ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Reference number
ISO/IEC 9314-25:1998(E)

PRICE CODE XP

*For price, see current catalogue*

# Contents





## Tables

**Figures**

# FOREWORD

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9314-25 was prepared by Joint Technical Committee ISO/IEC JTC 1 *Information technology*, Subcommittee SC 25, *Interconnection of information technology equipment.*

ISO/IEC 9314 consists of the following parts, under the general title *Information technology – Fibre Distributed Data Interface (FDDI):*

- *Part 1: Token Ring Physical Layer Protocol (PHY)*
- *Part 2: Token Ring Media Access Control (MAC)*
- *Part 3: Physical Layer Medium Dependent (PMD)*
- *Part 4: Single Mode Fibre Physical Layer Medium Dependent (SMF-PMD)*
- *Part 5: Hybrid Ring Control (HRC)*
- *Part 6: Station Management (SMT)*
- *Part 7: Physical Layer Protocol (PHY-2)*
- *Part 8: Media Access Control-2 (MAC-2)*
- *Part 9: Low-Cost Fibre – Physical Medium Dependent (LCF-PMD)*
- *Part 10: Token Ring Twisted Pair Physical layer Medium Dependent (TP-PMD)*
- *Part 13: Conformance Test Protocol Implementation Conformance Statement proforma (CT-PICS)*
- *Part 20: Physical Medium Dependent Conformance Testing (PMD-ATS)*
- *Part 21: Physical Layer Protocol Conformance Testing (PHY-ATS)*
- *Part 25: Abstract Test Suite for FDDI – Station Management Conformance Testing (SMT-ATS)*
- *Part 26: Media Access Control Conformance Testing (MAC-ATS)*

# INTRODUCTION

The International Organization for Standardization (ISO) has developed a standard to define the procedures required for Conformance Testing. These procedures are set forth in ISO 9646, Parts 1-7. Part 3 defines the language syntax to be used for writing Abstract Test Suites (ATS), that language is Tree and Tabular, Combined Notation (TTCN).

The Station Management (SMT) Abstract Test Suite (ATS) directly supports the FDDI Protocol Implementation Conformance Statement (PICS) Proforma and works in correlation with three other FDDI ATS standards.

This ATS for FDDI SMT provides the test procedures and test cases required to test the station management protocol described in the SMT standards. SMT specified the local portion of the system management application process for FDDI, including the control required for proper operation of an FDDI station in an FDDI ring. SMT provided services such as connection management, station insertion and removal, station initialization, configuration management, fault recovery, communication protocol for external authority, scheduling policies and the collection of statistics. SMT interact with PMD, PHY, and MAC for testing.

The three ATS standards when combined with SMT, that make up the complete Conformance Test for the FDDI Protocol are:

a) An ATS for FDDI Physical Medium Dependent (PMD) that provides a conformance test for FDDI PMD. PMD specifies the optical interface of FDDI stations. PMD is not a protocol standard and this ATS requires the measurement of physical quantities such as optical power, wavelength and signal jitter. The PMD ATS differs from the methodology of higher level protocol conformance tests written using the Tree and Tabular Combined Notation as specified by ISO 9643-3, because the TTCN notation does not provide a suitable vehicle for Physical Layer testing, where there is no concept of a protocol data unit and where physical quantities must be measured.



b) An ATS for the FDDI Physical Layer Protocol (PHY) that provides a conformance test for FDDI PHY. PHY specifies the upper sublayer of the Physical Layer for the FDDI, including the data encode/decode, framing and clocking, as well as the elasticity buffer, smoothing and repeat filter functions. FDDI PHY, however, does contain several state machines and implements a protocol at the level of FDDI code symbols. The only physical quantity that must be measured in this conformance test is frequency. The PHY ATS cannot use the TTCN notation. A unique notation is developed in the PHY ATS for specifying test patterns and expected results in terms of FDDI code symbol strings.

c) An ATS for FDDI Media Access Control (MAC) that provides a Conformance test for FDDI MAC. MAC specifies the lower sublayer of the Data Link Layer for FDDI. It specifies access to the medium, including addressing, data checking and data framing. MAC also specifies the receiver and transmitter state machines. Since MAC is a protocol that deals primarily with complete PDUs, the Tree and Tabular Combined Notation language specified in ISO 9643-3 is used to specify MAC protocol tests.

International Standard ISO/IEC 9314-25:1998, *Information technology - Fibre Distributed Data Interface (FDDI) - Station Management Conformance Testing (SMT-ATS)* was developed by ISO/IEC JTC 1/SC 25.

# INFORMATION TECHNOLOGY – FIBRE DISTRIBUTED DATA INTERFACE (FDDI) –

# Part 25: Abstract Test Suite for FDDI – Station Management Conformance Testing (SMT ATS)

## 1 Scope

This part of ISO/IEC 9314 contains the Abstract Test Suites for the Fiber Distributed Data Interface (FDDI) token ring Station Management (SMT) layer protocol. The SMT Protocol is extensive and very complex. In the development process, the protocol was broken into six separate areas. Those areas dealt with Physical Connection Management (PCM), Entity Coordination Management (ECM) Ring Management (RMT), Configuration Management (CMT), Frame Based Management (FBM) and Management Information Base (MIB).

This SMT ATS is divided along the same boundaries, with the exception that PCM and ECM are combined. Those two concepts are tested together. The formal description language used for Abstract Test Suite (ATS) development is Tree and Tabular Combined Notational (TTCN) and is defined in ISO 9646Framework. TTCN is intended for higher layer protocol testing and requires the use of discreet Protocol Data Units (PDUs). The TTCN notation is used in the test cases for RMT, FBM and MIB. It cannot be used for PCM, ECM and CFM. These three protocols use line states as the method of conveying information.

The TTCN (P) is similar in structure to TTCN but changes the paradigm from PDUs to line states. A description of the concept of TTCN (P) can be found in the beginning of section 6, PCM.

## 2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9314. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9314 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 9314-1:1989, *Information processing systems – Fibre Distributed Data Interface (FDDI) – Part 1: Token Ring Physical Layer Protocol (PHY)*

ISO/IEC 9314-2:1989, *Information processing systems – Fibre Distributed Data Interface (FDDI) – Part 2: Token Ring Media Access Control (MAC)*

ISO/IEC 9314-3:1990, *Information processing systems – Fibre Distributed Data Interface (FDDI) – Part 3: Physical Layer Medium Dependent (PMD)*

ISO/IEC 9314-6:1998 *Information technology – Fibre Distributed Data Interface (FDDI) – Part 6: Station Management (SMT)*

ISO/IEC 9646 (all parts), *Information technology – Open Systems Interconnection – Conformance testing methodology and framework*

ISO/IEC 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts*

ISO/IEC 9646-2:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 2: Abstract test suite specification*

## 3 Definitions

For the purposes of this part of ISO/IEC 9314, the following definitions apply.

**3.1** **Abstract Test Suite (ATS):** An ATS is a document, defined in ISO 9646, that depicts the suite of tests to be run by the test implementor to define conformance to a governing standard.

**3.2** **Frame Based Management (FBM):** FBM defines the frame formats and protocols used to manage FDDI stations on a ring in a peer-to-peer relationship.

**3.3** **Management Information Base (MIB):** The MIB provides the specification and capabilities of management information and its relationships to systems management.

**3.4** **Connection Management (CMT):** CMT controls the establishment and maintenance of an FDDI connection.

**3.5** **Physical Connection Management (PCM):** PCM controls initializations of a Physical connection, maintenance of the connection and closing of the connection.

**3.6** **Ring Management (RMT):** RMT monitors the MAC functions in an FDDI station and supports establishment and maintenance of an operational ring.

# 4 Convention and abbreviations

## 4.1 Conventions

## 4.2 Abbreviations

| | |
|---|---|
| ATS: | Abstract Test Suite; |
| CFM: | Configuration Management; |
| F: | Fail (when used in the verdict column of the Dynamic Behavior tables); |
| I: | Inconclusive (when used in the verdict column of the Dynamic Behavior tables); |
| ILS: | Idle Line State; |
| IUT: | Implementation Under Test; |
| MAC: | Media Access Control; |
| MLS: | Master Line State; |
| P: | Pass (when used in the verdict column of the Dynamic Behaviour tables); |
| PCM: | Physical Connection Management; |
| PDU: | Protocol Data Unit defined in terms of SMT and MAC Frames; |
| PHY: | Physical Layer Protocol; |
| PICS: | Protocol Implementation Conformance Statement; |
| PIXIT: | Protocol Implementation extra Information for Testing; |
| TTCN: | Tree and Tabular Combined Notation; |
| TTCN(P): | Tree and Tabular Combined Notation for PCM & CFM; |
| QLS: | Quiet Line State; |
| ALS: | Active Line State; |
| HLS: | Halt Line State; |
| ERR: | Error; |
| RMT: | Ring Management; |
| MIB: | Management Information Base. |



# 5 Timer definition

A set of timers, as described below, is used in the test suite, their values must be initialized prior to the beginning of test, unless a default value is specified.

**T_REQ:** the Target Token Rotation Timer
(TTRT) is configured in the IUT's MAC in units of µs. This value will be converted to units of 80 ns for MAC claim process.

**T_REQ1:** an alternate TTRT configured in the IUT's MAC in units of µs. This value will be converted to units of 80 ns for MAC claim process.

**T_REQ2:** an alternate TTRT configured in the Other's MAC in units of µs. This value will be converted to units of 80 ns for MAC claim process.

**T_Max:** the maximum token rotation time in µs.

**D_Max:** the maximum ring latency. The default value is 1773 µs.

**T_Non_Op:** time to allow ring recovery to occur before duplicate address conditions are examined. The default value is 1 s.

**RM_React:** maximum for the RMT state machine to recognize that transition conditions exist and to execute the appropriate transition. The default value is 83 ms.

**T_Jam:** time for which Jam Beacon is sent. The default value is 370 ms.

**T_DBJ:** time to start the second Beacon of the Double Beacon Jam after the first Beacon is sent. The default value is 82 ms.

**T-Direct:** time for which a Directed Beacon is sent before the Trace function is invoked. The default value is 370 ms.

**T_Stuck:** time to allow a Stuck Beacon to be sent, followed by the initiation of a Trace. The default value is 8 s.

**T_Rmode:** the maximum time allowable for Restricted Dialogue on the ring. The default value is zero seconds for Non-Used Restricted Dialogue.

**T_Announce:** the interval between sending Jam Beacons. The default value is 2 500 ms.

**T_Limit:** The rate-limiting interval for the Status Report Protocol. The default value is 2 s.



**Topr:** Time required for a test operator to initiate operation on the IUT, for example, triggering NIF request frame to be sent from the IUT. This is used in conjunction with the TTCN Implicit Send event for test coordination. This test suite uses a default value of 3 min.



The following are the expiration values of the timers used in PCM test cases. Whenever the name and the value correspond to ISO/IEC 9314-6 the reference is indicated.

**TB_Min**: Minimum Break time for link.

Range: TB_Min ≥ 4.823 ms with default values
Default: 5 ms (SMT PCM);

**TB_Max:** Break time before the BS_Flag is set. TB_Max shall be sufficiently large so that it will not be set inadvertently by noise generated by an optical bypass switch, which is bounded by MI_Max.

Range: TB_Max ≥ 30.0 ms with default values
Default: 50 ms (SMT PCM);

**MI_Max:** Maximum Optical Bypass media interruption time. The range and default value for MI_Max is specified in the PMD document.

Range: MI_Max ≤ 15.0 ms
Default: 15 ms (SMT PCM)

**C_Min:** Minimum time required to remain in the Connect State to ensure that the other end has recognized Halt Line State.

Range: C_Min ≥ 1.2 ms with default values
Default: 1.6 ms (SMT PCM);

**C_Second:** A timer used to check PCM wait for a change in the Connect State since it enters Connector State from Break State and has not yet received HLS.

Default: 1 s;

**PC_React:** Maximum time for PCM to make a state transition to Break upon receiving QLS.

Range: PC_React ≤ 3.0 ms
Default: 3 ms (SMT PCM);

**LS_Min:** Length of time continuous reception of ILS is required to be used by PCM.
Range: 25 µs ≥ LS_Min ≥ 0.48 µs with default values
Default: 0.48 µs(SMT PCM);

**LS_Max:** Maximum time to reestablish the correct line state as specified in the PHY document.

Range: LS_Max ≤ 25 µs
Default: 25 µs (SMT PCM);

**TL_Min:** Minimum time to transmit a PHY line state before advancing to the Next PCM state. TL-Min is set to twice the time required for line state recognition(LS_Max).

Range: TL_Min ≥ 50 µs with default values
Default: 50 µs (SMT PCM);

**N_Second:** A Timer used to check PCM wait to receive ILS in Next State

Default: 1 s;

**LS_Less:** A timer which measures the amount of time that it take the IUT to make a correct Line State transition.

Default: 0.24 µs;

**T_Out:** Signaling timeout. The minimum time that a PCM State Machine will remain in a state awaiting a line state change. When a line state change is expected and no transition is made in T_Out time, a transition shall be made to the Break State.



Range: T_Out ≥ 100 ms
Default: 100 ms (SMT PCM);

**LC_Short:** Short Link Confidence Test Time



Range: LC_Short > 5*10(4) ns
Default: 50 ms (SMT PCM);

**LC_Medium:** Medium Link Confidence Test Time.

Range: LC_Medium ≥ 50 *10(LER_Cutoff) ns
Default: 500 ms (SMT PCM);

**LC_Long:** Long Link Confidence Test Time.

Range: LC_Long ≥ 500 * 10 (LER_ Cutoff) ns
Default: 5 s (SMT PCM);

**LC_Extended:** Extended Link Confidence Test Time.

Range: LC_Extended ≥ 50 s
Default: 50 s (SMT PCM);

**T_Next(7):** LC_Test, Time for Link Confidence Test (SMT PCM);

**B_Second:** A timer used to check when IUT detects Link Error Rate exceeds the LER_Cutoff threshold; it enters Break State and transmits QLS.

Default: 1 s;

**T_Next(9):** Time for the optional MAC Local Loop to prevent deadlock. This allows sufficient time for MAC recovery process completion and the exchange of neighbor information frames.

Range: T_Next(9) ≥200 ms
Default: 200 ms (SMT PCM).

**NS_Max:** The maximum length of time that noise as measured by TNE, is allowed before a connection is broken down and restarted.

Range: 5.8 ms ≥ NS_Max ≥ 0.7255 ms
Default: 1.3 ms (SMT PCM);

**Trace_Max:** Maximum propagation time for a Trace on an FDDI topology. Trace_Max places a lower bound on the detection time for a nonrecovering ring (T_Stuck)

Range: Trace_Max ≥ 6.001773 s with default values
Default: 7.0 s

# 6 Physical Connection Management (PCM) & Entity Coordination Management (ECM) - Abstract Test Suites

## 6.1 Notation for PCM Tests

PCM is implemented as a complex state
machine and it is therefore highly desirable to use a formal notation to specify precisely conformance tests. One such formal notation is the Tree and Tabular Combined Notation (TTCN) as defined in ISO 9646-3. However PCM does not fit the TTCN paradigm well. The primary problem is that TTCN, which is intended for higher level protocol tests, requires that the protocol uses discrete Protocol Data Units (PDUs). It assumes that these PDUs are queued when received and that the TTCN "?" operator tests the PDU at the front of the receive queue. PCM signaling does not use PDUs but instead uses line states as the method of conveying information.

This document uses a notation called TTCN(P) to express the PCM tests. It is similar to TTCN, but changes the paradigm somewhat and simplifies the notation.
The key to understanding TTCN(P) is that the Implementation Under Test (IUT) is always considered to take on one of the following values:



- Quiet Line State (QLS)
- Idle Line State (ILS)
- Halt Line State (HLS)
- Master Line State (MLS)
- Active Line State (ALS)
- Noise Line State (NLS)
- Line State Unknown (LSU)

No other values can be set.

NLS is defined to be any condition which occurs in any of the other line states, which satisfies the conditions for termination of that state but does not satisfy the criteria for entry into any of the other line state. The tester never transmits NLS, and, in most cases the reception of NLS causes the IUT to fail the test.

The TTCN(P) "?" operator tests the contents of a received line state register in the tester, for example, the test ?ILS is satisfied if the current line state is ILS.

The tester is either transmitting a predefined MAC frame, repeating symbols received when its input is in ALS or is continuously transmitting one of the following:

- QLS
- ILS
- HLS
- MLS
- ERR
- Port_Type

ERR is a special pattern transmitted during the Link Confidence Test.

Port_Type is used to mean either HLS or MLS as appropriate. This notation is used to reduce the number of separate routines. In particular the port type, which is signaled in the Signal States when n=1 and n=2, is irrelevant to many test cases. Therefore the tests include subroutines, whose purpose is simply to step the IUT to the state where the actual test begins, use the Port_Type notation to allow the same routine to serve several port types.

A TTCN(P) test procedure consists of a sequence of event lines specifying an event or an action to take place at a given instant during the testing. The time progression is represented by the indentation number to the left of every event line in the form of [1], [2], etc. An event line may be one of the following: start transmitting line state symbols, check the current line state, start and test expiration of timers, invoke other test procedures (tree attachment), and a GOTO statement. These events are written in the format shown below:

[n] !line-state /* Start transmitting line-state symbols as specified, e.g. HLS */

[n] !Repeat /* Repeat input symbols received while input is in ALS, starting with the J symbol and stopping with the first V, I or H symbol. When V or H symbols are encountered the Repeat Filter rules of PHY are observed */

[n]!Packet /* cause the tester to transmit a single MAC Packet; when transmission of the packet is complete, the tester transmits ILS */

[n] ?line-state /* check if the current line state is in a specific state */

[n] ?OTHERWISE /* any line state */

[n] START timer-name /* start the timer with pre-specified duration */

[n] ?TIMEOUT timer-name /* test for expiration for the specified timer */

[n] +test-procedure-name /* call another test procedure */

[n] GOTO label /* goto another event line with indicated label */

As in many programming languages, a comment is a character string of the following form:

/* Text of comment */

A label for an event line is denoted by a sequence of letters ending with a ":", and appears after the indentation level number, for example,



[3] L1:?TIMEOUT C_min.

The "!" Transmit event means that the tester begins sending the indicated line state and continues sending it until another "!" operator is encountered.



The START timer event may be combined with the Transmit (!) or Line state check (?) event lines. Multiple timers may be started on the same event. For example,

[1] !QLS START TB_Min, START TB_Max.

The event lines are evaluated starting from the first indentation level, [1]. There may be several event lines at each indentation level. These event lines represent a set of alternatives and the tester must wait for at least one of them to occur before proceeding to the next indentation level. If multiple events occur at the same time, the event line appearing first applies. A transmit event line (!) is considered to have occurred or completed when the transmission of the specified line state symbols is initiated. A line state check event (?) is satisfied when the current line state matches the specified state.

When an event line is satisfied, the tester moves on to the next indentation level following that event line. If there is no higher level event line, then the test is complete. If a completed event line contains a verdict specification, the test is also considered completed, even if there is a higher indentation level event line following.

The event lines, with the exception of GOTO, SEND and Tree Attachment Event, may assign one of these verdicts: PASS, FAIL, or INCONCLUSIVE.

A GOTO event can only specify the labels appearing on the first line of an indentation level that is lower or equal to the current indentation level.

An event line that invokes another procedure is considered not satisfied if none of the first level event lines in that procedure have occurred.

Note that PCM is intended to operate in very noisy environments (perhaps as bad as a BER of $10^{-2}$ and does not generally react to brief noise events. We do not simulate noise in our PCM tests except for the Link Quality Tests; rather we expect that the IUT and the tester transmit nothing but clean line states without any errors. If the IUT transmits NLS or LSU it is always grounds for failure.

"TTCN(P) does not use the normal TTCN Constraints. This is because the Constraints section of TTCN uses PDUs rather than Line States to define Constraints.

An example illustrates the TTCN(P) notation. This example is the test case To_Next specified in 6.2.1.5:

**Procedure:**

```
[1]+Start_To_Connect
[2]  !HLS Start C_Min
[3]    A:?Timeout C_Min /*Comment*/
[4]      B:?HLS
[5]        Goto B
[4]      ?ILS                          Pass
[4]      ?Otherwise                    Fail(2)
[3]    ?HLS
[4]      Goto A
[3]    ?Otherwise                      Fail(1)
```

In **To_Next** the procedure **Start_To_Connect** is attached by the first statement:

**[1]+Start_To_Connect**, which is specified in 6.3.2, is:

**Procedure:**

```
[1]!QLS Start TB_Min
[2]  ?Timeout TB_Min
[3]    A:?QLS
[4]      Goto A
[3]    ?HLS      /*Comment*/
[3]    ?Otherwise                     Inconclusive
```

The key to understanding the attachment is that the attaching test is attached to each of the terminal leaves of the attached routine. A terminal leaf is any statement other than a GOTO that has no lower layer and is not qualified with Pass, Fail or Inconclusive verdict. In **Start_To_Connect** there is one terminal leaf:

```
[3]    ?HLS
```

Each test has a specific purpose. The purpose of **To_Next** is to verify that the IUT remains in the Connect State for at least C_Min before going to the Next State. **Start_To_Connect** is used simply to bring the IUT to the Connect State. Therefore it is a simple routine to progress a correct IUT to the Connect State and does not attempt to test all the requirements of PCM to get to the Connect State; these are tested elsewhere in an incremental fashion.

The first statement of **Start_To_Connect**, and the only statement at level 1 is:

[1]!QLS Start TB_Min

This statement causes the tester to transmit QLS and to start the TB_Min timer.

Control falls to the next statement at level 2:

[2]     ?Timeout TB_Min

This statement causes the tester to wait, still sending QLS, until the timer TB_Min expires, since there is no alternative at this level. This should cause the IUT to go to the Break State. There are three alternatives at level 3:

```
[3]     A:?QLS
[3]     ?HLS
[3]     ?Otherwise                    Inconclusive
```

Alternatives at the same level are evaluated in order and then the evaluation is repeated until one of the alternatives becomes true. One of the alternatives is always true because of the otherwise statement. At this point the IUT should either be transmitting QLS or HLS; if not, something is wrong. However, because this is not the point of the test, we label the otherwise "Inconclusive" rather than "fail". If we receive QLS, then we fall down to the following statement at level 4:

```
[4]     Goto A
```

This statement just returns the tester to level 3 at A and we loop back and evaluate the same three alternatives. Eventually the IUT should transmit HLS (signifying that it has entered the Connect State, the second alternative became true, Start_To_Connect has reached a terminal leaf), and control passes to the attaching test routine **To_Next**.

When the Start_To_Connect completes control passes to level 2, where there is one alternative:

```
[2]  !HLS Start C_Min
```

This statement causes the tester to transmit HLS and to start the timer C_Min. Control then passes to level 3 and the following alternatives:

```
[3]    A:?Timeout C_Min /*Comment*/
.
.
[3]    ?HLS
[3]    ?Otherwise   Fail(1)
```

The test purpose is to see if the IUT remains in the Connect State for at least C_Min before going to the Next State, therefore if any signal other than HLS (signifying Connect State) is received before C_Min expires, the Otherwise statement causes the test to fail. The "(1)" is a parameter used to distinguish between possibly different failure points. As long as HLS is received control will fall to a GOTO at level 4, which loops control back to B:

```
[4]     Goto A
```



Finally, when C_Min times out, control falls to:

```
[4]     B:?HLS
[5]       Goto B
[4]     ?ILS                      Pass
[4]     ?Otherwise                Fail(2)
```



As long as HLS is received, the tester continues to loop back to level 4. If ILS is received, then the IUT has made the transition to the Next State after waiting for at least C_Min and this test is passed. If anything else is received, then the test is failed.

Note that we can expand the attachment of Start_To_Connect by To_Next and get the following equivalent test routine:

**Procedure:**

```
[1]!QLS Start TB_Min
[2]  ?Timeout TB_Min
[3]   A:?QLS
[4]     Goto A
[3]   ?HLS
[4]     !HLS Start C_Min
[5]       B:?Timeout C_Min /*Comment*/
[6]         C:?HLS
[7]           Goto C
[6]         ?ILS                  Pass
[6]         ?Otherwise            Fail(2)
[5]       ?HLS
[6]         Goto B
[5]       ?Otherwise              Fail(1)
[3]   ?Otherwise                  Inconclusive
```

## 6.2 Test Cases

### 6.2.1 Connection Initialization Test:

#### 6.2.1.1 Case-ID: To_Connect_1

**Purpose:** Verify that IUT enters the Connect state after remaining in the Break state for at least TB_Min when it receives QLS.

**Test Setup:** See figure 1.

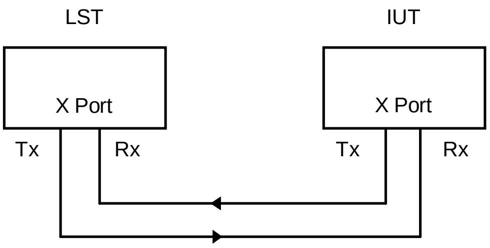


Figure 1 - Tester Configuration for Indicated Cases

**Procedure:**
```
[1]+Init            /*Comment(1)*/
[2]  !QLS Start TB_Min /*Comment(2)*/
[3]   A:?Timeout TB_Min
[4]    B:?QLS
[5]       Goto B
[4]    ?HLS                          Pass
[4]    ?Otherwise                    Fail(2)
[3]   ?QLS
[4]    Goto A
[3]   ?Otherwise   Fail(1)
```



**Extended Comments:**
(1) The Init function (6.3.1 of this document) is called to let the IUT start from the Break State.
(2) In the Break State, the Tester transmits QLS, starts the timer TB_Min to ensure that the IUT remains in the Break State for at least TB_Min. (PC-13)

**Reasons for Failure:**
(1) Initially, the IUT is in the Break State while QLS is transmitted for at least TB_Min. If the IUT does not transmit QLS within TB_Min, then the IUT fails.
(2) The IUT transmits QLS while in the Break State and HLS when it enters the Connect State. Any other output is invalid.

#### 6.2.1.2 Case-ID: To_Connect_2

**Procedure:**
```
[1]+Init/*Comment(1)*/
[2]  !HLS Start TB_Min /*Comment(2)*/
[3]   A:?Timeout TB_Min
[4]    B:?QLS
[5]      Goto B
[4]    ?HLS                           Pass
[4]    ?Otherwise                     Fail(2)
[3]   ?QLS
[4]    Goto A
[3]   ?Otherwise                      Fail(1)
```

**Extended Comments:**
(1) The Init function (6.3.1 of this document) is called to let the IUT start from the Break state.
(2) In the Break state, the Tester transmits HLS, starts the timer TB_Min to ensure that the IUT remains in the Break state for at least TB_Min. (PC-13)

**Reasons for Failure:**
(1) IUT must transmit QLS for TB_Min at the beginning of this procedure. If it does not meet this requirement then the IUT fails.
(2) After remaining in the Break State for longer than TB_Min, the IUT may either stay in Break or change to Connect. If the IUT transmits any line state other than QLS or HLS, the test fails.

### 6.2.1.3 Case-ID: Wait_for_Connect

**Purpose:** Show that the IUT progresses to Connect from the Break State and remains in the Connect State until HLS is received.

**Test Setup:** See figure 1.

**Procedure:**
```
[1]+Init
[2]  !QLS
[3]    A:?QLS
[4]      Goto A
[3]    ?HLS /*Comment(1)*/
[4]      Start C_Second /*Comment(2)*/
[5]        B:?Timeout C_Second          Pass
[5]        ?HLS
[6]          Goto B
[5]        ?Otherwise                   Fail(2)
[3]    ?Otherwise                       Fail(1)
```



**Extended Comments:**
(1) IUT enters the Connect State from the Break State and transmit HLS. (Connect_Actions)
(2) In this step, the IUT is in the Connect State, it is going to receive a sequence of HLS to process to the Next State. At this point, The Tester keeps sending QLS to the IUT to see what happen on the IUT. (PC-33)



**Reasons for Failure:**
(1) IUT transmit QLS when it enters the Break State. After TB_Min, it must enter the Connect State and transmit HLS. If the IUT does not transmit QLS followed by HLS then the IUT fails.
(2) When the IUT enters the Connect state, it will enter the Next state after receiving sufficient HLS. The tester remains in the Break state which should cause the IUT to remain in the Connect state forever. If the IUT transmits other than HLS then it is failed.

### 6.2.1.4 Case-ID: Connect_Error

**Purpose:** Verify that the IUT enters the Break
State if the ILS is received before HLS is received in the Connect State.

**Test Setup:** See figure 1.

**Procedure:**
```
[1]+Start_To_Connect /*Comment*/
[2]  !ILS  Start T_Out
[3]    A:?Timeout T_Out                Fail(1)
[3]    ?HLS
[4]      Goto A
[3]    ?QLS                            Pass
[3]    ?Otherwise   Fail(2)
```

**Extended Comments:** The IUT is in the Connect State (see 6.3.2).