 1985.

[2] Choi T.Y., "Formal Techniques for the Specification, Verification and Construction of Communication Protocols", IEEE Communications Magazine, Vol. 23, No. 10, oct 1985, 46-52.

[3] FTZ Richtlinie 1R6, "Kennzeichenaustausch zwischen DIVO(ISDN)-Vermittlungsstellen und ISDN-Teilnehmereinrichtungen – ISDN-D-Kanal-Protokoll –", Ausgabe 2.86.

[4] ISO/TC97/SC21/WG 16-1, Working draft for "OSI conformance testing methodology and framework", ISO/TC97/SC21 N410, Paris, February 1985. (ISO DP 9646/1 and DP 9646/2 since September 1986)

[5] Kirchner W., Wagner P., "ISDN D-Channel Protocol Simulator", Electrical Communication, Vol. 61, No. 1, 1987.

[6] König W., Truong H.L., Waldmann G., "D-Kanal Protokoll im ISDN-Pilotprojekt", NTG-Fachberichte 88, 1985, 112-119.

[7] Linn R.J., Nightingale J.S., "Testing OSI Protocols at the National Bureau of Standards", Proc. of the IEEE, Vol. 71, No.12, Dec. 1983, 1431-1434.

[8] Pavel J.R., Dwyer D.J., "Some Experiences of Testing Protocol Implementations", 4th Int. Workshop on "Protocol Specification, Testing and Verification", June 1984.

[9] Rafiq O., "Tools and Methodology for Testing OSI Protocol Entities", IEEE Int. Symposium on Fault Tolerant Computing, June 1985.

[10] Rayner D., "Standardizing Conformance Testing for OSI", 5th Int. Workshop on "Protocol Specification, Testing and Verification", June 1985.

[11] Sarikaya B., Bochmann G., "Some Experience with Test Sequence Generation for Protocols", 2nd Int. Workshop on "Protocol Specification, Testing and Verification", May 1982.

[12] Sabnani K., Dahbura A., "A New Technique for Generating Protocol Tests", ACM SIGCOMM Computer Communication Review, Vol. 15, No.4, Sept. 1985, 36-43.

[13] Schön H., "ISDN — A Challenge for the Deutsche Bundespost", ISS '84, Florence, May 1984.

[14] Sherif M.H., Hoover G.L., Wiederhold R.P., "X.25 Conformance Testing — a Tutorial", IEEE Communications Magazine, Vol. 24, No.1, January 1986, 16-27.

[15] Zeng H.X., Rayner D., "The Impact of the Ferry concept on Protocol Testing", 5th Int. Workshop on "Protocol Specification, Testing and Verification", June 1985.