BS ISO 17987-2:2016

BSI Standards Publication

Road vehicles — Local Interconnect Network (LIN)

Part 2: Transport protocol and network layer services

Documents
Document Sharing Hub

Full document details: BS ISO 17987-2:2016

National foreword

This British Standard is the UK implementation of ISO 17987-2:2016.

The UK participation in its preparation was entrusted to Technical Committee AUE/16, Data Communication (Road Vehicles).

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2016. Published by BSI Standards Limited 2016

ISBN 978 0 580 81729 8

ICS 35.240.60; 43.040.15

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 August 2016.

Amendments issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO 17987-2:2016

INTERNATIONAL STANDARD

ISO 17987-2

First edition
2016-08-15

Road vehicles — Local Interconnect Network (LIN) —

Part 2: Transport protocol and network layer services

Véhicules routiers — Réseau Internet local (LIN) —

Partie 2: Protocole de transport et couches de services réseau

Reference number
ISO 17987-2:2016(E)

Documents
Document Sharing Hub

© ISO 2016

Full document details: BS ISO 17987-2:2016

COPYRIGHT PROTECTED DOCUMENT

© ISO 2016, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 17987-2:2016

Contents

Page

Foreword ... **vi**

Introduction ... **vii**

1 Scope ... **1**

2 Normative references ... **2**

3 Terms, definitions, symbols and abbreviated terms ... **2**
3.1 Terms and definitions ... 2
3.2 Symbols ... 4
3.3 Abbreviated terms ... 4

4 Conventions ... **5**

5 Network management ... **5**
5.1 Network management general information ... 5
5.2 LIN node communication state diagram ... 5
5.3 Wake up ... 6
5.3.1 Wake up general information ... 6
5.3.2 Master generated wake up ... 6
5.3.3 Slave generated wake up ... 6
5.4 Go-to-sleep ... 8

6 Network layer overview ... **8**
6.1 General ... 8
6.2 Format description of network layer services ... 9
6.3 Internal operation of network layer ... 9
6.4 Service data unit specification ... 10
6.4.1 N_AI, address information ... 10
6.4.2 <Length> ... 11
6.4.3 <MessageData> ... 11
6.4.4 <N_Result> ... 11
6.5 Services provided by network layer to higher layers ... 12
6.5.1 Specification of network layer service primitives ... 12
6.5.2 N_USData.request ... 13
6.5.3 N_USData.confirm ... 13
6.5.4 N_USData_FF.indication ... 13
6.5.5 N_USData.indication ... 13

7 Transport layer protocol ... **14**
7.1 Protocol functions ... 14
7.2 Single frame transmission ... 14
7.3 Multiple frame transmission ... 14
7.4 Transport layer protocol data units ... 16
7.4.1 Protocol data unit types ... 16
7.4.2 SF N_PDU ... 16
7.4.3 FF N_PDU ... 16
7.4.4 CF N_PDU ... 16
7.4.5 Protocol data unit field description ... 16
7.5 Protocol control information specification ... 17
7.5.1 N_PCI ... 17
7.5.2 SingleFrame N_PCI parameter definition ... 18
7.5.3 FirstFrame N_PCI parameter definition ... 19
7.5.4 ConsecutiveFrame N_PCI parameter definition ... 19
7.6 Network layer timing ... 20
7.6.1 Timing constraints ... 20
7.6.2 Network layer timeouts ... 25
7.6.3 Network layer error handling ... 25
7.6.4 Unexpected arrival of N_PDU ... 26

© ISO 2016 – All rights reserved

AVA Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

8 Data link layer usage ... **27**
8.1 Data link layer service parameters ... 28
8.2 Data link layer interface services ... 28
8.2.1 L_Data.request ... 28
8.2.2 L_Data.confirm ... 28
8.2.3 L_Data.indication ... 28
8.3 Mapping of the N_PDU fields ... 28
8.4 Transport layer PDU structure and communication ... 29
8.4.1 PDU structure ... 29
8.4.2 Communication ... 31

9 Diagnostic communication requirements ... **31**
9.1 Definition of diagnostic classes ... 31
9.1.1 General ... 31
9.1.2 Diagnostic class I ... 31
9.1.3 Diagnostic class II ... 31
9.1.4 Diagnostic class III ... 31
9.1.5 Summary of slave node diagnostic classes ... 32
9.2 Diagnostic messages ... 32
9.3 Using the transport layer ... 32
9.4 Slave node diagnostic timing requirements ... 34
9.5 Response pending ... 36
9.6 Transport protocol handling in LIN master ... 36
9.6.1 General ... 36
9.6.2 Diagnostic master request schedule ... 36
9.6.3 Diagnostic slave response schedule ... 37
9.6.4 Diagnostic schedule execution ... 37
9.7 Transmission handler requirements ... 42
9.7.1 General ... 42
9.7.2 Master node transmission handler ... 42
9.7.3 Slave node transmission handler ... 46
9.8 Diagnostic service prioritization ... 48

10 LIN node capability language (NCL) ... **48**
10.1 General ... 48
10.2 Plug and play workflow concept ... 49
10.2.1 General ... 49
10.2.2 LIN node generation ... 50
10.2.3 LIN cluster design ... 50
10.2.4 Debugging ... 51

11 Node capability file (NCF) ... **51**
11.1 Overview of NCF syntax ... 51
11.2 Global structure definition ... 51
11.2.1 Node capability file marker ... 51
11.2.2 Language version number definition ... 52
11.2.3 NCF revision ... 52
11.2.4 Big-endian signal encoding variant ... 52
11.3 Node definition ... 52
11.3.1 General node definition ... 52
11.3.2 Diagnostic definition ... 53
11.3.3 Frame definition ... 54
11.3.4 Signal encoding type definition ... 55
11.3.5 Status management ... 56
11.3.6 Free text definition ... 56
11.4 NCF example ... 56

12 LIN description file (LDF) ... **57**
12.1 General ... 57
12.2 Overview of LDF syntax ... 57

AVA Documents
Document Sharing Hub

© ISO 2016 – All rights reserved

Full document details: BS ISO 17987-2:2016

12.3 LDF definition ... 58
12.3.1 Global structure definition ... 58
12.3.2 Signal definition ... 59
12.3.3 Frame definition ... 60
12.3.4 Node definition ... 62
12.3.5 Schedule table definition ... 65
12.3.6 Signal encoding type definition ... 67
12.3.7 Signal representation definition ... 69
12.4 LDF example ... 69

Bibliography ... 72

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

The committee responsible for this document is ISO/TC 22, *Road vehicles*, SC 31, *Data communication*.

A list of all parts in the ISO 17987 series can be found on the ISO website.

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

Introduction

This ISO 17987 (all parts) specifies the use cases, the communication protocol and physical layer requirements of an in-vehicle communication network called Local Interconnect Network (LIN).

The LIN protocol as proposed is an automotive focused low speed universal asynchronous receiver transmitter (UART) based network. Some of the key characteristics of the LIN protocol are signal based communication, schedule table based frame transfer, master/slave communication with error detection, node configuration and diagnostic service transportation.

The LIN protocol is for low cost automotive control applications, for example, door module and air condition systems. It serves as a communication infrastructure for low-speed control applications in vehicles by providing:

— signal based communication to exchange information between applications in different nodes;

— bitrate support from 1 kbit/s to 20 kbit/s;

— deterministic schedule table based frame communication;

— network management that wakes up and puts the LIN cluster into sleep mode in a controlled manner;

— status management that provides error handling and error signalling;

— transport layer that allows large amount of data to be transported (such as diagnostic services);

— specification of how to handle diagnostic services;

— electrical physical layer specifications;

— node description language describing properties of slave nodes;

— network description file describing behaviour of communication;

— application programmer's interface.

ISO 17987 (all parts) is based on the open systems interconnection (OSI) basic reference model as specified in ISO/IEC 7498-1 which structures communication systems into seven layers.

The OSI model structures data communication into seven layers called (top down) *application layer* (layer 7), *presentation layer*, *session layer*, *transport layer*, *network layer*, *data link layer* and *physical layer* (layer 1). A subset of these layers is used in ISO 17987 (all parts).

ISO 17987 (all parts) distinguishes between the services provided by a layer to the layer above it and the protocol used by the layer to send a message between the peer entities of that layer. The reason for this distinction is to make the services, especially the application layer services and the transport layer services, reusable also for other types of networks than LIN. In this way, the protocol is hidden from the service user and it is possible to change the protocol if special system requirements demand it.

ISO 17987 (all parts) provides all documents and references required to support the implementation of the requirements related to the following.

— ISO 17987-1: This part provides an overview of the ISO 17987 (all parts) and structure along with the use case definitions and a common set of resources (definitions, references) for use by all subsequent parts.

— ISO 17987-2: This part specifies the requirements related to the transport protocol and the network layer requirements to transport the PDU of a message between LIN nodes.

— ISO 17987-3: This part specifies the requirements for implementations of the LIN protocol on the logical level of abstraction. Hardware related properties are hidden in the defined constraints.

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

— ISO 17987-4: This part specifies the requirements for implementations of active hardware components which are necessary to interconnect the protocol implementation.

— ISO/TR 17987-5: This part specifies the LIN application programmers interface (API) and the node configuration and identification services. The node configuration and identification services are specified in the API and define how a slave node is configured and how a slave node uses the identification service.

— ISO 17987-6: This part specifies tests to check the conformance of the LIN protocol implementation according to ISO 17987-2 and ISO 17987-3. This comprises tests for the data link layer, the network layer and the transport layer.

— ISO 17987-7: This part specifies tests to check the conformance of the LIN electrical physical layer implementation (logical level of abstraction) according to ISO 17987-4.

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

INTERNATIONAL STANDARD ISO 17987-2:2016(E)

Road vehicles — Local Interconnect Network (LIN) —

Part 2: Transport protocol and network layer services

1 Scope

This document specifies a transport protocol and network layer services tailored to meet the requirements of LIN-based vehicle network systems on local interconnect networks. The protocol specifies an unconfirmed communication.

The LIN protocol supports the standardized service primitive interface as specified in ISO 14229-2.

This document provides the transport protocol and network layer services to support different application layer implementations like

— normal communication messages, and

— diagnostic communication messages.

The transport layer defines transportation of data that is contained in one or more frames. The transport layer messages are transported by diagnostic frames. A standardized API is specified for the transport layer.

Use of the transport layer is targeting systems where diagnostics are performed on the backbone bus (e.g. CAN) and where the system builder wants to use the same diagnostic capabilities on the LIN sub-bus clusters. The messages are in fact identical to the ISO 15765-2 and the PDUs carrying the messages are very similar.

The goals of the transport layer are

— low load on LIN master node,

— to provide full (or a subset thereof) diagnostics directly on the LIN slave nodes, and

— targeting clusters built with powerful LIN nodes (not the mainstream low cost).

A typical system configuration is shown in Figure 1.

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

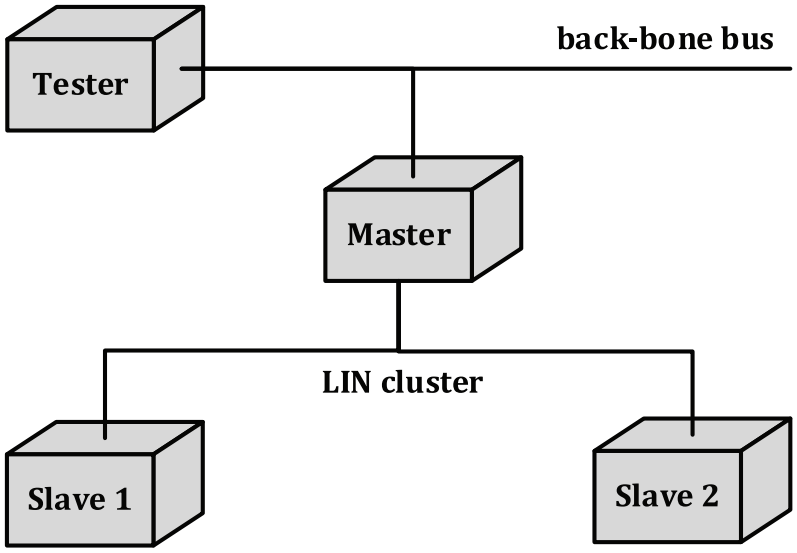

Figure 1 — Typical system setup for a LIN cluster using the transport layer

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 14229-1, *Road vehicles — Unified diagnostic services (UDS) — Part 1: Specification and requirements*

ISO 14229-2, *Road vehicles — Unified diagnostic services (UDS) — Part 2: Session layer services*

ISO 14229-7:2015, *Road vehicles — Unified diagnostic services (UDS) — Part 7: UDS on local interconnect network (UDSonLIN)*

ISO 17987-3:2016, *Road vehicles — Local Interconnect Network (LIN) — Part 3: Protocol specification*

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 7498-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1.1
broadcast NAD
slave node receiving a message with a NAD equal to the broadcast NAD $7F_{16}$ the message is received and processed

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

3.1.2
configured NAD
value in the range of (01_{16} to $7D_{16}$) which is assigned to each slave node

Note 1 to entry: The assignment of configured NAD to each slave node is defined in the LDF. The configured NAD is used for node configuration and identification services, as well as UDS services according to ISO 14229-7.

Note 2 to entry: When communication is initialized configured NADs of slave nodes may be identical. The master shall assign unique configured NADs to all slave nodes before diagnostic communication begins.

Note 3 to entry: Setting or altering the configured NAD in a slave node can be done by the following ways:

— the master node assigns a new configured NAD to a slave node supporting the "Assign NAD" service;

— an API call in a slave node assigns the configured NAD;

— the configured NAD is assigned with a static configuration.

3.1.3
functional NAD
($7E_{16}$)
used to broadcast diagnostic requests

3.1.4
initial NAD
constant/static value in the range of (01_{16} to $7D_{16}$)

Note 1 to entry: initial NAD value may be derived from a pin configuration, EEPROM or slave node position detection algorithm before entering the operational state (regular LIN communication).

Note 2 to entry: The combination of initial NAD, Supplier ID and Function ID unique for each slave node is used in the "Assign NAD" command allowing an unambiguous configured NAD assignment.

Note 3 to entry: If no initial NAD is defined for a slave node (LDF, NCF) the value is identical to the configured NAD.

3.1.5
P2 timing parameter
application timing parameter for the ECU(s) and the external test equipment

3.1.6
P2* timing parameter
enhanced response timing parameter for the ECU(s) application after response pending frame transmission

3.1.7
P4 timing parameter
timing parameter for the ECU(s) application defining the time between reception of a request and the final response

3.1.8
proprietary NAD
NAD values in the range [80_{16} – FF_{16}] are used for not standardized communication purpose, such as Tier-1 slave node diagnostics

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

3.2 Symbols

% percentage

µs microsecond

ms millisecond

| The vertical bar indicates choice. Either the left hand side or the right hand side of the vertical bar shall appear

3.3 Abbreviated terms

API application programmers interface

BNF Bachus-Naur format

CAN Controller Area Network

CF ConsecutiveFrame

FF FirstFrame

LDF LIN description file

L_Data data link data

MRF master request frame

N_AI network address information

N_As network layer timing parameter As

N_As_{max} timeout on As

N_Cr network layer timing parameter Cr

N_Cr_{max} timeout on Cr

N_Cs network layer timing parameter Cs

N_Cs_{max} timeout on Cs

N_Data network data

N_PCI network protocol control information

N_PCItype network protocol control information type

N_PDU network protocol data unit

N_SA network source address

N_SDU network service data unit

N_TAtype network target address type

N_USData network layer LIN data transfer service name

NAD node address for slave nodes

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

NCF	node capability file
NCL	node capability language
NRC	negative response code
NWL	network layer
OBD	on-board diagnostics
OSI	Open Systems Interconnection
PDU	protocol data unit
PID	protected identifier
RSID	response service identifier
SF	SingleFrame
SID	service identifier
SN	SequenceNumber
SRF	slave response frame
ST_{min}	SeparationTime minimum

4 Conventions

ISO 17987 (all parts) and ISO 14229-7 are based on the conventions specified in the OSI Service Conventions (ISO/IEC 10731) as they apply for physical layer, protocol, transport protocol and network layer services and diagnostic services.

5 Network management

5.1 Network management general information

Network management in a LIN cluster refers to cluster wake up and go-to-sleep only. Other network management features, for example, configuration detection and limp home management are left to the application.

5.2 LIN node communication state diagram

The state diagram in Figure 2 shows the behaviour model for LIN communication state.

— Bus Sleep

  Bus Sleep state is entered after first connection to power source and the system initialization, reset or when a go-to-sleep command is transmitted by the master or received by the slave node. The level on the bus is set to recessive. Only the wake up signal may be transmitted on the cluster.

— Operational

  The protocol behaviour (transmitting and receiving frames) specified in this document only applies to the Operational state.

NOTE The LIN “Bus Sleep” state does not necessarily correlate to the node’s power state.

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

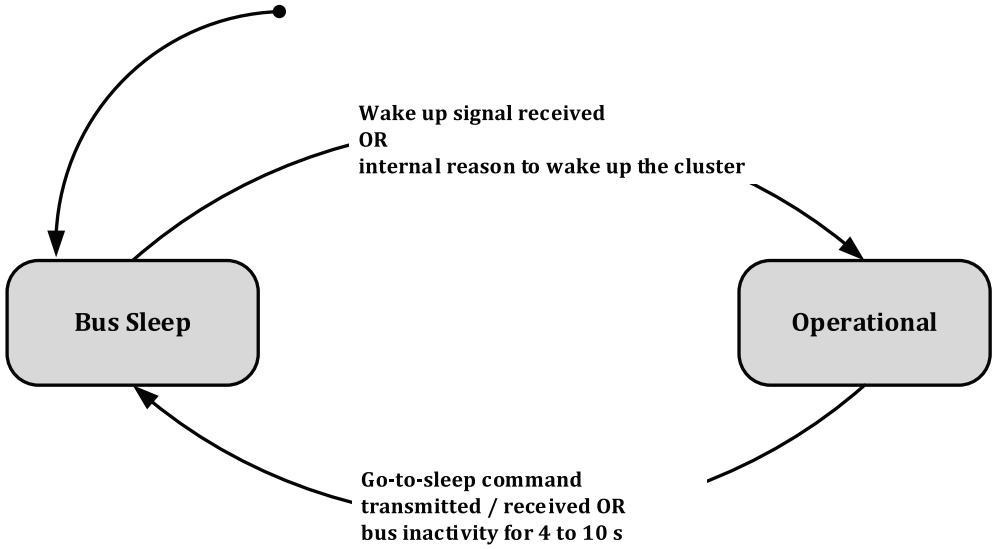

Figure 2 — LIN node communication state diagram

5.3 Wake up

5.3.1 Wake up general information

Any node in a sleeping LIN cluster may request a wake up, by transmitting a wake up signal. The wake up signal is started by forcing the bus to the dominant state for 250 µs to 5 ms, and is valid with the return of the bus signal to the recessive state.

5.3.2 Master generated wake up

The master node may issue a break field, e.g. by issuing an ordinary header since the break acts as a wake up signal (in this case, the master shall be aware of that this frame may not be processed by the slave nodes since they may not yet awake and ready to listen to headers).

Every slave node (connected to power) should detect the wake up signal (a dominant pulse longer than 150 µs followed by a rising edge of the bus signal) and be ready to listen to bus commands within 100 ms, measured from the ending edge of the dominant pulse (see Figure 3). The check for the rising edge shall be done by the transceiver and also could be done by the microcontroller LIN interface.

A detection threshold of 150 µs combined with a 250 µs pulse generation gives a detection margin that is enough for uncalibrated slave nodes. Following the detection of the wake up pulse, the Slave task machine (ISO 17987-3:2016, 7.5.3) shall start and enter into the idle state. During the idle state, the slave shall never issue a dominant level pulse on the bus until the state machine enters into an active state.

5.3.3 Slave generated wake up

If the node that transmitted the wake up signal is a slave node, it shall be ready to receive or transmit frames immediately. The master node shall also wake up and, when the slave nodes are ready (>100 ms), start transmitting headers to find out the cause (using signals) of the wake up.

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

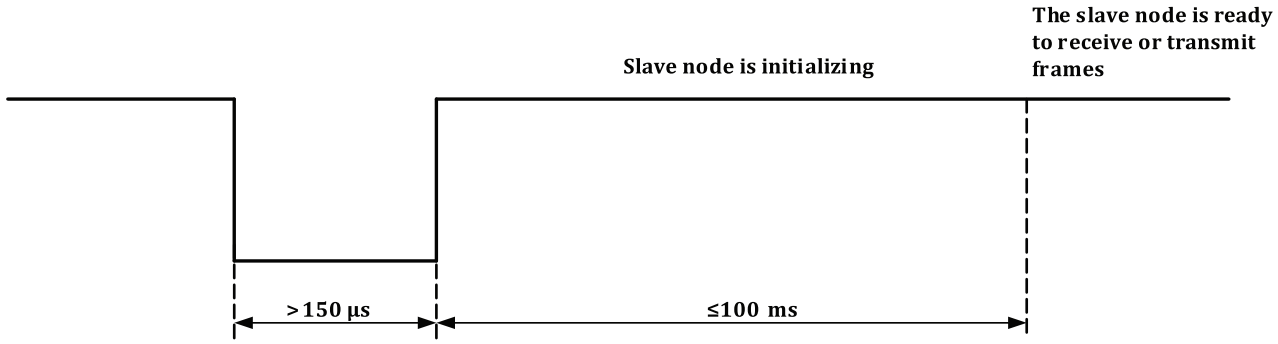

Figure 3 — Wake up signal reception in slave nodes

The master node shall detect the wake up signal (a dominant pulse longer than 150 µs followed by a rising edge of the bus signal) and be ready to start communication within a time that is decided by the cluster designer or application specific. The check for the rising edge shall be done by the transceiver and also could be done by the microcontroller LIN interface.

If the master node does not transmit a break field (i.e. starts to transmit a frame) or if the node issuing the wake up signal does not receive a wakeup signal (from another node) within 150 ms to 250 ms from the wake up signal, the node issuing the wake up signal shall transmit a new wake up signal (see Figure 4). In case the slave node transmits a wake up signal in the same time as the master node transmits a break field, the slave shall receive and recognize this break field.

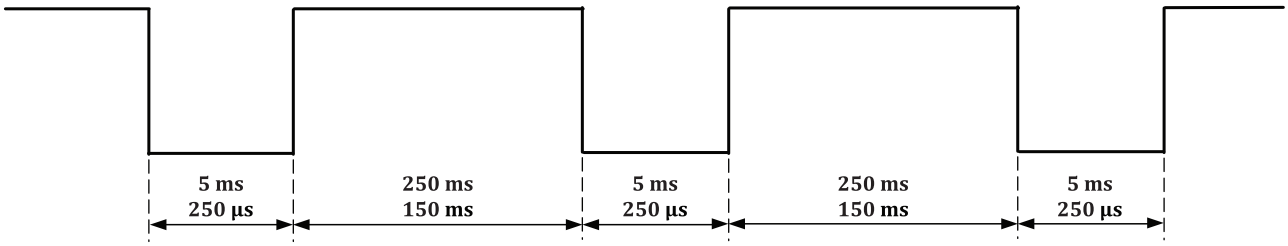

Figure 4 — One block of wake up signals

After three (failing) requests, the node shall wait minimum 1,5 s before issuing a fourth wake up signal. The reason for this longer duration is to allow the cluster to communicate in case the waking slave node has problems, e.g. if the slave node has problems with reading the bus it probably retransmit the wake up signal infinitely. There is no restriction of how many times a slave may transmit the wake up signal. However, it is recommended that a slave node transmits not more than one block of three wake up signals for each wake up condition. Figure 5 shows how wake up signals are transmitted over a longer time

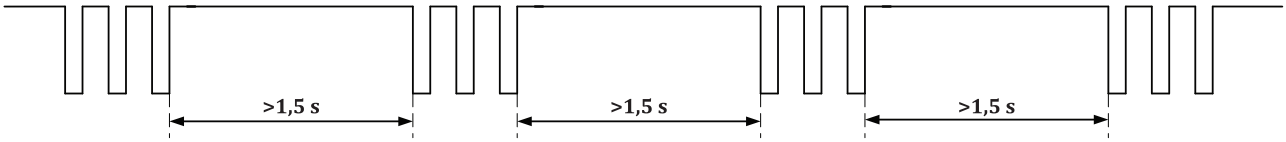

Figure 5 — Wake up signals over long time

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

5.4 Go-to-sleep

The master sets each node in the cluster instantly into bus sleep state by transmitting a go-to-sleep command. The request does not necessarily enforce the slave nodes into a low-power mode. The slave node application may still be active after the go-to-sleep command has been received. This behaviour is application specific. The go-to-sleep command is a master request frame with the first data field (Byte 1) set to 00_{16} and the rest set to FF_{16} (see Table 1). The slave nodes shall ignore the data fields Byte 2 to Byte 8 and interpret only the first data field (Byte 1).

Table 1 — Go-to-sleep command

LIN frame							
Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
00_{16}	FF_{16}	FF_{16}	FF_{16}	FF_{16}	FF_{16}	FF_{16}	FF_{16}

The normal way for setting the cluster to sleep is that the master node transmits the go-to-sleep command. In case of bus inactivity, a slave node shall be able to receive/transmit frames for 4 s. The slave node shall automatically enter bus sleep mode earliest 4 s and latest 10 s of bus inactivity. Bus inactivity is defined as no transitions between recessive and dominant bit values. Bus activity is the inverse.

NOTE LIN transceivers normally have filters to remove short spikes on the bus. The transition here refers to the signal after this filter.

6 Network layer overview

6.1 General

This document specifies an unconfirmed network layer communication protocol for the exchange of data between network nodes, e.g. from ECU to ECU, or between external test equipment and an ECU. If the data to be transferred do not fit into a single LIN frame, a segmentation method is provided.

In order to describe the function of the network layer, services provided to higher layers and the internal operation of the network layer shall be considered.

All network layer services have the same general structure. To define the services, three types of service primitives are specified:

— a service request primitive, used by higher communication layers or the application to pass control information and data required to be transmitted to the network layer;

— a service indication primitive, used by the network layer to pass status information and received data to upper communication layers or the application;

— a service confirmation primitive, used by the network layer to pass status information to higher communication layers or the application.

This service specification does not specify an application programming interface, but only a set of service primitives that are independent of any implementation.

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

6.2 Format description of network layer services

All network layer services have the same general format. Service primitives are written in the form:

```
service_name.type     (
                      parameter A,
                      parameter B
                      [,parameter C, ...]
                      )
```

where service_name is the name of the service, e.g. N_USData, type indicates the type of the service primitive, and "parameter A, parameter B [parameter C, ...]" are the N_SDU as a list of values passed by the service primitive. The brackets indicate that this part of the parameter list may be empty.

The service primitives define how a service user (e.g. diagnostic application) cooperates with a service provider (e.g. network layer). The following service primitives are specified in this document: request, indication and confirm.

— Using the service primitive request (`service_name.request`), a service user requests a service from the service provider.

— Using the service primitive indication (`service_name.indication`), the service provider informs a service user about an internal event of the network layer or the service request of a peer protocol layer entity service user.

— With the service primitive confirm (`service_name.confirm`), the service provider informs the service user about the result of a preceding service request of the service user.

6.3 Internal operation of network layer

The internal operation of the network layer provides methods for segmentation and reassembly. The main purpose of the network layer is to transfer messages that might or might not fit in a single LIN frame. Messages that do not fit into a single LIN frame are segmented into multiple parts, where each can be transmitted in a LIN frame.

Figures 6 and 7 show, respectively, an example of an unsegmented message transmission and of a segmented message transmission.

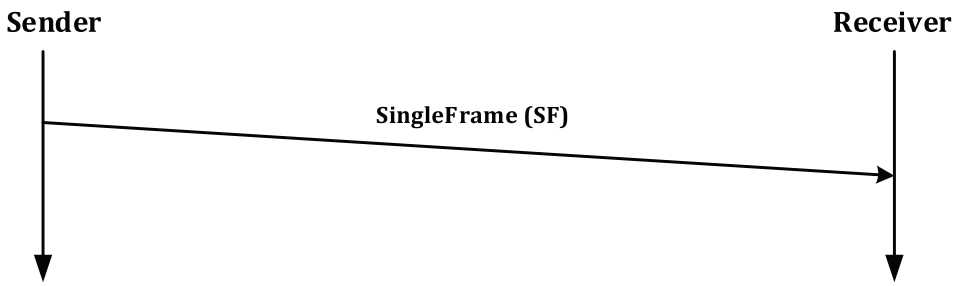

Figure 6 — Example of an unsegmented message

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

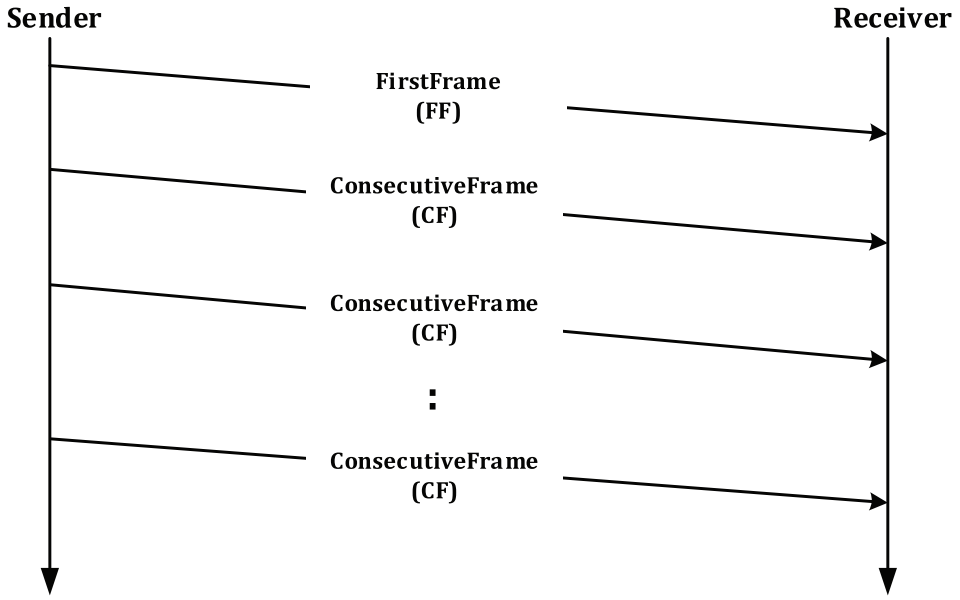

Figure 7 — Example of a segmented message

6.4 Service data unit specification

In the following subclauses, all network layer service parameters are described which are used by the LIN network layer services.

6.4.1 N_AI, address information

6.4.1.1 N_AI description

The `N_AI` is used to identify the communicating peer entities of the network layer. `N_AI` consists of a LIN ID and a `N_NAD`. Two dedicated LIN IDs are used for any transport layer protocol communication.

ID $3C_{16}$ (MasterReq) is assigned to the message transmissions sent by the master node.

ID $3D_{16}$ (SlaveResp) is assigned to the message transmissions sent by any slave node.

Because the node type (Master or Slave) is always fixed for all nodes in a LIN cluster, the LIN ID used is no subject of the service primitive parameters (6.5.1) but assigned to each node by this document.

Additionally, a `N_NAD` comprises the slave node addressed by the request/response and also the addressing type (physical, functional and broadcast).

These parameters refer to addressing information. As a whole, the `N_AI` parameters are used to identify the message senders and recipients as well as the communication model for the message (`N_TAtype`).

6.4.1.2 N_NAD, Network NAD

Type: 1 byte unsigned integer value

Range: 01_{16} – FF_{16}

Description: Two communication models are specified

— 1 to 1 communication, called physical addressing, and

— 1 to n communication called functional addressing or broadcast addressing.

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

Depending on the N_NAD value the communication model is given

— 00_{16}: not applicable, reserved for sleep command frame,

— 01_{16} – $7D_{16}$: physical addressing type,

— $7E_{16}$: functional addressing type,

— $7F_{16}$: broadcast addressing type, and

— 80_{16} – FF_{16}: proprietary addressing type.

Physical addressing shall be supported for all types of network layer messages. Responses on requests are expected. The N_NAD value shall always be validated against the configured NAD in a slave node.

Functional addressing shall only be supported for SingleFrame communication. No responses shall be transmitted on functional requests. The master node is the only node transmitting functional messages.

Broadcast addressing shall be supported for all types of network layer messages. Responses are unexpected, but supported, even if they could raise collisions. The master node shall be the only node transmitting broadcast messages.

The addressing type of NAD values in range 80_{16} – FF_{16} is unassigned by this document and shall be assigned by users when this range of NAD is used.

6.4.2 <Length>

Type: 12 bit

Range: 001_{16} to FFF_{16}

Description: This parameter includes the length of data to be transmitted/received.

6.4.3 <MessageData>

Type: string of bytes

Range: not applicable

Description: This parameter includes all data the higher layer entities exchange.

6.4.4 <N_Result>

Type: enumeration

Range: N_OK, N_TIMEOUT_As, N_TIMEOUT_Cs, N_TIMEOUT_Cr, N_WRONG_SN, N_UNEXP_PDU, N_ERROR, N_UNEXP_NEW_REQ

Description: This parameter contains the status relating to the outcome of a service execution. If two or more errors are discovered at the same time, then the network layer entity shall use the parameter value first found in this list in the error indication to the higher layers.

— N_OK

This value means that the service execution has completed successfully, it can be issued to a service user on both the sender and receiver side.

— N_TIMEOUT_As

This value is issued to the service user on the sender side when the timer N_As has passed its timeout value N_As_{max}.

— N_TIMEOUT_Cs

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

This value is issued to the protocol user on the sender side when the timer N_Cs has passed its timeout value N_Cs_{max}.

— N_TIMEOUT_Cr

This value is issued to the service user on the sender side when the timer N_Cr has passed its timeout value N_Cr_{max}.

— N_WRONG_SN

This value is issued to the service user upon reception of an unexpected SequenceNumber (N_PCI.SN) value, it can be issued to the service user on the receiver side only.

— N_UNEXP_PDU

This value is issued to the service user upon reception of an unexpected protocol data unit, it can be issued to the service user on the receiver or transmitter side.

— N_ERROR

This is the general error value. It shall be issued to the service user when an error has been detected by the network layer and no other parameter value can be used to better describe the error. It can be issued to the service user on both the sender and receiver side.

— N_UNEXP_NEW_REQ

This value is issued

a) to the lower priority service instances with an active diagnostic communication as parameter of N_USData.indication() upon service call of a new higher priority call of N_USData.request,

b) when a N_USData.request can't be accepted due to a currently active diagnostic communication with a higher priority, and

c) this value is issued to the service user upon service call of a new N_USData.request with the same priority.

6.5 Services provided by network layer to higher layers

6.5.1 Specification of network layer service primitives

The service interface defines a set of services that are needed to access the functions offered by the network layer, i.e. transmission/reception of data and setting of protocol parameters.

The communication services, of which the following are defined, enable the transfer of up to 4 095 bytes of data.

a) N_USData.request

This service is used to request the transfer of data. If necessary, the network layer segments the data.

b) N_USData_FF.indication

This service is used to signal the beginning of a segmented message reception to the upper layer.

c) N_USData.indication

This service is used to provide received data to the higher layers.

d) N_USData.confirm

This service confirms to the higher layers that the requested service has been carried out (successfully or not).

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

6.5.2 N_USData.request

The service primitive requests transmission of <MessageData> with <Length> bytes from the sender to the receiver peer entities identified by the address information in N_NAD (see 6.4.1.2 for parameter definition).

Each time the N_USData.request service is called, the network layer shall signal the completion (or failure) of the message transmission to the service user by means of the issuing of a N_USData.confirm service call:

```
N_USData.request     (
                     N_NAD
                     <MessageData>
                     <Length>
                     )
```

6.5.3 N_USData.confirm

The N_USData.confirm service is issued by the network layer. The service primitive confirms the completion of a N_USData.request service identified by the address information in N_NAD. The parameter <N_Result> provides the status of the service request (see 6.4.4 for parameter definition).

```
N_USData.confirm     (
                      N_NAD
                      <N_Result>
                      )
```

6.5.4 N_USData_FF.indication

The N_USData_FF.indication service is issued by the network layer. The service primitive indicates to the adjacent upper layer the arrival of a FirstFrame (FF) of a segmented message received from a peer protocol entity, identified by the address information in N_NAD (see 6.4.1.2 for parameter definition). This indication shall take place upon reception of the FirstFrame (FF) of a segmented message.

```
N_USData_FF.indication     (
                           N_NAD
                           <Length>
                           )
```

The N_USData_FF.indication service shall always be followed by an N_USData.indication service call from the network layer, indicating the completion (or failure) of the message reception.

A N_USData_FF.indication service call shall only be issued by the network layer if a correct FirstFrame (FF) message segment has been received.

If the network layer detects any type of error in a FirstFrame (FF), then the message shall be ignored by the network layer and no N_USData_FF.indication shall be issued to the adjacent upper layer.

If the network layer receives a FirstFrame (FF) with a data length value (FF_DL) that is greater than the available receiver buffer size, then this shall be considered as an error condition and no N_USData_FF.indication shall be issued to the adjacent upper layer.

6.5.5 N_USData.indication

The N_USData.indication service is issued by the network layer. The service primitive indicates <N_Result> events and delivers <MessageData> with <Length> bytes received from a peer protocol entity identified by the address information in N_NAD to the adjacent upper layer (see 6.4.1.2 for parameter definition).

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

The parameters <MessageData> and <Length> are only valid if <N_Result> equals N_OK.

```
N_USData.indication     (
                        N_NAD
                        <MessageData>
                        <Length>
                        <N_Result>
                        )
```

The N_USData.indication service call is issued after the reception of a SingleFrame (SF) message or as an indication of the completion (or failure) of a segmented message reception.

If the network layer detects any type of error in a SingleFrame (SF), then the message shall be ignored by the network layer and no N_USData.indication shall be issued to the adjacent upper layer.

7 Transport layer protocol

7.1 Protocol functions

The transport layer protocol performs the following functions:

— transmission/reception of messages up to 4 095 data bytes;

— reporting of transmission/reception completion/failure occurrence.

7.2 Single frame transmission

Figure 8 shows the transmission of a SingleFrame (SF) called LIN frame, which consists of 8 bytes (Byte 1 … Byte 8) with Byte 1 = N_ NAD, Byte 2 = N_PCI and Byte 3 … Byte 8 = N_Data. The actual message payload of a SF (see 7.4.2) is up to 6 data bytes (see also 8.3).

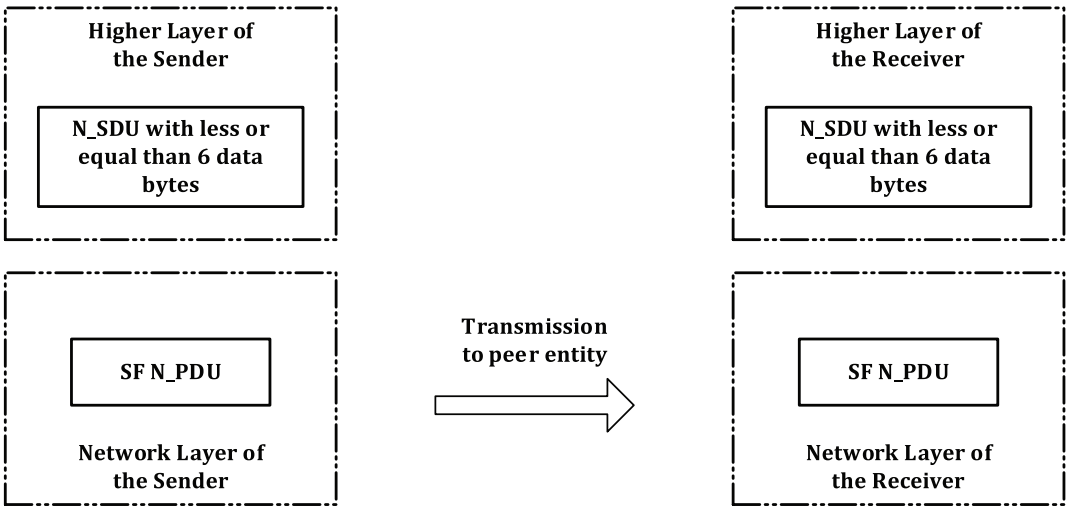

Figure 8 — Example of a SingleFrame (SF) transmission

7.3 Multiple frame transmission

Transmission of longer messages is performed via segmentation of the message into LIN frames (transmission of multiple N_PDUs). Reception of longer messages is performed via reception of multiple LIN frames (N_PDUs) and reassembly of the received data bytes (concatenation) into a message. The multiple N_PDUs are called FirstFrame (for the first N_PDU of the message) and ConsecutiveFrame (for all the following N_PDUs).

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

Messages longer than the allowed number of data bytes contained in a SF N_PDU are segmented into

— a FirstFrame protocol data unit (FF N_PDU), containing Byte 1 = N_ NAD, Byte 2 + Byte 3 = N_PCI and Byte 4 .. Byte 8 = N_Data. The actual payload of a FF (see 7.4.3 and 8.3) is 5 data bytes, and

— one or more ConsecutiveFrame protocol data units (CF N_PDU), containing Byte 1 = N_NAD, Byte 2 = N_PCI and Byte 3 .. Byte 8 = N_Data. The actual payload of a CF (see 7.4.4 and 8.3) is 6 data bytes. The last CF N_PDU contains the remaining valid payload data bytes N_Data. Unused data bytes in the last CF N_PDU are stuffed with value FF_{16}.

Figure 9 shows segmentation at the sender side and reassembly at the receiver side.

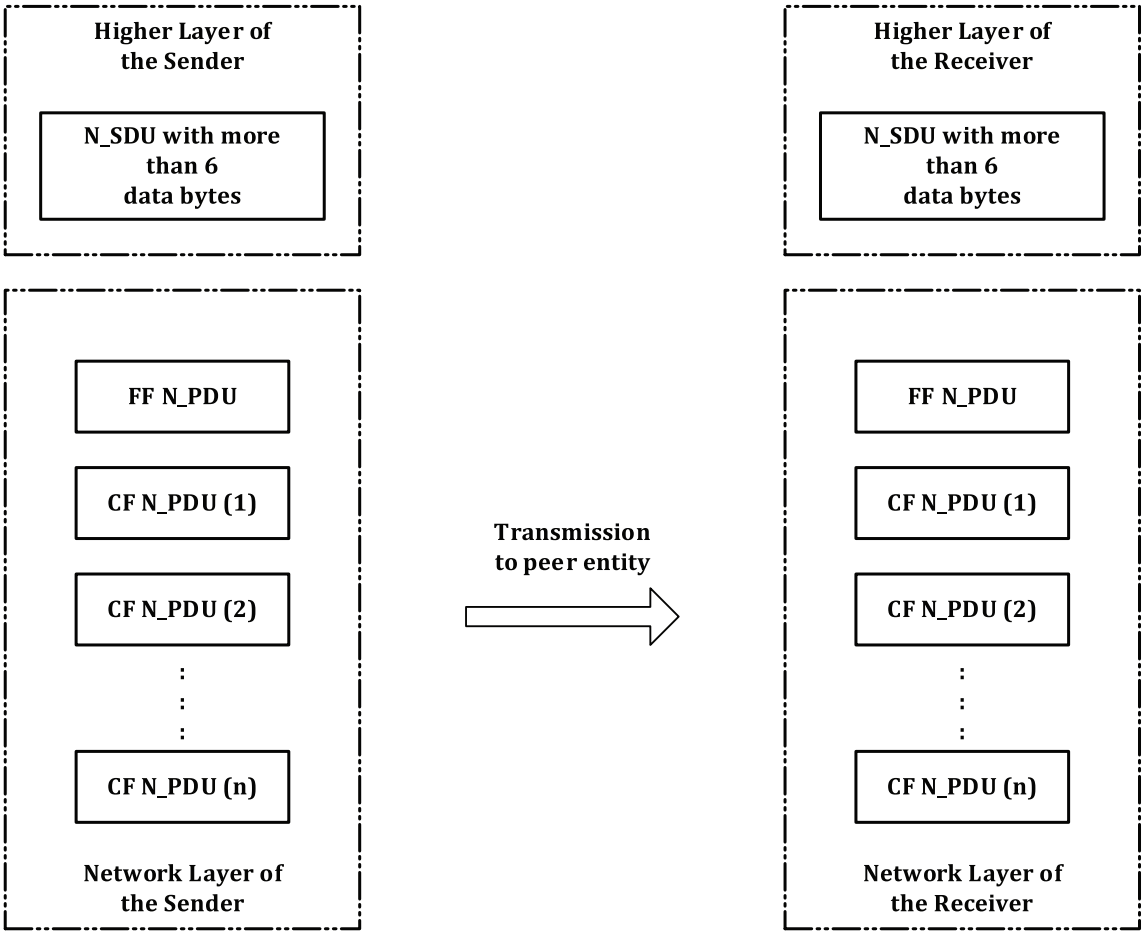

Figure 9 — Example of a multiple frame transmission (segmentation and reassembly)

The message length is transmitted in the FF N_PDU. All CF N_PDUs are numbered by the sender to support verification of correct order of transmission.

The master node application layer shall conform to the slave nodes capabilities. This can be realized by schedule table design, scheduling mode (diagnostic only vs. interleaved mode) or by application specific scheduling control. The transport layer of receiving slave nodes does not observe the ST_{min}.

These slave node capabilities are defined as follows.

SeparationTime minimum (ST_{min}) is the minimum time the master node shall wait between the transmissions of a FF and its first CF, as well as two CF. The separation time is statically specified for each slave node (LDF/NCF) when supporting transport layer communication. The measurement of the ST_{min} starts after completion of transmission of a ConsecutiveFrame (CF) and ends at the request for the transmission of the next CF.

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016

A slave node shall not take care on the capabilities of the master node (ST_{min}) because it is the responsibility of the master node application layer to apply an appropriate scheduling.

Capabilities of all nodes is that, each time the sender/receiver waits for an N_PDU from the receiver/sender, a timeout mechanism allows detection of a transmission failure (see 7.6.2).

7.4 Transport layer protocol data units

7.4.1 Protocol data unit types

The communication between the peer protocol entities of the network layer in different nodes is done by means of exchanging N_PDUs.

This document specifies three different types of transport layer protocol data units:

— SingleFrame (SF N_PDU);

— FirstFrame (FF N_PDU);

— ConsecutiveFrame (CF N_PDU);

which are used to establish a communication path between the peer network layer entities, to exchange communication parameters, to transmit user data and to release communication resources.

7.4.2 SF N_PDU

The SF N_PDU is identified by the SingleFrame protocol control information (SF N_PCI). The SF N_PDU shall be sent out by the sending network entity and received by all network entities. Only the one entity that is addressed by the N_AI shall proceed this request for the duration of the segmented message transmission. Other nodes which are not addressed shall reject already pending connections. It shall be sent out to transfer a service data unit that can be transferred via a single service request to the data link layer, and to transfer unsegmented messages.

7.4.3 FF N_PDU

The FF N_PDU is identified by the FirstFrame protocol control information (FF N_PCI). The FF N_PDU shall be sent out by the sending network entity and received by all network entities. Only the one entity that is addressed by the N_AI shall proceed this request for the duration of the segmented message transmission. Other nodes which are not addressed shall reject already pending connections. It identifies the first N_PDU of a segmented message transmitted by a network sending entity and received by a receiving network entity. The receiving network layer entity shall start assembling the segmented message on receipt of a FF N_PDU.

7.4.4 CF N_PDU

The CF N_PDU is identified by the ConsecutiveFrame protocol control information (CF N_PCI). The CF N_PDU transfers segments (N_Data) of the service data unit message data (`<MessageData>`). All N_PDUs transmitted by the sending entity after the FF N_PDU shall be encoded as CF N_PDUs. The receiving entity shall pass the assembled message to the service user of the network receiving entity after the last CF N_PDU has been received. The CF N_PDU shall be sent out by the sending network entity and received by a unique receiving network entity for the duration of the segmented message transmission.

7.4.5 Protocol data unit field description

7.4.5.1 N_PDU format

The protocol data unit (N_PDU) enables the transfer of data between the network layer in one node and the network layer in one or more other nodes (peer protocol entities). All N_PDUs consist of three fields, as specified in Table 2.

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 17987-2:2016