



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.634

Addendum 1

(09/98)

SERIES X: DATA NETWORKS AND OPEN SYSTEM
COMMUNICATIONS

OSI networking and system aspects – Efficiency

Information technology – Open Systems
Interconnection – Transport Fast Byte Protocol

Addendum 1: SDL specifications

ITU-T Recommendation X.634 – Addendum 1

(Previously CCITT Recommendation)

ITU-T X-SERIES RECOMMENDATIONS
DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION X.634

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – TRANSPORT FAST BYTE PROTOCOL

ADDENDUM 1

SDL specifications

Summary

This Addendum to Recommendation X.634 contains an SDL specification of the Transport Fast Byte Protocol description. The Transport Fast Byte Protocol eliminates the roundtrip delay associated with the establishment and release of a transport connection, and requires very low PCI overhead. The Transport Fast Byte Protocol is intended for use in situations in which enhancements to the network QoS are not required, and efficiency of operation (e.g. reduction of roundtrip delays on establishment and release) is of primary concern.

Source

Addendum to ITU-T Recommendation X.634, was prepared by ITU-T Study Group 7 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 25th of september 1998.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation the term *recognized operating agency (ROA)* includes any individual, company, corporation or governmental organization that operates a public correspondence service. The terms *Administration*, *ROA* and *public correspondence* are defined in the *Constitution of the ITU (Geneva, 1992)*.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 1999

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	<i>Page</i>
C.1 The system and block structure of the Transport Fast Byte Protocol entity.....	1
C.2 Interaction with Management and Control Planes	1
C.3 Procedure of the Transport Fast Byte Protocol entity	1

**INFORMATION TECHNOLOGY – OPEN SYSTEMS
INTERCONNECTION – TRANSPORT FAST BYTE PROTOCOL**

ADDENDUM 1

SDL specifications

(Geneva, 1998)

Annex C

SDL specification of the Transport Fast Byte Protocol entity

C.1 The system and block structure of the Transport Fast Byte Protocol entity

The SDL system diagram of the Transport Fast Byte Protocol entity is shown in Figure C.1 and the SDL block structure in Figure C.2.

C.2 Interaction with Management and Control Planes

C.2.1 Management Plane

No interactions with the Management Plane are specified.

C.2.2 Control Plane

No interactions with the Control Plane are specified.

C.3 Procedure of the Transport Fast Byte Protocol entity

The SDL diagrams of the procedure of the Transport Fast Byte Protocol entity are given in this subclause. If there exists any difference between the prose description also given in this subclause and the SDL diagrams, the SDL diagrams take precedence. On the other hand, if there exists differences between the specification in this Annex and the one in clause 6/X.634, the specification in clause 6/X.634 take precedence.

NOTE – In the SDL diagrams of this subclause, the octets in all PDUs and SDUs, i.e. the TSDU, are numbered from "1" to at most "65535".

The operation of the Transport Fast Byte Protocol entity is modeled as a state machine consisting of the following states:

- **Idle:** Each Transport Fast Byte Protocol entity is conceptually initiated in the Idle state and returns to this state upon the release of a connection.
- **Outgoing Connection Pending:** A Transport Fast Byte Protocol entity requesting a connection with its peer is in the Outgoing Connection Pending state until it receives acknowledgment from its peer.
- **Incoming Connection Pending:** A Transport Fast Byte Protocol entity that has received a connection request from its peer and is waiting for its user's response is in the Incoming Connection Pending state.
- **Data Transfer Ready:** Upon successful completion of the connection establishment, both peer Transport Fast Byte Protocol entities will be in Data Transfer Ready state and data transfer can take place. No TSDU is currently reassembled.
- **Data Transfer RAS:** Some but not all information for a TSDU currently being reassembled has arrived and is being buffered.
- **Data Transfer Abort:** The maximum permissible length of a TSDU has been exceeded during reassembly. The Transport Fast Byte Protocol entity remains in this state until an end of a TSDU is received (EOT-bit).

The state transition diagram for the Transport Fast Byte Protocol entity is shown in Figure C.3.

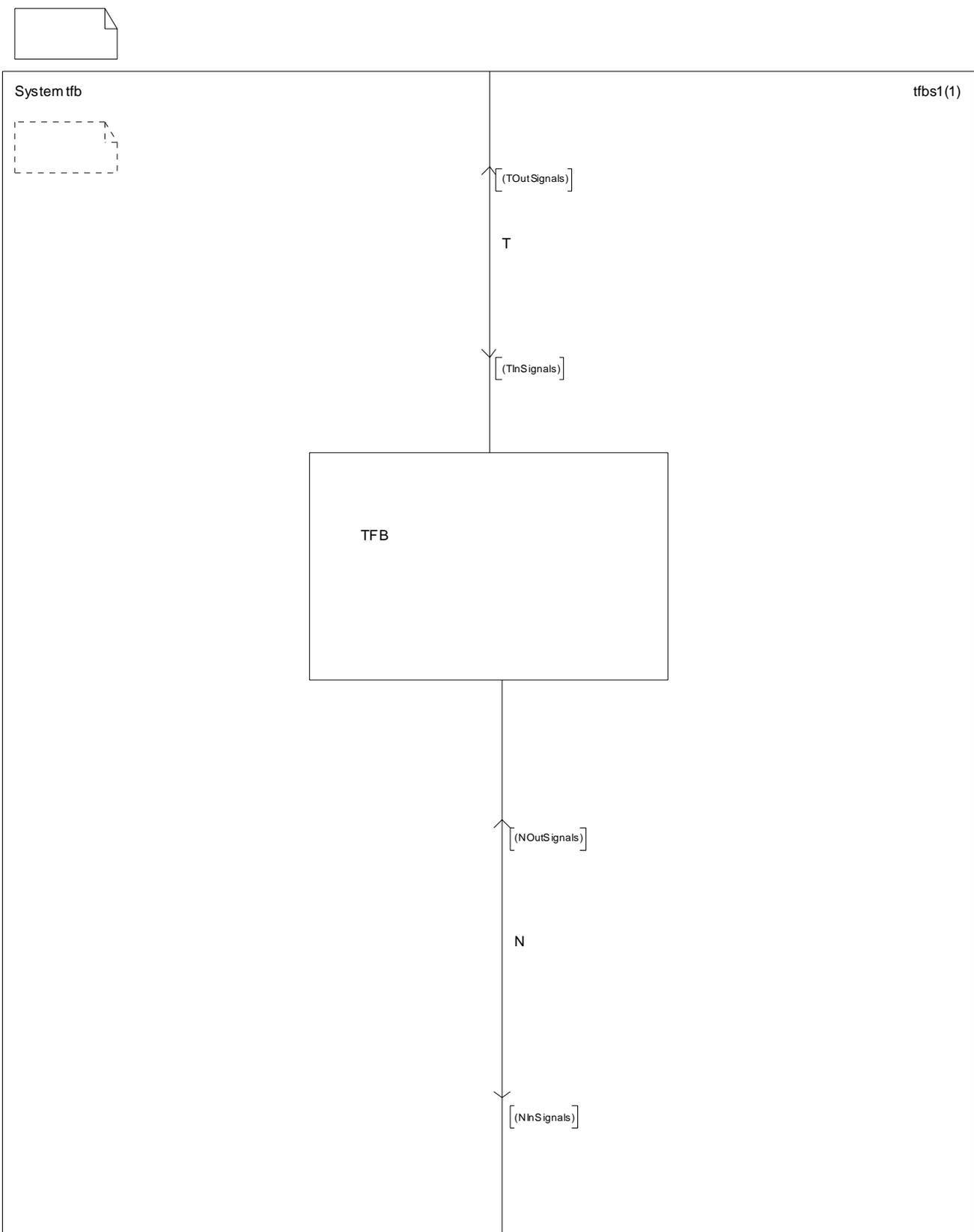


Figure C.1/X.634 – Transport Fast Byte system

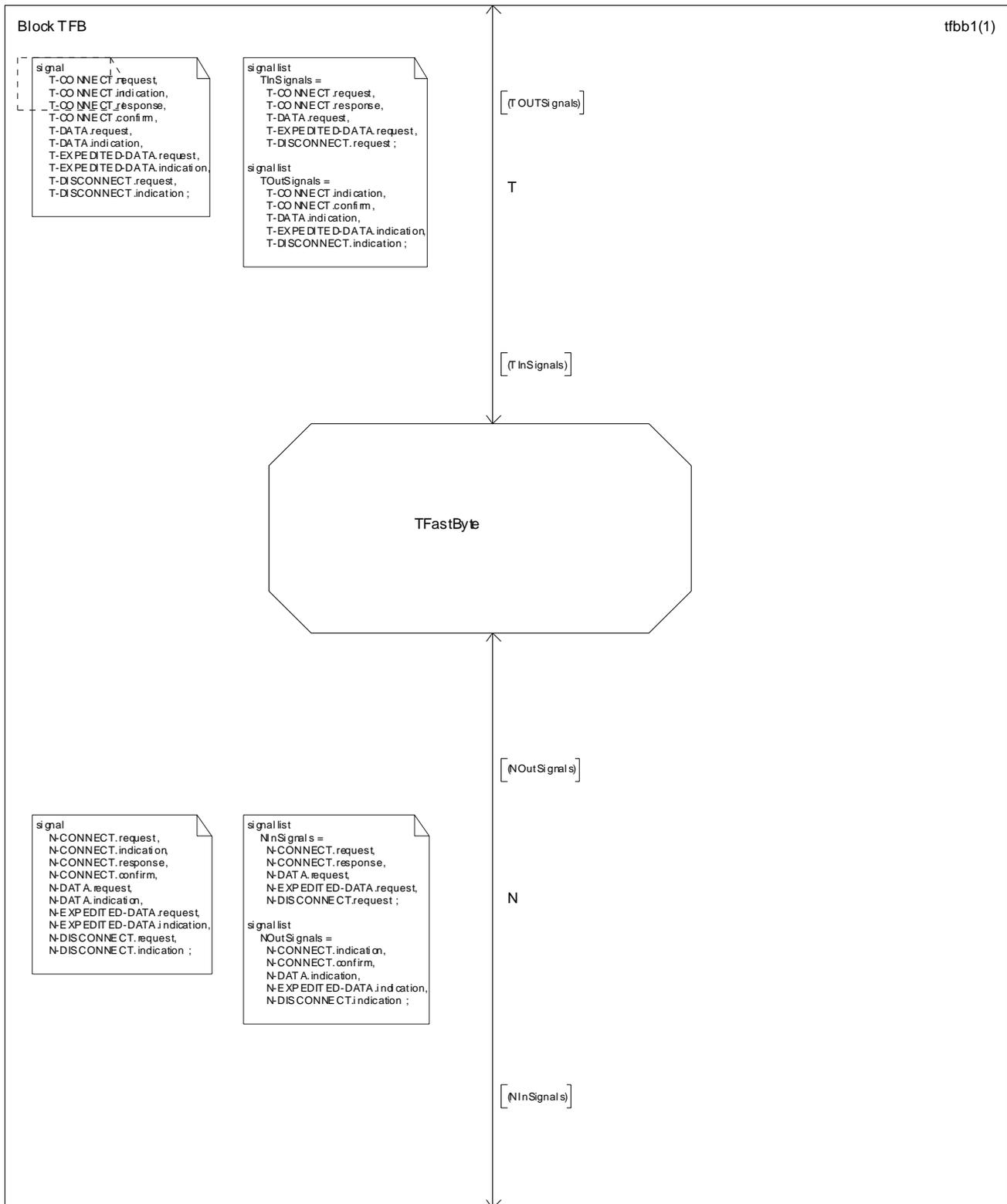
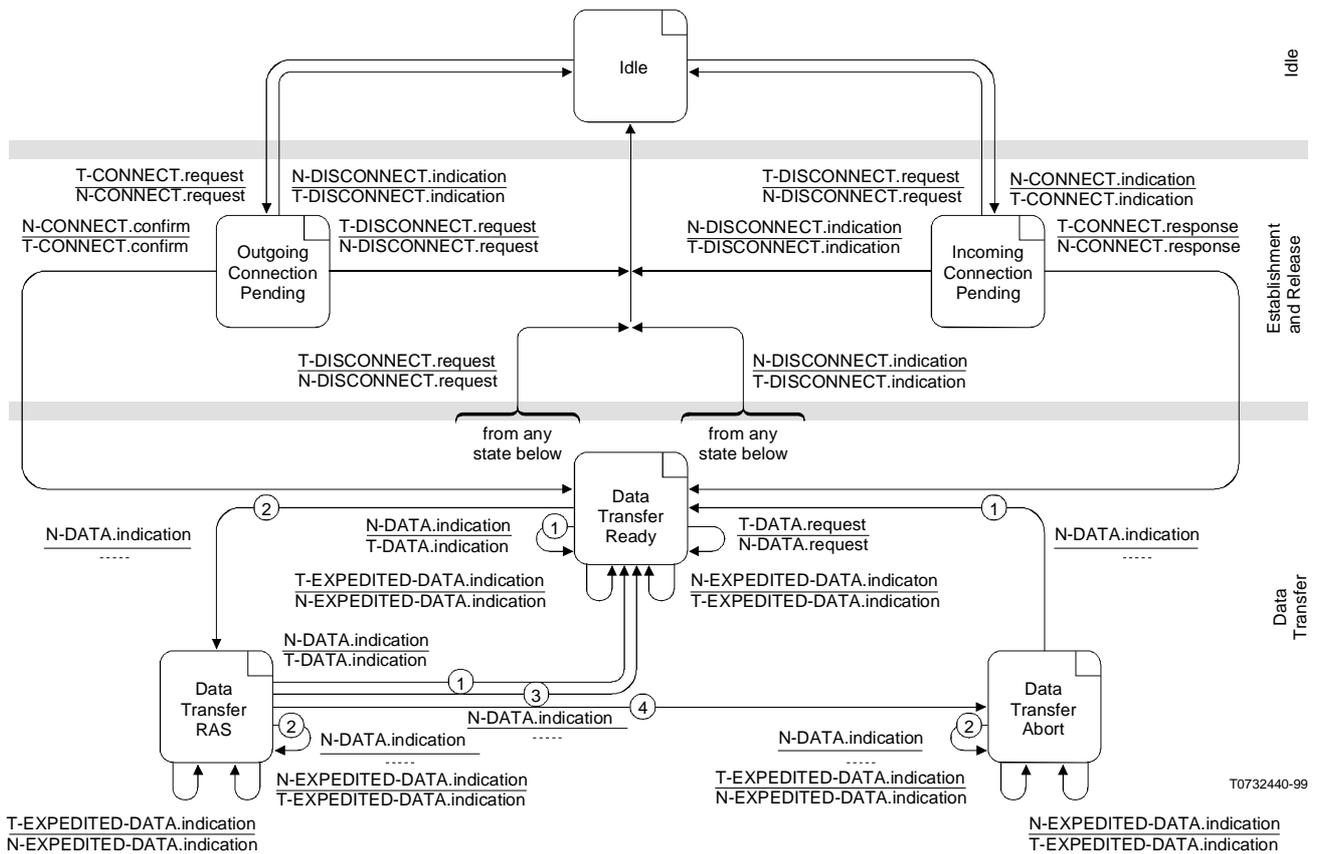


Figure C.2/X.634 – Transport Fast Byte block structure



T0732440-99

- 1) N-DATA.indication received and end of the TSDU detected.
- 2) N-DATA.indication received and no end of the TSDU detected.
- 3) ConnRcvTPDULength exceeded and end of the TSDU detected.
- 4) ConnRcvTPDULength exceeded and no end of the TSDU detected.

Figure C.3/X.634 – State transition diagram for the Transport Fast Byte Protocol entity

The description of the operations of the Transport Fast Byte Protocol entity make use of the following state variables:

CdAddr	A character string holding the "Called Address" as extracted out of a TPDU.
CgAddr	A character string holding the "Calling Address" as extracted out of a TPDU.
RsAddr	A character string holding the "Responding Address" as extracted out of a TPDU.
TPDU	An octet string holding the TPDU being constructed in the outgoing direction or the TPDU just having been received.
TSDU	An octet string holding the TSDU being reassembled in the incoming direction.
ptrRAS	An index into the variable TSDU indicating where to place the next information during reassembly.
ptrSEG	An index into the parameter TSUserData indicating where to extract data during segmentation.
ptrPDU	An index into the variable TPDU having been received indicating where to retrieve the next information during interpretation of the received TPDU.
lenPDU	The length of a TPDU having been received.
lenSDU	The (remaining) length of a TSDU being segmented.
len	Temporary variable holding the length of a character string.
EOTbit	A boolean variable being set to TRUE if the EOT-bit in the received TPDU has been set.
SndTPDULength	A temporary integer variable during connection establishment used for negotiation of the maximum size of a TPDU in the outgoing direction.
ConnSndTPDULength	An integer variable indicating the maximum size of a TPDU in the outgoing direction.

RcvTPDULength	A temporary integer variable during connection establishment used for negotiation of the maximum size of a TPDU in the incoming direction.
ConnRcvTPDULength	An integer variable indicating the maximum size of a TPDU in the incoming direction.
tmpNullPCI	A temporary boolean variable during connection establishment used for negotiation of the NullPCI capability.
ConnSndNullPCI	A boolean variable indicating the NullPCI capability in the outgoing direction.
ConnRcvNullPCI	A boolean variable indicating the NullPCI capability in the incoming direction.
Terminate	A boolean variable being set to TRUE after leaving state "Idle" if the process should stop after return to "Idle".

The Transport Fast Byte Protocol entity maintains the following parameters:

NmaxRcvLength	The maximum size of the NSUserData parameter in the incoming direction.
NmaxSndLength	The maximum size of the TSUserData parameter in the outgoing direction.
TmaxRcvLength	The maximum size of the TSUserData parameter in the incoming direction.
TmaxSndLength	The maximum size of the TSUserData parameter in the outgoing direction.
CPlaneNegotiation	A boolean value being set to TRUE if the parameter negotiation takes place outside the Transport Fast Byte Protocol entity, e.g. in the C-plane.
NullPCI	A boolean value being set to TRUE if NullPCI procedures are enabled.
Termination	A boolean value being set to TRUE if if the Transport Fast Byte Protocol entity should stop after return to the state "Idle".

The SDL definition of the Transport Fast Byte Protocol entity process is shown in Figure C.4.

C.3.1 Procedures in the state "Idle"

- 1) Upon receipt of a T-CONNECT.request, a TPDU is constructed that is then transferred in the NS-User-Data of the N-CONNECT.request; the process enters state "Outgoing Connection Pending".

If negotiation takes place outside the Transport Fast Byte Protocol entity, no TPDU is transmitted. Otherwise, the TPDU constructed follows the specification in clause 7/X.634. The NullPCI capability is enabled as indicated in the parameter "NullPCI", the TPDU lengths are taken from the parameter "TmaxSndLength" and "TmaxRcvLength", and the called and calling addresses were received as parameters of the T-CONNECT.request. If TS-User-Data was also received with the primitive, this is also copied into the TPDU.

NOTE 1 – The variable "Terminate" is set to TRUE after leaving state "Idle" if the parameter "Termination" has been set to TRUE also.

- 2) If the TPDU constructed above exceeds the maximum permissible length of the NS-User-Data, a T-DISCONNECT.indication is returned to the Transport Fast Byte Protocol entity user; the process remains in state "Idle".
- 3) Upon receipt of an N-CONNECT.indication, the TPDU received in the NS-User-Data is verified and the data extracted (if negotiation takes place outside the Transport Fast Byte Protocol entity, no TPDU is verified and no data is extracted). The data items are then sent in the parameters of the T-CONNECT.indication to the Transport Fast Byte Protocol entity user; the process enters state "Incoming Connection Pending".

The verification and extraction is specified in macro "Verify ICP TPDU" and is as follows:

- a) If no TPDU has been received or the first octet does not contain the value "10100010₂" or the extension bit in octet 2 is set, the connection establishment is rejected with an N-DISCONNECT.request.
- b) If octet 3 contains the value "00000001₂", the next 4 octets contain negotiation values for the maximum TPDU length; otherwise, the negotiation starts with the default maximum TPDU length values (512 octets).
- c) If the next octet contains the value "00000010₂", the next octets contain the called and calling address (these are temporarily stored in variables "CdAddr" and "CgAddr"); otherwise, both variables are set to the null character string.
- d) If the next octet contains the value "00000100₂", the information in the remaining octets is copied into the variable "TSDU" in preparation of sending it with the TS-User-Data in the T-CONNECT.indication primitive.

If during the data extraction an improper TPDU is detected (e.g. a truncated TPDU), the connection establishment is rejected with an N-DISCONNECT.request and a T-DISCONNECT.request.

NOTE 2 – The variable "Terminate" is set to TRUE after leaving state "Idle" if the parameter "Termination" has been set to TRUE also.

- 4) If upon return to state "Idle" the variable "Terminate" is TRUE, the process stops.

```
fpar
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
NullPCI BOOLEAN;
Termination BOOLEAN;
```

```
syntype
OCTETSTRING = CHARSTRING;
endsyntype

dcl
CdAddr CHARSTRING;
CgAddr CHARSTRING;
RsAddr CHARSTRING;
TPDU OCTETSTRING;
TSDU OCTETSTRING;
ptrRAS INTEGER;
ptrSEG INTEGER;
ptrPDU INTEGER;
lenPDU INTEGER;
lenSDU INTEGER;
len INTEGER;
EOTbit BOOLEAN;
SndTPDULength INTEGER;
ConnSndTPDULength INTEGER;
RcvTPDULength INTEGER;
ConnRcvTPDULength INTEGER;
tmpNullPCI BOOLEAN;
ConnSndNullPCI BOOLEAN;
ConnRcvNullPCI BOOLEAN;
Terminate BOOLEAN;
```

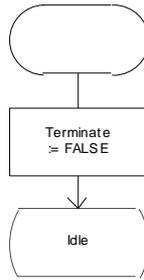


Figure C.4/X.634 (Sheet 1/15) – Transport Fast Byte process

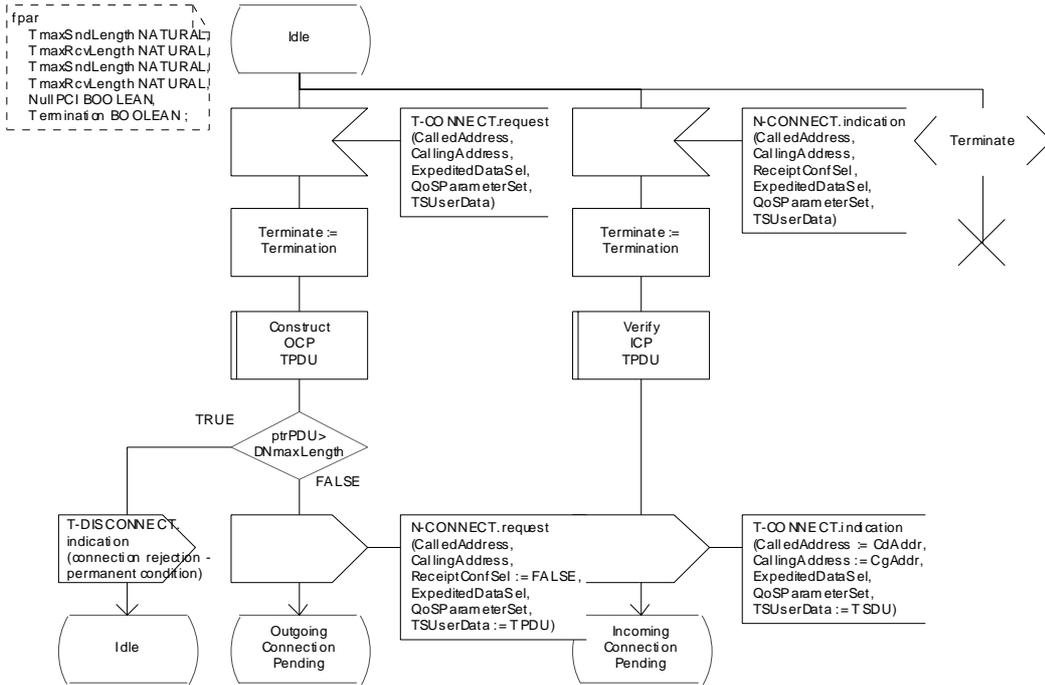


Figure C.4/X.634 (Sheet 2/15) – Transport Fast Byte process

```

fpar
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
NullPCI BOOLEAN;
Termination BOOLEAN;
    
```

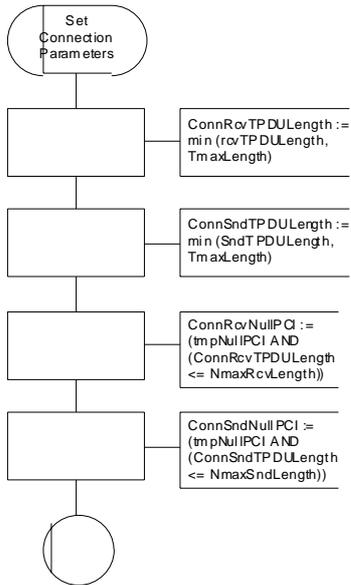
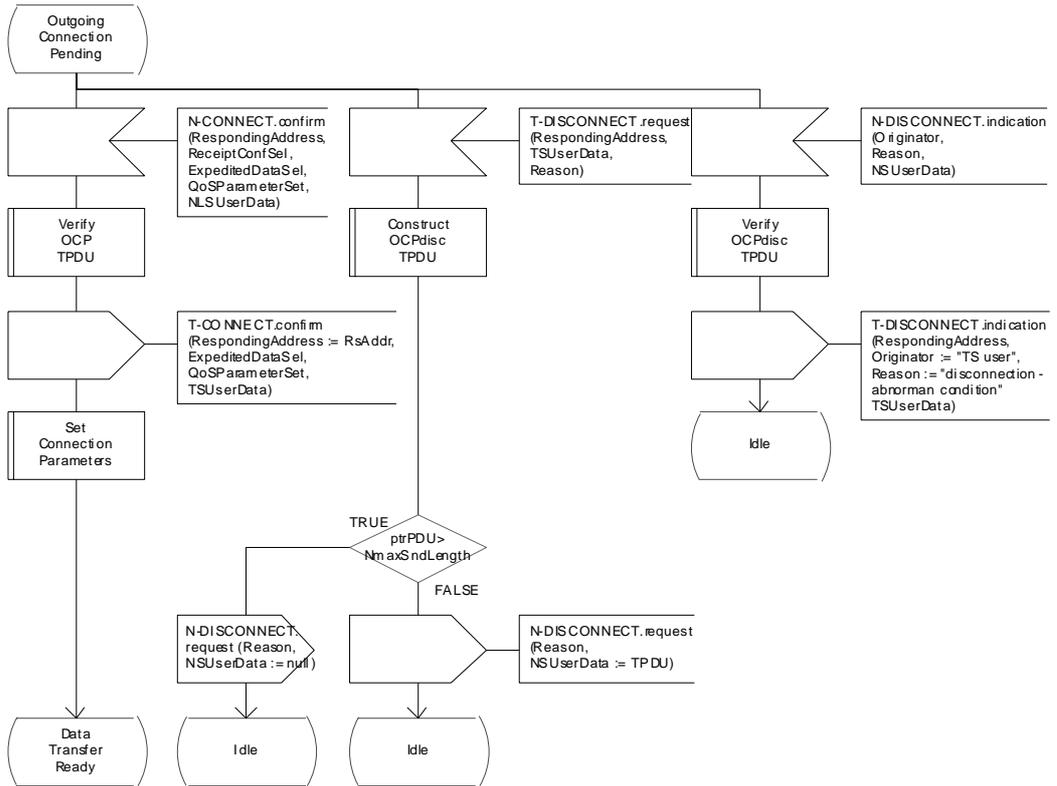


Figure C.4/X.634 (Sheet 3/15) – Transport Fast Byte process

```

    fpar
    T maxSndLength NATURAL;
    T maxRcvLength NATURAL;
    T maxSndLength NATURAL;
    T maxRcvLength NATURAL;
    NullPCI BOOLEAN;
    Termination BOOLEAN;
  
```

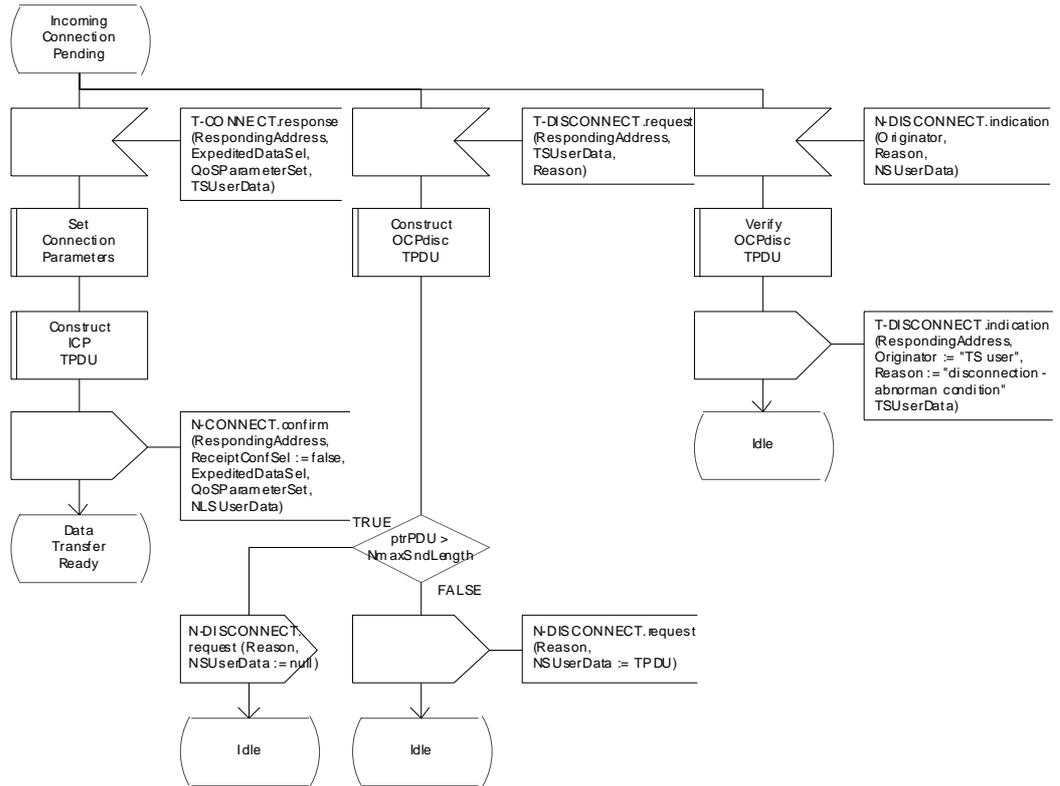


Figure C.4/X.634 (Sheet 4/15) – Transport Fast Byte process

```

fpar
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
NullPCI BOOLEAN;
Termination BOOLEAN;
    
```

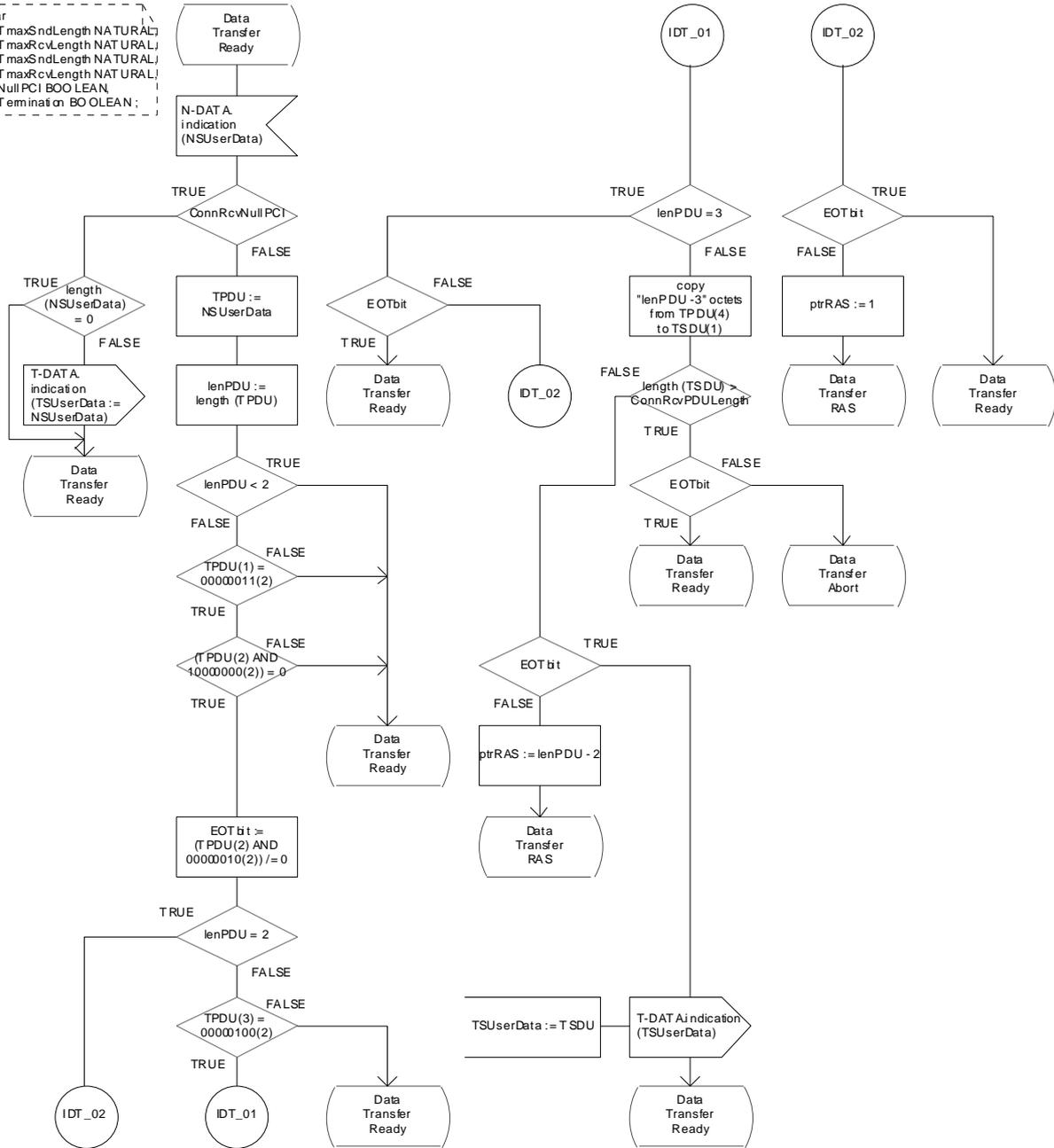


Figure C.4/X.644 (Sheet 5/15) – Transport Fast Byte process

```

par
  T maxSndLength NATURAL;
  T maxRcvLength NATURAL;
  T maxSndLength NATURAL;
  T maxRcvLength NATURAL;
  NullPCI BOOLEAN;
  Termination BOOLEAN;
endpar
  
```

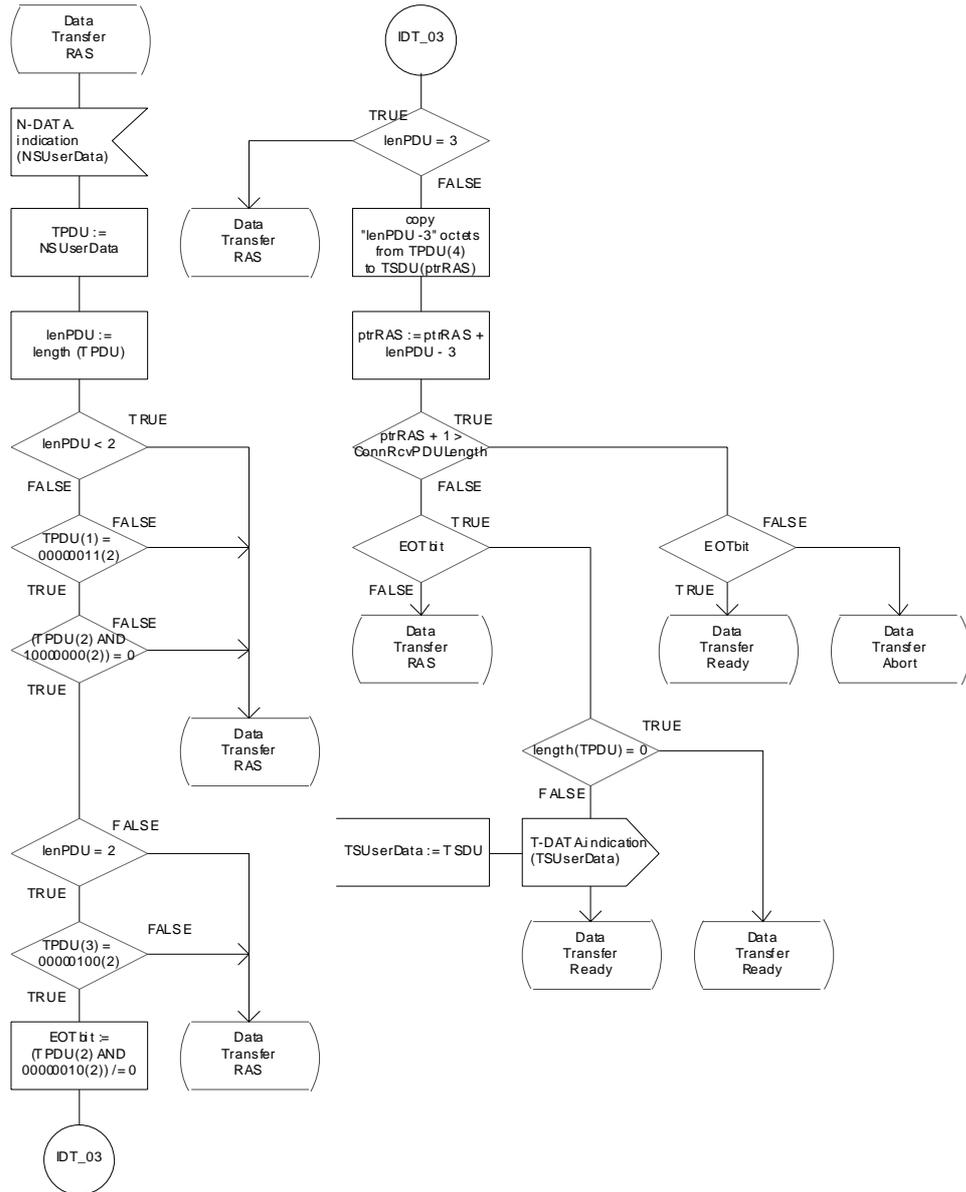


Figure C.4/X.634 (Sheet 6/15) – Transport Fast Byte process

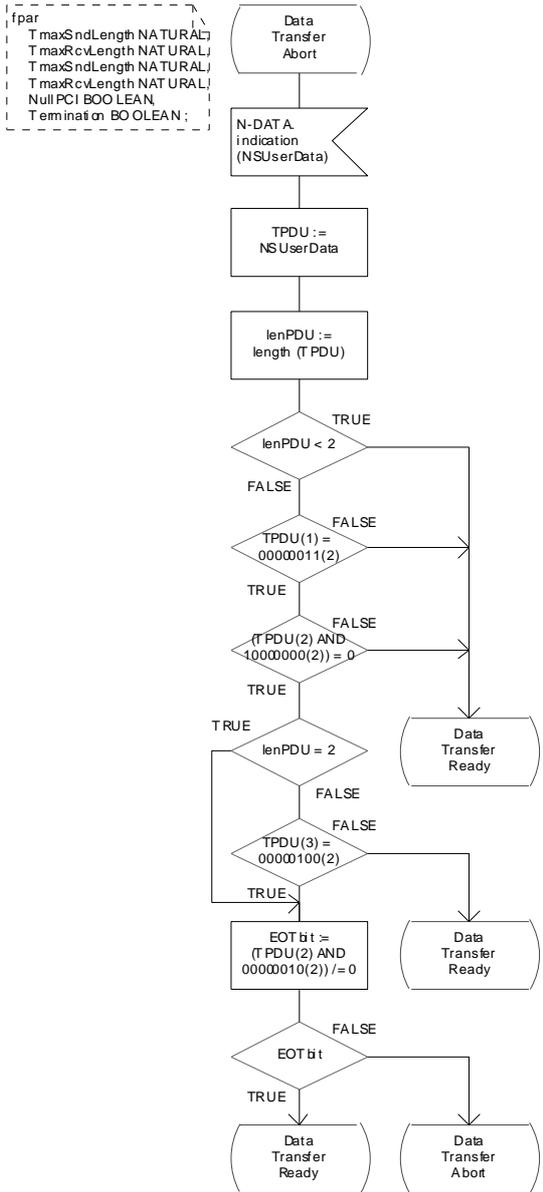


Figure C.4/X.644 (Sheet 7/15) – Transport Fast Byte process

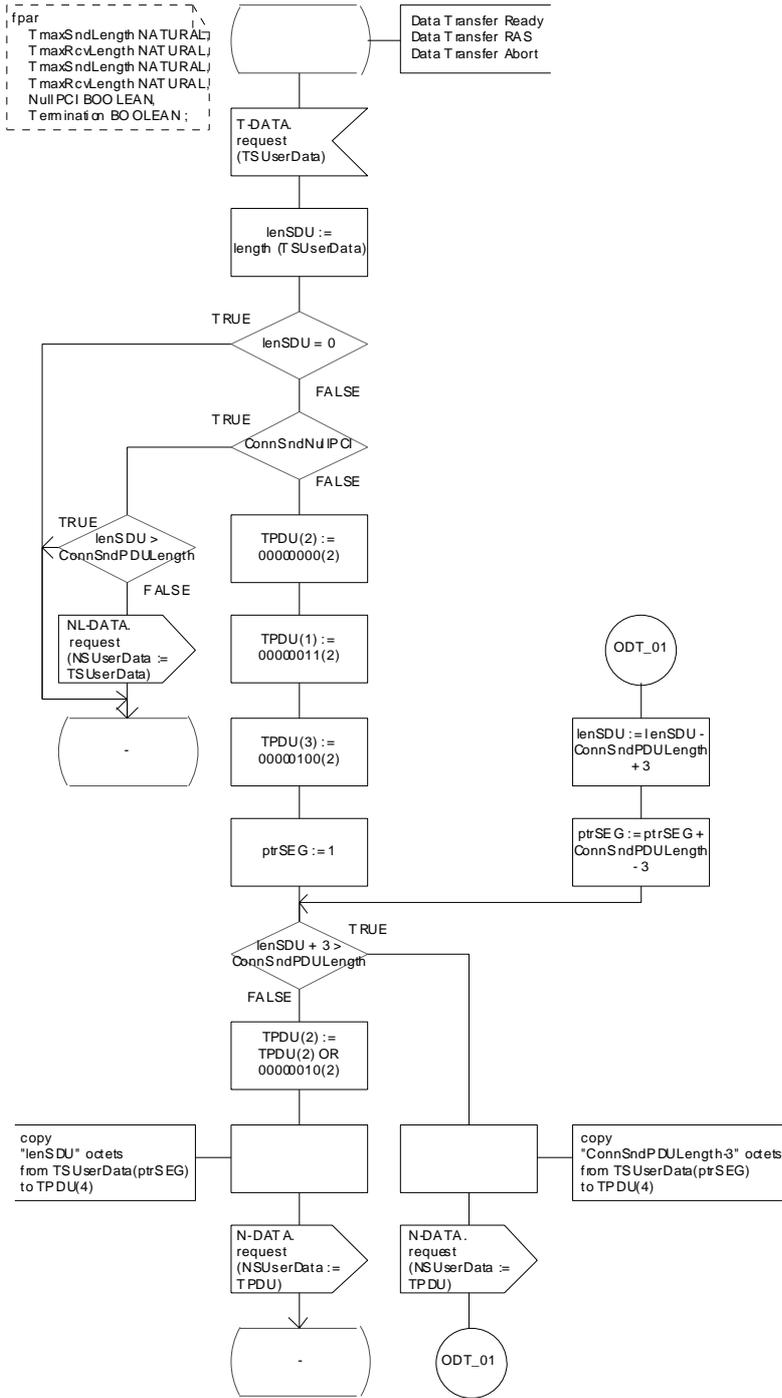


Figure C.4/X.634 (Sheet 8/15) – Transport Fast Byte process

```

fpar
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
NullPCI BOOLEAN;
Termination BOOLEAN;

```

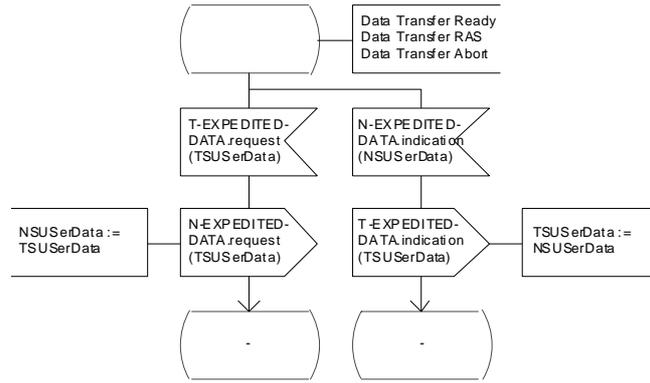


Figure C.4/X.634 (Sheet 9/15) – Transport Fast Byte process

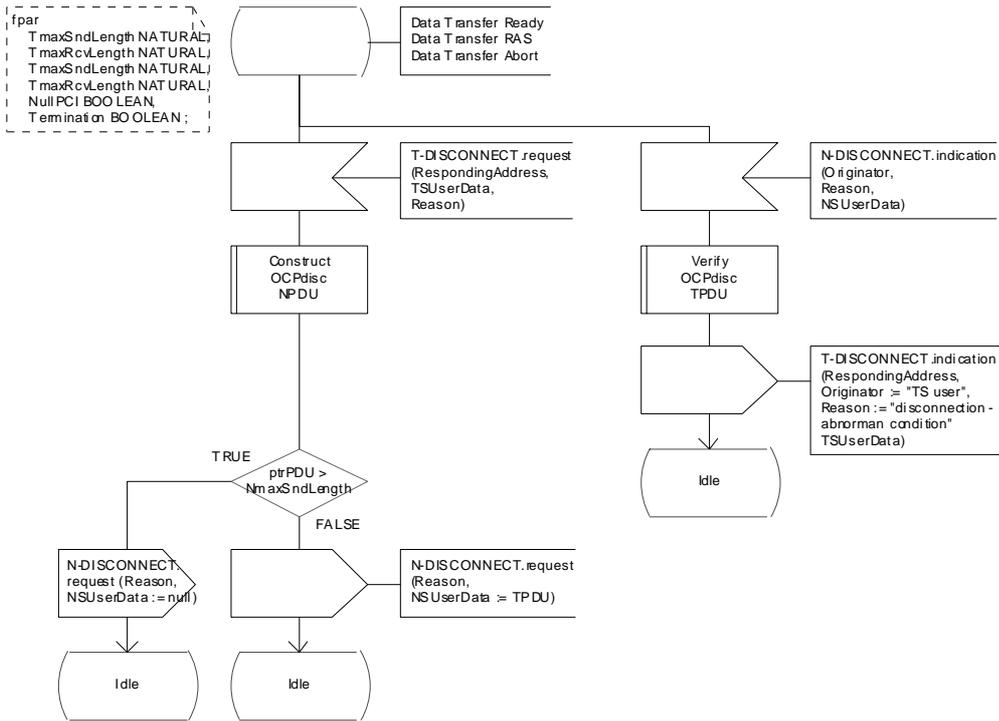


Figure C.4/X.634 (Sheet 10/15) – Transport Fast Byte process

```

fpar
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
TmaxSndLength NATURAL;
TmaxRcvLength NATURAL;
NullPCI BOOLEAN;
Termination BOOLEAN;
    
```

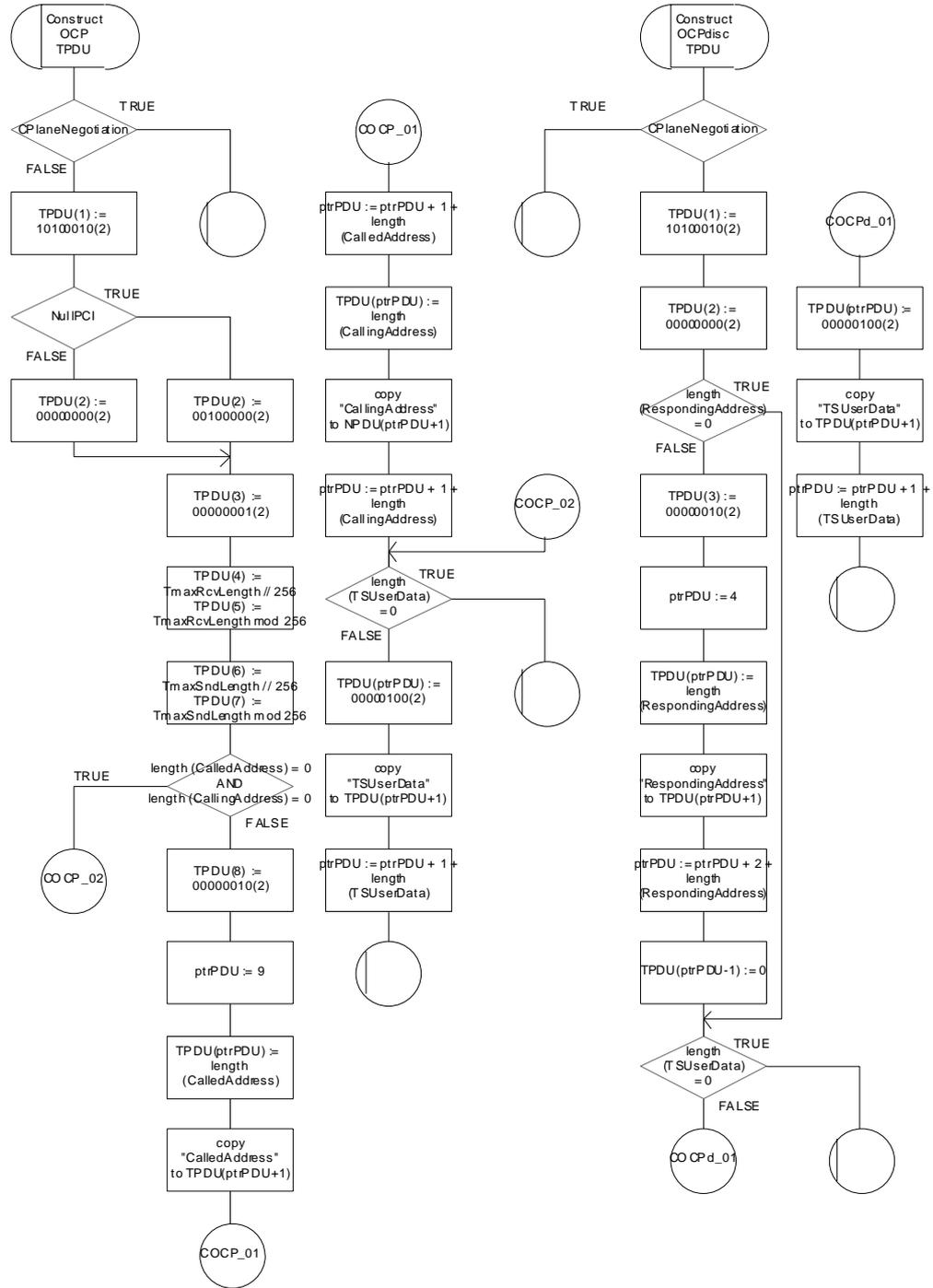


Figure C.4/X.634 (Sheet 11/15) – Transport Fast Byte process

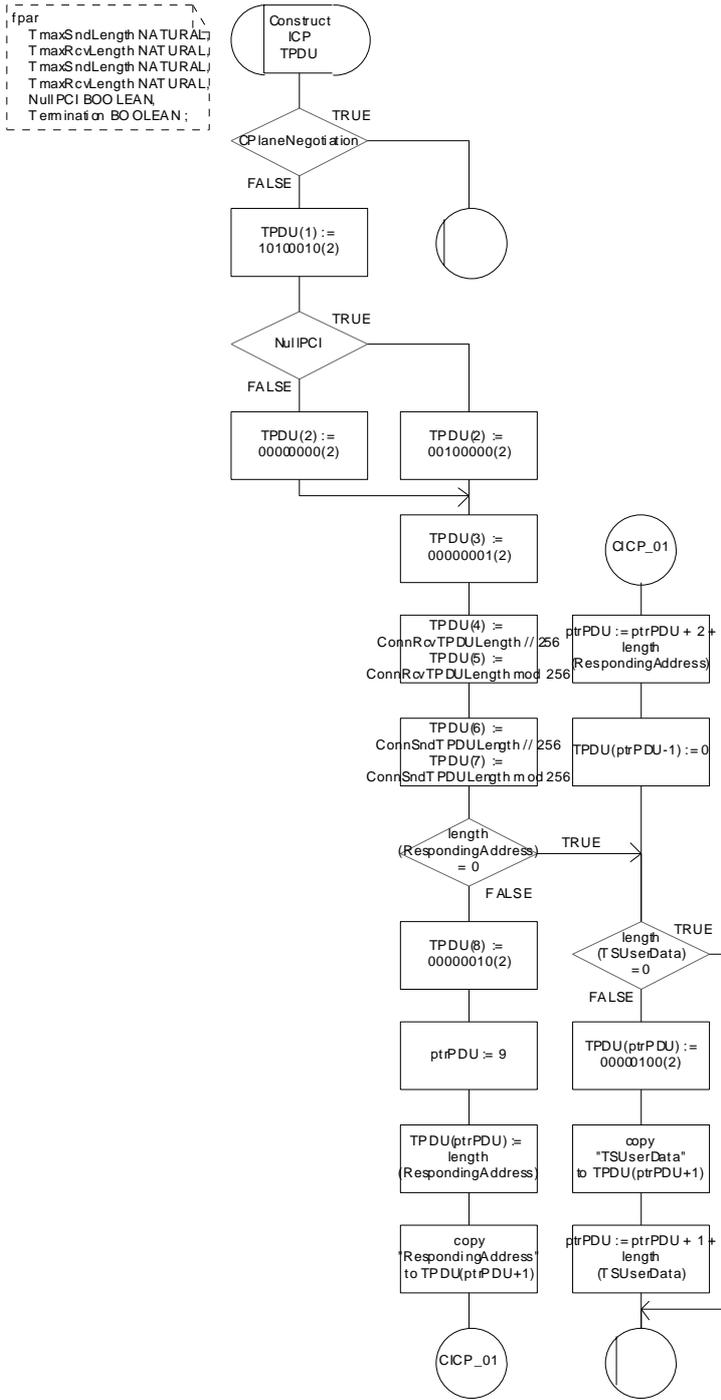


Figure C.4/X.634 (Sheet 12/15) – Transport Fast Byte process

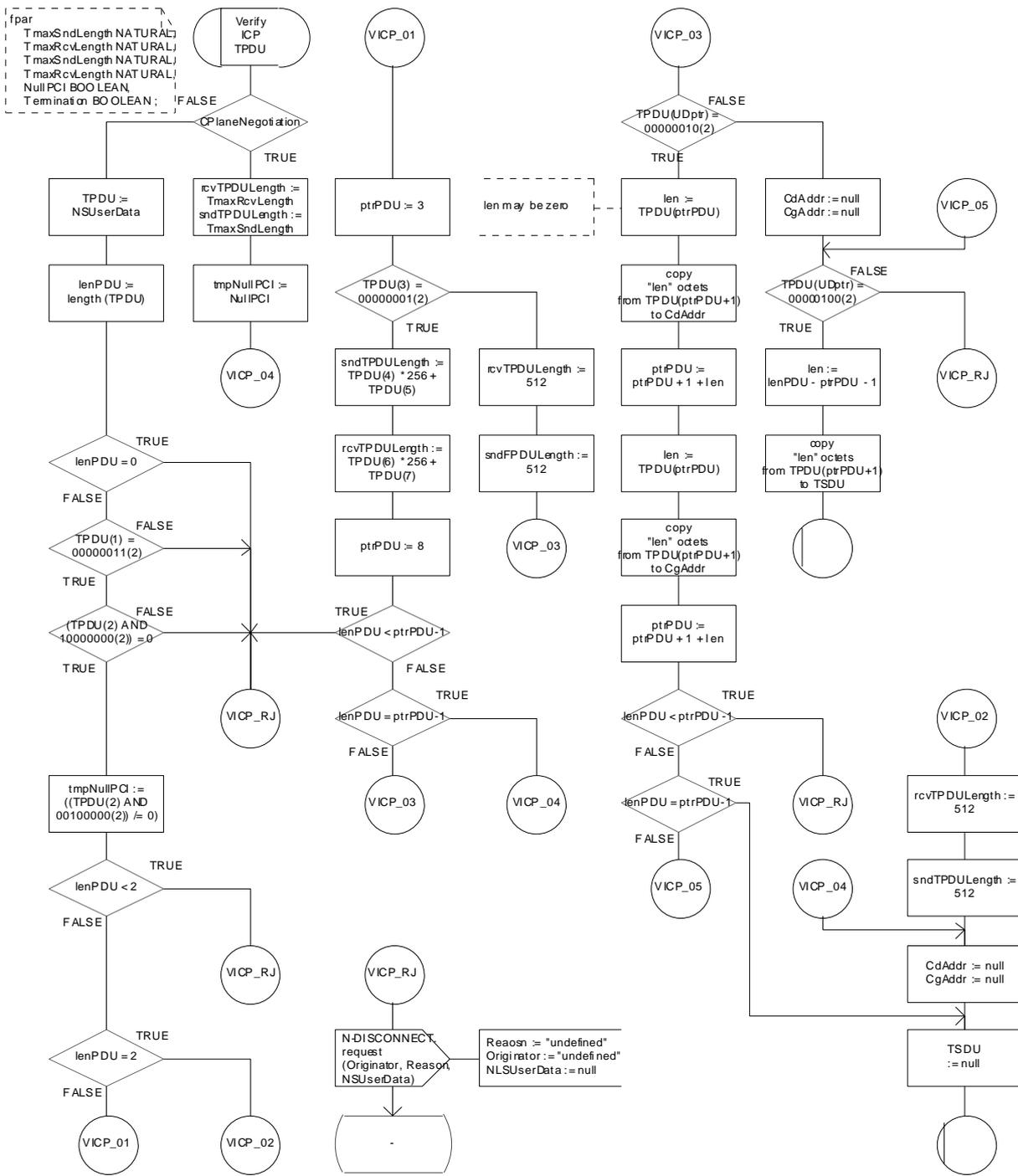


Figure C.4/X.634 (Sheet 13/15) – Transport Fast Byte process

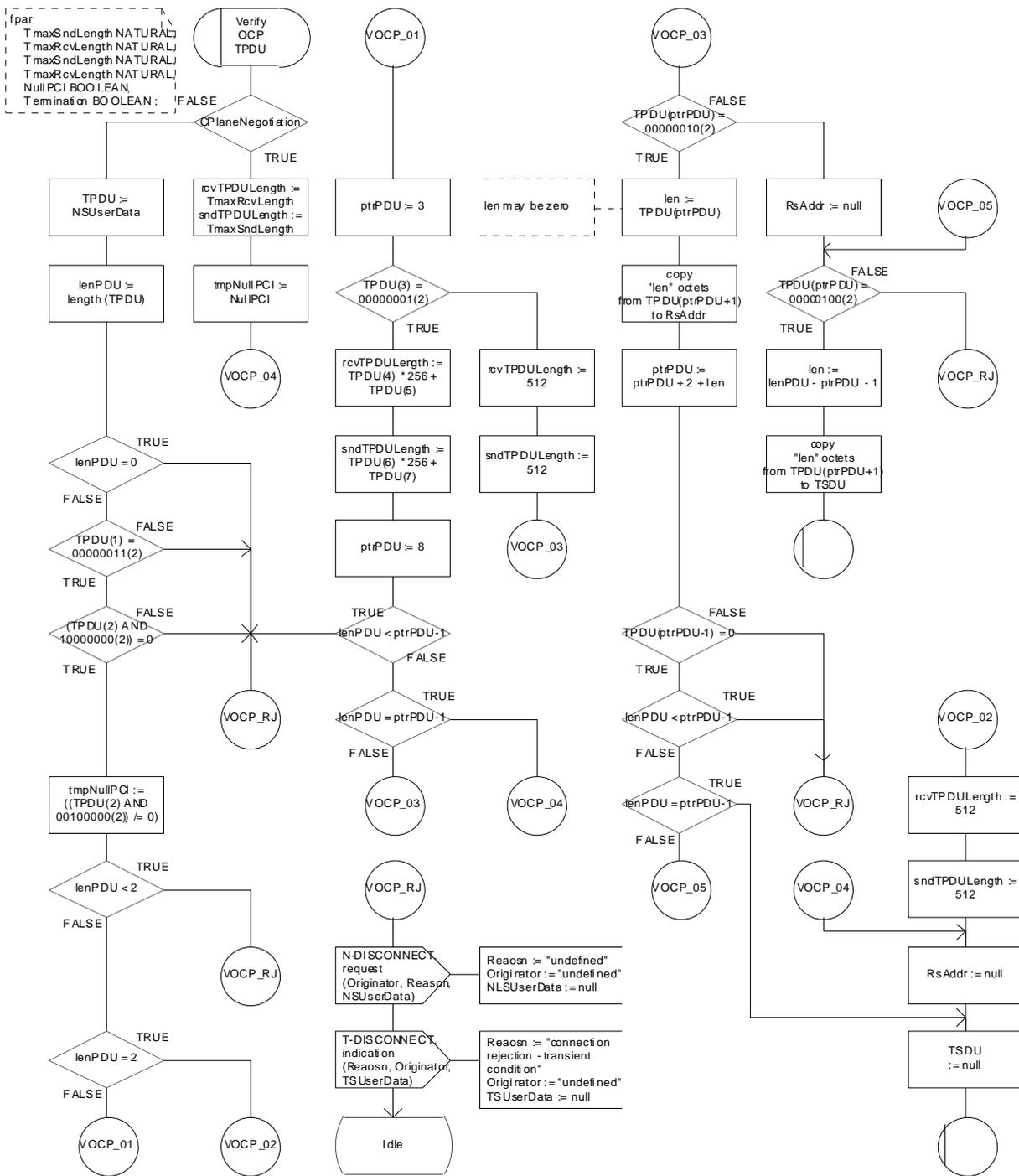


Figure C.4/X.634 (Sheet 14/15) – Transport Fast Byte process

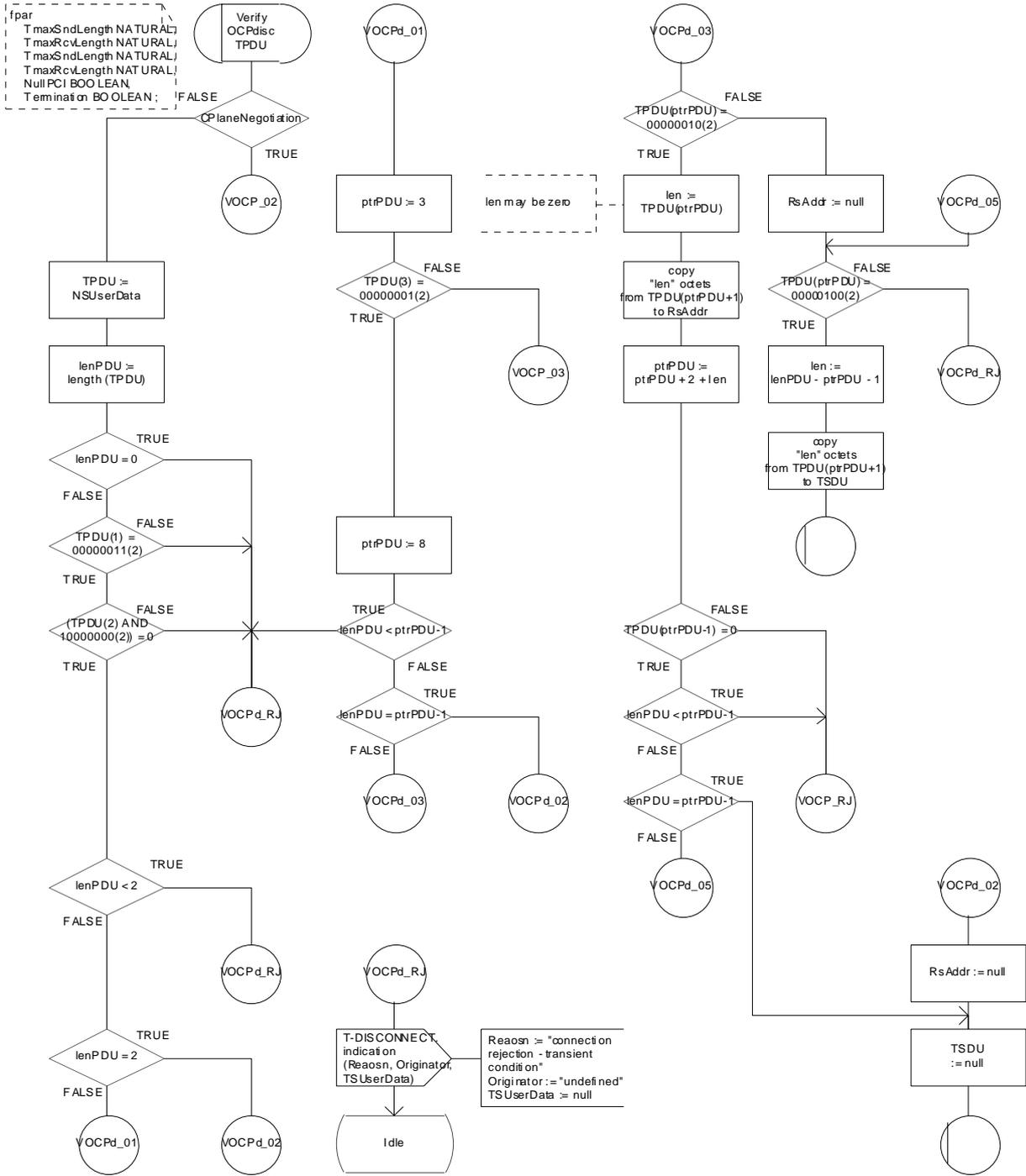


Figure C.4/X.634 (Sheet 15/15) – Transport Fast Byte process

C.3.2 Procedures in the state "Outgoing Connection Pending"

- 1) Upon receipt of an N-CONNECT.confirm, the TPDU received in the NS-User-Data is verified and the data extracted (if negotiation takes place outside the Transport Fast Byte Protocol entity, no TPDU is verified and no data is extracted). These data items are then sent in the parameters of the T-CONNECT.confirm to the Transport Fast Byte Protocol entity user. The connection parameters are set before the process enters state "Data Transfer Ready".

The verification and extraction is specified in macro "OCP TPDU" and is as follows:

- a) If no TPDU has been received or the first octet does not contain the value "10100010₂" or the extension bit in octet 2 is set, the connection establishment is rejected with an N-DISCONNECT.request and a T-DISCONNECT.request.
- b) If octet 3 contains the value "00000001₂", the next 4 octets contain negotiation values for the maximum TPDU length; otherwise, the negotiation starts with the default maximum TPDU length values (512 octets).
- c) If the next octet contains the value "00000010₂", the next octets contain the responding address (this is temporarily stored in variable "RsAddr"); otherwise, "RsAddr" is set to the null character string.
- d) If the next octet contains the value "00000100₂", the information in the remaining octets is copied into the variable "TSDU" in preparation of sending it with the TS-User-Data in the T-CONNECT.indication primitive.

If during the data extraction an improper TPDU is detected (e.g. a truncated TPDU), the connection establishment is rejected with an N-DISCONNECT.request and a T-DISCONNECT.request.

The following connection parameters are set:

- ConnSndTPDULength
- ConnRcvTPDULength
- ConnSndNullPCI
- ConnRcvNullPCI

- 2) If a T-DISCONNECT.request primitive is received, the Transport Fast Byte Protocol entity user abandons the connection establishment. A TPDU is constructed that is then transferred in the NS-User-Data of the N-DISCONNECT.request; the process returns to state "Idle".

If negotiation takes place outside the Transport Fast Byte Protocol entity, no TPDU is transmitted. Otherwise, the TPDU constructed follows the specification in clause 7/X.634. The second octet of the header is set to zero. If a responding address has been communicated in a parameter, the corresponding control part is constructed (no calling address is included). If TS-User-Data was also received with the primitive, this is also copied into the TPDU.

- 3) If the TPDU constructed above exceeds the maximum permissible length of the NS-User-Data, an N-DISCONNECT.request without NS-User-Data is sent to the Network layer; the process returns to state "Idle".
- 4) Upon receipt of an N-DISCONNECT.indication, connection establishment is rejected. The TPDU received in the NS-User-Data is verified and the data extracted (if negotiation takes place outside the Transport Fast Byte Protocol entity, no TPDU is verified and no data is extracted). These data items are then sent in the parameters of the T-DISCONNECT.indication to the Transport Fast Byte Protocol entity user; the process returns to state "Idle".

The verification and extraction is specified in macro "Verify OCPdisc TPDU" and is as follows:

- a) If no TPDU has been received or the first octet does not contain the value "10100010₂" or the extension bit in octet 2 is set, the connection establishment rejection is notified with a standard T-DISCONNECT.request.
- b) If octet 3 contains the value "00000001₂", the next 4 octets are skipped.
- c) If the next octet contains the value "00000010₂", the next octets contain the responding address (this is temporarily stored in variable "RsAddr"); otherwise, "RsAddr" is set to the null character string.
- d) If the next octet contains the value "00000100₂", the information in the remaining octets is copied into the variable "TSDU" in preparation of sending it with the TS-User-Data in the T-DISCONNECT.indication primitive.

If during the data extraction an improper TPDU is detected (e.g. a truncated TPDU), the connection establishment rejection is notified with a standard T-DISCONNECT.request.

C.3.3 Procedures in the state "Incoming Connection Pending"

- 1) Upon receipt of a T-CONNECT.response, the connection parameters are set and a TPDU is constructed that is then transferred in the NS-User-Data of the N-CONNECT. response; the process enters state "Data Transfer Ready".

The following connection parameters are set:

- ConnSndTPDULength
- ConnRcvTPDULength
- ConnSndNullPCI
- ConnRcvNullPCI

If negotiation takes place outside the Transport Fast Byte Protocol entity, no TPDU is transmitted. Otherwise, the TPDU constructed follows the specification in clause 7/X.634. The NullPCI capability is enabled as indicated in the parameter "NullPCI", the TPDU lengths are taken from the connection parameters above, and the responding address was received as parameters of the T-CONNECT.response. If TS-User-Data was also received with the primitive, this is also copied into the TPDU.

- 4) The procedures upon receipt of a T-DISCONNECT.request are described in C.3.2 items 2) and 3).
- 3) The procedures upon receipt of an N-DISCONNECT.indication are described in C.3.2 item 4).

C.3.4 Procedures in the state "Data Transfer Ready"

- 1) The Transport Fast Byte Protocol entity user submits data for transmission with the TS-User-Data parameter of a T-DATA.request. If no TS-User-Data is present, the primitive is ignored. If the NullPCI capability has been negotiated in the outgoing direction, the TS-User-Data is copied to the NS-User-Data of the N-DATA.request primitive; the process remains in the same state.
- 2) If the NullPCI capability in the outgoing direction is not enabled, a TPDU header is constructed (octets 1 and 2) and octet 3 introduces the data part. If this header and the complete TS-User-Data fits into a single TPDU, this is transmitted in a single N-DATA.request; the process remains in the same state.
- 3) If the header and the complete TS-User-Data does not fit into a single TPDU, a segmentation process is executed whereby as many maximum size TPDU's (with the EOT-bit set to zero) as necessary are sent before the last TPDU with the remainder of the TSDU, and the EOT-bit set is submitted with the N-DATA.request primitive for transmission. The process remains in the same state.
- 4) If the NullPCI capability has been negotiated in the incoming direction and an N-DATA.indication is received, the NS-User-Data is communicated in its entirety in the TS-User-Data parameter of a T-DATA.indication primitive to the Transport Fast Byte Protocol entity user. The process remains in state "Data Transfer Ready".
- 5) If the NullPCI capability in the incoming direction is not enabled, the header is verified (improper TPDU's are discarded and the process remains in state "Data Transfer Ready") and the EOT-bit is extracted. The remaining data is copied to the beginning of the variable "TSDU". If the EOT-bit is set, the TSDU is delivered to the Transport Fast Byte Protocol entity user in the TS-User-Data parameter of a T-DATA.indication primitive. If on the other hand the EOT-bit is not set, the variable "ptrRAS" is set to indicate the place where reassembly continues in the TSDU and the process enters state "Data Transfer RAS".
- 6) The procedures upon receipt of a T-DISCONNECT.request are described in C.3.2 items 2) and 3).
- 7) The procedures upon receipt of an N-DISCONNECT.indication are described in C.3.2 item 4).
- 8) Upon receipt of a T-EXPEDITED-DATA.request, the process issues an N-EXPEDITED-DATA.request primitive; the TS-User-Data is copied into the NS-User-Data. The process remains in the same state.
- 9) Upon receipt of an N-EXPEDITED-DATA.indication, the process issues a T-EXPEDITED-DATA.indication primitive; the NS-User-Data is copied into the TS-User-Data. The process remains in the same state.

C.3.5 Procedures in the state "Data Transfer RAS"

- 1) If an N-DATA.indication is received, the header is verified (improper TPDU's are discarded and the process returns to state "Data Transfer RAS") and the EOT-bit is extracted. The remaining data is copied to the variable "TSDU" (indicated by the variable "ptrRAS"). If the EOT-bit is set, the reassembled TSDU is delivered to the Transport Fast Byte Protocol entity user in the TS-User-Data parameter of a T-DATA.indication primitive. If on the other hand the EOT-bit is not set, the variable "ptrRAS" is set to indicate the place where reassembly continues in the TSDU and the process remains in state "Data Transfer RAS".
- 2) The procedures upon receipt of a T-DATA.request are described in C.3.4 items 1), 2), and 3).

- 3) The procedures upon receipt of a T-DISCONNECT.request are described in C.3.2 items 2) and 3).
- 4) The procedures upon receipt of an N-DISCONNECT.indication are described in C.3.2 item 4).
- 5) The procedures upon receipt of a T-EXPEDITED-DATA.request are described in C.3.4 item 8).
- 6) The procedures upon receipt of an N-EXPEDITED-DATA.indication are described in C.3.4 item 9).

C.3.6 Procedures in the state "Data Transfer Abort"

- 1) If an N-DATA.indication is received, the header is verified (improper TPDU's are discarded and the process returns to state "Data Transfer Ready") and the EOT-bit is extracted. If the EOT-bit is set, the process returns to state "Data Transfer Ready". If on the other hand the EOT-bit is not set, the process remains in state "Data Transfer Abort".
- 2) The procedures upon receipt of a T-DATA.request are described in C.3.4 items 1), 2), and 3).
- 3) The procedures upon receipt of a T-DISCONNECT.request are described in C.3.2 items 2) and 3).
- 4) The procedures upon receipt of an N-DISCONNECT.indication are described in C.3.2 item 4).
- 5) The procedures upon receipt of a T-EXPEDITED-DATA.request are described in C.3.4 item 8).
- 6) The procedures upon receipt of an N-EXPEDITED-DATA.indication are described in C.3.4 item 9).

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure
Series Z	Programming languages