System Conformance Testing

8

KNX Network, Transport, Application (Interface) Layer, Management Service Testing

3

Transport Layer Tests

4

Summary

This document contains the Transport Layer Test specifications.

Version 01.04.00 is a KNX Approved Standard.

This document is part of the KNX Specifications v2.1.

Document Updates

Version	Date	Modifications
1.0	2001.15.06	Approved Standard
1.1	2005.02.25	Approved Standard – integration of AN 002 – 009 – relevant parts of AN 041
1.2	2009.06	Readying document for publication as part of V2.0 of the KNX specifications
1.3	2009.10 DP	References to Application Notes replaced – correction of sequence 32 and 36 – Adding Test set-up for closed devices
1.3	2010.01 DV	No comments in release for voting – readying for final voting
1.3	2010.01	Resolving comments from Final voting – publication as AS
01.03.01	2013.10.25	Editorial updates.
01.04.00	2013.10.25	Editorial updates for the publication of KNX Specifications 2.1.

Filename:	08_03_04 System Conformance Testing - TL Tests v01.04.00 AS.docx
Version:	01.04.00
Status:	Approved Standard
Savedate:	2013.10.25
Number of pages:	50

Contents

1 General Transport Layer Tests[1]

1.1 Transport Layer tests for multicast communication

a) positive case: checking whether BDUT sends respectively accepts telegrams with correct Transport Control Field (TPCI) – Test implicitly carried out during Application Interface Layer Tests of Volume 8/3/7

b) negative case:

Purpose: Check BDUT´s acceptance of multicast-addressed frames with incorrect TPCI-coding.

Procedure: Use a telegram generator (e.g. EITT) to send

- multicast-addressed frames with incorrect TPCI coding for multicast communication

- multicast-addressed frames with correct TPCI coding and sequence number different from 0

Examples of stimuli (invalid TPCI coding for multicast communication):

```
IN      BC 1037 0001 E1 40  81  : Group Value Write with TPCI of T_Data_Connected

IN      BC 1037 0001 E2 80  40 FF  : Group Value Response with TPCI of T_(Dis)Connect

 IN      BC 1037 0001 E1 C0 00 : Group Value Read with TPCI of T_(N)Ack
```

Examples of stimuli (valid TPCI coding for multicast communication with sequence number different from 0):

```
IN      BC 1037 0001 E1 10  81  : Group Value Write with Sequence number 4
```

<u>Note</u>: it goes without saying that before sending of the connection-oriented telegrams in this document the necessary connection has to be established.

Acceptance: BDUT does not accept the frames. Check BDUT’s behaviour, e.g. by reading back the values respectively checking that the BDUT does not generate any responses to read telegrams.

Expected reactions to all stimuli examples above: BDUT does not take over the values or does not generate any responses.

1.2 Transport Layer test for broadcast communication

a) positive case: checking whether BDUT sends respectively accepts telegrams with correct Transport Control Field (TPCI) – Test implicitly carried out during Application Interface Layer Tests of Volume 8/3/7

b) negative case:

Purpose: Check BDUT´s acceptance of broadcast-addressed frames with incorrect TPCI-coding.

Procedure: Use a telegram generator (e.g. EITT) to send

- broadcast-addressed frames with incorrect TPCI coding for broadcast communication

- broadcast-addressed frames with correct TPCI coding and sequence number different from 0

Examples of stimuli (invalid TPCI coding for broadcast communication):

```
IN      BC 1037 0000 E3 40  C0 12 34  : IndAddrWrite(Addr=1234) with TPCI of T_Data_Connected

IN        BC 1037 0000 E1 83  E1  :DomainAddressRead ()with TPCI of T_(Dis)Connect

IN      BC 1037 0000 E1 C1  00  :IndAddrRead()with TPCI of T_(N)Ack
```

[1] These tests are compulsory for any development submitted to KNX certification from February 2006

Examples of stimuli (valid TPCI coding for multicast communication with sequence number different from 0):
IN BC 1037 0000 E3 10 C0 12 34 :IndAddrWrite(Addr=1234) with Sequence number 4

Acceptance: BDUT does not accept the frames. Check BDUT's behaviour, e.g. by reading back the values respectively checking that the BDUT does not generate any responses to read telegrams

Expected reactions to all stimuli examples above: BDUT does not take over the values or does not generate any responses.

1.3 Transport Layer tests point-to-point connection oriented communication

a) positive case: checking whether BDUT sends respectively accepts telegrams with correct Transport Control Field (TPCI) – Test implicitly carried out during Application Interface Layer Tests of Volume 8/3/7

b) negative case:

Purpose: Check BDUT´s acceptance of connection oriented frames with incorrect TPCI-coding.

Procedure: Use a telegram generator (e.g. EITT) to send connection oriented frames with incorrect TPCI coding for connection oriented communication

It shall also be tested that during a connection (T-Connect), that any sent group or broadcast telegrams do not disturb or lead to a break-down of the connection.

Examples of stimuli (invalid TPCI coding for connection oriented communication):
IN BC 1037 0000 E2 01 81 01 :ReadAdc(Channel=01, Count=01) with TPCI of T_Data_Broadcast

IN BC 1037 1234 E3 02 01 01 04 :MemoryRead(Count=01, Addr=0104) with TPCI of T_Data_Group

IN BC 1037 1234 66 03 D1 00 12 34 56 78 :AuthorizeRequest(12345678) with TPCI of T_Data_Individual

IN BC 1037 1234 66 83 D3 01 12 34 56 78 :SetKeyRequest(01, 12345678) with TPCI of T_(Dis)Connect

IN BC 1037 1234 66 C3 D7 04 01 10 01 FF :PropertyWrite(Obj=04, Prop=01, Count=1, Start=001, Data=FF) with TPCI of T_(N)Ack

Acceptance: BDUT does not accept the frames. Check BDUT's behaviour, e.g. by reading back the values respectively checking that the BDUT does not generate any responses to read telegrams.

Expected reactions to all stimuli examples above: BDUT does not take over the values or does not generate any responses.

Example of sequence with connection oriented communication interrupted by broadcast or group telegrams:

IN B0 AFFE 1001 60 80 :T-Connect(Addr=1001)

IN BC AFFE 1001 63 42 0A 01 A0 :MemoryRead(Count=0A, Addr=01A0)

OUT B0 1001 AFFE 60 C2 :T-Ack(Seq=0)

OUT BC 1001 AFFE 6D 42 4A 01 A0 01 23 45 67 89 AB CD EF 01 23 :MemoryResponse(Count=0A, Addr=01A0, Data=01 23 45 67 89 AB CD EF 01 23)

IN BC 1037 0001 E1 00 81 : Value Write command to group address supported by the device

IN BC 1037 0000 E1 01 00 :ReadPhysAddr()

IN B0 AFFE 1001 60 C2 :T-Ack(Seq=0)

IN B0 AFFE 1001 60 81 :T-Disconnect
Acceptance: BDUT does not interrupt the established transport connection (it however may write the relevant addressed group object).

1.4 Transport Layer tests point-to-point connectionless communication

c) positive case: checking whether BDUT sends respectively accepts telegrams with correct Transport Control Field (TPCI) – Test implicitly carried out during Application Interface Layer Tests of Volume 8/3/7

d) negative case:

Purpose: Check BDUT´s acceptance of connectionless frames with incorrect TPCI-coding.

Procedure: Use a telegram generator (e.g. EITT) to send connectionless frames with incorrect TPCI coding for connectionless communication:

Examples of stimuli (invalid TPCI coding for connectionless communication):
IN BC 1037 0000 E5 03 D5 04 01 10 01 :PropertyRead(Obj=04, Prop=01, Count=1, Start=001) with TPCI of T_Data_Broadcast

IN BC 1037 1234 E5 03 D5 04 01 10 01 :PropertyRead(Obj=04, Prop=01, Count=1, Start=001) with TPCI of T_Data_Group

IN BC 1037 1234 65 43 D5 04 01 10 01 :PropertyRead(Obj=04, Prop=01, Count=1, Start=001) with TPCI of T_Data_Connected

IN BC 1037 1234 65 83 D5 04 01 10 01 :PropertyRead(Obj=04, Prop=01, Count=1, Start=001) with TPCI of T_(Dis)Connect

IN BC 1037 1234 65 C3 D5 04 01 10 01 :PropertyRead(Obj=04, Prop=01, Count=1, Start=001) with TPCI of T_(N)Ack

Acceptance: BDUT does not accept the frames. Check BDUT's behaviour, e.g. by reading back the values respectively checking that the BDUT does not generate any responses to read telegrams.

Expected reactions to all stimuli examples above: BDUT does not take over the values or does not generate any responses.

Note: in the case where services may be supported both connection oriented as well as connectionless (for details see volume 3/3/7), the BDUT will always write the appropriate value respectively generate the necessary response.

1.5 Transport Layer tests unicast / Device oriented communication – connected

Transport Layer Minimum Requirements

Purpose: Check if TL minimum requirements are fulfilled.

Procedure: Use a telegram generator (e.g. EITT) to send a T_Connect.req to the BDUT

Acceptance: BDUT answers with a T_Disconnect.req

Note: This test is applicable only if the BDUT supports exclusively the minimum requirements on TL.

2 Testing of Transport Layer State Machine

2.1 Introduction

2.1.1 General

In each sequence it is explained which event is tested (purpose) and which telegram sequence is used with the appropriate initial state.

In each test step, the term **local** implies sending the messages via a PC tool connected to an RS232 with PEI-type10 directly to the BDUT (FT1.2 Protocol), while **remote** implies sending the messages via a second RS232 interface on bus (see relevant test set-ups)

2.1.2 Test environment when testing BCU (with PEI)

Test-Tools:

Hardware:	- one BCU with Serial Interface with PEI-Type10
	- Two-BCUs (optionally one additional BCU for observation of bus traffic) and two (optionally three) data interfaces (EDI)
	- Two (optionally three) PCs with Windows XP or later.
Software:	- EITT Version2.3 or upwards

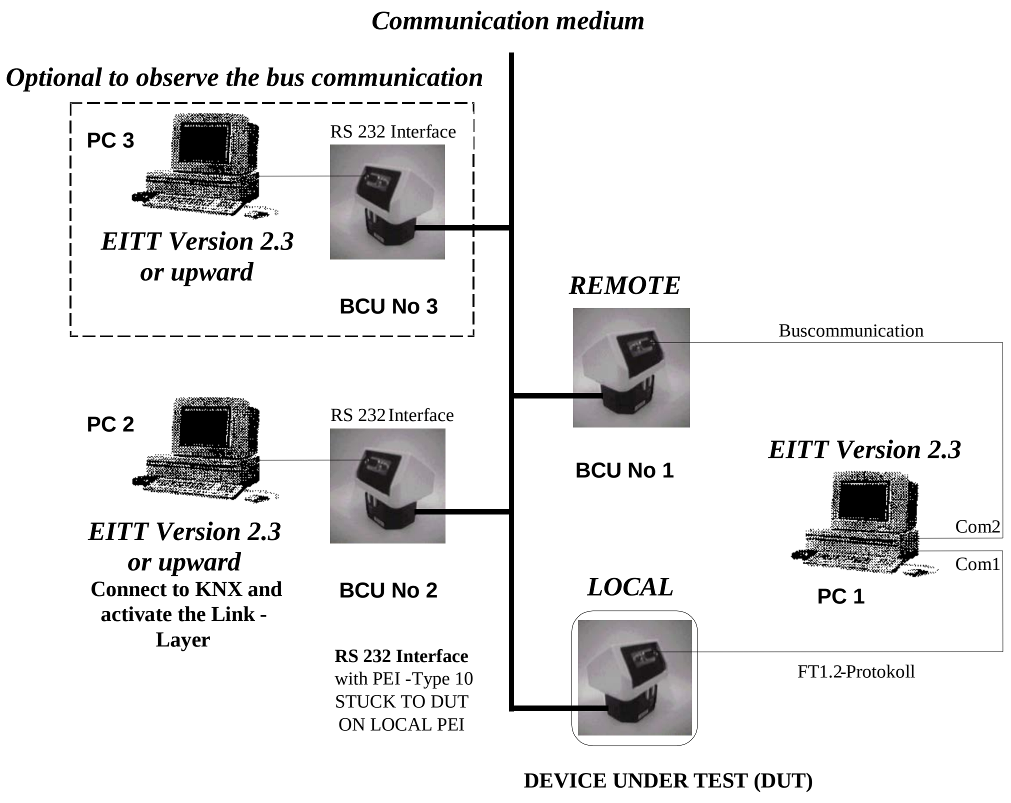

Communication medium
Optional to observe the bus communication
PC 3
RS 232 Interface
EITT Version 2.3 or upward
BCU No 3
REMOTE
Buscommunication
PC 2
RS 232 Interface
BCU No 1
EITT Version 2.3
Com2
Com1
EITT Version 2.3 or upward
Connect to KNX and activate the Link - Layer
BCU No 2
LOCAL
PC 1
RS 232 Interface with PEI -Type 10 STUCK TO DUT ON LOCAL PEI
FT1.2-Protokoll
DEVICE UNDER TEST (DUT)

Figure 1: Test equipment when testing BCU (with PEI)

2.1.3 Test environment when closed devices (without PEI)

Test-Tools:

Hardware: - one BCU with Serial Interface with PEI-Type10

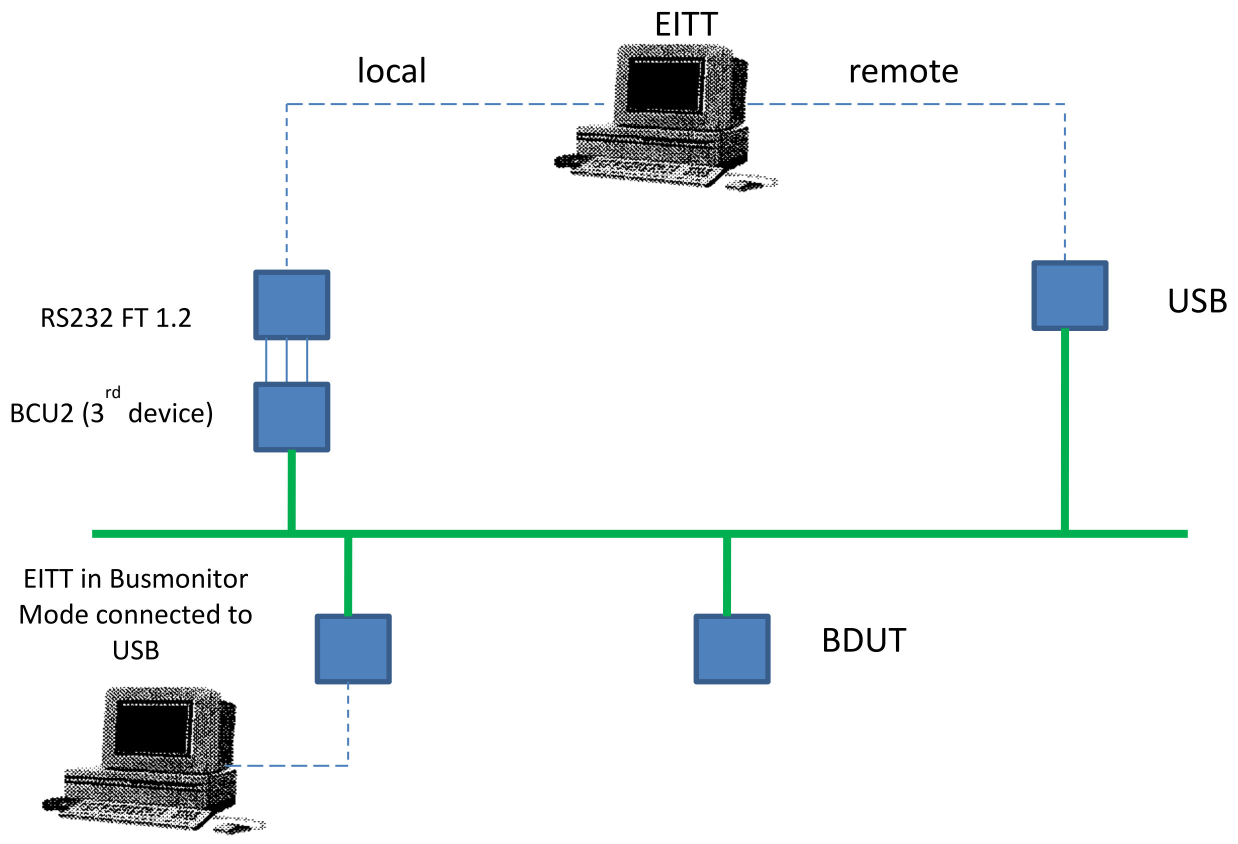
- One-BCUs (optionally two additional BCUs for observation of bus traffic) and one (optionally two) data interfaces (EDI)

- One (optionally two) PCs with Windows

Software: - EITT Version 2.3 or upwards

Figure 2: Test equipment when testing BCU (with PEI)

2.1.4 State machine of Connection-oriented Communication Relationship



An KNX end device only has a single one-to-one connection-oriented communication relationship.

To test the state machine of transport-layer the following services can only be used on a connection-oriented communication relationship:

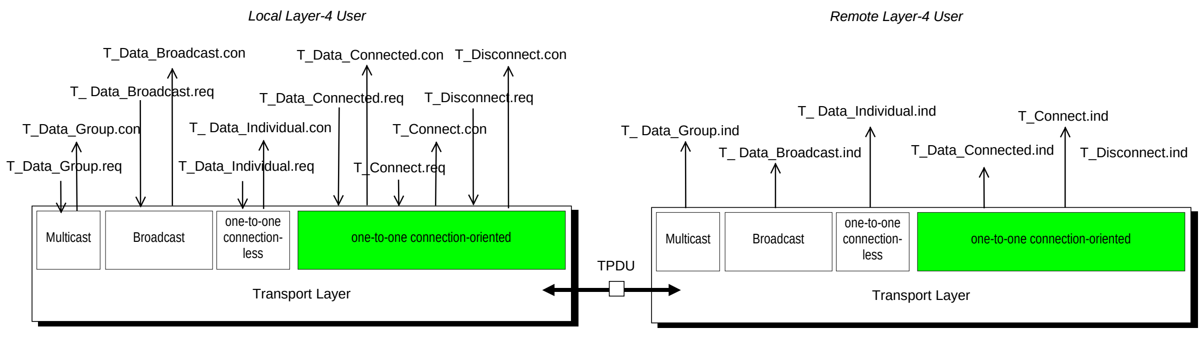


Figure 3: Interactivity of the Transport Layer

- T_Connect

- T_Disconnect
- T_Data_Connected

The transport-layer provides a supervision of the connection with a connection-time-out-timer. If the timer expires or if an unrecoverable error occurs, the transport-layer will release the connection immediately.

T_Data_Connected services are repeated up to three times if the T_Data_Connected.req is not acknowledged from the remote transport-layer entity within an acknowledgement-time-out-time. Repetitions of T_Data_Connected services are detected using a sequence number. The state machine is designed for only one connection at a time. To use more than one connection at a time several state machines are needed, one for each connection.

The state machine has the following states:

CLOSED	The connection is closed
OPEN_IDLE	The connection is open.
OPEN_WAIT	The state machine is waiting for a T_ACK when data have been sent to the remote partner

The following tests examine the actual state transition, the correct event and the executed action.

For further information see KNX Handbook (Volume 3: System Specifications, Part 3: Communication, Chapter 4: Transport Layer), which provides a specification of the complete state machine with reactions to all possible events in all states as well as the various existing styles of the TL state machine. The differences in the various state machine styles relate to the behaviour of the transport layer to errors. The normal behaviour to the expected events is identical between existing implementations and the specification. The main objective is to provide a unified handling of error conditions.

The state diagrams are inserted in these test specifications for explanatory reasons.

An example explains the structure of the state diagram.

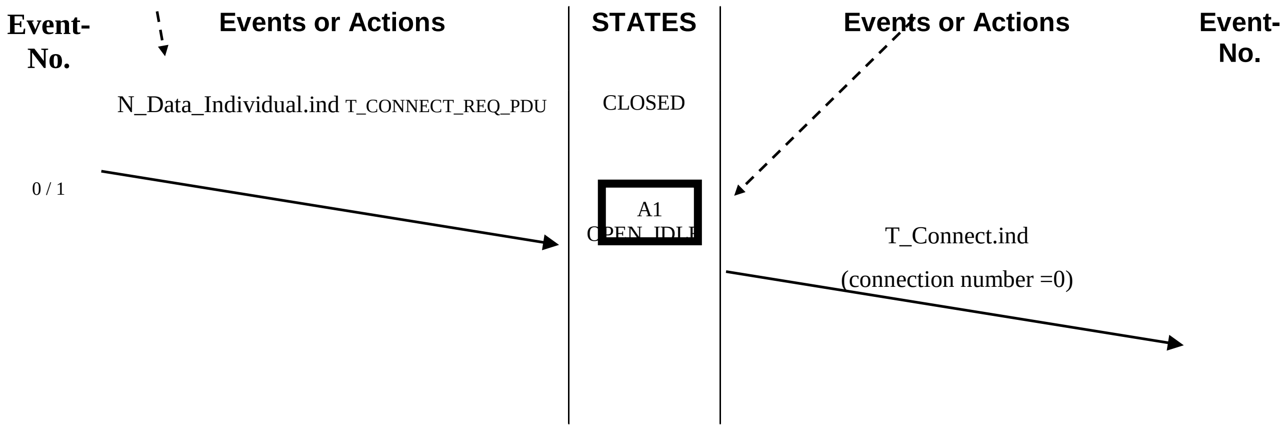


Figure 4: Example of a TL state diagram

All underneath telegram sequences contain local services conforming to the EMI2 protocol. For further information see KNX Handbook (Volume 3: System Specifications, Part 6: Application Interfaces, Chapter 3: External Message Interface)

Some of the underneath tests are marked with **'(without PEI)'**, if a test can be executed on devices without PEI. To achieve the initial state, in some cases the telegram sequence will however have to be modified. Stimulations are in this case always carried out via the bus and not via the PEI. E.g. the initial state OPEN_WAIT cannot be reached by sending a T_DATA_CONNECTED.req via the PEI in this case, but the BDUT shall be stimulated via the bus with a MaskVersion_Read frame in order to send a MaskVersion_Response. If the BDUT does not receive a T_ACK the transport layer remains in OPEN_WAIT.

In the heading of the different tests, it is stated for which styles[2] the underneath telegram sequence can be applied. Any deviating reactions from other styles are mentioned underneath the relevant telegram sequences.

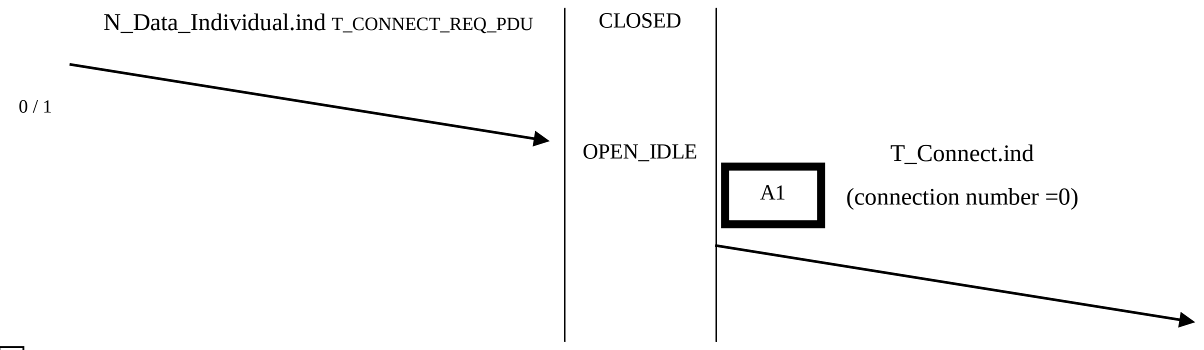
[2] The rationalised TL state machine as described in Volume 8/3/4 shall be complied with by all devices reporting compliance to TP1 mask 2.1

2.2 States caused by a Connect and Disconnect – Service

2.2.1 Connect from a remote device

2.2.1.1 Telegram Sequence 1: Procedure with initial state 'CLOSED' (all styles)

Purpose: Check whether the BDUT passes the following states.



☐

Figure 5: State transition telegram sequence 1

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req
@[tThe BDUT is in State 'CLOSED'.
@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind
@[tThe BDUT is in State 'OPEN_IDLE'.
```

2.2.1.2 Telegram sequence 2: Procedure with initial state ‘OPEN_IDLE’ (without PEI, style 1)

Purpose: Check whether the BDUT passes the following states. (OPEN_IDLE, OPEN_WAIT)

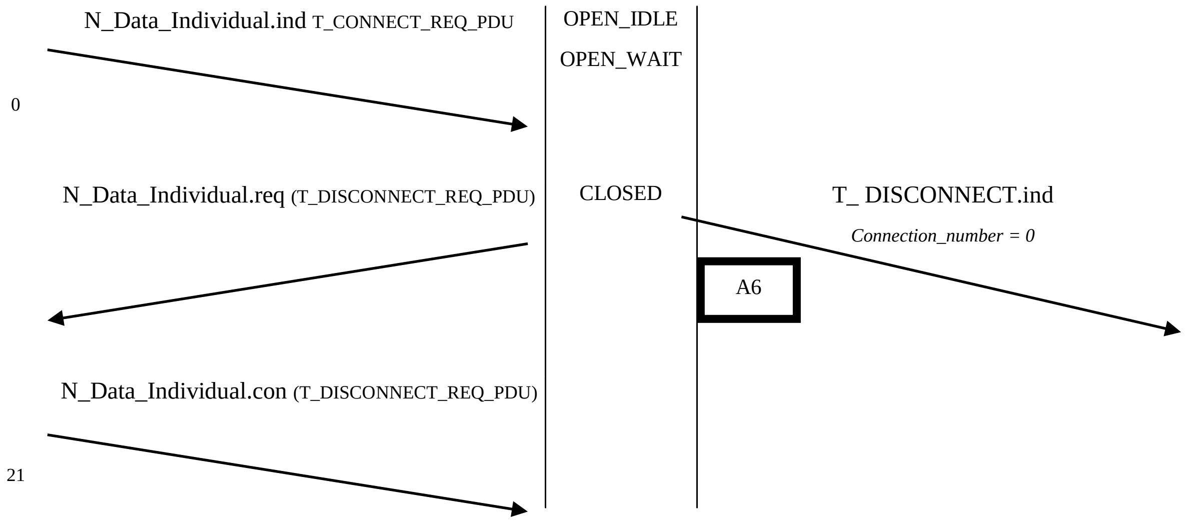

N_Data_Individual.ind T_CONNECT_REQ_PDU
OPEN_IDLE
OPEN_WAIT
0
N_Data_Individual.req (T_DISCONNECT_REQ_PDU)
CLOSED
T_ DISCONNECT.ind
Connection_number = 0
A6
N_Data_Individual.con (T_DISCONNECT_REQ_PDU)
21

Figure 6: State transition telegram sequence 2

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. The remote device sends a T_CONNECT.req to BDUT
(4)          IN   00:00:01.0  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t5. Check that BDUT sends a T_DISCONNECT.ind to the user.
(5)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
@[t6. Check that a telegram is sent addressed to remote BCU with
NSDU=T_DISCONNECT.req
(6)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tNow the BDUT is in State 'CLOSED'.
```

Note : Style 2/3: BDUT remains in OPEN_IDLE, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

2.2.1.3 Telegram sequence 3: Procedure with initial state ‘OPEN_WAIT’ (without PEI, style 1)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req
@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :
@[tThe BDUT is in State 'OPEN_IDLE'.
@[t4. Send a T_DATA_CONNECTED.req to remote BCU.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()
@[tThe BDUT is in State 'OPEN_WAIT'.
@[t6. The remote device sends a T_CONNECT.req to BDUT
(6)          IN   00:00:04.0  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t7. Check that BDUT sends a T_DISCONNECT.ind to the user.
(7)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
@[t8. Check that a telegram is sent addressed to remote BCU with
NSDU=T_DISCONNECT.req
(8)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tCheck that the BDUT is in State 'CLOSED'.
```

Note : Style 2/3 - BDUT remains in OPEN_WAIT, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

2.2.2 Connect from a remote device during an existing connection

2.2.2.1 Telegram sequence 4 : Procedure with initial state 'OPEN_IDLE' (without PEI, styles 1/3)

Purpose : Check whether the BDUT passes the following states.

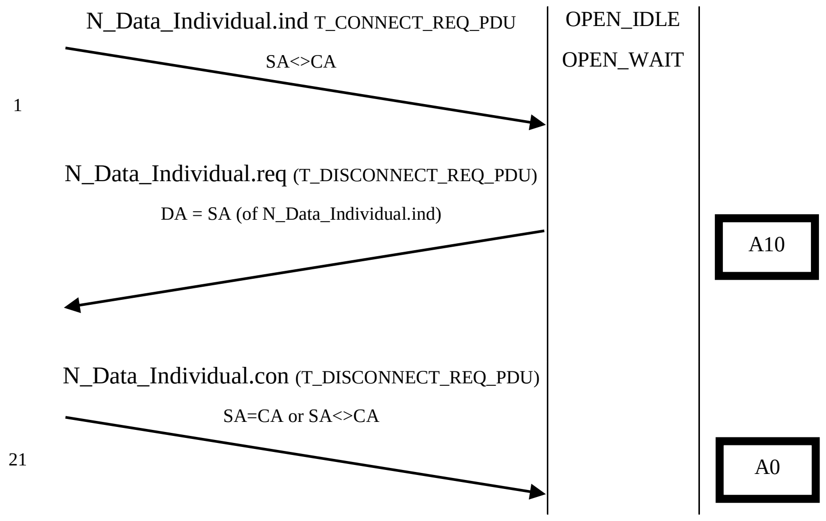

N_Data_Individual.ind T_CONNECT_REQ_PDU
SA<>CA
1
OPEN_IDLE
OPEN_WAIT
N_Data_Individual.req (T_DISCONNECT_REQ_PDU)
DA = SA (of N_Data_Individual.ind)
A10
N_Data_Individual.con (T_DISCONNECT_REQ_PDU)
SA=CA or SA<>CA
21
A0

Figure 7: State transition telegram sequence 4

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. Check that the BDUT conirms the T_CONNECT
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con
@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Connect from a remote device during an existing connection (remote BCU
<> third BCU).
(4)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t5. Check that a telegram is sent addressed to remote BCU with
NSDU=T_DISCONNECT.req
(5)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'OPEN_IDLE'.
```

Note : Style 2 - BDUT sends no Disconnect on the bus.

2.2.2.2 Telegram sequence 5: Procedure with initial state 'OPEN_WAIT' (without PEI, styles 1/3)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Set the BCU
in Link-Layer.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. Check that the BDUT conirms the T_CONNECT
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. BDUT sends a T_DATA_CONNECTED.req to the third BCU.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req

@[tThe BDUT is in State 'OPEN_WAIT'.

@[t5. Connect from a remote device during an existing connection (remote BCU
<> third BCU).
(5)          IN   00:00:04.0  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t6. Check that a telegram is sent addressed to remote BCU with
NSDU=T_DISCONNECT.req
(6)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'OPEN_WAIT'.
```

Note : Style 2 - BDUT sends no Disconnect on the bus.

2.2.3 Disconnect from a remote device

2.2.3.1 Telegram Sequence 6: Procedure with initial state 'CLOSED' (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.

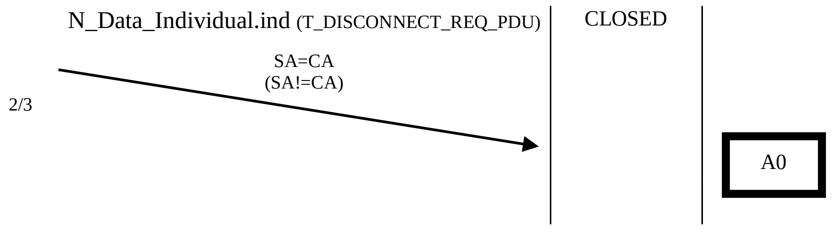


Figure 8: State transition for telegram sequence 6

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[tThe BDUT is in State 'CLOSED'.
```

```
@[t2. The remote device sends a T_DISCONNECT.req to BDUT
(2)          IN   00:00:00.2  B0 A001 A000 60 81  :T-Disconnect

@[tThe BDUT persists in State 'CLOSED'.
```

2.2.3.2 Telegram Sequence 7: Procedure with initial state 'OPEN_IDLE' (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.

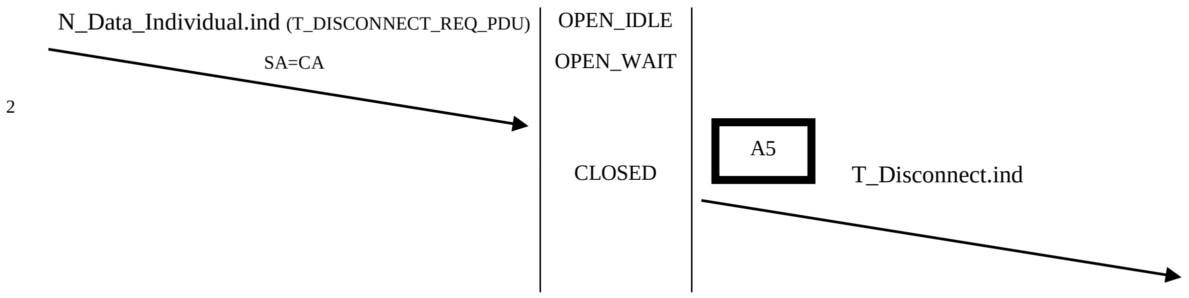


Figure 9: State transition for telegram sequence 7

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. The remote device sends a T_DISCONNECT.req to BDUT
(4)          IN   00:00:00.2  B0 A001 A000 60 81  :T-Disconnect
@[t5. Check that the BDUT sends a T_DISCONNECT_ind
(5)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
@[tThe BDUT is in State 'CLOSED'.
```

2.2.3.3 Telegram Sequence 8: Procedure with initial state 'OPEN_WAIT' (without PEI, all styles)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)         IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)         IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind
(3)         OUT  00:00:00.0  85 B0 A0 01 A0 00  :

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Send a T_DATA_CONNECTED.req to remote BCU.
(4)         IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
(5)         OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tThe BDUT is in State 'OPEN_WAIT'.

@[t6. The remote device sends a T_DISCONNECT.req to BDUT
(6)         IN   00:00:04.0  B0 A001 A000 60 81  :T-Disconnect
@[t7. Check that the BDUT sends a T_DISCONNECT_ind
(7)         OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
@[tNow the BDUT is in State 'CLOSED'.
```

2.2.4 Disconnect from a remote device during an existing connection

2.2.4.1 Telegram sequence 9: Procedure with initial state 'OPEN_IDLE' (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.

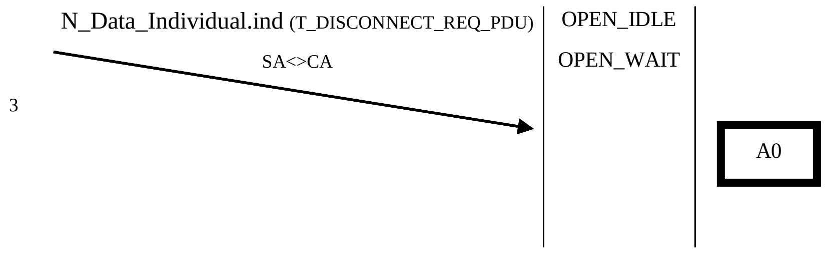


Figure 10: State transition for telegram sequence 9

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Activate the
Link-Layer.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)         IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. BDUT receives a T_CONNECT.con
(2)         IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)         OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con
@[tThe BDUT is in State 'OPEN_IDLE'.
```

```
@[t4. The BDUT receives a T_DISCONNECT.req from remote BCU (<> connected BCU
= third BCU)
(4)          IN   00:00:02.0  B0 A001 A000 60 81  :T-Disconnect
@[tNo reaction.
@[tThe BDUT persists in State 'OPEN_IDLE'.
```

2.2.4.2 Telegram sequence 10: Procedure with initial state ‘OPEN_WAIT’ (without PEI, all styles)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Activate the
Link-Layer.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. BDUT receives a T_CONNECT.con
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con
@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. BDUT sends a T_DATA_CONNECTED.req to the third BCU.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[tThe BDUT is in State 'OPEN_WAIT'.

@[t5. The BDUT receives a T_DISCONNECT.req from remote BCU (<> connected BCU
= third BCU)
(5)          IN   00:00:04.0  B0 A001 A000 60 81  :T-Disconnect
@[tNo reaction.
@[tThe BDUT persists in State 'OPEN_WAIT'.
```

2.2.5 Telegram sequence 11 : Connect from the local user to an existing device (styles 1/2)

Purpose: Check whether the BDUT passes the following states.

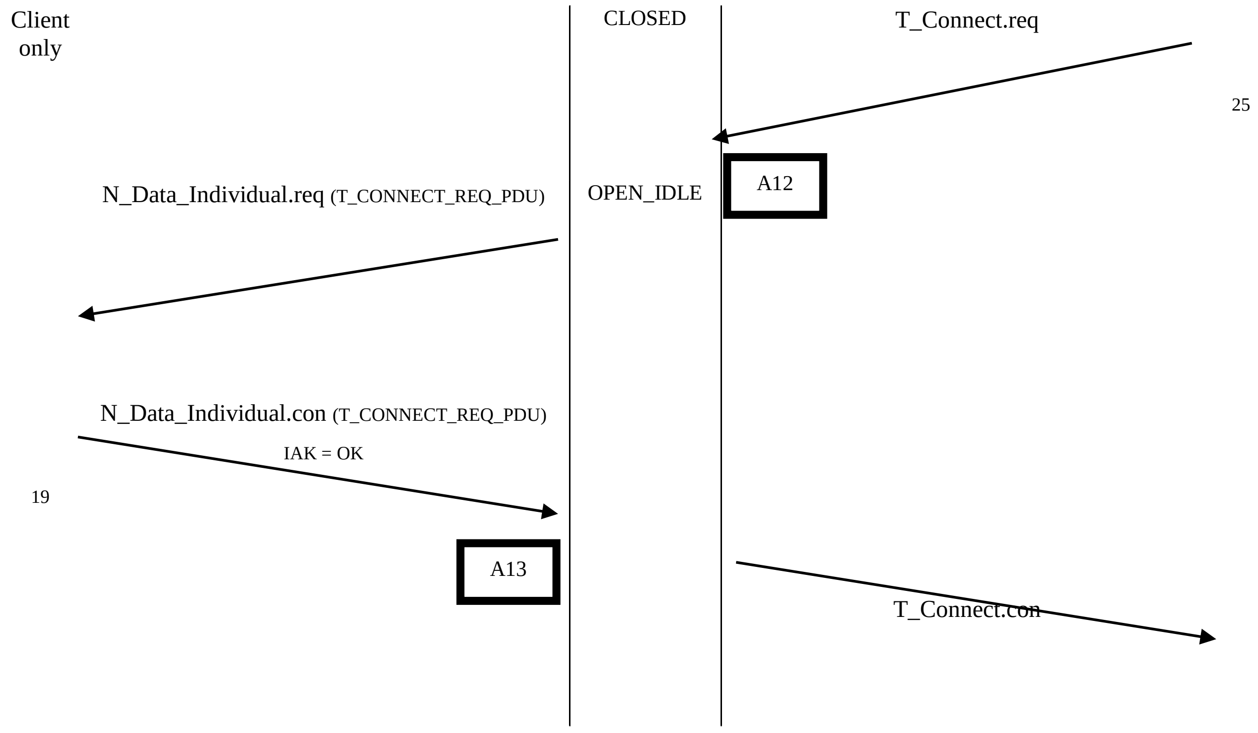

Client only
CLOSED
T_Connect.req
25
N_Data_Individual.req (T_CONNECT_REQ_PDU)
OPEN_IDLE
A12
N_Data_Individual.con (T_CONNECT_REQ_PDU)
IAK = OK
19
A13
T_Connect.con

Figure 11: State transition for telegram sequence 11

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to an existing device (remote BCU).
(2)          IN   00:00:00.2  43 00 00 00 A0 01  :T_CONNECT.req
@[t3. Check that a telegram is sent addressed to remote BCU with
NSDU=T_CONNECT.req
(3)          OUT  00:00:00.0  B0 A000 A001 60 80  :T-Connect(Addr=10.00.001)
@[t4. Check that the BDUT sends a T_CONNECT_con when the IAK=OK from the
existing device (remote BCU).
(4)          OUT  00:00:00.0  86 B0 A0 01 A0 00  :T_CONNECT.con

@[tThe BDUT is in State 'OPEN_IDLE'.
```

Note : Style 3 - On A12 BDUT changes to CONNECTING and on A13 to OPEN_IDLE.

2.2.6 Telegram sequence 12: Connect from the local user to an non existing device (style 1)

Purpose: Check whether the BDUT passes the following states.

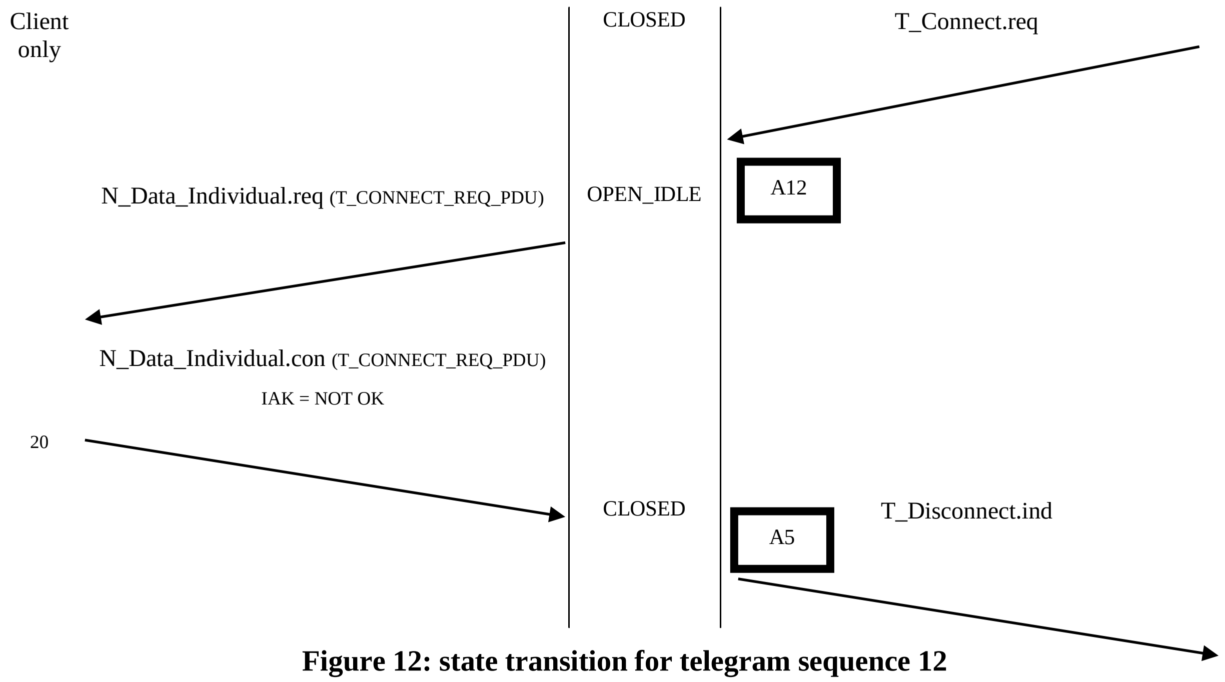

Client only
CLOSED
T_Connect.req
25
N_Data_Individual.req (T_CONNECT_REQ_PDU)
OPEN_IDLE
A12
N_Data_Individual.con (T_CONNECT_REQ_PDU)
IAK = NOT OK
20
CLOSED
A5
T_Disconnect.ind

Figure 12: state transition for telegram sequence 12

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)         IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[tThe BDUT is in State 'CLOSED'.

@[t2. Send a T_CONNECT to a non existing device (Address A005h).
(2)         IN   00:00:00.2  43 00 00 00 A0 05  :T_CONNECT.req

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t3. Check that a telegram is sent addressed to non existing BCU with
NSDU=T_CONNECT.req
(3)         OUT  00:00:00.0  B0 A000 A005 60 80  :T-Connect(Addr=10.00.005)
(only visible by third PC with EITT in Busmonitor mode)
@[t4. Check that a T_DISCONNECT_ind is sent, when receiving no IAK.
(4)         OUT  00:00:00.0  87 00 A0 00 A0 00  :T_DISCONNECT.ind

@[tThe BDUT is in State 'CLOSED'.
Note : Style 3 - On A12 BDUT changes to CONNECTING and on A5 to CLOSED -
Style 2: DUT sends no T.Disconnect.ind and remains in OPEN_IDLE”
```

2.2.7 Connect from the local user during an existing connection

2.2.7.1 Telegram Sequence 13: Procedure with initial state 'OPEN_IDLE' (styles 1/3)

Purpose: Check whether the BDUT passes the following states.

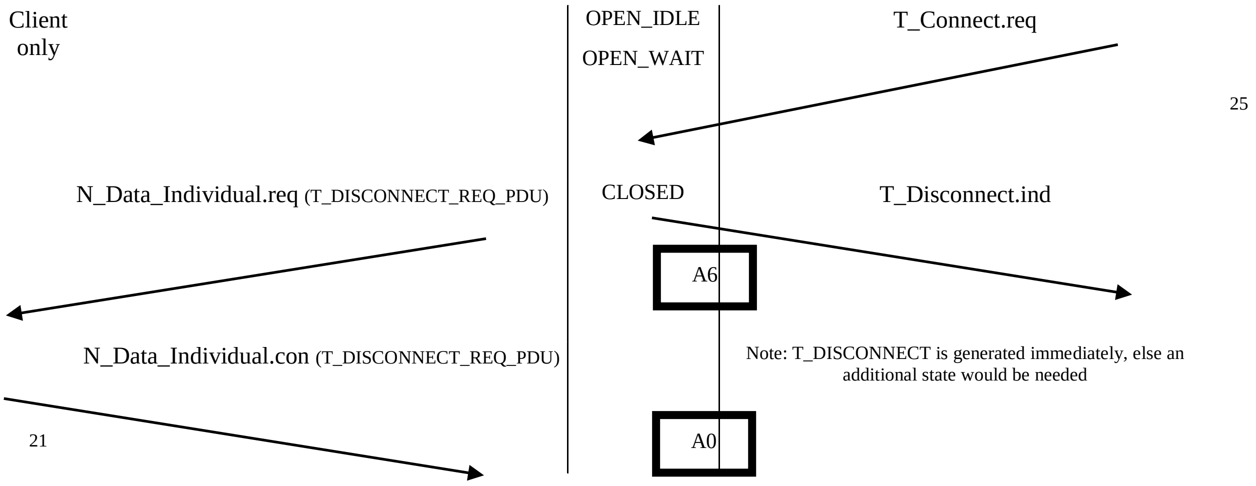

Client only
OPEN_IDLE
OPEN_WAIT
T_Connect.req
25
N_Data_Individual.req (T_DISCONNECT_REQ_PDU)
CLOSED
T_Disconnect.ind
A6
N_Data_Individual.con (T_DISCONNECT_REQ_PDU)
Note: T_DISCONNECT is generated immediately, else an additional state would be needed
21
A0

Figure 13: State transition for telegram sequence 13

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to remote BCU.
(2)          IN   00:00:00.2  43 00 00 00 A0 01  :T_CONNECT.req
@[t3. Check that a telegram is sending addressed to remote BCU with
NSDU=T_CONNECT.req
(3)          OUT  00:00:00.0  B0 A000 A001 60 80  :T-Connect(Addr=10.00.001)
@[t4. Check that the BDUT conirms the T_CONNECT when the IAK=OK from the
existing device (remote BCU).
(4)          OUT  00:00:00.0  86 B0 A0 01 A0 00  :T_CONNECT.con

@[tConnection established. State 'OPEN_IDLE'.

@[t5. Connect during an existing connection.
(5)          IN   00:00:00.2  43 00 00 00 A0 01  :T_CONNECT.req
@[t6. Check that a telegram is sent addressed to BCU with
NSDU=T_DISCONNECT.req
(6)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
@[t7. Check that a T_DISCONNECT_ind is sent
(7)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind

@[tThe BDUT goes in State 'CLOSED'.
```

Note : Style 2 - BDUT remains in OPEN_IDLE, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

2.2.7.2 Telegram Sequence 14: Procedure with initial state 'OPEN_WAIT' (styles 1/3)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to remote BCU.
(2)          IN   00:00:00.2  43 00 00 00 A0 01  :T_CONNECT.req
@[t3. Send telegram addressed to remote BCU with NSDU=T_CONNECT.req
(3)          OUT  00:00:00.0  B0 A000 A001 60 80  :T-Connect(Addr=10.00.001)
@[t4. Check that the BDUT sends a T_CONNECT_con when the IAK=OK from the
existing device (remote BCU).
(4)          OUT  00:00:00.0  86 B0 A0 01 A0 00  :T_CONNECT.con

@[tConnection established. State 'OPEN_IDLE'.

@[t5. Send a T_DATA_CONNECTED.req to remote BCU.
(5)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t6. Send telegram addressed to remote BCU with NSDU=T_DATA_CONNECTED.req
(6)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tBDUT is in State 'OPEN_WAIT'.

@[t7. Connect during an existing connection.
(7)          IN   00:00:00.2  43 00 00 00 A0 01  :T_CONNECT.req
@[t8. Check that a telegram is sent addressed to BCU with
NSDU=T_DISCONNECT.req
(8)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
@[t9. Check that a T_DISCONNECT_ind is sent
(9)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind

@[tNow the BDUT is in State 'CLOSED'.
```

Note : Style 2 - BDUT remains in OPEN_WAIT, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

2.2.8 Disconnect from the local user

2.2.8.1 Telegram sequence 15: Procedure with initial state ‘OPEN_IDLE’ (Styles 1/3)

Purpose: Check whether the BDUT passes the following states.

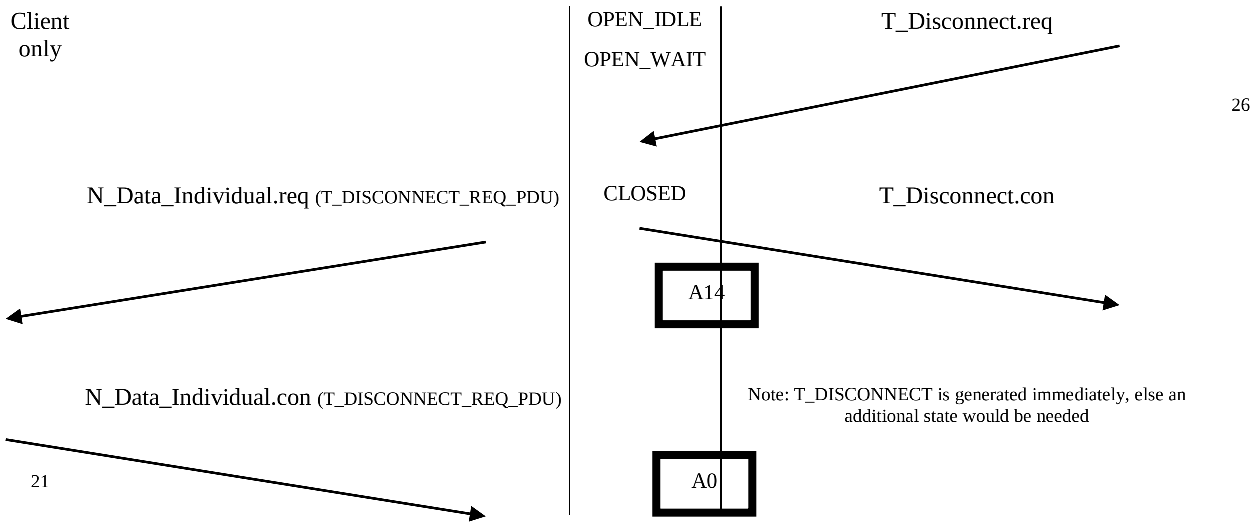

Client only
OPEN_IDLE
OPEN_WAIT
T_Disconnect.req
26
N_Data_Individual.req (T_DISCONNECT_REQ_PDU)
CLOSED
T_Disconnect.con
A14
N_Data_Individual.con (T_DISCONNECT_REQ_PDU)
Note: T_DISCONNECT is generated immediately, else an additional state would be needed
21
A0

Figure 14: State transition for telegram sequence 15

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT.req to remote BCU (client).
(2)          IN   00:00:00.2  43 00 00 00 A0 01  :T_CONNECT.req
@[t3. Send telegram addressed to remote BCU with NSDU=T_CONNECT.req
(3)          OUT  00:00:00.0  B0 A000 A001 60 80  :T-Connect(Addr=10.00.001)
@[t4. Check that the BDUT sends a T_CONNECT_con when the IAK=OK from the
existing device (remote BCU).
(4)          OUT  00:00:00.0  86 B0 A0 01 A0 00  :T_CONNECT.con

@[tConnection established. State 'OPEN_IDLE' (client).

@[t5. T_DISCONNECT.req from local user (BDUT).
(5)          IN   00:00:00.2  44 00 00 00 00 00  :T_DISCONNECT.req
@[t6. Check that the client sends an T_DISCONNECT_con to the local user.
(6)          OUT  00:00:00.0  88 00 00 00 00 00  :T_DISCONNECT.con
@[t7. Check that a telegram is sent addressed to remote BCU with
NSDU=T_DISCONNECT.req
(7)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tBDUT is in State 'CLOSED' –
Note : Style 2 - DUT remains in OPEN_IDLE and sends T_Disconnect.ind”
```

2.2.8.2 Telegram sequence 16: Procedure with initial state ‘OPEN_WAIT’ (Styles 1/3)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req
```

```
@[t2. Send T_CONNECT.req to remote BCU (client).
(2)         IN   00:00:00.2  43 00 00 00 A0 01  :T_CONNECT.req
@[t3. Send telegram addressed to remote BCU with NSDU=T_CONNECT.req
(3)         OUT  00:00:00.0  B0 A000 A001 60 80  :T-Connect(Addr=10.00.001)
@[t4. Check that the BDUT sends a T_CONNECT_con when the IAK=OK from the
existing device (remote BCU).
(4)         OUT  00:00:00.0  86 B0 A0 01 A0 00  :T_CONNECT.con

@[tConnection established. State 'OPEN_IDLE'.

@[t5. Send T_DATA_CONNECTED.req to remote BCU (client).
(5)         IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t6. A telegram is sent to remote BCU with NSDU=T_DATA_CONNECTED.req
(6)         OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tState 'OPEN_WAIT'.

@[t7. T_DISCONNECT.req from the local user (BDUT).
(7)         IN   00:00:00.2  44 00 00 00 00 00  :T_DISCONNECT.req
@[t8. Check that the client sends an T_DISCONNECT_con to the local user.
(8)         OUT  00:00:00.0  88 00 00 00 00 00  :T_DISCONNECT.con
@[t9. Check that a telegram is sent addressed to remote BCU with
NSDU=T_DISCONNECT.req
(9)         OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tBDUT is in State 'CLOSED'.
```

Note: : Style 2: DUT remains in OPEN_WAIT, DUT sends Disconnect on the bus + disconnect.ind after leaving OPEN_WAIT

2.2.9 Telegram sequence 17: Disconnect from the local user without an existing connection (styles 1/3)

Purpose: Check whether the BDUT passes the following states.

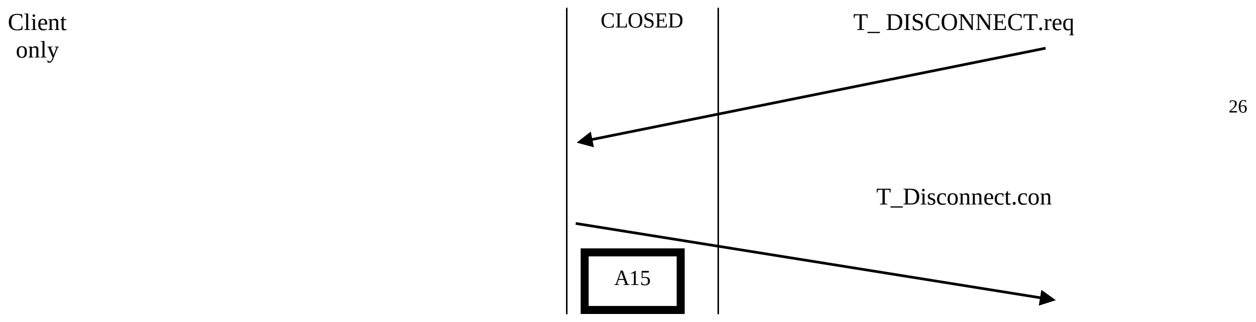


Figure 15: State transition for telegram sequence 17

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t Set PEI of BDUT to Transport-Layer (remote)
(1)         IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[tState 'CLOSE'(client).

@[t2. T_DISCONNECT.req from local user (BDUT).
(2)         IN   00:00:00.2  44 00 00 00 00 00  :T_DISCONNECT.req
@[t3. Check that the client sends an T_DISCONNECT_con to the local user.
(3)         OUT  00:00:00.0  88 00 00 00 00 00  :T_DISCONNECT.con
```

Note : Style 2 - BDUT sends no T_Disconnect.con.

2.2.10 Connection timeout

2.2.10.1 Telegram sequence 18: Procedure with initial state 'OPEN_IDLE' (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.

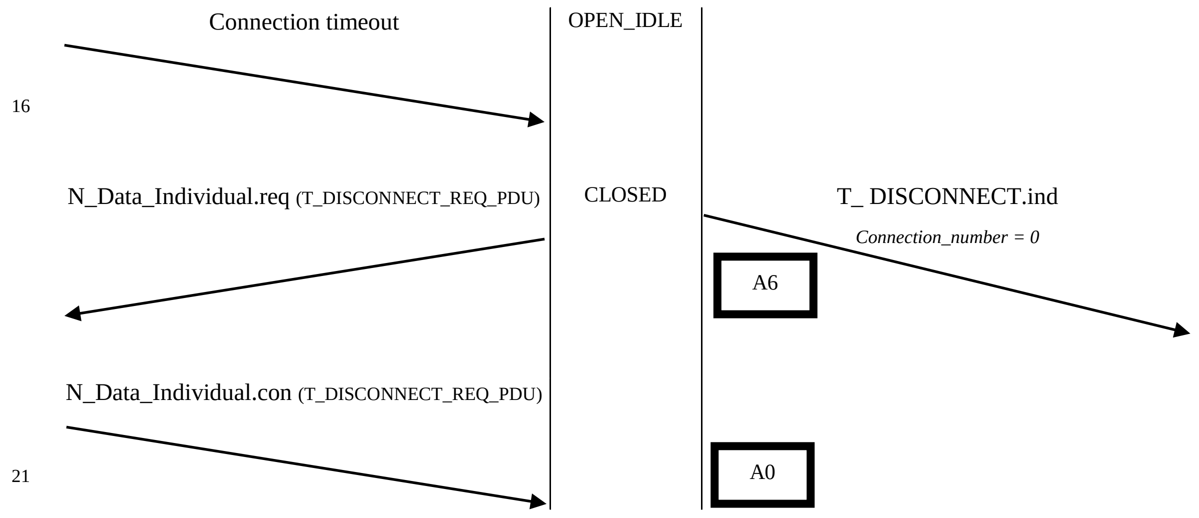

Connection timeout
OPEN_IDLE
16
N_Data_Individual.req (T_DISCONNECT_REQ_PDU)
CLOSED
T_ DISCONNECT.ind
Connection_number = 0
A6
N_Data_Individual.con (T_DISCONNECT_REQ_PDU)
21
A0

Figure 16: State transition for telegram sequence 18

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in state 'OPEN_IDLE'.

@[tThe connection timeout timer has started.
@[t4.+5. After the time interval of 6s (connection timeout timer is released)
the BDUT will close the connection by sending a T_DISCONNECT.ind and a
telegram with NSDU=T_DISCONNECT_REQ.
(4)          OUT  00:00:06.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
(5)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
```

2.2.11 Acknowledgement timeout

2.2.11.1 Telegram sequence 19: Procedure with initial state 'OPEN_WAIT' (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.

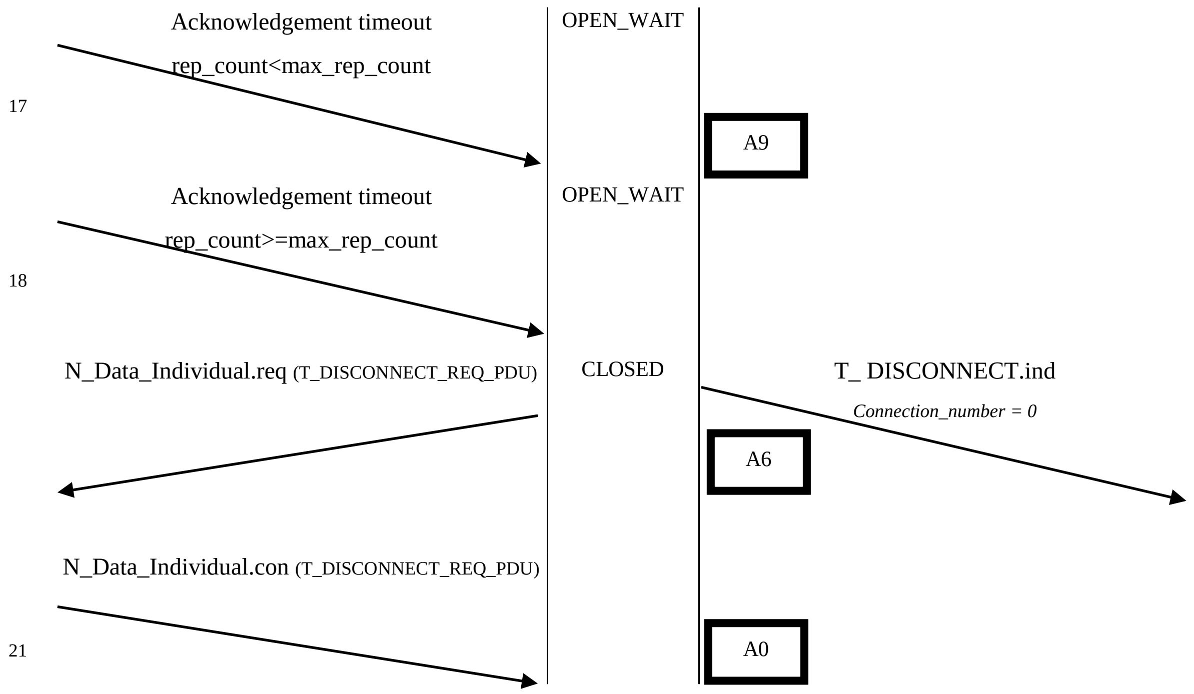


Figure 17: State transition for telegram sequence 19

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in state 'OPEN_IDLE'.

@[t5. Send T_DATA_CONNECTED.req to remote BCU (client).
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t6. A telegram is sent to remote BCU with NSDU=T_DATA_CONNECTED.req
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tThe BDUT is in state 'OPEN_WAIT'.
------------------------------------------------------------------------
@[tThe acknowledgement timeout timer has started.
(6)          OUT  00:00:03.0  B0 A000 A001 61 43  00  :MaskVersionRead()
(7)          OUT  00:00:03.0  B0 A000 A001 61 43  00  :MaskVersionRead()
(8)          OUT  00:00:03.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tIf the repeat counter = 3 (maximum of T_DATA_CONNECTED.req repetitions),
the acknowledgement timeout timer stops and the user gets a T_DISCONNECT.ind.
```

```
(9)         OUT  00:00:00.2  87 00 A0 01 A0 00  :T_DISCONNECT.ind
(10)        OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT is in state 'CLOSED'.
-------------------------------------------------------------------------
Or alternatively for step (6) to step (10) for Style 1 rationalised

@[tThe DUT is in state 'OPEN_WAIT'.
@[tThe connection timeout timer has started. If no T_ACK is received, the
BDUT closes the connection after 6 sec.
(6)         OUT  00:00:06.0  87 00 00 00 00 00:T_DISCONNECT.ind
(7)         OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
@[tThe DUT is in state 'CLOSED'.
```

2.3 Reception of data

2.3.1 Telegram sequence 20a: Reception of a correct N_Data_Individual – Procedure with initial state ‘OPEN_IDLE’ (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.

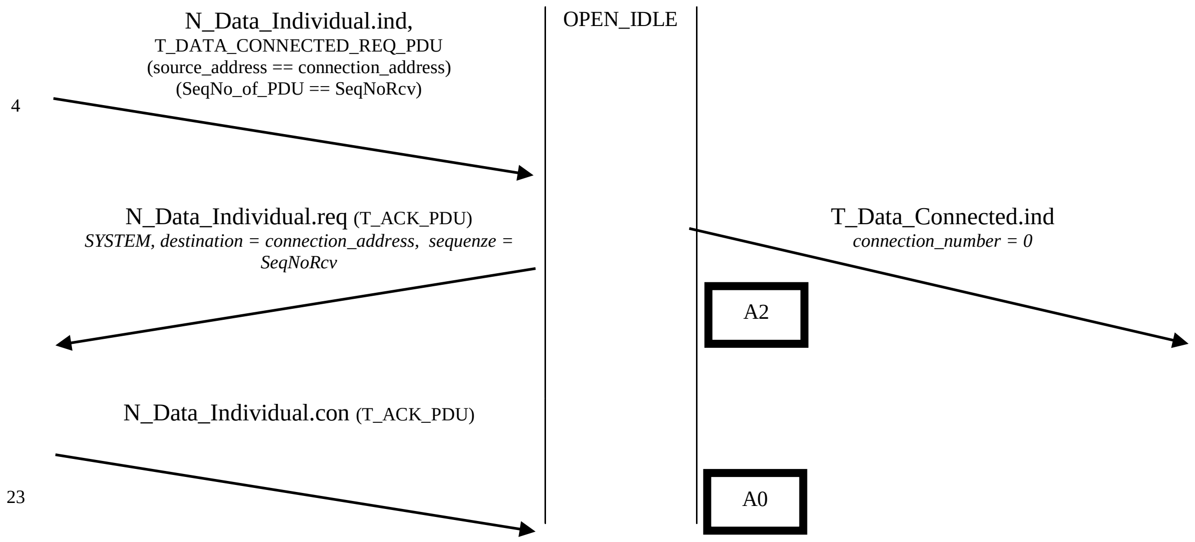

N_Data_Individual.ind,
T_DATA_CONNECTED_REQ_PDU
(source_address == connection_address)
(SeqNo_of_PDU == SeqNoRcv)
4
OPEN_IDLE
N_Data_Individual.req (T_ACK_PDU)
SYSTEM, destination = connection_address, sequenze = SeqNoRcv
T_Data_Connected.ind
connection_number = 0
A2
N_Data_Individual.con (T_ACK_PDU)
23
A0

Figure 18: State Transition for telegram sequence 20

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)         IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)         IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)         OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.
```

```
@[t4. Now the BDUT receives a NSDU=T_DATA_CONNECTED_REQ (MaskVersionRead).
(4)          IN   00:00:00.2  B0 A001 A000 61 43  00  :MaskVersionRead()
@[t5. Check that the BDUT is sent a T_DATA_CONNECTED.ind and a telegram with
NSDU=T_ACK.
(5)          OUT  00:00:00.0  89 B0 A0 01 A0 00 61 43 00
:T_DATA_CONNECTED.ind
(6)          OUT  00:00:00.0  B0 A000 A001 60 C2  :T-Ack(Seq=0)

@[tThe BDUT persists in State 'OPEN_IDLE'.
```

2.3.2 Telegram sequence 20b: Reception of a correct N_Data_Individual – Procedure with initial state ‘OPEN_WAIT’ (without PEI, all styles)

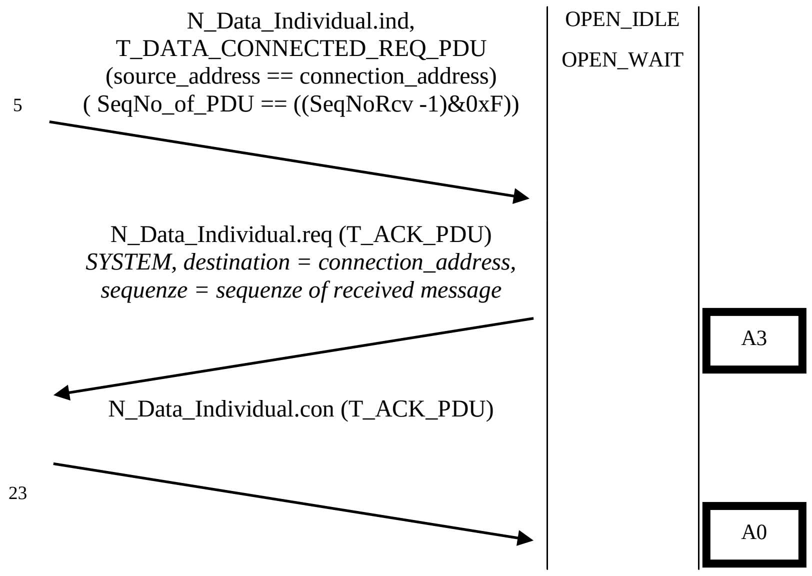
```
Note: the ?? characters denote wildcards for the reception of data in EITT.

@[t Purpose: check whether a repetition of an N_DataIndividual with
SeqNo_of_PDU = SeqNoRcv is accepted and the data are stored.
@[t Initial state: Send T_Connect and MaskVersionRead to BDUT to establish
OPEN_WAIT as the initial state for the sequence.
(1)          IN        start  B0 AFFE 1001 60 80  :T-Connect(Addr=1001)
@[t BDUT is in State OPEN_IDLE
@[t Now the BDUT receives a MaskVersionRead
(2)          IN   00:00:02.0  B0 AFFE 1001 61
43  00  :DeviceDescriptorRead(DescType=00)
(3)          OUT  00:00:00.0  B0 1001 AFFE 60 C2  :T-Ack(Seq=0)
@[t BDUT is in State OPEN_IDLE
(4)          OUT  00:00:00.0  B0 1001 AFFE 63 43  40 ??
??  :DeviceDescriptorResponse(DescType=00, Descriptor=?? ?? )
@[t BDUT is in State OPEN_WAIT
(5)          IN   00:00:03.5  B0 AFFE 1001 63 46  01 01
FE  :MemoryRead(Count=01, Addr=01FE - may be any memory location)
(6)          OUT  00:00:00.0  B0 1001 AFFE 60 C6  :T-Ack(Seq=1)
@[t BDUT persists in state OPEN_WAIT
(7)          OUT  00:00:00.2  B0 1001 AFFE 63 43  40 ??
??  :DeviceDescriptorResponse(DescType=00, Descriptor=?? ?? )
(8)          IN   00:00:06.5  B0 AFFE 1001 60 C2  :T-Ack(Seq=0)
@[t BDUT is now in OPEN_IDLE and sends the stored Memory-Response
(9)          OUT  00:00:00.0  B0 1001 AFFE 64 46  41 01 FE
??  :MemoryResponse(Count=01, Addr=01FE, Data=?? )
(10)         OUT  00:00:03.5  B0 1001 AFFE 64 46  41 01 FE
??  :MemoryResponse(Count=01, Addr=01FE, Data=?? )
(11)         OUT  00:00:03.5  B0 1001 AFFE 64 46  41 01 FE
??  :MemoryResponse(Count=01, Addr=01FE, Data=?? )
(12)         OUT  00:00:03.5  B0 1001 AFFE 64 46  41 01 FE
??  :MemoryResponse(Count=01, Addr=01FE, Data=?? )
(13)         OUT  00:00:03.5  B0 1001 AFFE 60 81  :T-Disconnect
```

2.3.3 Reception of a repeated N_Data_Individual

2.3.3.1 Telegram sequence 21: Procedure with initial state 'OPEN_IDLE' (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.



N_Data_Individual.ind,
T_DATA_CONNECTED_REQ_PDU
(source_address == connection_address)
(SeqNo_of_PDU == ((SeqNoRcv -1)&0xF))
5
OPEN_IDLE
OPEN_WAIT
N_Data_Individual.req (T_ACK_PDU)
SYSTEM, destination = connection_address,
sequenze = sequenze of received message
A3
N_Data_Individual.con (T_ACK_PDU)
23
A0

Figure 19: State diagram for telegram sequence 21

```
Purpose : check whether a repetition of an N_Data_Individual with
SeqNo_of_PDU = '-1" is accepted and the data is ignored.

Initial state: A T_Connect is sent to establish the initial OPEN_IDLE
state

IN    B0 AFFE 1001 60 80 : T_Connect (Addr=1001)

BDUT is now in State Open_Idle – now the BDUT receives a
MaskVersionRead

IN    B0 AFFE 1001 61 7F 00 : MaskVersionRead()

OUT   B0 1001 AFFE 60 FE : T_Ack (Seq=F)

BDUT is in State Open_Idle for a few seconds

OUT   B0 1001 AFFE 60 81 : T_Disconnect

End State : closed
```

2.3.3.2 Telegram sequence 22: Procedure with initial state 'OPEN_WAIT' (without PEI, all styles)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Send T_DATA_CONNECTED.req to remote BCU.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t5. A telegram is sent to remote BCU with NSDU=T_DATA_CONNECTED.req
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tThe BDUT is in state 'OPEN_WAIT'.

@[t6. Now the BDUT receives a NSDU=T_DATA_CONNECTED_REQ (MaskVersionRead).
(6)          IN   00:00:00.2  B0 A001 A000 61 43  00  :MaskVersionRead()
@[t7. Check that the BDUT sends a T_DATA_CONNECTED.ind and
@[t8. a telegram with NSDU=T_DATA_ACK.
(7)          OUT  00:00:00.0  89 B0 A0 01 A0 00 61 43 00
:T_DATA_CONNECTED.ind
(8)          OUT  00:00:00.0  B0 A000 A001 60 C2  :T-Ack(Seq=0)

@[t9. The BDUT receives a repeated NSDU=T_DATA_CONNECTED_REQ
(MaskVersionRead).
(9)          IN   00:00:00.2  B0 A001 A000 61 43  00  :MaskVersionRead()
@[t10. Check that the BDUT sends a telegram with NSDU=T_DATA_ACK.
(10)         OUT  00:00:00.0  B0 A000 A001 60 C2  :T-Ack(Seq=0)

@[tThe BDUT persists in state 'OPEN_WAIT'.
```

2.3.4 Reception of data N_Data_Individual with wrong sequence number

2.3.4.1 Telegram sequence 23: Procedure with initial state 'OPEN_IDLE' (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.

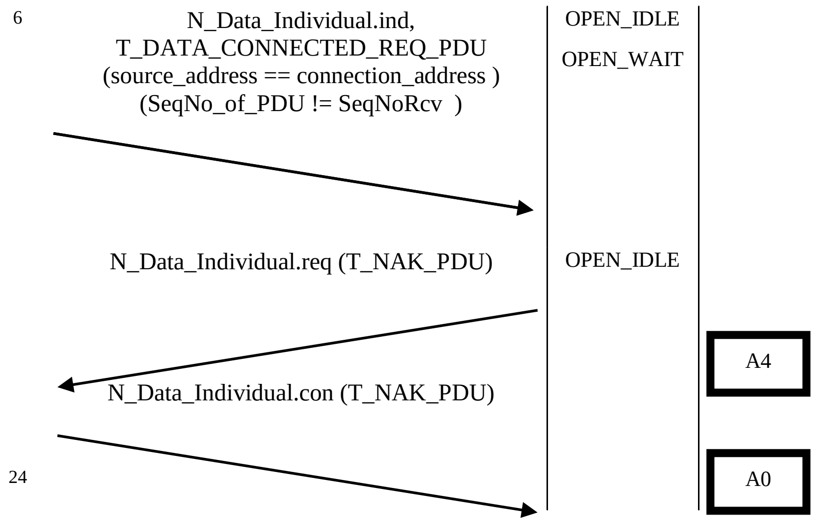

6
N_Data_Individual.ind,
T_DATA_CONNECTED_REQ_PDU
(source_address == connection_address)
(SeqNo_of_PDU != SeqNoRcv)
OPEN_IDLE
OPEN_WAIT
N_Data_Individual.req (T_NAK_PDU)
OPEN_IDLE
A4
N_Data_Individual.con (T_NAK_PDU)
24
A0

Figure 20: State transition for telegram sequence 23

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind
----------------------------------------------------------------
@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. The BDUT receives a MaskVersionRead with a wrong sequence number.
(4)          IN   00:00:00.2  B0 A001 A000 61 57  00  :MaskVersionRead()
@[t5. Check that the BDUT sends a T_NAK. (Seq=5)
(5)          OUT  00:00:00.0  B0 A000 A001 60 D7  :T-Nack(Seq=5)

@[tThe BDUT persists in State 'OPEN_IDLE'.
-----------------------------------------------------------------
Alternatively for Style 1 rationalised

@[tThe DUT is in State 'OPEN_IDLE'.
@[t4. The DUT receives a MaskVersionRead with a wrong sequence number.
(4)          IN   00:00:00.2  B0 A001 A000 61 57  00  :MaskVersionRead()

@[t5. Check that the DUT sends a T_Disconnect on the bus and a
T_Disconnect.ind locally.
(5)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
(6)          OUT  00:00:00.0  87 00 00 00 00 00  :T_DISCONNECT.ind
@[tThe DUT changes to "CLOSED".
```

2.3.4.2 Telegram sequence 24: Procedure with initial state ‘OPEN_WAIT’ (without PEI, all styles)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Send T_DATA_CONNECTED.req to remote BCU.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t5. A telegram is sent to remote BCU with NSDU=T_DATA_CONNECTED.req
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()
------------------------------------------------------------------------
@[tThe BDUT is in state 'OPEN_WAIT'.

@[t6. The BDUT receives a MaskVersionRead with a wrong sequence number.
(6)          IN   00:00:00.2  B0 A001 A000 61 57  00  :MaskVersionRead()
@[t7. Check that the BDUT sends a T_NAK. (Seq=5)
(7)          OUT  00:00:00.0  B0 A000 A001 60 D7  :T-Nack(Seq=5)

@[tThe BDUT is still in state 'OPEN_WAIT'.
------------------------------------------------------------------------
Alternatively for Style 1 rationalised
@[tThe DUT is in state 'OPEN_WAIT'.
@[t6. The DUT receives a MaskVersionRead with a wrong sequence number.
(6)          IN   00:00:00.2  B0 A001 A000 61 57  00  :MaskVersionRead()

@[t7. Check that the DUT sends a T_Disconnect on the Bus and a
T_Disconnect.ind locally.
(7)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
(8)          OUT  00:00:00.0  87 00 00 00 00 00  :T_DISCONNECT.ind
@[tThe DUT changes to "CLOSED".
```

2.3.5 Reception of data N_Data_Individual with wrong source address

2.3.5.1 Telegram sequence 25: Procedure with initial state 'OPEN_IDLE' (without PEI, style 1)

Purpose: Check whether the BDUT passes the following states.

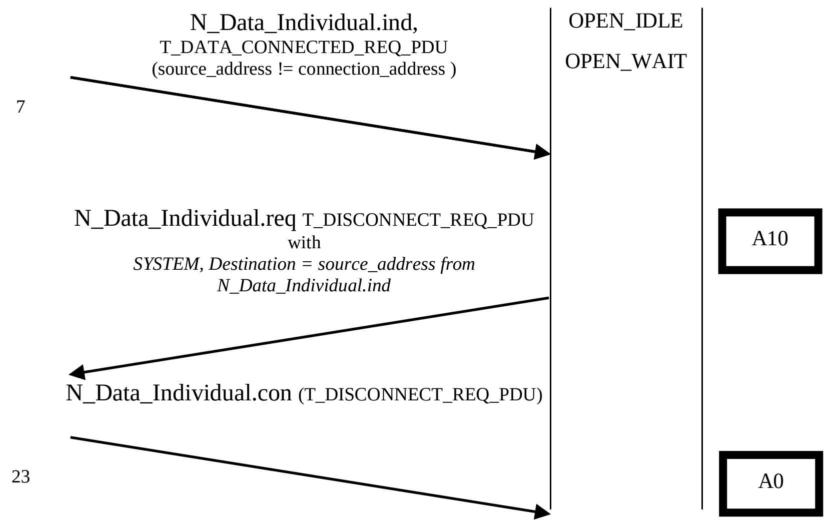

N_Data_Individual.ind,
T_DATA_CONNECTED_REQ_PDU
(source_address != connection_address)
OPEN_IDLE
OPEN_WAIT
7
N_Data_Individual.req T_DISCONNECT_REQ_PDU
with
SYSTEM, Destination = source_address from N_Data_Individual.ind
A10
N_Data_Individual.con (T_DISCONNECT_REQ_PDU)
23
A0

Figure 21: State transition for telegram sequence 25

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Activate the
Link-Layer.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. BDUT receives a T_CONNECT.con
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. The BDUT receives a MaskVersionRead from remote BCU (<> connected BCU =
third BCU)
(4)          IN   00:00:00.2  B0 A001 A000 61 43  00  :MaskVersionRead()
@[t5. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT_REQ.
(5)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'OPEN_IDLE'.
```

Note: Style 2/3 - BDUT sends no Disconnect on the bus.

2.3.5.2 Telegram sequence 26: Procedure with initial state 'OPEN_WAIT' (without PEI, style 1)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Activate the
Link-Layer.
```

```
@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. BDUT receives a T_CONNECT.con
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. BDUT sends a T_DATA_CONNECTED.req to the third BCU.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req

@[tThe BDUT is in State 'OPEN_WAIT'.

@[t5. The BDUT receives a MaskVersionRead from remote BCU (<> connected BCU =
third BCU)
(5)          IN   00:00:04.0  B0 A001 A000 61 43  00  :MaskVersionRead()
@[t5. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT_REQ.
(6)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'OPEN_WAIT'.
```

Note: Style 2/3 - BDUT sends no Disconnect on the bus.

2.4 Transmission of data

2.4.1 T_DATA_CONNECTED-reqest from the local user

2.4.1.1 Telegram sequence 27: Procedure with initial state ‘OPEN_IDLE’ (style 1/3)

Purpose: Check whether the BDUT passes the following states.

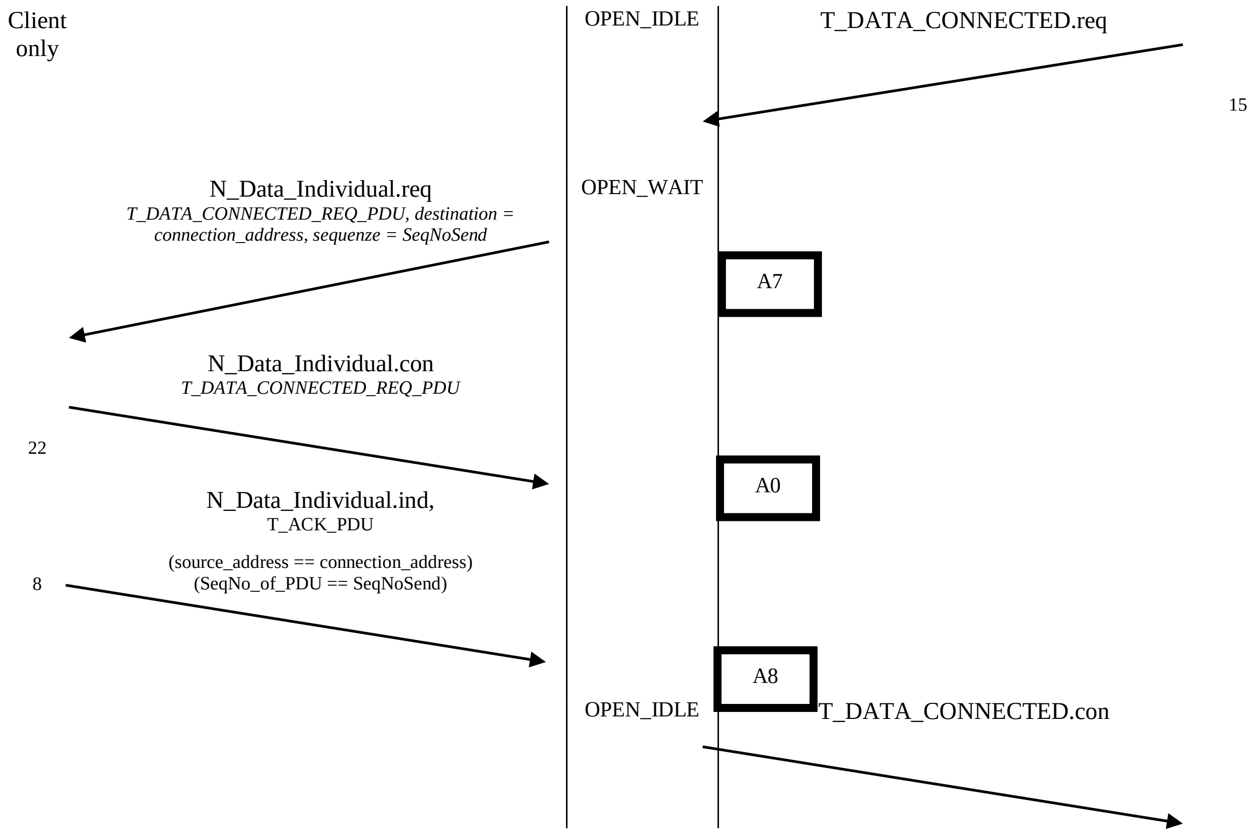


Figure 22: State transition for telegram sequence 27

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT gets a T_DATA_CONNECTED.req.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t5. Check that the BDUT sends a telegram with NSDU=T_DATA_CONNECTED.req.
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tThe BDUT is in State 'OPEN_WAIT'.
```

```
@[t6. Now the BDUT receives a T_ACK.
@[t7. Check that the BDUT goes to 'OPEN_IDLE' and sends a
T_DATA_CONNECTED.con
(6)          IN   00:00:00.2  B0 A001 A000 60 C2  :T-Ack(Seq=0)
(7)          OUT  00:00:00.0  8E B0 A0 00 00 00 61 43 00
:T_DATA_CONNECTED.con
```

Note : Style 2 - BDUT sends no T_Data_Connected.con.

2.4.1.2 Telegram sequence 28: Procedure with initial state ‘CLOSED’ to check event 15 (style 1)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[tThe BDUT is in State 'CLOSED'.

@[t2. Now the BDUT gets a T_DATA_CONNECTED.req
(2)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t3. Check that a T_DISCONNECT_ind is sent.
(3)          OUT  00:00:00.0  87 00 A0 00 A0 00  :T_DISCONNECT.ind

@[tThe BDUT persists in State 'CLOSED'.
```

Note: Style 2/3 - BDUT sends no T_Disconnect.ind.

2.4.1.3 Telegram sequence 29: Procedure with initial state ‘OPEN_WAIT to check event 15 (style 1)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT gets a T_DATA_CONNECTED.req.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t5. The BDUT sends a telegram with NSDU=T_DATA_CONNECTED.req.
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()
```

```
---------------------------------------------------------------------------

@[tThe BDUT is in State 'OPEN_WAIT'.

@[t6. Now the BDUT gets a T_DATA_CONNECTED.req again.
(6)          IN   00:00:07.0  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t7. Check that a telegram is sent addressed to BCU with
NSDU=T_DISCONNECT.req
(7)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
@[t8. Check that a T_DISCONNECT_ind is sent
(8)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind

@[tThe BDUT is in State 'CLOSED'.
---------------------------------------------------------------------------
Alternatively for Style 1 rationalised

@[tThe DUT is in State 'OPEN_WAIT'.
@[t6. The DUT receives a second T_DATA.requ.
(6)          IN   00:00:03.0  41 B0 00 00 00 00 01 03 00  :T_DATA.req

@[t7. No reaction to the second T_Data.req, until BDUT receives a T_ACK.
(7)          IN   00:00:02.0  B0 A001 A000 60 C2  :T-Ack(Seq=0)
(8)          OUT  00:00:00.0  8E B0 A0 00 AF 00 61 43 00  :T_DATA.conf
(9)          OUT  00:00:00.0  B0 A000 A001 61 47  00  :MaskVersionRead()

@[t8. BDUT is in OPEN_WAIT, timeout after 6s.
(10)         OUT  00:00:05.8  B0 A000 A001 60 81  :T-Disconnect
(11)         OUT  00:00:00.0  87 00 00 00 00 00  :T_DISCONNECT.ind
@[tThe DUT is in State 'CLOSED'.
```

Note: Style 2/3 - BDUT remains in OPEN_WAIT, BDUT sends T_Data_Connected on the bus after leaving OPEN_WAIT.

2.4.2 Reception of a T_ACK_PDU

2.4.2.1 Telegram sequence 30: Procedure with initial state 'OPEN_IDLE' (without PEI, styles 1/2)

Purpose: Check whether the BDUT passes the following states.

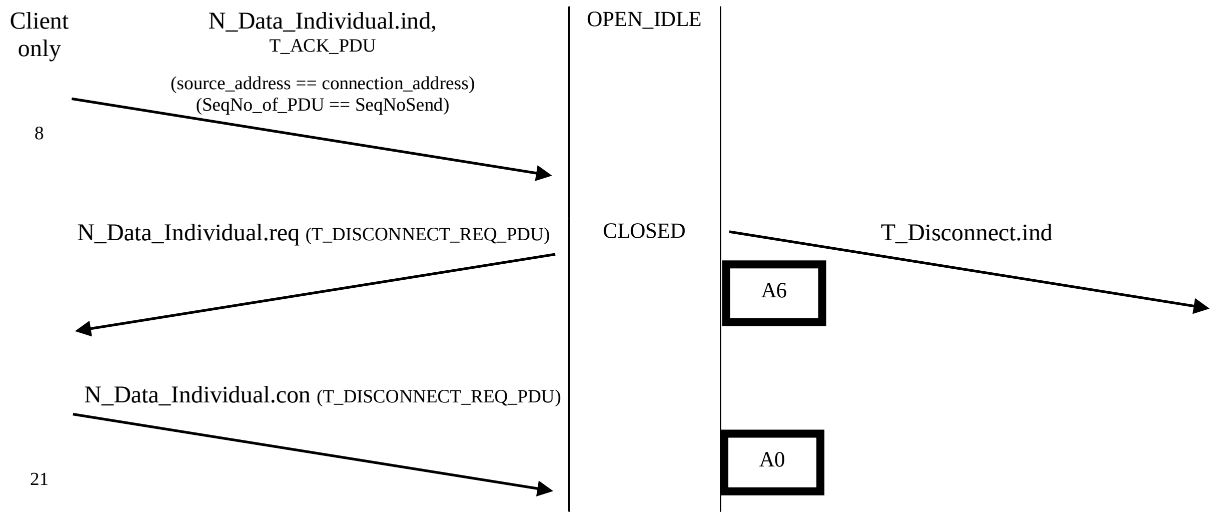

Client only
N_Data_Individual.ind,
T_ACK_PDU
(source_address == connection_address)
(SeqNo_of_PDU == SeqNoSend)
8
OPEN_IDLE
N_Data_Individual.req (T_DISCONNECT_REQ_PDU)
CLOSED
T_Disconnect.ind
A6
N_Data_Individual.con (T_DISCONNECT_REQ_PDU)
21
A0

Figure 23: State transition for telegram sequence 30

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN       start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req
@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT receives a T_ACK.
(4)          IN   00:00:00.2  B0 A001 A000 60 C2  :T-Ack(Seq=0)
@[t5. Check that a telegram is sent addressed to BCU with
NSDU=T_DISCONNECT.req
(5)          OUT  00:00:00.2  B0 A000 A001 60 81  :T-Disconnect
@[t6. Check that a T_DISCONNECT_ind is sent
(6)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind

@[tThe BDUT is in State 'CLOSED'.
```

Note: Style 3 - BDUT remains in OPEN_IDLE, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

2.4.3 Reception of an T_ACK_PDU with wrong sequence number

2.4.3.1 Telegram sequence 31: Procedure with initial state 'OPEN_IDLE' (without PEI, styles 1/2[3])

Purpose: Check whether the BDUT passes the following states.

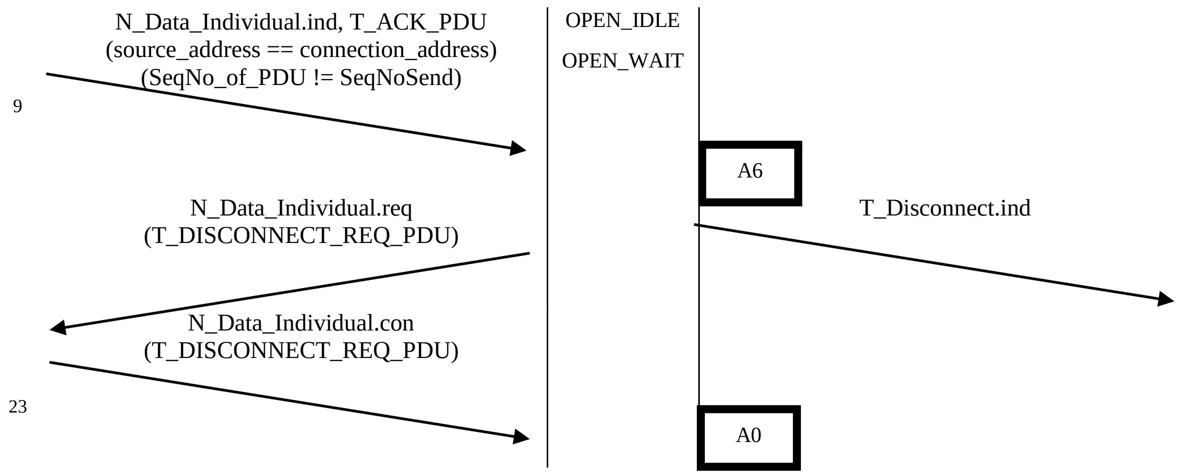


Figure 24: State transition for telegram sequence 31

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN       start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req
```

[3] Current System 1 implementations (see volume 6) transmit faulty telegrams. This is not allowed for updated versions or new developed system 1 implementations.

```
@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT receives a T_ACK with wrong sequence number.
(4)          IN   00:00:00.2  B0 A001 A000 60 D6  :T-Ack(Seq=5)
@[t5. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT.req.
(5)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
@[t6. Check that a T_DISCONNECT_ind is sent
(6)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind

@[tThe BDUT goes into state ‘CLOSED'.
```

Note: Style 3 - BDUT remains in OPEN_IDLE, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

2.4.3.2 Telegram sequence 32: Procedure with initial state ‘OPEN_WAIT’ (without PEI, styles 1/3)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT gets a T_DATA_CONNECTED.req.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t5. The BDUT sends a telegram with NSDU=T_DATA_CONNECTED.req.
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tThe BDUT is in State 'OPEN_WAIT'.

@[t6. Now the BDUT receives a T_ACK with wrong sequence number.
(6)          IN   00:00:00.2  B0 A001 A000 60 D6  :T-Ack(Seq=5)
@[t7. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT.req.
(7)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
@[t8. Check that a T_DISCONNECT_ind is sent
(8)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
```

Note: Style 2 - BDUT remains in OPEN_WAIT and sends no T_Disconnect.ind and no Disconnect on the bus.

2.4.4 Reception of T_ACK_PDU with wrong connection address

2.4.4.1 Telegram sequence 33: Procedure with initial state 'OPEN_IDLE' (without PEI, style 1)

Purpose: Check whether the BDUT passes the following states.

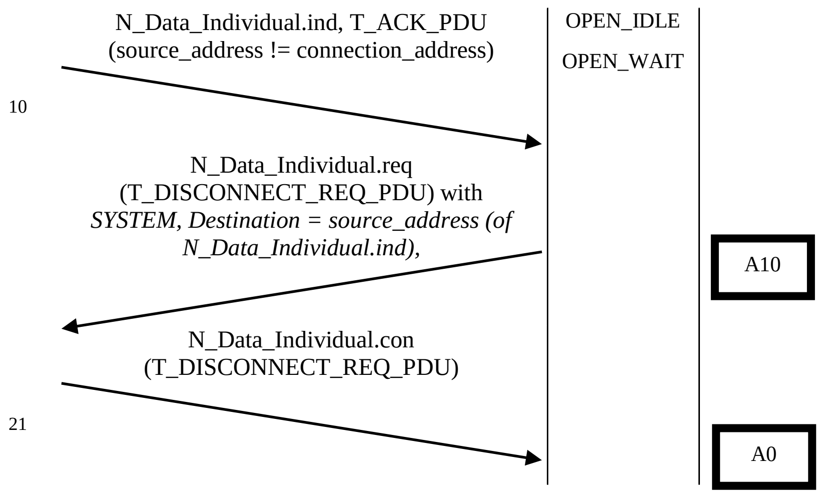

N_Data_Individual.ind, T_ACK_PDU
(source_address != connection_address)
OPEN_IDLE
OPEN_WAIT
10
N_Data_Individual.req
(T_DISCONNECT_REQ_PDU) with
SYSTEM, Destination = source_address (of N_Data_Individual.ind),
A10
N_Data_Individual.con
(T_DISCONNECT_REQ_PDU)
21
A0

Figure 25: State transition for telegram sequence 33

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Activate the
Link-Layer.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. BDUT receives a T_CONNECT.con
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. The BDUT receives a T_ACK from remote BCU (<> connected BCU = third
BCU)
(4)          IN   00:00:00.2  B0 A001 A000 60 C2  :T-Ack(Seq=0)
@[t5. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT.req.
(5)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'OPEN_IDLE'.
```

Note: Style 2/3 - BDUT sends no Disconnect on the bus.

2.4.4.2 Telegram sequence 34: Procedure with initial state ‘OPEN_WAIT’ (without PEI, style 1)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Activate the
Link-Layer.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. BDUT receives a T_CONNECT.con
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. BDUT sends a T_DATA_CONNECTED.req to the third BCU.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req

@[tThe BDUT is in State 'OPEN_WAIT'.

@[t5. The BDUT receives a T_ACK from remote BCU (<> connected BCU = third
BCU)
(5)          IN   00:00:04.0  B0 A001 A000 60 C2  :T-Ack(Seq=0)
@[t6. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT.req.
(6)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'OPEN_WAIT'.
```
Note: Style 2/3 - BDUT sends no Disconnect on the bus.

2.4.5 Reception of T_NAK_PDU with wrong sequence number

2.4.5.1 Telegram Sequence 35: Procedure with initial state 'OPEN_IDLE' (without PEI, style 1/2[4])

Purpose: Check whether the BDUT passes the following states.

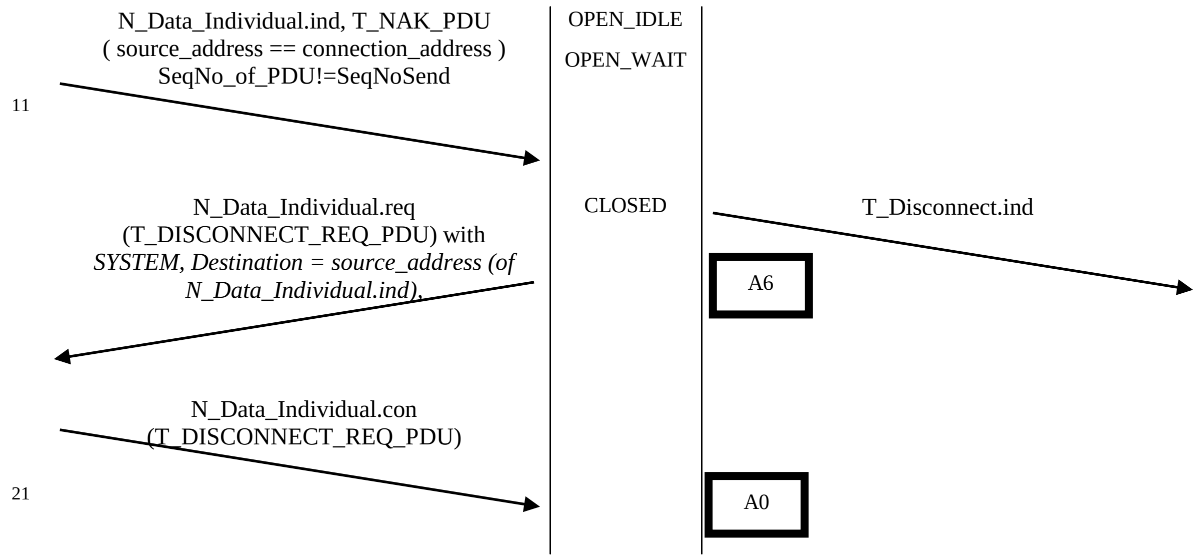

N_Data_Individual.ind, T_NAK_PDU
(source_address == connection_address)
SeqNo_of_PDU!=SeqNoSend
11
OPEN_IDLE
OPEN_WAIT
N_Data_Individual.req
(T_DISCONNECT_REQ_PDU) with
SYSTEM, Destination = source_address (of N_Data_Individual.ind).
CLOSED
T_Disconnect.ind
A6
N_Data_Individual.con
(T_DISCONNECT_REQ_PDU)
21
A0

Figure 26: State transition for sequence 35

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT receives a T_NACK with wrong sequence number.
(4)          IN   00:00:00.2  B0 A001 A000 60 D7  :T-Nack(Seq=5)
@[t5. Check that the BDUT sends a T_DISCONNECT.ind
(5)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
@[t6. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT.req.
(6)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT takes the State 'CLOSED'.
```

Note: Style 3 - BDUT remains in OPEN_IDLE, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

[4] Current System 1 implementations (see volume 6) transmit faulty telegrams. This is not allowed for updated versions or new developed system 1 implementations.

2.4.5.2 Telegram Sequence 36: Procedure with initial state ‘OPEN_WAIT’ (without PEI, style 1/3)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT gets a T_DATA_CONNECTED.req.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t5. The BDUT sends a telegram with NSDU=T_DATA_CONNECTED.req.
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tThe BDUT is in State 'OPEN_WAIT'.

@[t6. Now the BDUT receives a T_NACK with wrong sequence number.
(6)          IN   00:00:00.2  B0 A001 A000 60 D7  :T-Nack(Seq=5)
@[t7. Check that the BDUT sends a T_DISCONNECT.ind
(7)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
@[t8. Send telegram addressed to remote BCU with NSDU=T_DISCONNECT.req
(8)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tNow the BDUT is in State 'CLOSED'.
```

Note: Style 2 - BDUT remains in OPEN_WAIT, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

2.4.6 Telegram Sequence 37 : Reception of T_NAK_PDU with correct sequence number - Procedure with initial state 'OPEN_IDLE' (without PEI, style 1/3)

Purpose: Check whether the BDUT passes the following states.

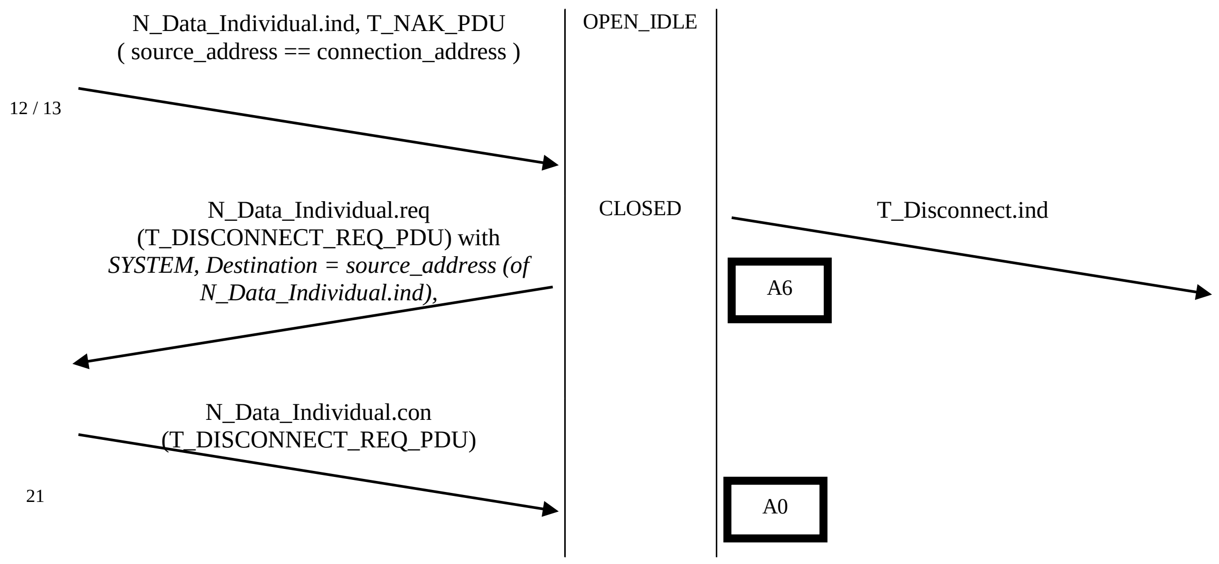

N_Data_Individual.ind, T_NAK_PDU
(source_address == connection_address)
OPEN_IDLE
12 / 13
N_Data_Individual.req
(T_DISCONNECT_REQ_PDU) with
SYSTEM, Destination = source_address (of N_Data_Individual.ind),
CLOSED
T_Disconnect.ind
A6
N_Data_Individual.con
(T_DISCONNECT_REQ_PDU)
21
A0

Figure 27: State transition for telegram sequence 37

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT receives a T_NACK with correct sequence number.
(4)          IN   00:00:00.2  B0 A001 A000 60 C3  :T-Nack(Seq=0)
@[t5. Check that the BDUT sends a T_DISCONNECT.ind
(5)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
@[t6. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT.req.
(6)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT takes the State 'CLOSED'.
```

Note: style 2 - BDUT remains in OPEN_IDLE, BDUT sends no T_Disconnect.ind and no Disconnect on the bus.

2.4.7 Telegram Sequence 38: Reception of T_NAK_PDU and maximum number of repetitions is not reached - Procedure with initial state 'OPEN_WAIT' (without PEI, all styles)

Purpose: Check whether the BDUT passes the following states.

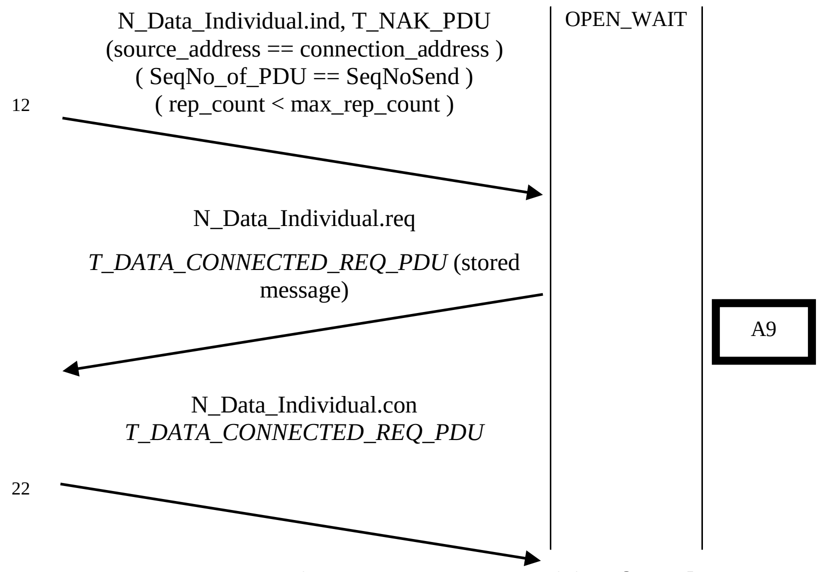

N_Data_Individual.ind, T_NAK_PDU
(source_address == connection_address)
(SeqNo_of_PDU == SeqNoSend)
(rep_count < max_rep_count)
12
OPEN_WAIT
N_Data_Individual.req
T_DATA_CONNECTED_REQ_PDU (stored message)
A9
N_Data_Individual.con
T_DATA_CONNECTED_REQ_PDU
22

Figure 28: State transition for telegram sequence 38

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. Now the BDUT gets a T_DATA_CONNECTED.req.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t5. The BDUT sends a telegram with NSDU=T_DATA_CONNECTED.req.
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()
-----------------------------------------------------------------------
@[tThe BDUT is in State 'OPEN_WAIT'.

@[t6. Now the BDUT receives a T_NACK.
(6)          IN   00:00:00.2  B0 A001 A000 60 C3  :T-Nack(Seq=0)
@[t7. The BDUT sends a telegram with NSDU=T_DATA_CONNECTED.req (stored
message).
(7)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()

@[tThe BDUT persists in State 'OPEN_WAIT'.
-----------------------------------------------------------------------
```

```
Alternatively for Style 1 rationalised
@[tThe DUT is in State 'OPEN_WAIT'.
@[t4. Now the DUT receives a T_NACK with the correct sequence number.
(6)          IN   00:00:00.2  B0 A001 A000 60 C3  :T-Nack(Seq=0)

@[t5. Check that the DUT sends a T_DISCONNECT.ind
(7)          OUT  00:00:00.0  87 00 00 00 00 00  :T_DISCONNECT.ind
@[t6. Check that the DUT sends a telegram with NSDU=T_DISCONNECT.req.
(8)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect
@[tThe DUT goes to 'CLOSED'.
```

2.4.8 Telegram Sequence 39: Reception of T_NAK_PDU and maximum number of repetitions is reached - Procedure with initial state 'OPEN_WAIT' (without PEI, all styles – optional style 1 rationalised)

Purpose: Check whether the BDUT passes the following states.

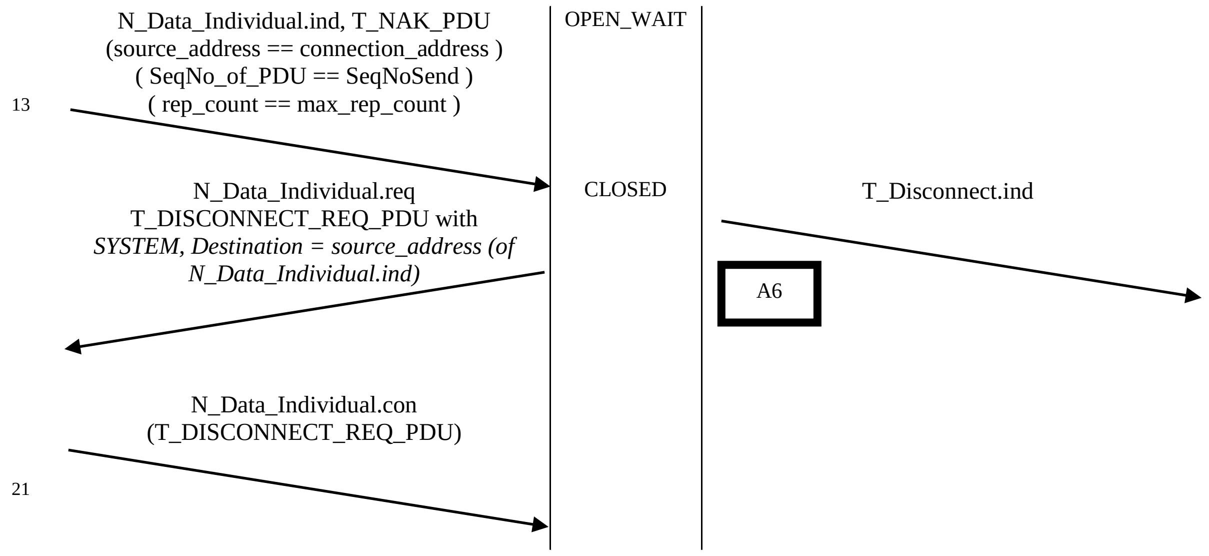

N_Data_Individual.ind, T_NAK_PDU
(source_address == connection_address)
(SeqNo_of_PDU == SeqNoSend)
(rep_count == max_rep_count)
13
OPEN_WAIT
CLOSED
T_Disconnect.ind
N_Data_Individual.req
T_DISCONNECT_REQ_PDU with
SYSTEM, Destination = source_address (of N_Data_Individual.ind)
A6
N_Data_Individual.con
(T_DISCONNECT_REQ_PDU)
21

Figure 29: State transition for telegram sequence 39

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Send T_CONNECT to BDUT.
(2)          IN   00:00:00.2  B0 A001 A000 60 80  :T-Connect(Addr=10.00.000)
@[t3. Check that the BDUT sends a T_CONNECT_ind to remote BCU
(3)          OUT  00:00:00.0  85 B0 A0 01 A0 00  :T_CONNECT.ind

@[tThe BDUT is in State 'OPEN_IDLE'.
```

```
@[t4. Now the BDUT gets a T_DATA_CONNECTED.req.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req
@[t5. The BDUT sends a telegram with NSDU=T_DATA_CONNECTED.req.
(5)          OUT  00:00:00.0  B0 A000 A001 61 43  00  :MaskVersionRead()
@[tThe BDUT is still in State 'OPEN_WAIT'.
(6)          OUT  00:00:03.0  B0 A000 A001 61 43  00  :MaskVersionRead()
(7)          OUT  00:00:03.0  B0 A000 A001 61 43  00  :MaskVersionRead()
(8)          OUT  00:00:03.0  B0 A000 A001 61 43  00  :MaskVersionRead()
@[t6. Now the BDUT receives a T_NACK.
(9)          IN   00:00:10.0  B0 A001 A000 60 C3  :T-Nack(Seq=0)
@[t7. Check that the BDUT sends a T_DISCONNECT.ind
(10)          OUT  00:00:00.0  87 00 A0 01 A0 00  :T_DISCONNECT.ind
@[t8. Send telegram addressed to remote BCU with NSDU=T_DISCONNECT.req
(11)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT is in State 'CLOSED'.
```

2.4.9 Reception of T_NAK_PDU with wrong connection address

2.4.9.1 Telegram sequence 40: Procedure with initial state ‘OPEN_IDLE’ (without PEI, style 1)

Purpose: Check whether the BDUT passes the following states.

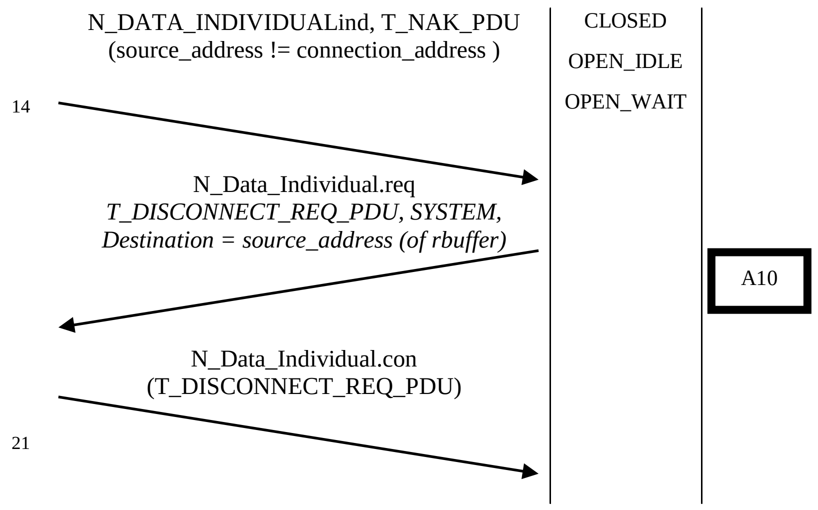

N_DATA_INDIVIDUALind, T_NAK_PDU
(source_address != connection_address)
CLOSED
OPEN_IDLE
OPEN_WAIT
14
N_Data_Individual.req
T_DISCONNECT_REQ_PDU, SYSTEM,
Destination = source_address (of rbuffer)
A10
N_Data_Individual.con
(T_DISCONNECT_REQ_PDU)
21

Figure 30: State transition for sequence 40

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Activate the
Link-Layer.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. BDUT receives a T_CONNECT.con
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con

@[tThe BDUT is in State 'OPEN_IDLE'.
```

```
@[t4. The BDUT receives a T_NACK from remote BCU (<> connected BCU = third
BCU)
(4)          IN   00:00:00.2  B0 A001 A000 60 C3  :T-Nack(Seq=0)
@[t5. Check that a telegram is sent addressed to remote BCU with
NSDU=T_DISCONNECT.req
(5)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'OPEN_IDLE'.
```

Note: Style 2/3 - BDUT sends no Disconnect on the bus.

2.4.9.2 Telegram sequence 41: Procedure with initial state ‘OPEN_WAIT’ (without PEI, style 1)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.
@@PREPARATION: Install a third BCU with physical address A007H. Activate the
Link-Layer.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[t2. Connect BDUT with third BCU
@[t3. BDUT receives a T_CONNECT.con
(2)          IN   00:00:00.2  43 00 00 00 A0 07  :T_CONNECT.req
(3)          OUT  00:00:00.0  86 B0 A0 07 A0 00  :T_CONNECT.con

@[tThe BDUT is in State 'OPEN_IDLE'.

@[t4. BDUT sends a T_DATA_CONNECTED.req to the third BCU.
(4)          IN   00:00:00.2  41 B0 00 00 00 00 01 03 00
:T_DATA_CONNECTED.req

@[tThe BDUT is in State 'OPEN_WAIT'.

@[t5. The BDUT receives a T_NACK from remote BCU (<> connected BCU = third
BCU)
(5)          IN   00:00:04.0  B0 A001 A000 60 C3  :T-Nack(Seq=0)
@[t6. Check that a telegram is sent addressed to remote BCU with
NSDU=T_DISCONNECT.req
(6)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'OPEN_WAIT'.
```

Note : Style 2/3 - BDUT sends no Disconnect on the bus.

2.5 Events started in state ‚CLOSED’

The following telegram sequence checks the events No. 4, No. 8 and No. 11 representative for event No. 4 till No.14 started with initial state ‚CLOSED’.

2.5.1 Telegram Sequence 42: Procedure with initial state ‘CLOSED’ (without PEI, style 1)

```
@@PREPARATION: Set physical address of BDUT to A000H. The remote BCU is set
to A001H.

@[t1. Set PEI of BDUT to Transport-Layer (remote)
(1)          IN        start  A9 00 12 34 48 88 0A  :PEI_SWITCH.req

@[tThe BDUT is in State 'CLOSED'.
```

```
@[t2. Now the BDUT receives a NSDU=T_DATA_CONNECTED_REQ (MaskVersionRead).
(2)          IN   00:00:00.2  B0 A001 A000 61 43  00  :MaskVersionRead()
@[t3. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT_REQ.
(3)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'CLOSED'.

@[t4. Now the BDUT receives a T_ACK.
(4)          IN   00:00:00.2  B0 A001 A000 60 C2  :T-Ack(Seq=0)
@[t5. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT_REQ.
(5)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'CLOSED'.

@[t6. Now the BDUT receives a T_NAK.
(6)          IN   00:00:00.2  B0 A001 A000 60 C3  :T-Nack(Seq=0)
@[t7. Check that the BDUT sends a telegram with NSDU=T_DISCONNECT_REQ.
(7)          OUT  00:00:00.0  B0 A000 A001 60 81  :T-Disconnect

@[tThe BDUT persists in State 'CLOSED'.
```

Note: Style 2/3 - BDUT sends no Disconnect on the bus.