BS EN 61158-4-19:2014

BSI Standards Publication

Industrial communication networks — Fieldbus specifications

Part 4-19: Data-link layer protocol specification — Type 19 elements

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of EN 61158-4-19:2014. It is identical to IEC 61158-4-19:2014. It supersedes BS EN 61158-4-19:2012 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee AMT/7, Industrial communications: process measurement and control, including fieldbus.

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.

Published by BSI Standards Limited 2014

ISBN 978 0 580 79447 6

ICS 25.040.40; 35.100.20; 35.110

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 30 November 2014.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 61158-4-19

October 2014

ICS 25.040.40; 35.100.20; 35.110

Supersedes EN 61158-4-19:2012

English Version

Industrial communication networks - Fieldbus specifications - Part 4-19: Data-link layer protocol specification - Type 19 elements (IEC 61158-4-19:2014)

Réseaux de communication industriels - Spécifications des bus de terrain - Partie 4-19: Spécification du protocole de la couche liaison de données - Eléments de type 19 (CEI 61158-4-19:2014)

Industrielle Kommunikationsnetze - Feldbusse - Teil 4-19: Protokollspezifikation des Data Link Layer (Sicherungsschicht) - Typ 19-Elemente (IEC 61158-4-19:2014)

This European Standard was approved by CENELEC on 2014-09-19. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Ref. No. EN 61158-4-19:2014 E

Foreword

The text of document 65C/762/FDIS, future edition 3 of IEC 61158-4-19, prepared by SC 65C “Industrial networks” of IEC/TC 65 “Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61158-4-19:2014.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2015-06-19
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2017-09-19

This document supersedes EN 61158-4-19:2012.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 61158-4-19:2014 was approved by CENELEC as a European Standard without any modification.

In the official version, for bibliography, the following notes have to be added for the standards indicated:

IEC 61131 Series	NOTE	Harmonised as EN 61131 Series
IEC 61158-1:2014	NOTE	Harmonised as EN 61158-1:2014
IEC 61158-3-19:2014	NOTE	Harmonised as EN 61158-3-19:2014
IEC 61158-5-16:2007	NOTE	Harmonised as EN 61158-5-16:2007
IEC 61158-5-19:2014	NOTE	Harmonised as EN 61158-5-19:2014
IEC 61158-6-16:2007	NOTE	Harmonised as EN 61158-6-16:2007
IEC 61158-6-19:2014	NOTE	Harmonised as EN 61158-6-19:2014
IEC 61784-1	NOTE	Harmonised as EN 61784-1
IEC 61784-2	NOTE	Harmonised as EN 61784-2
IEC 61800-7 Series	NOTE	Harmonised as EN 61800-7 Series
IEC 61800-7-201	NOTE	Harmonised as EN 61800-7-201
IEC 61800-7-202	NOTE	Harmonised as EN 61800-7-202
IEC 61800-7-203	NOTE	Harmonised as EN 61800-7-203
IEC 61800-7-204	NOTE	Harmonised as EN 61800-7-204

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

Publication	Year	Title	EN/HD	Year
IEC 61158-4-16	2007	Industrial communication networks - Fieldbus specifications Part 4-16: Data-link layer protocol specification - Type 16 elements	EN 61158-4-16	2008
ISO/IEC 7498-1	-	Information technology - Open Systems Interconnection - Basic reference model: The basic model	-	-
ISO/IEC 7498-3	-	Information technology - Open Systems Interconnection - Basic reference model: Naming and addressing	-	-
ISO 8601	-	Data elements and interchange formats - Information interchange - Representation of dates and times	-	-
ISO/IEC 8802-3	-	Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications	-	-
IEEE 802.3	-	IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications	-	-
IETF RFC 879	-	The TCP Maximum Segment Size and Related Topics	-	-

CONTENTS

INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC 61158-1.

The data-link protocol provides the data-link service by making use of the services available from the physical layer. The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer data-link entities (DLEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

a) as a guide for implementors and designers;

b) for use in the testing and procurement of equipment;

c) as part of an agreement for the admittance of systems into the open systems environment;

d) as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

NOTE Attention is drawn to the fact that use of the associated protocol type(s) is restricted by its (their) intellectual-property-right holder(s). In all cases, the commitment to limited release of intellectual-property-rights made by the holder(s) of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the profile parts. Use of the various protocol type(s) in other combinations may require permission from their respective intellectual-property-right holders.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning Type 19 elements and possibly other types given in this document as follows:

DE 102 00 502 4759.8-32	[BR]	Verfahren zur Laufzeitkorrektur in einer Kommunikationsstruktur
DE 102 37 097	[RI]	Korrektur von Signallaufzeiten in verteilten Kommunikationssystemen

IEC takes no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured IEC that they are willing to negotiate licenses either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holders of these patent rights is registered with IEC. Information may be obtained from:

[BR] BoschRexrothAG
Zum Eisengiesser 1
D-97816 Lohr
Germany

[RI] Rexroth Indramat GmbH
Bgm.-Dr.-Nebel-Str. 2
D - 97816 Lohr
Germany

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 4-19: Data-link layer protocol specification – Type 19 elements

1 Scope

1.1 General

The data-link layer provides basic time-critical messaging communications between devices in an automation environment.

This protocol provides communication opportunities to all participating data-link entities

a) in a synchronously-starting cyclic manner, according to a pre-established schedule, and
b) in a cyclic or acyclic asynchronous manner, as requested each cycle by each of those data-link entities.

Thus this protocol can be characterized as one which provides cyclic and acyclic access asynchronously but with a synchronous restart of each cycle.

1.2 Specifications

This standard specifies

a) procedures for the timely transfer of data and control information from one data-link user entity to a peer user entity, and among the data-link entities forming the distributed data-link service provider;
b) the structure of the fieldbus DLPDUs used for the transfer of data and control information by the protocol of this standard, and their representation as physical interface data units.

1.3 Procedures

The procedures are defined in terms of

a) the interactions between peer DL-entities (DLEs) through the exchange of fieldbus DLPDUs;
b) the interactions between a DL-service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives;
c) the interactions between a DLS-provider and a Ph-service provider in the same system through the exchange of Ph-service primitives.

1.4 Applicability

These procedures are applicable to instances of communication between systems which support time-critical communications services within the data-link layer of the OSI or fieldbus reference models, and which require the ability to interconnect in an open systems interconnection environment.

Profiles provide a simple multi-attribute means of summarizing an implementation’s capabilities, and thus its applicability to various time-critical communications needs.

1.5 Conformance

This standard also specifies conformance requirements for systems implementing these procedures. This part of this standard does not contain tests to demonstrate compliance with such requirements.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE All parts of the IEC 61158 series, as well as IEC 61784-1 and IEC 61784-2 are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

IEC 61158-4-16:2007, *Industrial communication networks – Fieldbus specifications – Part 4-16: Data-link layer protocol specification – Type 16 elements*

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

ISO/IEC 7498-3, *Information technology – Open Systems Interconnection – Part 3: Basic Reference Model: Naming and addressing*

ISO/IEC 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

IEEE 802.3: *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*

Internet Engineering Task Force (IETF), *Request for Comments (RFC): RFC 879, The TCP Maximum Segment Size and Related Topics* (available at <http://www.ietf.org/rfc/rfc0879.txt>)

3 Terms, definitions, symbols, acronyms, abbreviations and conventions

For the purposes of this document, the following terms, definitions, symbols, abbreviations and conventions apply.

3.1 Reference model terms and definitions

This standard is based in part on the concepts developed in ISO/IEC 7498-1 and ISO/IEC 7498-3, and makes use of the following terms defined therein.

3.2 Additional Type 19 terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.2.1
broadcast
transmission to all devices in a network without any acknowledgment by the receivers

3.2.2
communication cycle
fixed time period between two master synchronization telegrams in which real-time telegrams are transmitted in the RT channel and non-real-time telegrams are transmitted in the IP channel

3.2.3
control unit
control device (for example, a PLC as specified in the IEC 61131 standard family)

3.2.4
control word
two adjacent octets inside the master data telegram containing commands for the addressed device

3.2.5
cross communication
direct multicast data transfer between devices

3.2.6
cycle time
duration of a communication cycle

3.2.7
cyclic communication
periodic exchange of telegrams

3.2.8
cyclic data
part of a telegram, which does not change its meaning during cyclic operation of the network

3.2.9
cyclic operation
operation in which devices in the communication network are addressed and queried one after the other at fixed, constant time intervals

3.2.10
device
a slave in the communication network, (for example, a power drive system as defined in the IEC 61800 standard family, I/O stations as defined in the IEC 61131 standard family)

3.2.11
device address field
address field (eight bits) containing the address of the device

3.2.12
device control
four adjacent octets inside the master data telegram containing commands for each device

3.2.13
device status
four adjacent octets inside the acknowledge telegram containing status information for each device

3.2.14
DLE station identifier
network address assigned to a DLE

3.2.15
DLE station slot
unit (granularity of one) of position dependent mapping (for cyclic data field) of which a DLE may occupy one or more, delineated by the range beginning at the DLE station identifier with a length equal to the configured number of occupied slots

3.2.16
element
part of IDNs – each IDN has 7 elements, whereas each one has a specific meaning (for example: number, name, data)

3.2.17
EtherType
part of the Type 19 specific telegram header

3.2.18
forwarding
mode by which a device passes on a received telegram to the other port, either changed or unchanged

3.2.19
identification number
IDN
designation of operating data under which a data block is preserved with its attribute, name, unit, minimum and maximum input values, and the data

3.2.20
line
line structure
network topology, in which the transmission medium is routed from station to station in the form of a line; the information is transmitted in one direction from the master down to the last slave in the line, and then flows back to the master via all the slaves in the reverse order (CP16/3)

3.2.21
loopback
mode by which a device passes on a received telegram to the same port and to the other port, either changed or unchanged

3.2.22
master
node, which assigns the other nodes (i.e., slaves) the right to transmit

3.2.23
master data telegram
MDT
telegram, in which the master inserts its data

3.2.24
master DLE
DLE that performs the functions of network master

3.2.25
master synchronization telegram
MST
telegram, or part of a telegram, in which the master inserts a time synchronization signal

3.2.26
MDT0 telegram
telegram, in which the master transmits its synchronization data, as well as parts or all of its real-time data, to the slaves

3.2.27
participant
node which is connected to the network

3.2.28
physical layer
first layer of the ISO-OSI reference model

3.2.29
protocol
convention about the data formats, time sequences, and error correction in the data exchange of communication systems

3.2.30
real-time data
part of the telegram that does not change its meaning during cyclic operation of the interface

3.2.31
RT channel
defined time slot within the communication cycle, which passes the CPF16 real-time telegrams

3.2.32
service channel
SVC
non real-time transmission of information upon master request during RT channel

3.2.33
slave
node, which is assigned the right to transmit by the master

3.2.34
slave DLE
DLE that performs the functions of network slave

3.2.35
station
node which is connected to the network

3.2.36
status word
two adjacent octets inside the acknowledge telegram containing status information of a device

3.2.37
S-0-nnnn
designation of IDNs

3.2.38
telegram
DLPDU

3.2.39
topology
physical network architecture with respect to the connection between the stations of the communication system

3.2.40
topology index
position of a slave in a Type 19 network using a daisy chain numeration starting with value 1 at the first slave after the master

3.3 Symbols

ADR	device address ($1 \leq ADR \leq 511$) adjusted directly on the device, for example using a selector switch
AT MST	header in AT
AT0...3	acknowledge telegrams
INFO	service channel information
JtScyc	jitter in tScyc
MDT0	master data telegram with synchronization data that the slaves evaluates
MDT1...3	master data telegrams without synchronization data
P1	port 1
P2	port 2
RxD	received data
SLKN	slave identification parameter, slave arrangement
SVC	service channel
t1	AT transmission starting time
t1min	shortest AT transmission starting time
t3	command value valid time
t4	feedback acquisition capture point
t5	minimum feedback processing time
tcable	time, by which the transmitted signal is delayed by the cable, for each unit of length (approx., 5 ns/m)
trep	time, by which the received signal is delayed by a forwarding slave (input-output)

tRing	time, which a master telegram needs, until it has passed through the network and reached the master again
tScyc	communication cycle time
TxD	transmitted data

3.4 Acronyms and abbreviations

AHS	service transport handshake of the device (acknowledge HS)
AT	acknowledge telegram
C-CON	connection control
C-DEV	device control
C1D	class 1 diagnostic
CP	communication phase
CPS	communication phase switching
CRC	cyclic redundancy check
CSoS	CIP Safety on SERCOS
FCS	DLPDU check sequence
FG	Function Group
FSP	Function-specific Profile
GDP	Generic Device Profile
HP	Hot-Plug
HP0	Hot-Plug phase 0
HP1	Hot-Plug phase 1
HP2	Hot-Plug phase 2
HS	service channel handshake (see AHS and MHS)
IDN	identification number
IP	Internet protocol
MAC	media access control
MDT	master data telegram
MDT MST	header in MDT

MHS	service transport handshake of the master
MS	communication from slave to master
MST	master synchronization telegram
NRT	non real-time
P-channel	primary channel
P-telegram	primary telegram
PL	parametrization level
RT	real-time
RTC	real-time channel
RTD	real-time data in MDT or AT
S-DEV	device status
S-channel	secondary channel
S-telegram	secondary telegrama
SCH	session control header
SCP	Type 19 Communication Profile
SE	structure element
SERCOS	serial real-time communication system interface
SFD	start DLPDU delimiter
SI	structure instance
SMP	Type 19 Messaging Protocol
SVC	service channel
UCC	unified communication channel

3.5 Additional conventions

All data types are assigned identification numbers (IDNs). They include real-time data (commands and feedback values), parameters, and procedures. Most IDNs are similar to those for Type 16 (see IEC 61158-4-16, 3.6). Several IDNs relate to the application and are defined in their relevant standards (for example, IEC 61800-7-20x for Power Drive Systems).

Refer to Annex A for additional information, as well as to IEC 61158-4-16, Clause A.1 for detailed IDN specification.

4 DL-protocol overview

4.1 Overview

This protocol type provides a highly optimized means of interchanging fixed-length real-time data and variable-length segmented messages between a single master device and a set of slave devices, interconnected in a ring or a line topology. The ring topology provides for redundant communication paths, and in case of a fault it automatically switches to a set of two lines without disturbing the communication.

This protocol type also provides for direct real-time data transmission between slaves, inside the real-time channel (RTC), within each communication cycle.

The exchange of real-time data is totally synchronous by configuration and is unaffected by the messaging traffic.

The device addresses are set by the user, using a selector for example. Additional devices may be added whenever required, even during operation, without affecting the address selections, which already exist. The determination of the number, identity and characteristics of each device may be configured or may be detected automatically at start-up.

Slave interfaces shall be used to connect the slave devices to the network. At the physical layer, a slave represents the connection of one or more devices to the network. Logically, one slave with several devices shall act the same as several slaves with one device each.

This protocol type also provides a unified communication channel (UC channel), in which any Standard Ethernet DLPDUs can be exchanged between Type 19 devices and any other connected Ethernet network nodes.

There are two classes of Type 19 DLE:

a) master DLE;
b) slave DLE.

Only the master DLE is able to initiate cyclic transmission.

Type 19 telegrams are Ethernet DLPDUs according to ISO/IEC 8802-3. Type 19 real-time telegrams shall be transmitted in the real-time part of the communication cycle time. They mainly transport input and output data, for example command and feedback values. The Type 19 header specifies two types type 19 telegrams:

a) Master data telegram (MDT), in which the master transmits real-time data to the slaves;
b) Acknowledge telegram (AT), in which the slaves transmit real-time data to the master and other slaves.

Other Ethernet DLPDUs can be transmitted in the UC channel.

Type 19 specifies 4 MDTs (MDT0 to MDT3). The MDTs shall be transmitted by the master and received by each slave. The MDTs shall contain all information (for example: synchronization, command values, digital outputs) which is sent from the master to the slaves through the real-time channel.

MDT0 shall always be transmitted. MDT1 through MDT3 shall be transmitted only if required depending on the configuration of the application data to be transmitted. The master shall always send the same number of MDTs during each communication cycle.

Type 19 specifies 4 ATs (AT0 to AT3). The ATs shall be transmitted by the master with the configured AT length. The AT data fields are set to 0, except the application data of cross communication. Each slave shall insert its data into its allocated data field within the AT. The ATs shall contain all information (for example: feedback values, digital inputs) which is sent from the slaves to the master as well as to other slave devices through the real-time channel.

AT0 shall always be transmitted. AT1 through AT3 shall be transmitted only if required depending on the configuration of the application data to be transmitted. The master shall always send the same number of ATs during each communication cycle.

The allocations of the service channels (SVC), the device control (C-DEV), the device status (S-DEV) and the connections in the MDT as well as in the AT shall be configured with parameters. The lengths of connections in the MDTs and the ATs shall depend on the amount of application data and may be different for each slave depending upon configuration. The number of MDTs and ATs may also be different because of the configuration. This configuration shall meet the following requirements.

a) A SVC of a slave shall be transmitted within one MDT or one AT and shall not be spread into different MDTs or ATs.
b) A connection of a slave shall be transmitted within one MDT or one AT and shall not be spread into two different MDTs or ATs.
c) Each connection and SVC shall start at an even address in the MDT and AT.
d) Each device control and device status shall start at an even address in the MDT and AT.
e) All other combination of configurations of SVCs, connections, device control and device status are possible.

Devices in a Type19 network use the order of little endian for the serial transmission of data. Little endian describes the least significant bit of the least significant octet of the least significant word is sent first, followed by the rest of the bits of this octet, then by the rest of the octets of this word, and so on in the same order.

4.2 General DLPDU identification

4.2.1 Introduction

DLPDUs shall be identified as specified in Table 1.

Table 1 – Ethernet DLPDU identification

DLPDU field	Data type	Value/description
Dest MAC	octet[6]	Destination MAC address
Src MAC	octet[6]	Source MAC address
EtherType	WORD	0x88CD (Type 19)

4.2.2 Destination address (Dest MAC)

The master shall transmit DLPDUs to all slaves using the broadcast address 0xFFFF FFFF FFFF as the destination address.

4.2.3 Source address (Src MAC)

The source address shall always be the MAC address of the master.

4.2.4 EtherType

The EtherType for real-time DLPDUs shall contain the value 0x88CD, which is the unique type field number that has been allocated by the IEEE EtherType Field Registration Authority for Type 19 telegrams.

NOTE This field number refers to Type 19 communication.

4.3 General DLPDU structure

4.3.1 Introduction

The data structure in a DLPDU shall consist of the following data entries as specified in Table 2.

Table 2 – Data structure in a DLPDU

Data field	Data type	Value/description
Header	octet[6]	Defines the DLPDU type
Payload	octet[40-1494]	Data fields are padded, if less than 40 octets

4.3.2 DLPDU header

The DLPDU header shall specify two types of telegrams, as specified in 4.4:

- **Master data telegram (MDT):** MDTs shall transmit data from the master to the slaves;
- **Acknowledge telegram (AT):** ATs shall transmit data from the slaves to the master, as well as to other slaves within the Type 19 network.

4.3.3 DLPDU payload

All transmitted data are permitted to have arbitrary bit sequences.

Padding octets shall be added if the Type 19 data is less than 40 octets, in order to reach a total data field length of at least 46 octets.

The DLPDU payload shall be as described in 4.5 and 4.6.

4.4 DLPDU header

4.4.1 Introduction

The DLPDU header shall distinguish the various DLPDUs. It shall be coded in the telegram whether the DLPDU is transmitted in the primary or secondary channel, whether it is an MDT or an AT, and which one (MDT0 to MDT3, respectively AT0 to AT3).

In a line topology, the master shall decide whether the telegrams are marked as primary or secondary telegrams, depending upon configuration.

The DLPDU header structure is shown in Table 3:

Table 3 – DLPDU payload header

Data field	Data type	Value/description
DLPDU type	octet[1]	see 4.4.2
Reserved	octet[1]	—
Reserved	octet[4]	—

4.4.2 DLPDU type

The DLPDU type shall be generated by the master and transmitted in every MDT and AT. Its content shall be as shown in Table 4.

Table 4 – DLPDU type

Bit number	Bit value	Description
7	—	Primary or secondary telegram
	0	Telegram on the primary channel (P-Telegram)
	1	Telegram on the secondary channel (S-Telegram)
6	—	MDT or AT
	0	MDT
	1	AT
5	—	Cycle CNT
	0	Cycle CNT is disabled
	1	Cycle CNT is enabled (Cycle CNT is defined in MDT phase (bit 6-4)
4	—	(reserved)
3-2		(reserved for Telegram number 4 to 15)
1-0	—	Telegram number
	00	Telegram number 0
	01	Telegram number 1
	10	Telegram number 2
	11	Telegram number 3

4.5 MDT DLPDU

4.5.1 MDT MST field summary

The MDT shall be as specified in Table 5. The MDT header of MDT0 is called MST.

Table 5 – MDT header

DLPDU part	Data field	Data type	Value/Description
MDT header	MDT type	octet[1]	see 4.5.3
	MDT phase	octet[1]	see 4.5.4
	MDT CRC	octet[4]	see 4.5.5

4.5.2 Evaluation of MDT header in the slaves

The MDT header shall be generated by the master and evaluated by the slaves. Each slave shall evaluate the MDT header according to Table 6.

Table 6 – MDT header to be considered by the slave

	MDT type	MDT phase	MDT CRC
MDT0 = MST	Yes	Yes	Yes
MDT1	Yes	No	Yes
MDT2	Yes	No	Yes
MDT3	Yes	No	Yes

NOTE The slave has to evaluate the MDT type and MDT phase only if the MDT CRC is valid.

4.5.3 MDT type

Refer to 4.4.2, whereas bit #6 shall be 0.

4.5.4 MDT phase

The MDT phase shall contain the status of the Type 19 communication during initialization and during CP4. The phase shall be generated by the master and transmitted in every MDT. The structure is shown in Table 7.

Table 7 – MDT phase

Bit number	Bit value	Description
7	—	Communication phase switching (CPS)
	0	Current CP Communication phase (bit 3-0) contains current CP.
	1	New CP Communication phase (bit 3-0) contains the target CP for phase switching.
6-4	—	Cycle CNT (shall be enabled in MDT DLPDU type, bit 5 is set to 1)
	0-7	Value of Cycle CNT (is incremented by 1 in each communication cycle by the master)
3-0	—	Communication phase (CP)
	0000	CP0
	0001	CP1
	0010	CP2
	0011	CP3
	0100	CP4
	0101 to 1111	(reserved)

4.5.5 MDT CRC

The cyclic redundancy check (CRC) shall be used by the transmit and receive algorithms to generate a CRC value for the MDT CRC field. The MDT CRC field shall contain a 4-octet (32-bit) cyclic redundancy check (CRC) value. This value shall be computed as a function of the contents of the destination address (see 4.2.2), source address (see 4.2.3), EtherType (see 4.2.4), Type 19 type (see 4.4.2) and phase (see 4.5.4). The encoding shall be as defined by the Standard Ethernet CRC generating polynomial (see ISO/IEC 8802-3).

The MDT CRC shall be generated by the master and transmitted in every MDT (MDT0 to MDT3). This CRC shall be evaluated in every MDT by the slave (see Table 6).

4.5.6 MDT payload during initialization

4.5.6.1 General

The content of the MDT data field depends on the communication phase (CP) and is described in the following subclauses.

4.5.6.2 CP0

The master shall always transmit MDT0 telegrams only, and no MDT1, MDT2, nor MDT3 telegrams. MDT0 shall be structured as stated in Table 8.

Table 8 – MDT0 structure in CP0

DLPDU part	Data field	Data type	Value/Description
MDT	MDT type	octet[1]	MDT0, see 4.5.3
	MDT phase	octet[1]	CP0, see 4.5.4
	MDT CRC	octet[4]	see 4.5.5
MDT payload	Communication version	octet[4]	see Table 9
	MDT Data Field	octet[36]	Shall be padded and not used

Table 9 – Communication version

Bit number	Bit value	Description
31-23	—	(reserved: 0x00 as valid combination shall be checked by slaves)
22	—	SWC
	0	Non-Type 19 (Industrial Ethernet) devices not used by application
	1	Non-Type 19 (Industrial Ethernet) devices used by application: last slave in line shall not forward Type 19 telegrams to the inactive port (loopback without forward shall be activated), only last slave shall set S-0-1032 Communication control.bit 3 = 1
21	—	Fast CP switch
	0	Transmission of MST (MDT0) interrupted during CP switch for CPS delay time (120ms)
	1	CPS delay time reduced to the reconfiguration time of the master (shall be acknowledged by the slave in the topology index field of AT0-CP0)
20	—	Transmission of communication parameters in MDT0 of CP0
	0	No transmission of parameters
	1	Transmission of the following parameters: - AT0 transmission starting time (t1-CP1&CP2) - Beginning of UC channel (t6-CP1&CP2) - End of UC channel (t7-CP1&CP2)
19-18	—	(reserved: 0x00 as valid combination shall be checked by slaves)
17-16	—	Structure and number of MDTs and ATs in CP1 and CP2
	00	2 MDTs and 2 ATs in CP1 and CP2 (include SVCs, C-DEV, S-DEV) – up to 255 slaves
	01	4 MDTs and 4 ATs in CP1 and CP2 (include SVCs, C-DEV, S-DEV) – up to 511 slaves
	10	(reserved)
	11	(reserved)
15-8	—	(reserved)
7-1	—	(reserved: 0x00 as valid combination shall be checked by slaves)
0	—	Address allocation
	0	Without address allocation (used for Type 19 Version 1.0 only)

Bit number	Bit value	Description
	1	Address allocation (shall be used for Type 19 Version 1.1.1 and greater)

4.5.6.3 CP1 and CP2

The master shall choose between two communication sequences used in CP1 and CP2:

a) If the master supports 255 slaves or less, it may transmit either MDT0 to MDT3 or MDT0 and MDT1 only, for example to save initialization time.

b) If the master supports 256 slaves or more (up to 511 slaves) it shall transmit MDT0 to MDT3.

The slaves shall support both sequence options. It shall select the required one by evaluating bit 17 and bit 16 of the communication version (see Table 9).

The MDT data fields shall contain the service channel (see 6.2) and the device control (see 4.5.7.4.2) of the topology indices as shown in Table 10, Table 11, Table 12 and Table 13 respectively.

In CP1 a slave shall behave as requested if the handshake bit (MHS) is set to 1 in the corresponding SVC control. The MDT SVC INFO is “don’t care”. The content of device control shall be valid.

Telegrams in CP2 shall have the same structure as in CP1, but the contents of SVC INFO shall be valid only in CP2.

Table 10 – MDT0 in CP1 and CP2 (topology indices 0 to 127)

DLPDU part	Data field	Data type	Value/Description
MDT MST	MDT type	octet[1]	MDT0, see 4.5.3
	MDT phase	octet[1]	CP1 or CP2, see 4.5.4
	MDT CRC	octet[4]	see 4.5.5
MDT data field	MDT SVC for topology index #0	octet[6]	—
	…	…	(And so on for topology indices #1 to #126)
	MDT SVC for topology index #127	octet[6]	—
	Device control for topology address #0	octet[2]	—
	Reserved for topology index #0	octet[2]	—
	…	…	(And so on for topology indices #1 to #126)
	Device control for topology index #127	octet[2]	—
	Reserved for topology index #127	octet[2]	—

Table 11 – MDT1 in CP1 and CP2 (topology indices 128 to 255)

DLPDU part	Data field	Data type	Value/Description
MDT MST	MDT type	octet[1]	MDT1, see 4.5.3
	MDT phase	octet[1]	CP1 or CP2, see 4.5.4
	MDT CRC	octet[4]	see 4.5.5
MDT data field	SVC of topology index #128	octet[6]	—
	…	…	(And so on for topology index #129 to #254)
	SVC of topology index #255	octet[6]	—
	Device control of topology index #128	octet[2]	—
	Reserved for topology index #128	octet[2]	—
	…	…	(And so on for topology index #129 to #254)
	Device control of topology index #255	octet[2]	—
	Reserved for topology index #255	octet[2]	—

Table 12 – MDT2 in CP1 and CP2 (topology indices 256 to 383)

DLPDU part	Data field	Data type	Value/Description
MDT MST	MDT type	octet[1]	MDT2, see 4.5.3
	MDT phase	octet[1]	CP1 or CP2, see 4.5.4
	MDT CRC	octet[4]	see 4.5.5
MDT data field	SVC of topology index #256	octet[6]	—
	…	…	(And so on for topology index #257 to #382)
	SVC of topology index #383	octet[6]	—
	Device control of topology index #256	octet[2]	—
	Reserved for topology index #256	octet[2]	—
	…	…	(And so on for topology index #257 to #382)
	Device control of topology index #383	octet[2]	—
	Reserved for topology index #383	octet[2]	—

Table 13 – MDT3 in CP1 and CP2 (topology indices 384 to 511)

DLPDU part	Data field	Data type	Value/Description
MDT MST	MDT type	octet[1]	MDT3, see 4.5.3
	MDT phase	octet[1]	CP1 or CP2, see 4.5.4
	MDT CRC	octet[4]	see 4.5.5
MDT data field	SVC of topology index #384	octet[6]	—
	…	…	(And so on for topology index #385 to #510)
	SVC of topology index #511	octet[6]	—
	Device control of topology index #384	octet[2]	—
	Reserved for topology index #384	octet[2]	—
	…	…	(And so on for topology index #385 to #510)
	Device control of topology index #511	octet[2]	—
	Reserved for topology index #511	octet[2]	—

4.5.6.4 CP3 and CP4

In CP3 and CP4, the master shall transmit MDTs with the same structure (see 4.5.7).

In CP3, only the service channel and the device control shall be valid. The configured application data in the connections of the MDTs shall not be evaluated, but they shall have the number of octets required for CP4. The positions of the service channels and the connections in the MDT relevant to the individual slaves shall be as transmitted by the master to the slaves during CP2 with the corresponding communication parameters.

In CP4, the configured application data shall be valid and filled with command values as determined by the parameters that the master transmitted to the slaves during CP2. The connection control and the resource control which depend on the application profile shall be valid.

4.5.7 MDT payload in normal operation

4.5.7.1 Introduction

The MDT payload of MDT0 (see Table 14) shall always contain

a) one MDT0 hot-plug field (see 4.5.7.2) and depending on the configuration;
b) one MDT0 extended field;
c) several service channels, one per slave only (see 5.5.7.2.4);
d) several device controls, one per slave only;
e) several connections (see 4.5.7.4).

The MDT payload of MDT1 to MDT3 (see Table 14) can contain depending on the configuration

a) several service channels, one per slave only (see 5.5.7.2.4);
b) several device controls, one per slave only;
c) several connections (see 4.5.7.4).

Table 14 – MDT data field

DLPDU part	Data field	Data type	Value/Description
MDT data field	MDT0 hot-plug field	octet[8]	In MDT0 only, see 4.5.7.2
	MDT0 extended field	octet[4]	In MDT0 only, see 4.5.7.2
	MDT service channels	octet[see 4.5.7.3]	Optional. see 4.5.7.3
	MDT real-time data	octet[see 4.5.7.4]	Optional. see 4.5.7.4

For each slave,

- IDN S-0-1013 (SVC offset in MDT) shall set the offset for its service channel;
- IDN S-0-1009 (Device Control offset in MDT) shall set the offset for the device control;
- IDN S-0-1050.x.3 (Telegram assignment) shall set the offset for the connection data;
- IDN S-0-1050.x.5 (Current length of connection) shall contain the length of the connection data;
- IDN S-0-1010 (Lengths of MDTs) shall contain the lengths of all MDTs (in Figure 1 shown for MDT0 only).

These parameters shall be transmitted by the master to the slaves in CP2.

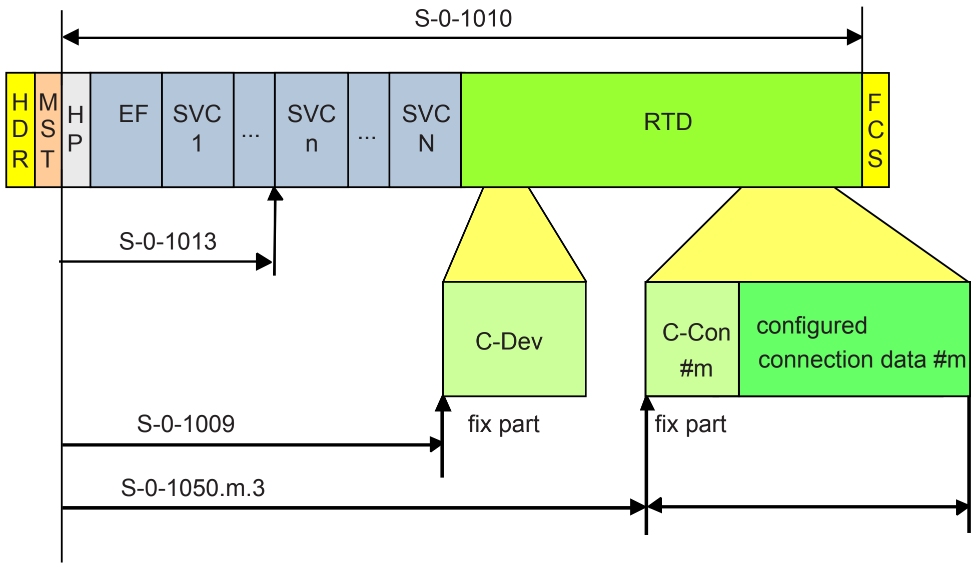

Figure 1 – Example of offsets within MDT payload

4.5.7.2 MDT hot-plug field

4.5.7.2.1 MDT hot-plug field summary

The structure of the MDT hot-plug field shall be as specified in Table 15, depending upon hot-plug phases (see 5.5).

Table 15 – MDT hot-plug field

DLPDU part	Data field	Data type	Value/Description
MDT hot-plug field	MDT-HP address	octet[2]	see 4.5.7.2.2
	HP control	octet[2]	see 4.5.7.2.3
	MDT-HP INFO	octet[4]	see 4.5.7.2.4

4.5.7.2.2 MDT-HP address field

The content of the HP address field in the MDT shall be as specified in Table 16:

Table 16 – HP address in MDT-HP field

Bit number	Value	Description
15-12	—	Slave index specifies the number of slaves within one device (device can contain multiple slaves)
	0-15	Slave index (set by master in HP0)
11-0	—	Device addresses
	0-511	Device addresses for operation (set by master in HP1 and HP2)
	512-4092	(reserved)
	4093	(reserved)
	4094	(reserved)
	4095	Broadcast address (set by master in HP0, each HP slave shall react to this address)

4.5.7.2.3 HP control field (in HP0 and HP1)

The content of HP control field shall be as specified in Table 17.

Table 17 – HP control field (in HP0 and HP1)

Bit number	Value	Description
15	—	Hot-Plug support (the master shall set this bit to 1 in CP4, if hot-plug is supported)
	0	Hot-Plug not supported by master (slave shall switch from HP0, HP1 or HP2 to NRT mode)
	1	Hot-Plug supported by master (HP slave prepared HP function)
14-10	—	(reserved)
9	—	Enable / disable Hot-plug
	0	Hot-Plug disabled (HP slave shall switch to HP0)
	1	Hot-Plug enabled (HP slave shall evaluate the HP field)
8	—	HP field vs. SVC communication
	0	HP field is used only (the master uses only the HP field to communicate with the HP slave)
	1	HP field and SVC are used (the master uses the HP field and the SVC to communicate with the HP slave)
7-0	—	Coding of HP0 parameters (0 to 127)
	0	No HP parameter (MDT-HP INFO is don’t care)
	1	Communication cycle time (tScyc, S-0-1002)

Bit number	Value	Description
	2	Beginning of UC channel (t6, S-0-1017, list element 0)
	3	End of UC channel (t7, S-0-1017, list element 1)
	4	Requested MTU (S-0-1027.0.1)
	5	Communication version (content in MDT0 of CP0)
	6-15	(reserved)
	16	MDT0 length (S-0-1010, list element 0)
	17	MDT1 length (S-0-1010, list element 1)
	18	MDT2 length (S-0-1010, list element 2)
	19	MDT3 length (S-0-1010, list element 3)
	20-31	(reserved for MDT length, MDT4 to MDT15)
	32	AT0 length (S-0-1012, list element 0)
	33	AT1 length (S-0-1012, list element 1)
	34	AT2 length (S-0-1012, list element 2)
	35	AT3 length (S-0-1012, list element 3)
	36-47	(reserved for AT length, AT4 to AT15)
	48-127	(reserved)
	—	Coding of HP1 parameters (128 to 255)
	128	MDT-SVC offset (S-0-1013)
	129	AT-SVC offset (S-0-1014)
	130	Topology index (S-0-1042, if supported)
	131-255	(reserved)

4.5.7.2.4 MDT HP INFO fields (in HP0 and HP1)

The MDT HP INFO field shall always be 4 octets long. If hot-plug data with 2 octets only are transmitted, then the data shall be in the low word, and the high word value shall be “don’t care”.

The MDT HP INFO field shall be the container for the hot-plug data exchange from the master to the HP slave device, which takes place in steps in the MDT HP field.

4.5.7.2.5 MDT Extended field

The Extended field (EF) can be transmitted in the MDT0 behind the hot-plug field (see Table 18). The extended field supports the transmission of time fragments and the TSref-counter.

1) Time fragments: The master transmits its own time (8 octets) in 4 fragments with 2 octets via this field to the slaves. The fragmentation is controlled by 2 bits and the Cycle CNT of MST.
   - Time fragment valid bit is set to 1 by the master if the data in the field "Time fragments" are valid and the Cycle CNT of MST is 0. The master may set this bit to 0 at any time.
   - Activate time: In a continuous transfer of time fragments this bit is toggled when the master has transmitted a new time and the Cycle CNT of MST is 0.
2) TSref-counter: This counter is 14 bits long and counts modulo using the value of parameter S-0-1061.
   - The initial value of the TSref-counter is 0.
   - The master increments the TSref-counter in each communication cycle by 1.

- The master calculates the Least Common Multiple (LCM) of all producer cycle times and writes the LCM value in S-0-1061. If the TSref-counter reaches the calculated LCM value, then the master set the TSref-counter to 0, and the function is started again (modulo function).

If the SCP_Classes SCP_SYNC 0x02 or SCP_SysTime are activated by the master the Extended field is present in the MDT0 of CP3 and CP4.

Table 18 – Extended Field (EF)

Bit number	Value	Description
31	—	Activate time
	toggle	Master has continuously transmitted a new time. Slave shall set the new time in its internal time system.
30	—	Time fragment valid
	0	Not valid (time fragment not valid. Slave shall not use this time fragment)
	1	Valid (time fragment valid. Slave shall use this time fragment)
29-16		TSref-counter (modulo counter with the Least Common Multiple of different producer cycle times which shall be synchronized)
15-0		Time fragment (fragments of Type 19 current time, multiplexed with Cycle CNT of MST)

4.5.7.3 MDT service channel (SVC) field

4.5.7.3.1 MDT service channel field summary

The MDT service channel field (see Table 19) shall contain all service channels (SVC) of the configured devices within a Type 19 network. Only configured slave devices shall have their own dedicated service channel, depending upon the application, whereas there shall be no restriction regarding device address order.

The service channel for each device shall be as specified in Table 20.

Table 19 – MDT service channel field

DLPDU part	Data field	Data type	Value/Description
MDT service channel field	MDT SVC for slave device #1	octet[6]	—
	MDT SVC for slave device #2	octet[6]	—
	…	…	(And so on for slave device 3 to slave device (N-1))
	MDT SVC for slave device #N	octet[6]	—

Table 20 – MDT SVC (for each slave)

DLPDU part	Data field	Data type	Value/Description
MDT SVC of slave device #k	SVC control	octet[2]	Offset in the MDT defined by S-0-1013 SVC offset in MDT of the assigned slave
	MDT SVC INFO	octet[4]	—

4.5.7.3.2 SVC control

The content of service channel (SVC) control word shall be as specified in Table 21.

Table 21 – SVC control word (DLL)

Bit number	Bit value	Control word description
15-6	—	(reserved)
5-3	—	Data block element
	000	Service channel not active, close service channel or break a transmission in progress
	001	IDN of the operation data. The service channel is closed for the previous IDN and opened for a new IDN
	010	Name of operation data
	011	Attribute of operation data
	100	Unit of the operation data
	101	Minimum value
	110	Maximum value
	111	Operation data
2	—	Bit last transmission
	0	Transmission in progress
	1	Last transmission
1	—	R/W (read/write)
	0	Read SVC INFO
	1	Write SVC INFO
0	—	MHS (master handshake bit)
	toggle	Service transport handshake of the master

4.5.7.3.3 MDT SVC INFO

The MDT SVC INFO field shall always be 4 octets long. If only 2 octets are transmitted in a step, then the data shall be in the low word, and the high word value shall be “don’t care”.

The MDT SVC INFO field shall be the container for the non-cyclic data exchange from the master to the slave device, which takes place in steps in the MDT SVC field.

4.5.7.4 MDT real-time data field

4.5.7.4.1 General

Each slave shall have only one device control as specified in Table 22.

Each slave can have several connections as specified in Table 23.

The connection data to any one slave device shall not be spread into two different MDTs. There shall be no restriction to the offset parameters (S-0-1009, ,S-0-1050.x.03) regarding device address order.

The master shall at least process the device control and the application data in the corresponding producer cycle time. If a master has several connections with different producer cycle times to one slave, then the device control shall be updated by the master with the fastest producer cycle.

Table 22 – MDT device control

DLPDU part	Data field	Data type	Value/Description
Real-time data #k	Device control	octet[2]	Offset in the MDT defined by S-0-1009 Device Control (C-DEV) Offset in MDT of the assigned slave

The device control shall be present for each slave exactly once.

Table 23 – MDT application data

DLPDU part	Data field	Data type	Value/Description
Real-time data #k	Application data	Container	Offset in the MDT defined by S-0-1050.x.03 Telegram Assignment of the assigned slave

NOTE The application data may be present for each slave 0-255 times.

The functionality of a connection is described in 4.7

4.5.7.4.2 Device control

The content of device control field shall be as specified in Table 24.

Table 24 – Device control field (C-DEV)

Bit number	Bit value	Description
15	—	Identification
	0	No Identification request
	1	Identification request (slave shows the condition of this bit at the Type 19 LED or at the display). This function is used for the remote address allocation or for configuration errors between master and slave.
14	—	Topology HS (Initial value is 0 in every CP)
	toggle	The master toggles every time it requires a topology change.
13-12	—	Topology control (Master selects the new topology)
	00	Fast-Forward on both ports
	01	Loopback with Forward of P-Telegrams
	10	Loopback with Forward of S-Telegrams
	11	(reserved: slave shall ignore this bit combination)
11	—	Control physical topology (If the slave detects a toggle, then it shall drop the source address table. The control physical topology is used in the UC channel only)
	0	physical ring is broken
	1	physical ring is closed
10-9	—	(reserved)
8	—	Master valid (indicates if the master is processing data. In CP1 the slave detects the support of this function if this bit is set to 1 by the master)
	0	Master is not valid (the master shall set this bit = 0, if the master activates NRT state or CP0. The contents of device control C-DEV are invalid. Producer ready of all producer connections shall be set to 0)
	1	Master is valid (If supported, the master shall set this bit = 1, if the master activates CP1.The contents of

Bit number	Bit value	Description
		device control C-DEV are valid)
7-0	—	(reserved)

4.6 AT DLPDU

4.6.1 AT header field summary

The AT is a Type 19 telegram and shall be as specified in Table 25. The Type 19 header is called AT header.

Table 25 – AT MST header

DLPDU part	Data field	Data type	Value/Description
AT header	AT type	octet[1]	see 4.6.3
	AT phase	octet[1]	see 4.6.4
	AT CRC	octet[4]	see 4.6.5

4.6.2 Evaluation of AT header in the slaves

The AT header shall be generated by the master and evaluated by the slaves. Each slave shall evaluate the AT header according to Table 26.

Table 26 – AT header fields to be considered by the slave

	AT type	AT phase	AT CRC
AT0	Yes	No	Yes
AT1	Yes	No	Yes
AT2	Yes	No	Yes
AT3	Yes	No	Yes

NOTE The slave has to evaluate the AT type only if the AT CRC is valid.

4.6.3 AT type

Refer to 4.4.2, whereas bit #6 shall be 1.

4.6.4 AT phase

The AT phase shall contain the status of the Type 19 communication during initialization and during CP4. The phase shall be generated by the master and transmitted in every AT. The structure is the same as for MDT phase (see 4.5.4).

The phase of an AT shall not be evaluated by the slave (see Table 26).

4.6.5 AT CRC

The AT CRC shall be generated by the master as the MDT CRC (see 4.5.5).

The AT CRC shall be evaluated by the slave (see Table 26).

4.6.6 AT Payload during initialization

4.6.6.1 General

The content of the AT payload depends on the communication phase (CP) as described in the following subclauses.

4.6.6.2 AT payload in CP0

The master shall always transmit AT0 telegrams, and no AT1, AT2, nor AT3 telegrams. AT0 shall be structured as stated in Table 27. Table 27 also specifies the AT0 payload data that the master shall fill with initial values. The master shall

- set the sequence counter to 0x0001 and / or to 0x8001 (depending on the related topology);
- fill all topology index fields with 0xFFFF.

Port Px and Py in Table 27 are interchangeable, that means port Px and Py can be port P1 or P2.

Table 27 – AT0 structure in CP0

DLPDU part	Data field	Data type	Value/Description
AT MST	AT type	octet[1]	AT0, see 4.6.3
	AT phase	octet[1]	CP0, see 4.6.4
	AT CRC	octet[4]	see 4.6.5
AT payload	Sequence counter	octet[2]	Initial values:0x0001 (Port Px), 0x8001 (Port Py) Received value (Port Px): 0x08001+number of slaves (with ring) 0x0001+(2*number of slaves)-1 (with line) Received value (Port Py): 0x0001+number of slaves (with ring) 0x8001+(2*number of slaves)-1 (with line)
	Topology index #1	octet[2]	Initial value 0xFFFF (for Port Px and Py) Received value (Port Px and Py): device address if slave present.
	Topology indices #2 to #510	octet[2*509]	Initial value 0xFFFF (for Port Px and Py) Received value (Port Px and Py): device address if slave present.
	Topology address #511	octet[2]	Initial value 0xFFFF (for Port Px and Py) Received value (Port Px and Py): device address if slave present.

The content of topology index field in the AT0 of CP0 shall be as specified in Table 28.

Table 28 – Topology address in AT0-CP0

Bit No.	Value	Description
Master (15-0)	0xFFFF	set by the master
Slave (15)	—	Support of requested functions in Communication version of MDT0-CP0 (see 5.2.4, Table 9)
	0	Slave doesn't support one or more of the requested functions (default values are activated)
	1	Slave supports all requested functions (values of all requested functions are activated)
Slave (14-9)	—	(reserved)
Slave (8-0)	—	Device addresses
	0-511	Valid device addresses

4.6.6.3 AT payload in CP1 and CP2

The master shall choose between two communication sequences in CP1 and CP2:

a) If the master supports 255 slaves or less, it may transmit either AT0 to AT3 telegrams, or AT0 and AT1 telegrams only, for example to save initialization time. The slaves shall support both sequence options by evaluating the Communication Version (see Table 9).

b) If the master supports 256 slaves or more (up to 511 slaves) it shall transmit AT0 to AT3 telegrams.

The AT data field of the AT0 shall contain the service channel (see 4.6.7.3) and the device status (see 4.6.7.4.2) of topology indices as shown in Table 29, Table 30, Table 31 and Table 32 respectively.

In CP1 the requested slave shall respond by setting the handshake bit (AHS) and valid bit (SVC valid) to 1 in the corresponding SVC status. The AT SVC INFO is "don't care".

In the device status the slave shall process the following bits:

a) slave valid bit is set to 1;

b) topology bits are updated;

c) parametrization level and communication error interface are valid.

In CP2 the telegrams shall have the same structure as in CP1, but the contents of AT SVC INFO shall be valid in CP2.

Table 29 – AT0 in CP1 and CP2 (topology indices 0 to 127)

DLPDU part	Data field	Data type	Value/Description
AT header	AT type	octet[1]	AT0, see 4.6.3
	AT phase	octet[1]	CP1 or CP2, see 4.6.4
	AT CRC	octet[4]	see 4.6.5
AT data field	AT SVC for topology index #0	octet[6]	—
	…	octet[6*126]	(and so on for topology indices #1 to #126)
	AT SVC for topology index #127	octet[6]	—
	Device status for topology index #0	octet[2]	—
	Reserved for topology index #0	octet[2]	—
	…	octet[2*126]	(And so on for topology indices #1 to #126)
	Device status for topology index #127	octet[2]	—
	Reserved for topology index #127	octet[2]	—

Table 30 – AT1 in CP1 and CP2 (topology indices 128 to 255)

DLPDU part	Data field	Data type	Value/Description
AT header	AT type	octet[1]	AT1, see 4.6.3
	AT phase	octet[1]	CP1 or CP2, 4.6.4
	AT CRC	octet[4]	see 4.6.5
AT data field	AT SVC for topology index #128	octet[6]	—
	…	octet[6*126]	(And so on for topology indices #129 to #254)
	AT SVC for topology index #255	octet[6]	—
	Device status for topology index #128	octet[2]	—
	Reserved for topology index #128	octet[2]	—
	…	octet[2*126]	(And so on for topology indices #129 to #254)
	Device status for topology index #255	octet[2]	—
	Reserved for topology index #255	octet[2]	—

Table 31 – AT2 in CP1 and CP2 (topology indices 256 to 383)

DLPDU part	Data field	Data type	Value/Description
AT header	AT type	octet[1]	AT3, see 4.6.3
	AT phase	octet[1]	CP1 or CP2, 4.6.4
	AT CRC	octet[4]	see 4.6.5
AT data field	AT SVC for topology index #256	octet[6]	—
	…	octet[6*126]	(And so on for topology indices #257 to #382)
	AT SVC for topology index #383	octet[6]	—
	Device status for topology index #256	octet[2]	—
	Reserved for topology index #256	octet[2]	—
	…	octet[2*126]	(And so on for topology indices #257 to #382)
	Device status for topology index #383	octet[2]	—
	Reserved for topology index #383	octet[2]	—

Table 32 – AT3 in CP1 and CP2 (topology indices 384 to 511)

DLPDU part	Data field	Data type	Value/Description
AT header	AT type	octet[1]	AT3, see 4.6.3
	AT phase	octet[1]	CP1 or CP2, 4.6.4
	AT CRC	octet[4]	see 4.6.5
AT data field	AT SVC for topology index #384	octet[6]	—
	…	octet[6*126]	(And so on for topology indices #385 to #510)
	AT SVC for topology index #511	octet[6]	—
	Device status for topology index #0	octet[2]	—
	Reserved for topology index #0	octet[2]	—
	…	octet[2*126]	(And so on for topology indices #385 to #510)
	Device status for topology index #511	octet[2]	—
	Reserved for topology index #511	octet[2]	—

4.6.6.4 AT payload in CP3

In CP3, the master shall transmit ATs with the same structure as in CP4. The slaves insert their data in the corresponding data fields.

In CP3, only the service channel and the device status shall be valid. The configurable application data in the connections of the ATs shall not be evaluated, but they shall have the number of octets required for CP4. The positions of the service channels and the connections in the AT relevant to the individual slaves shall be as transmitted by the master to the slaves during CP2 with the corresponding communication parameters.

4.6.7 AT payload in CP4

4.6.7.1 Introduction

In CP4, the master shall transmit ATs with the same structure as in CP3. The slaves insert their data in the corresponding data fields.

In CP4, the configurable real-time data shall be valid and filled with actual values as determined by the parameters that the master transmitted to the slaves during CP2. The connection control and the resource status which depend on the application profile shall be valid.

The AT payload of AT0 (see Table 33) shall always contain

- one AT0 hot-plug field and depending on the configuration;
- several service channels, one per slave only;
- several device status, one per slave only;
- several connections.

The AT payload of AT1 to AT3 (see Table 33) can contain depending on the configuration

- several service channels, one per slave only;
- several device status, one per slave only;
- several connections.

The AT payload structure shall be as specified in Table 33.

Table 33 – AT data field

DLPDU part	Data Field	Data Type	Value/Description
AT data field	AT0 hot-plug field	octet[8]	in AT0 only, see 4.6.7.2
	AT service channels	octet[see 4.6.7.3]	Optional. see 4.6.7.3
	AT real-time data	octet[see 4.6.7.4]	Optional. see 4.6.7.4

For each slave,

- IDN S-0-1014 (SVC offset in AT) shall set the offset for its service channel;
- IDN S-0-1011 (Device status S-DEV) shall set the offset for the device status;
- IDN S-0-1050.x.3 (Telegram assignment) shall set the offset for the connection data;
- IDN S-0-1050.x.5 (Current length of connection) shall contain the length of the connection data;
- IDN S-0-1012 (Length of ATs) shall contain the length of the ATs (see Figure 2).

These parameters shall be transmitted from the master to the slaves in CP2.

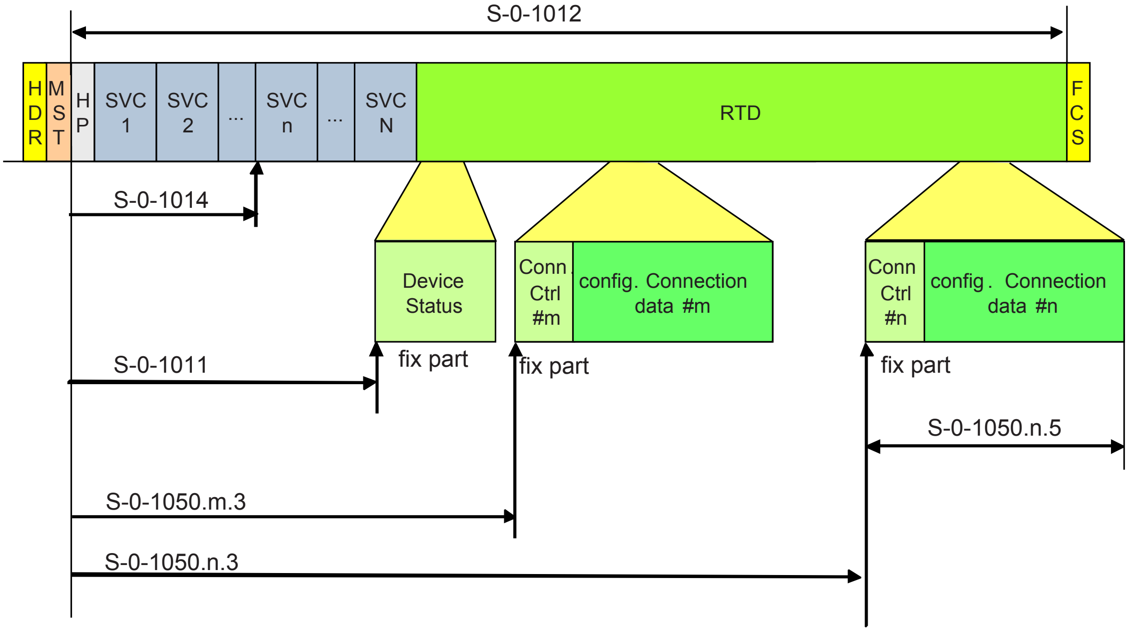

Figure 2 – Example of Offsets within AT payload

4.6.7.2 AT hot-plug field

4.6.7.2.1 AT hot-plug field summary

The structure of the AT hot-plug field shall be as specified in Table 34, depending upon hot-plug phases (see 5.5).

Table 34 – AT hot-plug field in HP0 and HP1

DLPDU part	Data field	Data type	Value/Description
AT hot-plug field	AT-HP address	octet[2]	see 4.6.7.2.2
	HP status	octet[2]	S-HP, see 4.6.7.2.3
	AT-HP INFO	octet[4]	(reserved, see 5.5)

4.6.7.2.2 AT-HP address field

The content of HP address field in the AT shall be as specified in Table 35.

Table 35 – HP address in AT-HP field

Bit number	Value	Description
15-12	—	Slave index specifies the number of slaves within one device (device can contain multiple slaves)
	0..15	Slave index (set by slave in HP1)
11-0	—	Device addresses
	0-511	Device addresses for operation (set by slave in HP1 and HP2)
	512-4092	(reserved)
	4093	End of scanning of slave indices (an unused index was detected by the HP slave), set by slave in HP1
	4094	(reserved)
	4095	(reserved)

4.6.7.2.3 HP status field (in HP0 and HP1)

The content of HP status field shall be as specified in Table 36.

Table 36 – HP status field (in HP0 and HP1)

Bit number	Value	Description
15-9	—	(reserved)
8	—	HP Condition
	0	Acknowledgment in HP1 (code see bit 7-0)
	1	Error in HP1 (error code see bit 7-0)
7-0	—	HP1 acknowledgment or error codes
	0	(reserved)
	1	Acknowledgment: No data in AT-HP INFO (bit 8=0, AT-HP INFO is “don’t care”)
	2	Error: SVC activation (bit 8 = 1, error occurs during activating of SVC in the HP slave)
	3	(reserved)
	4	Error: device address (bit 8 = 1, master does not transmit the sercos address of the HP slave)
	5	Error: Slave scan (bit 8 = 1, master did not scan all slaves in a multi slave device)
	6 to 127	(reserved)
	128	MDT-SVC pointer (shall be supported bit 8 = 0 --> no error, parameter valid bit 8 = 1 --> error, parameter invalid)
	129	AT-SVC pointer (shall be supported bit 8 = 0 --> no error, parameter valid bit 8 = 1 --> error, parameter invalid)
	130	Topology index
	131 to 254	(reserved)
	255	Error: HP slave receives its own device address on its inactive port (bit 8 = 1 --> next HP slave connected to this HP slave has the same device address)

4.6.7.3 AT service channel field

4.6.7.3.1 AT service channel fields summary

The AT service channel field (see Table 37) shall contain all service channels (SVC) of the configured devices within a Type 19 network.

Each configured slave shall have its own dedicated service channel, depending upon the application, whereas there shall be no restriction regarding device address order.

The service channel for each slave shall be as specified in Table 38.

Table 37 – AT service channel field

DLPDU part	Data field	Data type	Value/Description
AT service channel field	AT SVC for slave # 1	octet[6]	see Table 38
	AT SVC for slave #2	octet[6]	see Table 38
	…	…	(And so on for slaves #3 to slaves #(N-1))
	AT SVC for slave #N	octet[6]	see Table 38

Table 38 – AT SVC (for each slave)

DLPDU part	Data field	Data type	Value/Description
AT SVC of slave device #k	SVC status	octet[2]	Offset in the AT defined by S-0-1014 (SVC offset in AT) of the assigned slave.
	AT SVC INFO	octet[4]	—

4.6.7.3.2 AT SVC status

The structure of SVC status shall be as specified in Table 39.

Table 39 – AT SVC status description (DLL)

Bit number	Bit value	Description
15-4	—	(reserved)
3	—	SVC processing
	0	SVC invalid (slave doesn’t process AT SVC in this P channel or S channel)
	1	SVC valid (slave processes AT SVC in this P channel or S channel)
2	—	SVC error
	0	No error
	1	Error in SVC (error message in SVC INFO)
1	—	Busy
	0	Step finished (slave ready for new step)
	1	Step in process (new step not allowed)
0	—	AHS
	toggle	SVC transport handshake of the slave (toggle bit)

4.6.7.3.3 AT SVC INFO

The AT SVC INFO field shall always be 4 octets long. If only 2 octets are transmitted in one step, then the data shall be in the low word, and the high word value shall be “don’t care”.

The AT SVC INFO field shall be the container for the non-cyclic data exchange from any slave to the master, which takes place in steps in the AT SVC field of the telegram.

4.6.7.4 AT real-time data field

4.6.7.4.1 General

Each slave shall have only one device status (see Table 40).

Each slave can have several connections as specified in Table 41.

The connection data to any one slave device shall not be spread into two different ATs. There shall be no restriction to the offset parameters (S-0-1009, S-0-1050.x.03) regarding device address order.

The slave shall at least process the device status and the application data in the corresponding producer cycle time. If a slave has several connections with different producer cycle times to the master or to other slaves, then the device status shall be updated by the slave with the slowest producer cycle.

Table 40 – AT device status

DLPDU part	Data field	Data type	Value/Description
Real-time data #k	Device status	octet[2]	Offset in the AT defined by S-0-1011 Device Status S-DEV Offset in AT of the assigned slave

The device status shall be present for each slave exactly once.

Table 41 – AT connection data

DLPDU part	Data field	Data type	Value/Description
Real-time data #k	Application data	Container	Offset in the AT defined by S-0-1050.x.03 (Telegram Assignment) of the assigned slave

NOTE The application data may be present for each slave 0-255 times.

The functionality of a connection is described in 4.7

4.6.7.4.2 Device status (S-DEV)

The content of device status field shall be as specified in Table 42.

The fastest reaction time to any event affecting device status except bits 11-10 (Status of inactive port) shall be within the slowest producer cycle time but at most 200 ms.

Table 42 – Device status field

Bit number	Bit value	Description
15	—	Communication warning interface
	0	No warning
	1	Communication warning occurred (for example: number of permitted MST losses has exceeded the half value of S-0-1003)
14	—	Topology HS
	toggle	Initial value is 0 in every CP. Slave toggles, if the request of the master has been recognized, that means, the topology status may be updated after the toggle.
13-10	—	Topology status / Port status
	00-00	Fast-Forward on both ports (Diagnostic not available)
	01-00	Loopback with Forward of P-Telegrams (no link on inactive port --> no device connected)
	01-01	Loopback with Forward of P-Telegrams (LINK on inactive port --> device connected)
	01-10	Loopback with Forward of P-Telegrams (P LINK: P telegrams on inactive port --> Type 19 device connected)
	01-11	Loopback with Forward of P-Telegrams (S LINK: S telegrams on inactive port --> Type 19 device connected)
	10-00	Loopback with Forward of S-Telegrams (no link on inactive port --> no device connected)
	10-01	Loopback with Forward of S-Telegrams (LINK on inactive port --> device connected)
	10-10	Loopback with Forward of S-Telegrams (P-LINK: P telegrams on inactive port --> Type 19 device connected)
	10-11	Loopback with Forward of S-Telegrams (S-LINK: S telegrams on inactive port --> Type 19 device connected)
	11-xx	store & forward or cut-through
	00-xx	Additional bit combinations:
	00-01	fast-forward on both ports (Diagnostic supported)
	00-10	fast-forward on both ports (error in P channel)
	00-11	fast-forward on both ports (error in S channel)
9	—	Error connection
	0	Error-free connection
	1	Error in the connection occurred (consumer recognized an error in a connection)
8	—	Slave valid (indicates if a slave is processing data)
	0	Slave not valid (Set to 0 when entering CP0. Modified during CPS. The contents of device status S-DEV are invalid. Producer ready of all producer connections shall be set to 0)
	1	Slave valid (CP > CP0. Modified during CPS the contents of device status (S-DEV) are valid)
7	—	Error (C1D), inclusive sub-device and resource errors
	0	No error
	1	Error (detailed information is shown in S-0-0390)
6	—	Warning (C2D), inclusive sub-device and resource warnings

Bit number	Bit value	Description
	0	No Warning
	1	Warning (detailed information is shown in S-0-0390)
5	—	Procedure command change bit
	0	No change in procedure command acknowledgement
	1	Changing procedure command acknowledgment (procedure command is positive or negative acknowledged)
4	—	Sub-device level
	0	Operating level (OL) is active
	1	Parametrization level (PL) is active
3	—	(reserved)
2	—	(reserved)
1-0	—	(reserved)

4.7 Mechanisms of connections

4.7.1 Introduction

The communication connects all participants in a network. Because of this, it is possible that each participant is able to communicate with any other. A connection determines which participants communicate together. The Type 19 network supports the application data exchange between the master and all slaves, and between the slaves in both directions. Not all participants need to communicate with each other, therefore, the application data exchange between the participants are configured via connections. The connection functionality is based on a producer-consumer model. That means, for each transmission of application data, one connection is required.

A connection shall have one producer only, but it can have no consumer or several consumers. In the S-0-1050.x.01 (Connection setup) the participant gets the info, whether the connection is configured as a producer or consumer. The connection control is a fixed part of each connection and is configured automatically. The transmitted application data of the connection are configurable.

Description and definitions of Figure 5:

- write: Application writes application data and signals it to the producer
- produce & set: First, the producer inserts application data into the connection, then it sets the connection control (C-Con) related to producer state machine
- evaluate & consume: First, the consumer evaluates the connection control (C-Con), then it consumes application data from the connection and signals it to the application
- read: Application reads application data from the consumer.

Flow of application data from application to application:

a) The application (e.g. control unit, drive, IO, etc.) generates the application data and writes it to the producer.
b) The producer is informed also and selects the corresponding state of the state machine and produces the application data into the connection and sets the connection control based on the producer state's conditions.
c) The communication transmits all connections with the application data within the Type 19 telegrams.
d) The consumer evaluates the connection control and consumes the application data based on the consumer state's conditions.

e) The application is informed by the consumer and reads the application data.

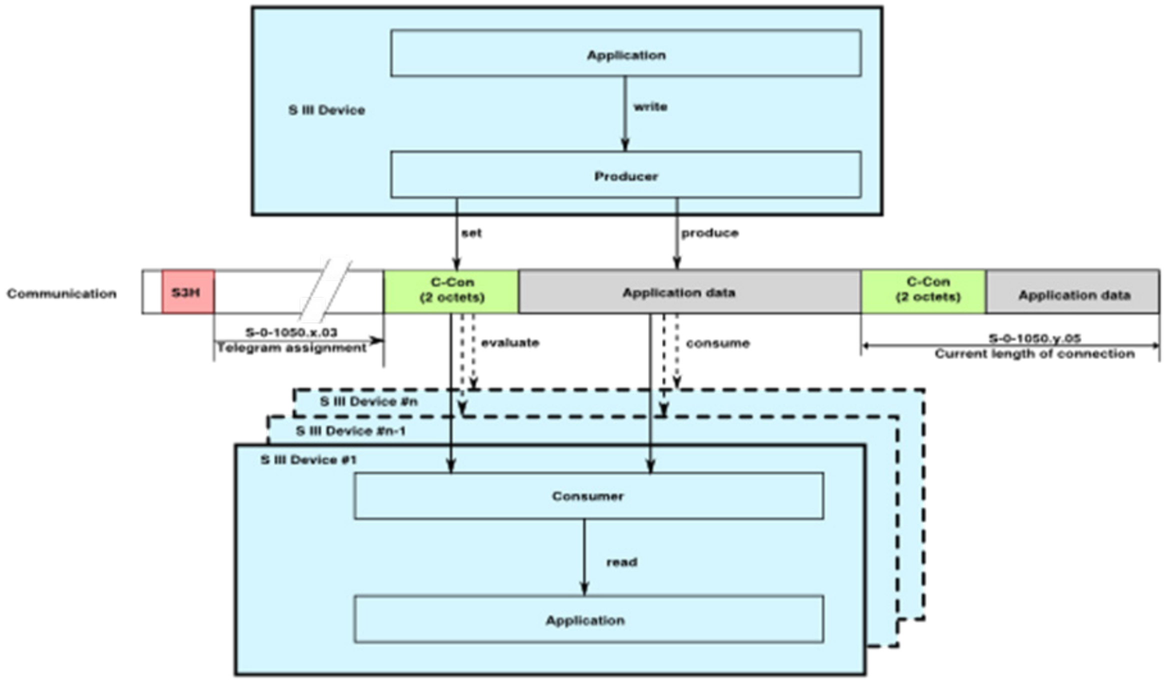

Figure 3 – Flow of application data

4.7.2 Configuration of connections

The configuration of the application data is done using S-0-1050 (Connections). It shall be assigned to a slave during initialization in CP2. All configured application data in a connection shall be write protected in CP3 and CP4.

The configuration rules of connections are the following:

- Only operation data shall be used.
- For operation data with fixed length the determined data length in the attribute is used.
- For operation data with variable length, the current data length used in the connection shall be round up to an even number of octets. The 4 octets length indicators are not part of the application data.
- Each operation data shall start at an even number of octets within the connection.
- The structure of the connection shall be determined by S-0-1050.x.06 (Configuration List), see Table 43.

The connection shall be as configured in S-0-1050.x.01 (Connection setup). This connection connects one producer to all consumers. Depending upon configuration, the application data shall be either:

- configurable by the master during initialization; see S-0-1050.x.06 (Configuration List);
- configurable by the slave during initialization, for example S-0-1500.x.05 (Container OutputData) or S-0-1500.x.09 (Container InputData);
- as specified in the telegram type parameter S-0-0015 (Telegram type).

The position of the connection within a Type 19 telegram is defined by the S-0-1050.x.03 (Telegram Assignment) and the S-0-1050.x.05 (Current length of connection), see Figure 4.

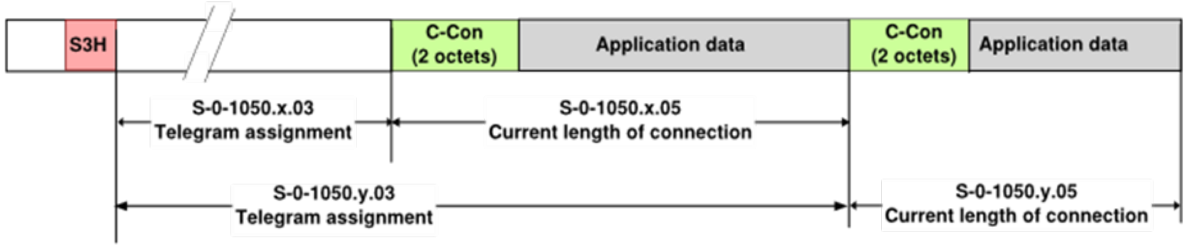

Figure 4 – Telegram assignment and connection length

Table 43 – Structure of the connection

Frame part	Data field	Data type	Value/Description
Slave #k configurable application data	Connection control	octet[2]	Offset in the MDT or AT defined by S-0-1050.x.03 (Telegram Assignment) of the assigned slave
	Operation data IDN …	octet[depending upon IDN]	Number and length of operation data k shall be configured in S-0-1050.x.06 (Configuration List) or by the selected standard telegram S-0-0015 (Telegram type)
	Operation data IDN …	octet[depending upon IDN]	
	…	…	
	Operation data IDN …	octet[depending upon IDN]	

4.7.3 Connection control

The connection control shall be set as specified in Table 44. The Connection Control (C-CON) are the first two octets of each connection.

Table 44 – Connection control (C-CON)

Bit number	Bit value	Description
15-12		Counter: - initial value of this counter is 0 in CP4 - every change of this counter announces new application data in the connection and the application data may be processed - at a cycle synchronous connection this counter shall always be increased in the related communication cycle - at a non-synchronous connection the change of this counter triggers a watchdog, i.e. after change of this counter the monitoring time (tPcyc) is always started again. This counter shall be increased once per monitoring time (tPcyc), after producer ready (bit 0) is set to 1 - Bit 12 shall be equal to bit 1 (new data)
11-8	—	(reserved)
7		Real-time bit 2 (part of application data, description see FG_RTB)
6		Real-time bit 1 (part of application data, description see FG_RTB)
5	—	(reserved)
4	—	Flow-control (description see 4.7.4 and 4.7.5)
	0	Run (producing and consuming are active)
	1	Stop (the producing is canceled, the consumer shall not generate an error)
3	—	(reserved)
2	—	Data field delay (the consumer shall prefer taking the application data of the port at which this bit has the value 0)
	0	No delay (application data are transmitted without delay in the same

Bit number	Bit value	Description
		communication cycle)
	1	Delay (master has copied the application data and therefore the application data are transmitted with additional delay of one communication cycle)
1	—	New application data
	toggle	- initial value of this bit is 0 in CP4 - every toggle of this bit announces new application data in the connection and then the application data are exchanged between connection and application. This implies, that in the consumer - when the application data of a certain producer cycle has not been received, the prospected value for this bit has also to be toggled for the next communication cycle. - at a clock synchronous or cyclic connection this bit shall always be toggled in the related communication cycle. - at a non-synchronous connection this bit triggers a watchdog, i.e. after toggle of this bit the monitoring is always started again with the monitoring time (tPcyc). This bit shall be toggled once per monitoring time (tPcyc), after producer ready (bit 0) is set to 1. - Bit 1 shall be equal to bit 12 (LSB of counter).
0	—	Producer ready If master valid resp. slave valid is set to 0, this bit shall be set to 0 also.
	0	Not valid (the producer does not generate any application data in this connection yet)
	1	Valid The producer generates application data in this connection. The consumer can process the application data if the producer has toggled the new data (bit 1). The producer ready bit shall be evaluated in CP4 only.

The connection control state machines of producer and consumer describe the dynamic behavior of establishing and releasing connections. The producer is the active part that provides application data and determines when the connection is activated or deactivated. The consumer waits for valid application data and reacts on state switches of the producer. The producer and the consumer provide their connection state in S-0-1050.x.09 (Connection State). The producer also provides the S-0-1050.x.08 (Connection Control C-CON).

4.7.4 Producer state machine

In addition to S-0-1050.x.09 (Connection State), the producer signals its current state to the consumer by setting corresponding bits in the connection control (C-CON). The initial state of the producer is “init”. In this state the producer is waiting for CP4. By reaching CP4, the producer shall change to state “network in operation (CP4)”. As long as the system is in CP4, the producer shall stay in this state. The state “network in operation (CP4)” includes a sub-state machine, which describes the current producing state of the producer. The producer can change its sub-states among “prepare”, “ready”, ”producing”, “stopping” and "waiting". If the communication leaves CP4, the producer state machine shall go back to “init” state (see Figure 5).

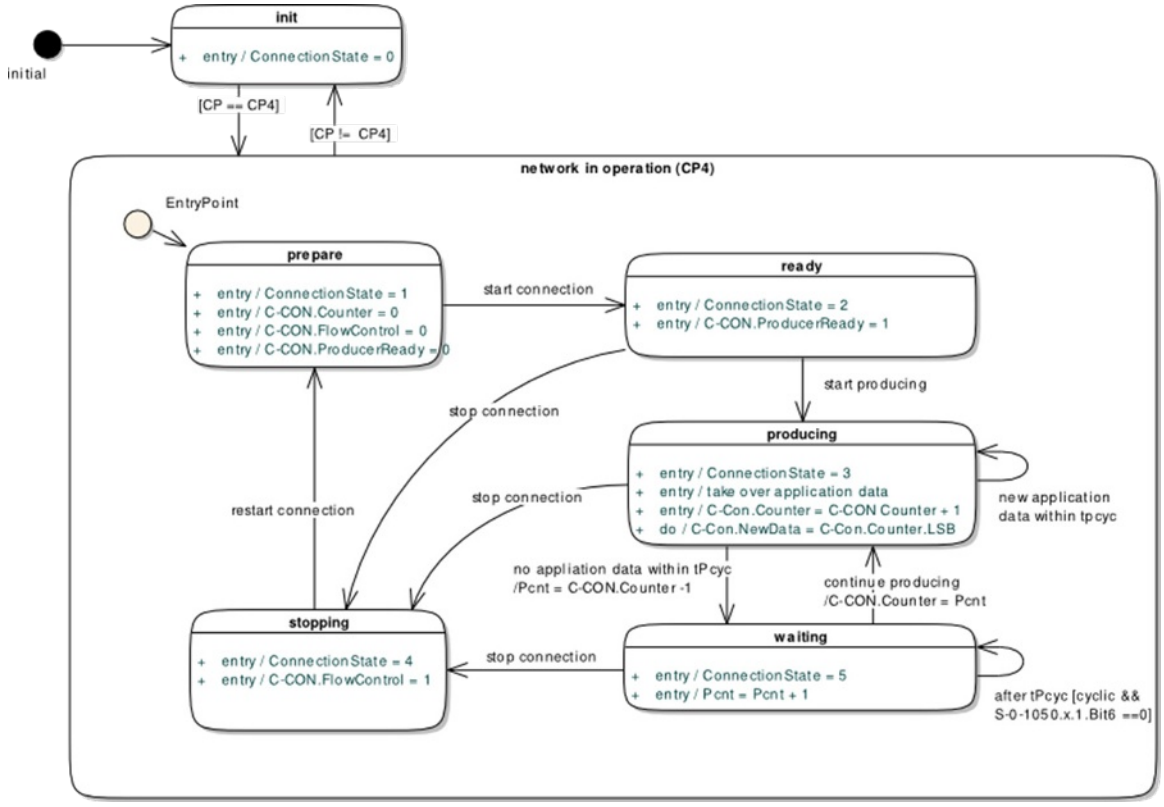

Figure 5 – Connection control state machine producer

NOTE Pcnt is an internal counter of the producer.

Table 45 shows valid bit combination's of the connection control in each producer state.

Table 45 – Connection control combinations

State	Name	Connection control (C-CON)		
		Producer Ready	Flow control	Counter / New data
0	Init	x	x	X
1	Prepare	0	0	0
2	Ready	1	0	0
3	Producing	1	0	+1
4	Stopping	1	1	no change
5	Waiting	1	0	no change

Table 46 shows the states of the producer state machine.

Table 46 – States of the producer state machine

State	Description
init	Starting the state machine, the producer shall start and stay in this state as long as the communication phase is not CP4. Entering this state, the connection state shall be set to 0. The connections are configured by the master or configurator and checked by the slaves.
Network in operation (CP4)	In this state the communication phase is CP4 and the producer handles the sub-state machine depending on the application data.

Table 47 shows the states of the producer sub-state machine.

Table 47 – States of the producer sub-state machine

State	Description
Prepare	In this state the producer prepares the connection. Entering the “prepare” state, the producer shall reset the C-CON.FlowControl, C-CON.ProducerReady and C-CON.Counter to 0. The connection state shall be set to 1. Having prepared the connection, the producer is allowed to start the connection and switch to “ready” state.
Ready	The producer is ready to transmit application data and shall set the C-CON.ProducerReady to 1. Entering this state the connection state shall be set to 2. The transition to state “producing” shall be performed by the producer dependent on the application. NOTE As fast as possible the application shall start producing independent of the state of all consumer connections.
Producing	In this state the producer shall produce valid application data and shall increase the C.CON.Counter by 1 as configured in S-0-1050.x.01. The C-CON.NewData is set to C-CON.Counter.LSB. Entering this state, the connection state shall be set to 3.
Waiting	The producer waits for new application data. The Pcnt is initialized with the C-CON.Counter – 1. Entering this state the internal producer counter Pcnt is incremented by 1 in each producer cycle as configured in S-0-1050.x.01. The connection state shall be set to 4.
Stopping	The producer stops producing application data. Entering this state the C-CON.FlowControl shall be set to 1 and the application data becomes invalid. The connection state shall be set to 5.

The producer transitions can be described as followed:

Table 48 – Producer transitions

Transition			Description
Source	Target	Condition	
Init	Network in operation (CP4)	CP == CP4	If the communication reaches CP4, then the producer shall switch from “init” to “network in operation (CP4)”. The sub-state machine "network in operation (CP4)" starts with sub-state “prepare”.
Network in operation (CP4)	Init	CP != CP4	If the communication leaves CP4, then the producer shall switch to the state “init”.
Prepare	Ready	Start connection	The producer is ready to transmit application data and switches to “ready” state autonomous.
Ready	Producing	Start producing	The producer takes over new application data.
Producing	Producing	tPcyc	In every producer cycle time (tPcyc) the producer application data and remains in state “producing”.
Producing	Waiting	No new application data	The producer didn’t receive new application data within this producer cycle and shall copy

Transition			Description
Source	Target	Condition	
			the C.CON.Counter - 1 to Pcnt. The producer shall switch to state "waiting".
Waiting	Waiting	After tPcyc	In every producer cycle time (tPcyc) and S-0-1050.x.01, bit 6 = 0 (with expectation) the producer increments the Pcnt by 1 and remains in the state “waiting”.
Waiting	Producing	Continue producing	The producer receives new application data continues producing and shall copy the Pcnt to the C-CON.Counter.
Ready	Stopping	Stop connection	The producer stops the connection and switches to state “stopping”. In order to produce application data again, the connection has to be restarted.
Producing	Stopping	Stop connection	The producer stops producing application data and switches to state “stopping”. In order to produce application again, the connection has to be restarted.
Waiting	Stopping	Stop connection	The producer stops the connection and switches to state “stopping”. The producer shall set the C-CON.FlowControl to 1. In order to produce application data again, the connection has to be restarted.
Stopping	Prepare	Restart connection	The producer restarts the connection and switches to state “prepare”.

4.7.5 Consumer state machine

The initial state of the consumer is “init” (Figure 6). In this state the consumer shall wait until the communication is switched to CP4. If the communication is in CP4, the consumer shall switch the consumer state to “network in operation (CP4)” and enter the consumer sub-state machine in state “prepare”. Further state switches of the consumer are driven by the producer. The consumer reacts on changing connection control bits (C-CON) of the producer or the communication leaves CP4. The only exception is the recovery from state “error” to “prepare”, which can be initiated by executing procedure command S-0-0099 (Reset class 1 diagnostic).

The standard procedure of consuming application data is switching from “prepare”, through “waiting” to “consuming” state. A consumer in “consuming” state can be stopped using the state “stopped”. If application data losses occur, the consumer shall change from “consuming” to “warning” state and increments the S-0-1050.x.12 Error Counter Data Losses. If errors occur, the consumer shall change from “warning” to “error” state. The consumer shall signal errors using the device status bit S-DEV.ConnError.

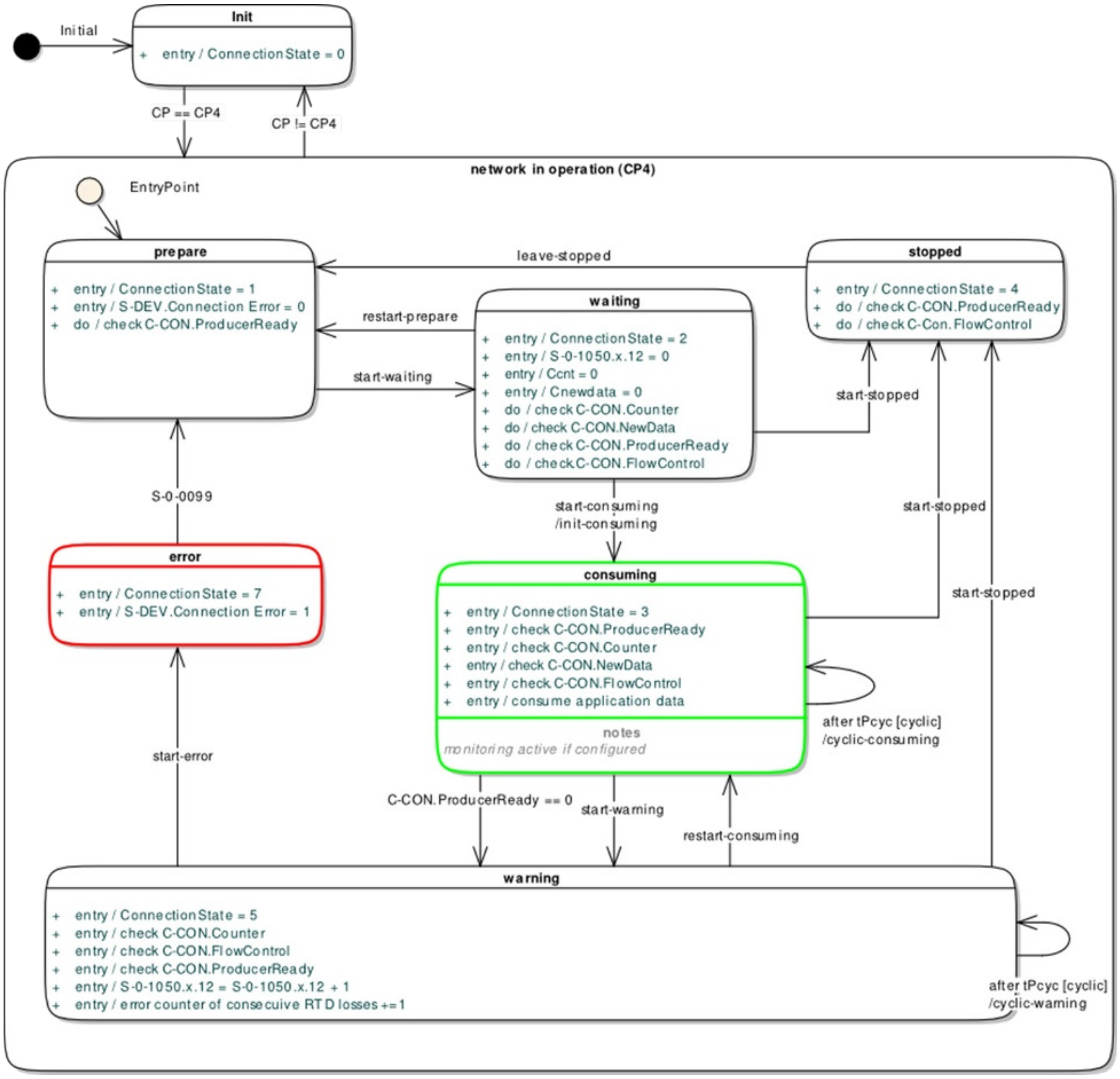

Figure 6 – Connection control state machine consumer

NOTE 1 Ccnt is an internal counter of the consumer and is used to check the C-CON.Counter.

NOTE 2 Cnewdata is an internal bit of the consumer and is used to check the C-CON.NewData.

Table 49 and Table 50 show the states of the consumer state machine and the consumer sub-state machine.

Table 49 – States of the consumer state machine

State	Description
init	Starting the state machine, the consumer shall start and stay in this state as long as the communication phase is not CP4. Entering this state, the connection state shall be set to 0. The connections are configured by the master or configurator and checked by the slaves.
network in operation (CP4)	In this state the communication phase is CP4 and the consumer reacts on information given by the producer. If the communication leaves CP4, then the consumer switches to “init”.

Table 50 – States of the consumer sub-state machine

State	Description
prepare	The consumer prepares the connection, shall set the connection state to 1 and shall reset the device status bit (S-DEV.Connection Error) to 0. The consumer shall check the C-CON.Producer-ready.
waiting	Entering this state, the connection state shall be set to 2, the S-0-1050.x.12 (Error Counter Data Losses) and the consumer internal Cnewdata bit resp. counter Ccnt shall be set to 0. Within this state, the consumer shall check the connection control (C-CON) defined as follows: - The consumer shall check the C-CON.Counter and C-CON.NewData to determine if the C-CON.Counter is supported by the producer. If this is the case, then the consumer shall use the C-CON.Counter for the consumer state machine only. - The consumer shall check the C-CON.ProducerReady. - The consumer shall check the C-CON.FlowControl.
consuming	In state "consuming", the connection state shall be set to 3 and the application data of the producer shall be consumed by the consumer. In addition, the consumer shall check the connection control (C.CON) defined as follows: - If the producer cancels the producing and sets the C-CON.ProducerReady = 0, then the application data becomes invalid. The consumer shall not consume the application data. - The consumer shall check the C-CON.FlowControl. - If the corresponding expectation is not fulfilled of the configured method, then the consumer recognizes losses of application data in this producer cycle and shall not consume the application data. - For non-synchronous consumer with watch-dog: If the consumer evaluates the C-CON.Counter in a frequency of its modulo (16 times), then the consumer recognizes losses of application data. - If the corresponding expectation is fulfilled of the configured method in S-0-1050.x.01, bit 6, then the consumer shall consume the application data. - The internal "error counter of consecutive data losses" shall be set to 0, if a change of C-CON.NewData or C-CON.Con-counter is detected.
warning	In state "warning" the connection state shall be set to 5. In addition, the consumer shall check the connection control (C.CON) defined as follows: - The consumer shall check C-CON.FlowControl. - The consumer shall increment the S-0-1050.x.12 Error Counter Data Losses by 1 in each producer cycle (tPcyc). - The consumer shall increment the internal "error counter of consecutive data losses" by 1 in each producer cycle (tPcyc). - If the C-CON.ProducerReady = 0, then the consumer interrupts the consuming and waits until the C-CON.ProducerReady is set to 1 again by the producer. - The consumer shall check if the internal "error counter of consecutive data losses" exceeds S-0-1050.x.11 Allowed Data Losses. - The consumer shall check if the internal "error counter of consecutive data losses" didn't exceed S-0-1050.x.11 Allowed Data Losses and the application data are valid again (C-CON.ProducerReady = 1).
stopped	In this state the connection state shall be set to 4 and the consumer shall wait for a connection restart. The consumer shall check the C-CON.ProducerReady and the C-CON.FlowControl.
error	In this state the connection state shall be set to 7 and the consumer shall set the status device bit S-DEV.ConnError to 1 and set the S-0-0390 Diagnostic number to 0xC30F4002. The consumer shall check if S-0-0099 (Reset class 1 diagnostic) is activated.

Table 51 shows the consumer transitions.

Table 51 – Consumer transitions

Transition			Description
Source	Target	Condition	
init	network in operation (CP4) state = prepare	CP == CP4	If the communication reaches CP4, then the consumer shall switch to state “network in operation (CP4)”. The sub-state machine starts with sub-state “prepare”.
network in operation (CP4)	init	CP != CP4	If the communication leaves CP4, then the consumer shall switch to state “init”.
prepare	waiting	start-waiting C-CON.ProducerReady == 1	If the C-CON.ProducerReady is set to 1, then the consumer shall switch to state “waiting”.
waiting	prepare	restart-prepare C-CON.ProducerReady == 0	If the C-CON.ProducerReady is set to 0 on both ports (P1 and P2), then the consumer shall switch to state “prepare”. All other bits in the C-CON are don't care.
waiting	stopped	start-stopped C-CON.ProducerReady == 1 && C-CON.FlowControl == 1	The producer stops the connection by setting the C-CON.FlowControl to 1 and the C-CON.ProducerReady is 1, then the consumer shall switch to state “stopped”.
warning	stopped	start-stopped C-CON.ProducerReady == 1 && C-CON.FlowControl == 1	The producer stops the connection by setting the C.CON.FlowControl to 1. In this case the consumer shall switch to state “stopped”.
consuming	stopped	start-stopped C-CON.ProducerReady == 1 && C-CON.FlowControl == 1	The producer stops the connection by setting the C.CON.FlowControl to 1. In this case the consumer shall switch to state “stopped”.
stopped	prepare	leave-stopped C-CON.ProducerReady == 0 or C-CON.FlowControl == 0	The producer restarts the connection by setting the C.CON.ProducerReady to 0. In this case the consumer shall switch to state “prepare”.
consuming	warning	C-CON.ProducerReady == 0 (shall be fulfilled on both ports)	If the C-CON.ProducerReady is set to 0 on both ports (P1 and P2) during consuming, then the consumer doesn’t receive valid application data anymore and shall switch to state "warning". All other bits in the C-CON are don't care.
warning	error	start-error ("error counter of consecutive data losses") > S-0-1050.x.11 Allowed Data Losses	If the C-CON.ProducerReady is 1 and C-CON.FlowControl is 0 and the internal "error counter of consecutive data losses" exceeds S-0-1050.x.11 Allowed Data Losses, then the consumer shall switch to state “error”.
error	prepare	S-0-0099 Reset class 1 diagnostic	The error is reset with executing S-0-0099 Reset class 1 diagnostic. The consumer shall set the status device bit S-DEV.ConnError to 0 and shall switch to state “prepare”.
waiting	consuming	start-consuming (init consuming) C-CON.ProducerReady==1 && C-CON.FlowControl==0	C-CON.NewData!=Cnewdata: If the C-CON.NewData ≠ Cnewdata, then the consumer shall initialize the internal bit Cnewdata = C-CON.NewData and switch to state “consuming”. C-CON.Counter!=Ccnt: If the C-CON.Counter ≠ Ccnt, then the consumer shall initialize the internal Ccnt = C-CON.Counter and switch to state “consuming”.
warning	consuming	C.CON.Counter != CNT && C.CON.ProducerReady == 1	The producer produces valid application data again and the amount of consecutive RTD losses didn’t exceed the S-0-1050.x.11 (Allowed Data Losses), then the

Transition			Description
Source	Target	Condition	
			consumer switches to state "consuming".
consuming	consuming	after tPcyc cyclic consuming C-CON.ProducerReady == 1 && C-CON.FlowControl == 0	If C-CON.ProducerReady is 1 and C-CON.FlowControl is 0, then the consumer consumes application data in every producer cycle time (tPcyc) and remains in the state "consuming".
consuming	warning	start-warning C-CON.ProducerReady == 1 && C-CON.FlowControl == 0	If the check of C-CON.NewData was erroneous, then the producer doesn't increase the C-CON.NewData correctly. In this case the application data are invalid and the consumer shall switch to state "warning". If the check of C-CON.Counter was erroneous, then the producer doesn't increase the C-CON.Counter correctly. In this case the application data are invalid and the consumer shall switch to state "warning".
warning	warning	after tPcyc cyclic warning C-CON.ProducerReady == 0	If C-CON.ProducerReady is 0, then the consumer shall increment the S-0-1050.x.12 Error Counter Data Losses and the internal "error counter of consecutive data losses" by 1 in every producer cycle time (tPcyc). The consumer remains in the state "warning".
warning	consuming	restart-consuming C-CON.ProducerReady == 1 && C-CON.FlowControl == 0	If the producer produces application data again (C-CON.ProducerReady is 1 and C-CON.FlowControl is 0) and the internal "error counter of consecutive data losses" didn't exceed the S-0-1050.x.11 Allowed Data Losses, then the consumer switches to state "consuming". The following condition shall correspond to the expectation.

5 DL management

5.1 Overview

DL-management procedures are functionally processed in response to DL-management service requests submitted by the DL-user and events caused by the network.

5.2 Initialization of cyclic communication

5.2.1 Introduction

Upon an Initiate_cyclic_communication (ICC) request by the DL user in the master device, the so-called phase upshift is initiated.

A Notify_cyclic_communication (NCC) indication is generated for the DL user in the slave device if the phase upshift has been successfully completed.

Upon a Disable_cyclic_communication (DCC) request by the DL user in the master device the so-called phase downshift is initiated.

A Notify_cyclic_communication_disabled (NCCD) indication is generated for the DL user in the slave device if the cyclic communication has been disabled.

A Notify_error (NER) indication is generated for the DL user in a master and a slave device if an error has occurred in the cyclic communication.

5.2.2 Communication phases (CP)

5.2.2.1 General

Initialization shall be divided into five communication phases and NRT state:

a) After a station has been powered up, and internal checks are completed and error-free, it shall operate in Non-Real-Time (NRT) state (see 5.2.2.2.2).
b) initialization of a Type 19 network shall always beginning with CP0;
c) CP0 shall be used for recognizing the participating slaves;
d) CP1 shall be used to configure the slave devices for non-cyclic communication;
e) CP2 shall be used to configure the slave devices for cyclic communication and for parameter setting in the slave via non-cyclic communication;
f) CP3 shall be used to further configure the slave devices, the cyclic communication shall already be running but shall not be used;
g) in CP4 the initialization process is complete and the Type 19 network shall be in operation.

It shall also be possible to enter CP0 from any higher phase. It shall not be possible to enter other phases except when leaving the previous one in ascending order.

The master shall initiate a specific CP by setting the MDT phase in the Type 19 DLPDUs (see 4.5.4). The slaves shall follow accordingly. Only in the case of a communication error, the slaves shall switch to NRT state.

If a slave is connected to an already operational network and receives a MST that indicates CP4, then it shall enter the hot-plug procedure (see 5.5) if it is supported. Otherwise, the slave remains in NRT state.

The communication phase state machine is shown on the left side of Figure 7.

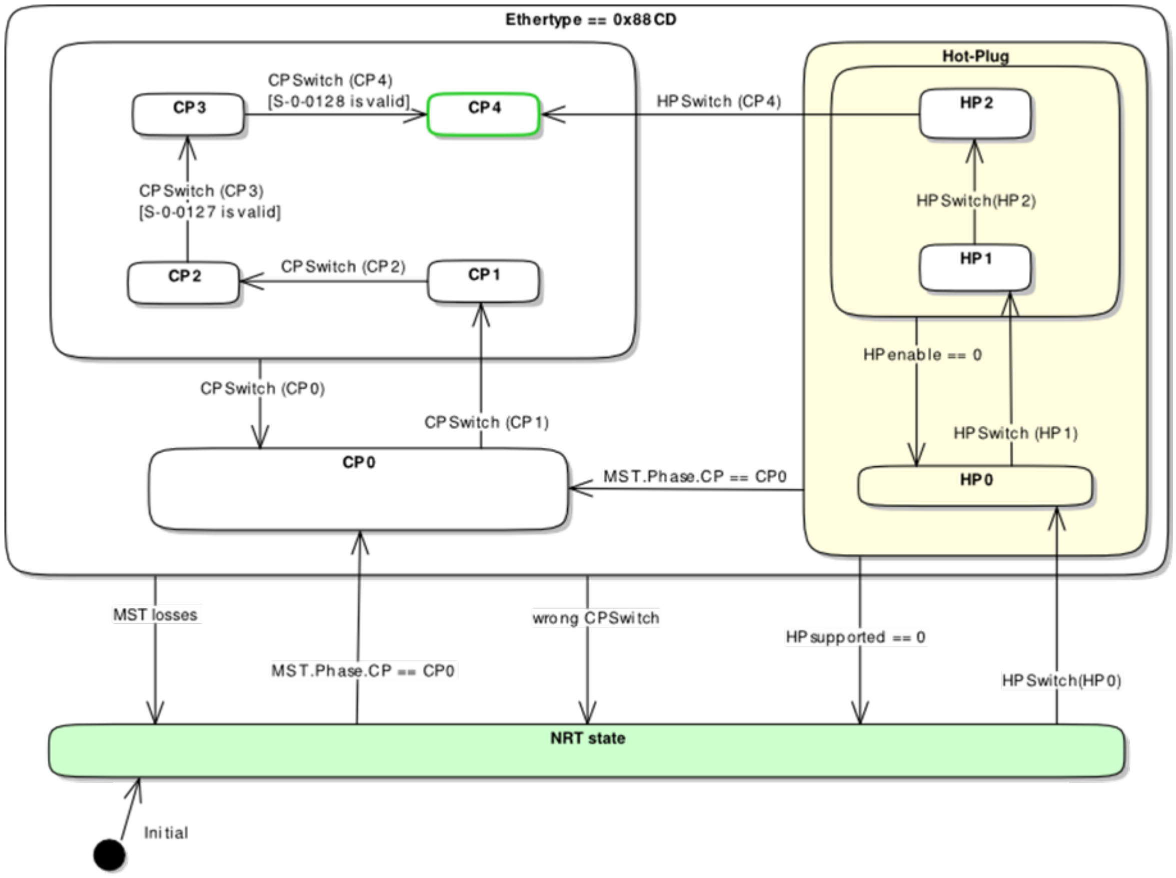

Figure 7 – Communication phase (CP) state machine

5.2.2.2 States of the CP state machine

5.2.2.2.1 General

The states of the CP state machine are described in Clauses 5.2.2.2.2 (NRT state), 5.2.2.2.3 (CP0), 5.2.2.2.4 (CP1), 5.2.2.2.5 (CP2), 5.2.2.2.6 (CP3) and 5.2.2.2.7 (CP4). The LED pattern shall be adjusted to the corresponding communication states.

5.2.2.2.2 Non-real-time state (NRT)

Upon powering on, the master and each slave shall activate NRT state independently.

NRT mode is activated during NRT state.

The collision buffer shall be administrated as described in the topology state machine.

The master shall leave NRT state to CP0 upon request by its DL user.

The slave shall check the MST of the type 19 telegrams. If the slave recognizes a MST with CP = 0, then it shall activate CP0 and change the topology. If a slave recognizes a MST with CP = 4, then it shall activate HP0, if hot-plug is supported.

Before entering NRT state from a state different to CP0 or HP0 the slave shall generate a link down on both ports.

In NRT state the slave may activate pattern #1 of Type 19 LED.

5.2.2.2.3 Communication phase 0 (CP0)

During CP0, the master shall send MDT0 and AT0 as specified on one or both of its ports, depending on the given topology, in order to

- check the topology (for example check if the network is established);
- check if all slaves required by the application are present in the network.

The slave shall

- check the communication version;
- support the address allocation;
- change the topology state depending on the received MST automatically at time T7cp0 between
  - NRT state (store&forward or cut-through),
  - RT state (loopback with forward and fast-forward).

5.2.2.2.4 Communication phase 1 (CP1)

The topology indices which are determined in CP0 are used for the addressing of the service channels in CP1.

In CP1, the master shall initialize the service channels of all identified slaves of CP0.

During the request procedure the master shall send MDT0 and MDT1 (additional MDT2 and MDT3, if more than 255 slaves are identified) and set MHS=1 in the SVC control to request each used slave to insert data into the corresponding AT. AT0 and AT1 (additional AT2 and AT3, if more than 255 slaves are identified) shall be transmitted by the master with an allocated AT data field (contents are 0).

In the device control, the master shall adjust the following bits and all identified slaves shall evaluate them:

- C-DEV.identification
- C-DEV.topology
- C-DEV.status physical topology, only if UC channel is supported

If the C.DEV.Master valid = 1 in CP1, then the slaves shall evaluate the C-DEV in CP2 to CP4 only if C-DEV.Master valid is set to 1. If the C-DEV.Master valid = 0 in CP1, then this bit is not supported by the master and the slave shall evaluate the C-DEV in CP2 to CP4 also if the C-DEV.Master valid is set to 0.

During the request procedure the following sequence shall be handled by the master and all identified slaves.

a) At first the slave shall set Slave valid = 1 in the device status to indicate the availability in the topology.
b) If the slave is ready to communicate via SVC, then it shall set SVC valid = 1 into the corresponding SVC status.
c) Thereafter, the master shall set the MHS = 1 in the corresponding SVC control.
d) If a slave responds with AHS=1 in the SVC status within the handshake timeout (10 communication cycles), then the SVC is correct initialized.
e) If a slave does not respond with AHS=1 in the SVC status within the handshake timeout, then the master shall generate an error message and switches to CP0.

In the device status, the slave shall adjust the following bits and the master shall evaluate them, if S-DEV.Slave valid = 1:

- S-DEV.Slave valid=1
- S-DEV.topology
- S-DEV.parameterization level
- S-DEV.communication warning interface
- S-DEV.C1D error of device
- S-DEV.C2D warning of device

All identified slaves shall behave as described here, even those with Sub-device (S-0-1040) address = 0.

5.2.2.2.5 Communication phase 2 (CP2)

During CP2, the slaves shall be addressed specifically using their corresponding service channel. For CP2 and higher phases, they shall support complete service channel functionality.

As a minimum, the master shall transmit to all present slaves:

- the communication parameters required for CP3 and CP4;
- the length of all MDTs and ATs;
- the offsets of their service channel and real-time data.

In the device control (C-DEV), the master shall adjust the following bits and all present slaves shall evaluate them, if C-DEV.Master valid is equal to CP1:

- C-DEV.Master valid = C-DEV.Master valid of CP1
- C-DEV.Identification
- C-DEV.Topology
- C-DEV.Status physical topology, only if UC channel is supported

In the device status, the slave shall adjust the following bits and the master shall evaluate them, if Slave valid = 1:

- S-DEV.Slave valid=1
- S-DEV.Topology
- S-DEV.Parameterization level
- S-DEV.Communication warning interface
- S-DEV.C1D error in device
- S-DEV.C2D warning in device
- S-DEV.procedure command change bit

The entire information exchange takes place via the mechanisms of the service channel (see 6.2). The reliability of transmission shall be guaranteed by the MHS and AHS bits as well as the HS timeout. Further parameter exchanges can take place in CP2 or CP3.

The master shall transmit the ring delays and activate S-0-1024 SYNC delay measuring procedure command of all slaves, which shall be synchronized. The slaves shall adjust the synchronization time depending on the ring delays.

If the master sets t6 = 0 the UC channel is deactivated in CP3 and CP4, and the time t7 is don't care.

The transition from CP2 to CP3 shall be prepared according to the following procedure:

a) The master shall activate the procedure command S-0-0127 CP3 transition check as defined.
b) The slave shall then determine the validity of the parameters for CP3. The validity check of the parameters by the slave shall refer only to general criteria (e.g., minimum, maximum). It shall not recognize if all parameters that have been transmitted by the master are correct with respect to the master real-time data and the total installation. This means that even if a slave acknowledges the "CP3 transition check" positively, there can be incorrect communication parameters with respect to the total installation which can lead to a disruption of the communication.
c) If the slave detects any error, then it shall continue the process with the "Procedure with error". If there are additional invalid parameters still present after the procedure command has been processed, the slave shall
   - save the IDNs of the invalid data into the S-0-0021 IDN-list of invalid operation data for CP2 and shall
   - respond with the procedure command acknowledgment: "Error, procedure command execution impossible".

   After the negative procedure command acknowledgment,

   - the master may read the diagnostic parameters (e.g. S-0-0021, S-0-0390 Diagnostic number ...) and display an error message.
   - Before the master activates the S-0-0127 CP3 transition check again, it shall delete this procedure command in the slave.
   - In this faulty case, the master shall remain in CP2 and depending on its capabilities, try again to set the parameters identified as invalid or send an error message to allow further initialization by means of an operator intervention.

   After the master has written further parameters which are write protected in CP3 according to their attribute, the procedure command S-0-0127 CP3 transition check shall be activated once more.

d) If the slave does not detect any error, then it shall continue the process with the "Procedure without error".
   - The slave shall acknowledge the procedure command positively (e.g., "Procedure command executed correctly") and set CPS ready = 1.
   - After receiving the positive procedure command acknowledgment, the master shall delete the procedure command in the slave.
   - At parameters with variable length which can be configured also as application data, the current length indicator shall not be changed any more.
   - If the master has written further parameter which are write protected in CP3, after the positive procedure command acknowledgment, then the slave shall set CPS ready = 0 and the master shall repeat the procedure command S-0-0127 CP3 transition check again.
   - Otherwise, the master may then initiate the switching to CP3 (see 5.2.3).

5.2.2.2.6 Communication phase 3 (CP3)

Starting with CP3, the exchange of real-time data shall be done via the telegrams defined for CP4. The master shall send the configured MDTs and ATs to all slaves.

To support the IP switch, the master shall transmit the time slot parameters of the UC channel with broadcast address in the MDT HP field. Directly after the start of CP3 and immediately after ring recovery the master shall transmit t6 and t7 with the defined scan cycles of each parameter. The contents of the MDT HP field are shown in Table 52.

Table 52 – MDT hot-plug field in CP3 and after ring recovery

Sub-device address	HP control	MDT HP INFO	Scan cycles of each parameter
4095 (broadcast)	0x02 (coding of t6)	value of t6 (beginning of UC channel)	tScyc ≤ 2ms --> scan cycles = (2ms/tScyc) * 10 cycles tScyc > 2ms --> scan cycles = 10 cycles
4095 (broadcast)	0x03 (coding of t7)	value of t7 (end of UC channel)	tScyc ≤ 2ms --> scan cycles = (2ms/tScyc) * 10 cycles tScyc > 2ms --> scan cycles = 10 cycles

During CP3, the master may transmit application parameters (for example: Limits, thresholds, machinery parameter, etc.) for the slaves via the service channel. Transmission reliability for the service channel shall be guaranteed by the SVC control, SVC status and the HS timeout.

In CP3, the function specific profile may be activated.

The transition from CP3 to CP4 shall be performed according to the following procedure:

a) The master shall activate the procedure command S-0-0128 CP4 transition check as defined.
b) The slave shall then determine the validity of the parameters for CP4.
c) Afterwards, the slave shall complete the processing of the parameters that are required for operating the slave.
d) The slave shall then activate the synchronization.
e) If the slave detects any error, then it shall continue the process with the “Procedure with error”. If there are additional invalid parameters still present after the procedure command has been processed, the slave shall
    - save the IDNs of the invalid data into the S-0-0022 IDN-list of invalid operation data for CP3 and shall
    - respond with the procedure command acknowledgment: “Error, procedure command execution impossible”.

    After the negative procedure command acknowledgment,

    - the master may read the diagnostic parameters (e.g. S-0-0022, S-0-0390 Diagnostic number ...) and display an error message.
    - Before the master activates the S-0-0128 CP4 transition check again, it shall delete this procedure command in the slave.
    - In this faulty case, the master shall remain in CP3 and, depending on the capabilities of the master, try to re-establish the parameters identified as invalid or send an error message indicating that human intervention (e.g., operator) is required.

    After the master has written further parameters (depending on S-0-0022 IDN-list of invalid operation data for CP3) in the slave during CP3, the procedure command S-0-0128 CP4 transition check shall be activated once more.
f) If the slave does not detect any error, then it shall continue the process with the “Procedure without error”.

- The slave shall acknowledge the procedure command positively (e.g., “procedure command executed correctly”) and set CPS ready = 1.
- After receiving the positive procedure command acknowledgment, the master shall delete the procedure command in the slave.
- If the master has written further parameter which are write protected in CP4, after the positive procedure command acknowledgment, then the slave shall set CPS ready = 0 and the master shall repeat the procedure command S-0-0128 CP4 transition check again.
- Otherwise, the master may then initiate the switching to CP4 (see 5.2.3).

5.2.2.2.7 Communication phase 4 (CP4)

Upon switching to CP4, the initialization is complete. In CP4 the master may enable all slaves and the application is ready to operate. The master shall send the configured MDTs and ATs to all slaves. The exchange of valid real-time data shall be done via the prepared connections defined for CP4.

The hot-plug function is activated in the master, if supported.

Transmission reliability for the service channel shall be guaranteed by the SVC control, SVC status and the HS timeout.

5.2.2.3 Transitions of CP state machine

Table 53 describes the transitions of the CP state machine.

Table 53 – Transitions of CP state machine

Transition		Condition	Description
Source	Target		
NRT state	CP0	MST.Phase.CP=CP0	The master shall leave NRT state to CP0 upon request by its DL user. In this case the master shall transmit MST with CP0. If the slave receives a MST (MDT0) with CP0 while it is in NRT state the slave shall activate CP0 as well as the Loopback with forward at the port at which the slave has received the MST.
Ethertype = 0x88CD	NRT state	MST losses	During CP0, CP1 or CP2: If a slave does not receive any MST in CP0, CP1 or CP2 within the MST timeout (130ms), then it shall switch to NRT state. During CP3 or CP4: If the maximum number of MST losses (communication error) exceeds the S-0-1003 Allowed MST losses in CP3&CP4, then the slave switches to NRT state. Before entering NRT state from a state different to CP0 or HP0 the slave shall generate a link down on both ports.
Ethertype = 0x88CD	NRT state	wrong CPSwitch	If an error occurs during phase switching, then - the slave switches to NRT state. - the master switches to CP0. Detailed description see 5.2.3. Before entering NRT state from a state different to CP0 or HP0 the slave shall generate a link down on both ports.
CP0	CP1	CPSwitch(CP1)	As soon as the master has received at least 100 AT0 with the same content of sequence counter and the number of modified topology index fields corresponds to the SEQCNT according to the

Transition		Condition	Description
Source	Target		
			given topology and all recommended slaves are present, the master may initiate to switch to CP1. The address allocation shall be canceled in the slave if it recognizes MST.Phase.CPS = 1.
CP1	CP2	CPSwitch(CP2)	After the master has initialized the service channel of all slaves on the Type 19 network, the master shall switch to CP2 (see 5.2.3). If the initialization time of the service channels is exceeded, then master may respond with an error message depending upon configuration and switch to CP0.
CP2	CP3	CPSwitch(CP3)	The transition from CP2 to CP3 shall be performed only if the checks of S-0-0127 CP3 transition check have passed successfully.
CP3	CP4	CPSwitch(CP4)	The transition from CP3 to CP4 shall be performed only if the checks of S-0-0128 CP4 transition check have passed successfully.
CP1-CP4	CP0	CPSwitch(CP0)	The only possibility of leaving the Communication phases CP1 to CP4 (other than up-shift phases) shall be a return to CP0. The reason for this can be operator intervention. Any slave which recognizes CP0 shall shut-down itself in the best possible manner. The method of shutting down the slaves is part of the function specific profiles.

NOTE Details about the CPSwitch(CPx) are described in the corresponding state machine.

5.2.3 Switching of communication phases (CPS)

5.2.3.1 Sequence of CP switching in the master

Figure 8 describes the CPS state machine of a master. The LED pattern shall be adjusted to the corresponding topology state.

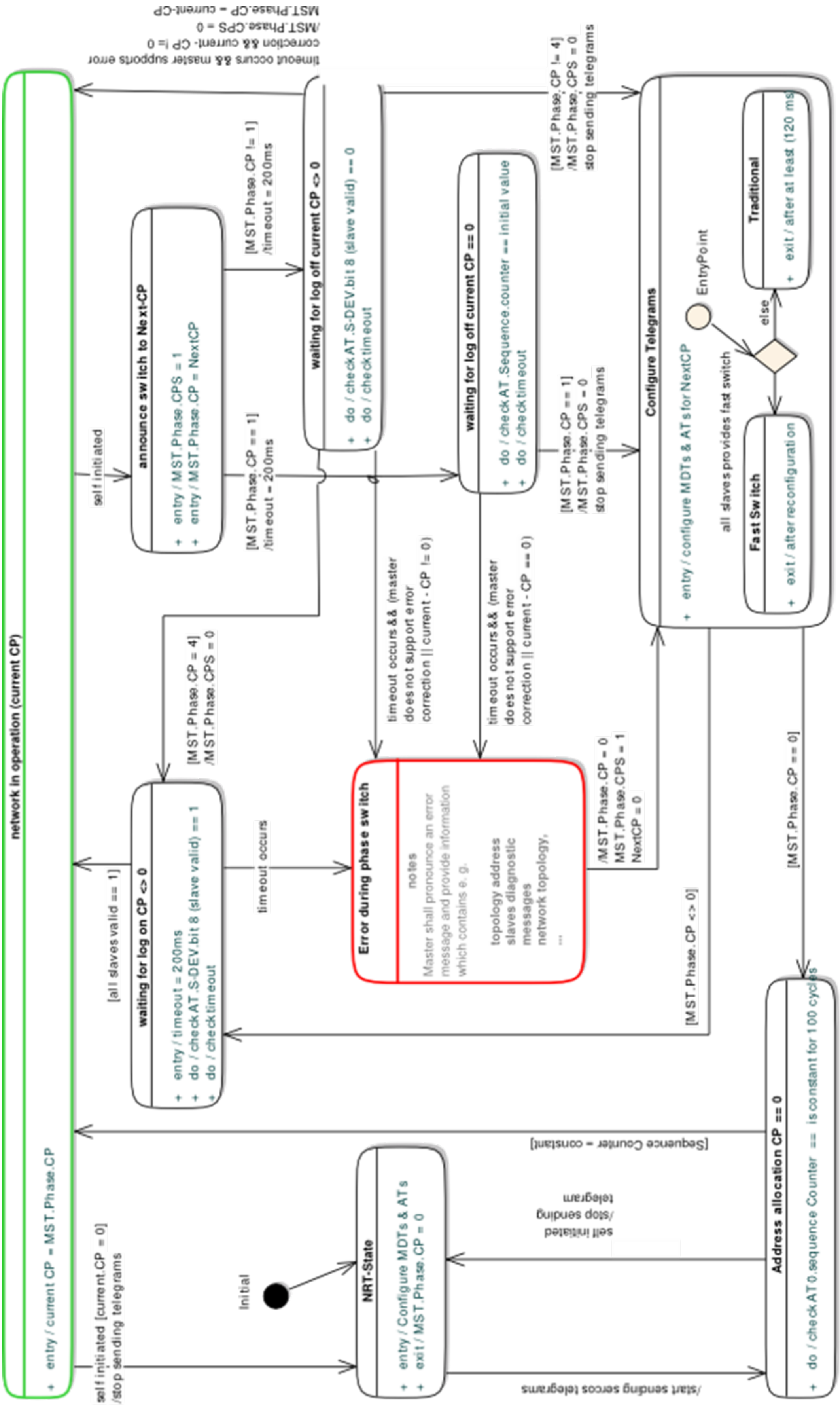

Figure 8 – CPSwitch state machine master

NOTES

- current CP = at the time active CP
- next CP = CP0 or current CP or current CP + 1 (only valid, if current CP < 4)
- CPS master timeout = 200 ms
- CPS delay time = 120 ms
- SEQCNT = sequence counter in AT0 of CP0

5.2.3.2 States of CPSwitch state machine of the master

The states of the state machine are described in Table 54.

Table 54 – States of master CPSwitch state machine

State	Description
NRT state	Upon powering on, the master and each slave shall activate NRT state independently. NRT mode is activated during NRT state. The collision buffer shall be administrated as described in the topology state machine. The master shall leave NRT state to CP0 upon request by its DL user. The slave shall check the MST of the Type 19 telegrams. If the slave recognizes a • MST with CP = 0, then it shall activate CP0 and change the topology (see 5.3.6) • MST with CP = 4, then it shall activate HP0, if hot-plug is supported. Before entering NRT state from a state different to CP0 or HP0 the slave shall generate a link down on both ports. In NRT state the slave may activate pattern #1 of Type 19 LED.
Address allocation CP == 0	The master shall set the CPS master timeout (200 ms) and waits until the received SEQCNT of AT0 is constant. The master shall check the received sequence counter (SEQCNT) of AT0. If the sequence counter is not changed within 100 successive communication cycles, then the master may switch to "network in operation".
waiting for log on CP ≠ 0	The master shall set the CPS master timeout (200 ms) and waits until all slave valid bits are set to 1. In CP1 to CP4 the master shall check the slave valid bit. If the slave valid bit is set to 1 by each slave, then the master switches to "network in operation".
Network in operation (current CP)	The master transmit MDTs and ATs with the timing of the current CP.
Announce switch to next CP	In order to switch the communication phase, the master shall set the condition in the MST (CPS = 1 and next CP). The master shall determine the next CP, see notes.
Waiting for log off current CP == 0	The master shall set the CPS Master timeout (200 ms) and waits until the received SEQCNT is equal to the transmitted SEQCNT (initial value of transmitted SEQCNT = 1). In CP0 (MST.Phase.CPS = 1 && MST.Phase.CP = 1) the master checks the received sequence counter of AT0. If the sequence counter is not changed by any slave, then the master switches to "Configure telegrams".
Waiting for log off current CP ≠ 0	The master shall set the CPS Master timeout (200 ms) and waits until the slaves do not set the slave valid to 1. As long as any slave sets the slave valid to 1 the master remains in this state and shall transmit MDTs and ATs. In CP1 to CP4 (MST.Phase.CP ≠ 1 && MST.Phase.CPS = 1) the master shall check the slave valid bit. If the slave valid bit is set to 0 by each slave, then the master switches to "Configure telegrams".
Configure telegrams	The master shall stop the transmission of MDTs and ATs. The master shall configure the MDTs, ATs and the timing for the next CP. If all slaves on the network supports the CPS switching without CPS delay time, then the master activates the state "Fast switch". Otherwise the master shall activate the state "Traditional".
Fast switch	The master switches to the announced state immediately (CPS delay time = 0ms).
Traditional	The master waits the CPS delay time (120ms) and then it switches to the announced state.
Error during phase switch	The master shall generate an error message, which may contain one of the following items e.g. • device address • topology index • slave diagnostics • network diagnostics • and switches to CP0.

5.2.3.3 Transitions of CPSwitch state machine of the master

The transitions of the state machine are described in Table 55.

Table 55 – Transitions of master CPSwitch state machine

Transition			Description
Source	**Target**	**Condition**	
NRT state	Address allocation CP == 0	Start sending Type 19 telegrams	The master leaves the NRT state and switches to CP0 by sending MST with CP0.
Network in Operation	NRT state	self initiated current CP = 0 stop sending telegrams	The master shall only switch from CP0 to NRT state by transmitting no more Type 19 telegrams.
Address allocation CP == 0	Network in operation	SEQCNT = constant	If the sequence counter is constant for 100 cycles, then the next CP (CP1) maybe activated by the master and the address allocation is finished.
Address allocation CP == 0	NRT state	self initiated stop sending telegrams	If the master does not recognize a constant SEQCNT within 100 cycles, then the check is repeated up to 10 times. If no constant result is achieved during this time, then the Master generates an error message and switches to NRT state.
Network in operation	Announce switch to next CP	Self initiated	If the master would like to change the communication phase, then it activates the state "Announce switch to next CP". The master shall set CPS = 1 and the MST.Phase.CP to next CP (see notes).
Announce switch to next CP	Waiting for log off, current CP = 0	MST.Phase.CP = 1 CPS master timeout = 200ms	After the master has announced the next CP with CP1, the master sets the CPS master timeout and activates the state "Waiting for log off, current CP = 0".
Announce switch to next CP	Waiting for log off, current CP ≠ 0	MST.Phase.CP ≠ 1 CPS master timeout = 200ms	After the master has announced the next CP with CP ≠ 1, the master sets the CPS master timeout and activates the state "Waiting for log off, current CP ≠ 0".
Waiting for log off, current CP = 0	Configure telegrams	MST.Phase.CP = 1, CPS = 0 Stop sending telegrams	If the sequence counter is not changed by any slave in CP0, the master stops the transmission of MDTs and ATs, activates the state "Configure telegrams" and prepares the next CP.
Waiting for log off, current CP ≠ 0	Configure telegrams	MST.Phase.CP ≠ 4, CPS = 0 Stop sending telegrams	If slave valid = 0 of all slaves in CP1 to CP3, the master stops the transmission of MDTs and ATs, activates the state "Configure telegrams" and prepares the next CP.
Waiting for log off, current CP ≠ 0	waiting for log on, CP ≠ 0	MST.Phase.CP = 4, CPS = 0	If switching from CP3 to CP4 the master does not stop transmitting MDTs and ATs because the structure of the Type 19 telegrams and the timing are identical.
Configure telegrams	Waiting for log on, current CP ≠ 0	MST.Phase.CP ≠ 0	After the Type 19 telegrams are configured for the next CP (CP1 to CP3) and the CPS delay time occurs the master shall transmit the Type 19 telegrams with the structure and timing of the next CP.
Configure telegrams	Address allocation CP = 0	MST.Phase.CP = 0	After the Type 19 telegrams are configured for CP0 and the CPS delay time occurs the master shall transmit the Type 19 telegrams with the structure and timing of CP0.
Waiting for log on, current CP ≠ 0	Network in operation	slave valid = 1 of all slaves	If slave valid = 1 of all slaves, then the next CP (CP0, CP2 to CP4) maybe activate by the master.
Waiting for log off, current CP ≠ 0	network in operation	CPS master timeout occurs and master supports error correction	If the current CP ≠ 0, the master activates the current CP again by setting the MST.Phase.CP = current CP and CPS = 0. Now the master can check why not have one or more slaves logged off. If the cause of the error can be fixed, the phase switching shall continue, if it cannot be fixed, the master shall

Transition			Description
Source	Target	Condition	
			switch to CP0.
Errors during phase switching			
waiting for log on, CP ≠ 0	Error during phase switch	CPS master timeout occurs	After the CPS master timeout is exceeded in CP1 to CP4 and the master did not receive the slave valid = 1 of each slave, then the master shall produce an error message showing e.g. the respective device addresses and topology indices. After deleting the error by the control unit, the master shall switch to CP0.
Waiting for log off, current CP = 0	Error during phase switch	CPS master timeout occurs and master does not support error correction	After the CPS master timeout is exceeded in CP0 and the master still receives a changed SEQCNT in AT0, then the master shall produce an error message showing e.g. the respective device addresses and topology indices. After deleting the error by the control unit, the master shall switch to CP0.
Waiting for log off, current CP ≠ 0	Error during phase switch	CPS master timeout occurs and master does not support error correction	After the CPS master timeout is exceeded in CP1 to CP4 and the master still receives slave valid = 1 of one or more slaves, then the master shall produce an error message showing e.g. the respective device addresses and topology indices. After deleting the error by the control unit, the master shall switch to CP0.
Error during phase switch	Configure telegrams	Next CP = 0	The master leaves the error state by setting the MST.Phase.CP = 0 and MST.Phase.CPS = 1. The master switches to CP0 via "configure telegrams".

5.2.3.4 Sequence of CP switching in the slave

Figure 9 shows the CPSwitch state machine of the slave.

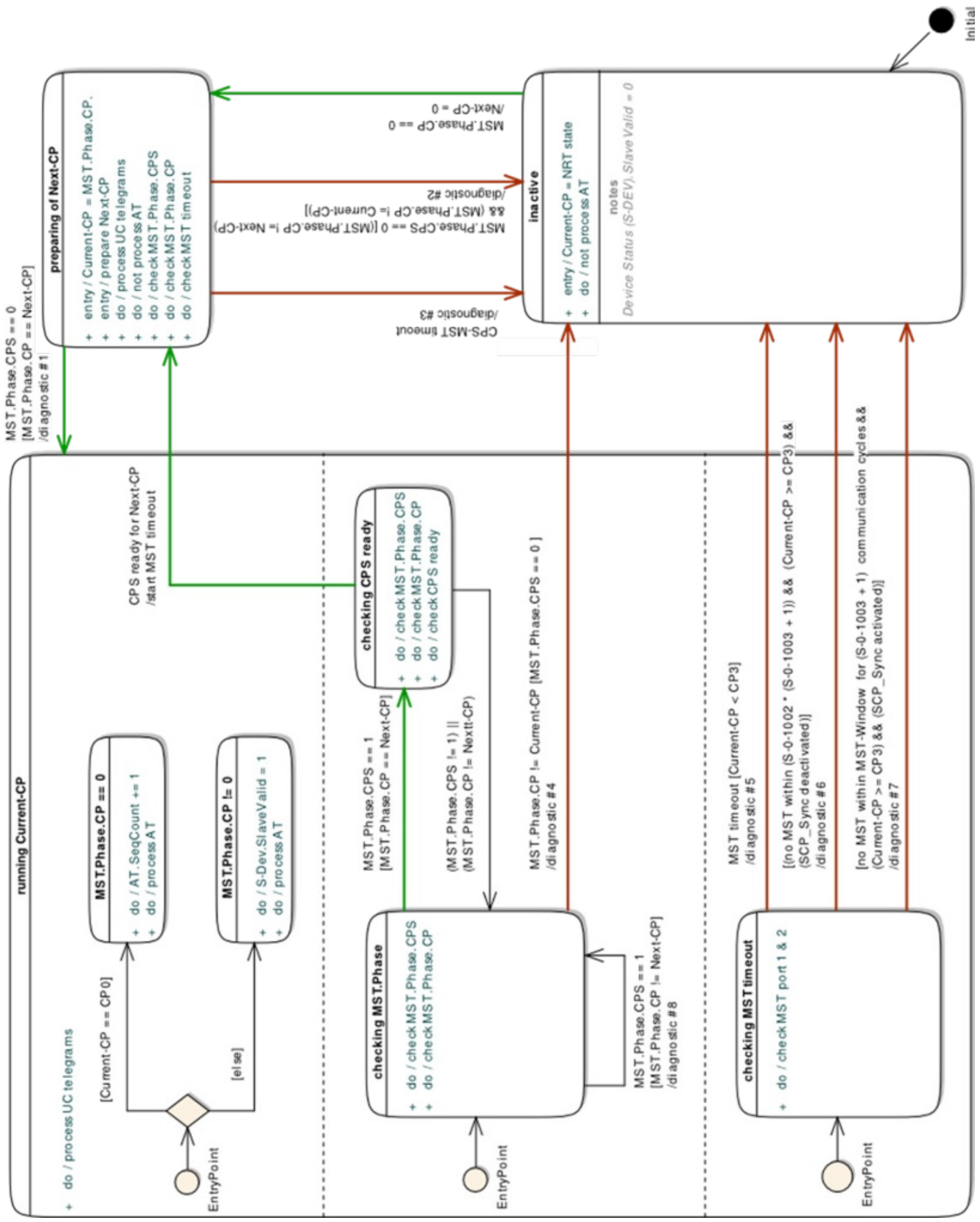

Figure 9 – CPSwitch state machine of the slave

NOTES

- current CP = at the time active CP
- next CP = CP0 or Current-CP or Current-CP + 1 (only valid, if current CP < 4)
- SEQCNT = sequence counter in AT0 of CP0
- CPS-MST timeout = 500ms
- MST timeout = 130ms

5.2.3.5 States of CP switching in the slave

Table 56 shows the states of the CPSwitch state machine.

Table 56 – States of slave CPSwitch state machine

State	Description
running Current-CP	This state contains the following states that are processed simultaneously: • MST.Phase.CP == 0 • MST.Phase.CP != 0 • checking MST.Phase • checking MST timeout • checking CPS ready The slave shall process UC telegrams.
MST.Phase.CP == 0	The slave activates CP0 and shall evaluate the Communication Version in the MDT0 of CP0 and shall activate the Address allocation.
MST.Phase.CP != 0	Depending on Current-CP the slave activates CP1, CP2, CP3 or CP4. The slave shall set S-DEV.SlaveValid to 1 and shall process MDT and AT as defined in the corresponding communication phase.
checking MST.Phase	The slave shall recognize the phase switching in the MST with CPS = 1 and Next-CP. • Next-CP shall be equal to • Current-CP + 1 (valid only, if Current-CP < 4) or • CP0.
checking MST timeout	The slave shall check the MST on both ports related to the CP.
checking CPS ready	If the slave recognizes the following conditions • MST.Phase.CPS is 1 and • MST.Phase.CP is Next-CP and the • CP related procedure command has been finished positively (CP2 and CP3 only) and • SVC valid is 1, then the slave generates "CPS ready for Next-CP" by setting slave valid to 0.
preparing of Next-CP	The slave shall not write data to the ATs any more. The slave writes the MST.Phase.CP to Current-CP. The slave prepares the Next-CP internally and activates the CPS MST timeout. This watchdog is triggered with every received MST. The slave shall wait to a MST with MST.Phase.CPS is 0 and MST.Phase.CP is Next-CP. The slave shall process the UC telegrams.
inactive	The slave shall set Slave valid = 0 and shall not write data to the ATs any more. The slave shall activate NRT state and shall process UC telegrams. Before entering NRT state from a state different to CP0 or HP0 the slave shall generate a link down on both ports.

5.2.3.6 Transitions of CP switching in the slave

The transitions of the state machine are shown in Table 57, Table 58 and Table 59.

Table 57 – Transitions of slave CPSwitch state machine

Transition with warning			Description
Source	Target	Condition	
Inactive	preparing of Next-CP	MST.Phase.CP == 0	Next CP = 0, slave switches to CP0
preparing of Next-CP	running Current-CP	MST.Phase.CPS == 0 &&	The master finished the phase switching correctly. The slave generates diagnostic #1 and

Transition with warning			Description
Source	Target	Condition	
		MST.Phase.CP == Next CP	activates the state "running Current-CP".
running Current-CP	MST.Phase.CP == 0	Current-CP == 0	The slave switches to CP0.
running Current-CP	MST.Phase.CP != 0	Current-CP != 0	Depending on Current-CP the slave switches to CP1, CP2, CP3 or CP4.
checking MST.Phase	checking CPS ready	MST.Phase.CPS == 1 && MST.Phase.CP == Next-CP	The master announces the next CP with CPS = 1 correctly. The slave activates the state "checking CPS ready".
checking CPS ready	checking MST.Phase	MST.Phase.CPS != 0 or MST.Phase.CP != Next-CP	If the slave detects a change in the MST.Phase.CPS or in the MST.Phase.CP, then it shall return to state "checking MST.Phase". **Recommended:** The master should move back the slave with the condition MST.Phase.CPS = 0 && MST.Phase.CP = Current-CP, in order to generate no error. The master may check the diagnostics of the slave to see why it has not set the CPS ready for Next-CP.
checking CPS ready	preparing of Next-CP	CPS ready for Next-CP	If the slave determines CPS ready for Next-CP, then it shall set the Slave valid to 0 and activate the state "preparing of Next-CP".

Table 58 – Transitions of slave CPSwitch state machine (transitions with warning)

Transition with warning			Description
Source	Target	Condition	
checking MST.Phase	checking MST.Phase	MST.Phase.CPS == 1 && MST.Phase.CP != Next CP	The master announces the Next-CP with CPS = 1 incorrectly, that means the Next-CP is not Current-CP or Current-CP+1 (vaild only, if Current-CP < 4) or CP0. In this case the slave generates diagnostic #8 and remains in the same state.

Table 59 – Transitions of slave CPSwitch state machine (transitions with error)

Transition with error			Description
Source	Target	Condition	
checking MST.Phase	inactive	MST.Phase.CPS == 0 && MST.Phase.CP != Current CP	The Master changes the phase without CPS = 1. The slave generates diagnostic #4 and activates state "inactive".
checking MST timeout	inactive	MST timeout && Current CP < CP3	MST timeout in CP0 to CP2: The slave does not receive a MST within the MST timeout (**130ms**). The slave generates diagnostic #5 and activates state "inactive".
checking MST timeout	inactive	no MST within (S-0-1002 * (S-0-1003 + 1)) && (Current CP ≥ CP3) && (SCP_Sync deactivated)	MST losses in CP3 and CP4 without SCP_Sync: The slave without SCP_Sync does not receive a MST within the MST timeout defined by S-0-1002 Communication Cycle time (tScyc) and S-0-1003 Allowed MST losses in CP3&CP4. MST timeout = tScyc * (allowed MST losses + 1). The slave generates diagnostic #6 and activates state "inactive".

Transition with error			Description
Source	Target	Condition	
checking MST timeout	inactive	no MST within MST Window for (S-0-1003 + 1) communication cycles && (Current CP ≥ CP3) && (SCP_Sync activated)	MST losses in CP3 and CP4 with SCP_Sync: The slave with SCP_Sync recognizes more successive MST losses as defined by the S-0-1003 Allowed MST losses in CP3&CP4 (allowed MST losses + 1). The slave generates diagnostic #7 and activates state "inactive".
preparing of Next-CP	Inactive	CPS-MST timeout (500ms)	If the CPS-MST timeout occurs, then the slave generates diagnostic #3 and activates state "inactive".
preparing of Next-CP	Inactive	MST.Phase.CPS == 0 && (MST.Phase.CP != Next CP) && (MST.Phase.CP != Current-CP)	During phase switching is in process the master changes the phase to an invalid CP. The slave generates diagnostic #2 and activates state "inactive".

5.2.3.7 Diagnosis of the CPS state machine

During the CPS transitions the slave generates the diagnostics shown in Table 60.

Table 60 – Diagnostics of CPS state machine slave

Diagnostic	Device status	Pattern of LED	S-0-0390	S-0-0014	Description
#1	Bit 7 = 0 Bit 6 = 0	MST.Phase.CP	---	Bit 2..0 = MST.Phase.CP	Switching of communication phase finished without error
#2 (to be used in new implementations of Type 19 devices)	Bit 7 = 1	Communication error	0xC30F4019	Bit 13 = 1	• During phase switching the master sets MST.Phase.CPS = 0 and MST.Phase.CP is set to one of the following invalid conditions: • MST.Phase.CP > 4, invalid CP • MST.Phase.CP ≠ current CP + 1, invalid sequence during the phase upshift. • MST.Phase.CP ≠ CP0, invalid sequence during the phase downshift.
#2a (used until V1.1.2 of CP16/3)	Bit 7 = 1	Communication error	0xC30F4003	Bit 5 = 1	During phase switching the master sets MST.Phase.CPS = 0 and MST.Phase.CP > 4, invalid CP.
#2b (used until V1.1.2 of CP16/3)	Bit 7 = 1	Communication error	0xC30F4004	Bit 6 = 1	During phase switching the master sets MST.Phase.CPS = 0 and MST.Phase.CP ≠ current CP + 1, invalid sequence during the phase upshift.
#2c (used until V1.1.2 of CP16/3)	Bit 7 = 1	Communication error	0xC30F4005	Bit 7 = 1	During phase switching the master sets MST.Phase.CPS = 0 and MST.Phase.CP ≠ CP0, invalid sequence during the phase downshift.
#3	Bit 7 = 1	Communication error	0xC30F4017	Bit 12 = 1	During phase switching is active the master did not send the MSTs again and the CPS-MST timeout occurs in the slave.
#4 (to be used in new implementations of Type	Bit 7 = 1	Communication error	0xC30F4019	Bit 13 = 1	• The master changes the MST.Phase.CP with MST.Phase.CPS = 0 and MST.Phase.CP is set to one of the following invalid conditions:

Diagnostic	Device status	Pattern of LED	S-0-0390	S-0-0014	Description
19 devices)					• MST.Phase.CP > 4, invalid CP. • MST.Phase.CP ≠ current CP + 1, invalid sequence during the phase upshift. • MST.Phase.CP ≠ CP0, invalid sequence during the phase downshift.
#4a (used until V1.1.2 of CP16/3)	Bit 7 = 1	Communication error	0xC30F4003	Bit 5 = 1	The master changes the CP with MST.Phase.CPS = 0 and set MST.Phase.CP > 4, --> invalid CP.
#4b (used until V1.1.2 of CP16/3)	Bit 7 = 1	Communication error	0xC30F4004	Bit 6 = 1	The master changes the CP with MST.Phase.CPS = 0 and set MST.Phase.CP ≠ current CP + 1, --> invalid sequence during the phase upshift.
#4c (used until V1.1.2 of CP16/3)	Bit 7 = 1	Communication error	0xC30F4005	Bit 7 = 1	The master changes the CP with MST.Phase.CPS = 0 and set MST.Phase.CP ≠ CP0, --> invalid sequence during the phase downshift.
#4d (used until V1.1.2 of CP16/3)	Bit 7 = 1	Communication error	0xC30F4006	--	The master changes the current-CP with MST.Phase.CPS = 0.
#6	Bit 7 = 1	Communication error	0xC30F4001	Bit 3 = 1	In CP3 and CP4 the slave did not receive a MST within the time defined by S-0-1002 Communication Cycle time (tScyc) and S-0-1003 Allowed MST losses in CP3&CP4, [S-0-1002 * (S-0-1003 + 1)], SCP_Sync is deactivated.
#7	Bit 7 = 1	Communication error	0xC30F4001	Bit 3 = 1	In CP3 and CP4 the successive MST losses exceeds the value defined in S-0-1003 Allowed MST losses in CP3&CP4. SCP_Sync is activated.
#8	Bit 6 = 1	---	0xC30E4019	Bit 2..0 = current CP	The master changes the MST.Phase.CP with CPS = 1 and MST.Phase.CP is set to one of the following invalid conditions: MST.Phase.CP > 4, invalid CP. MST.Phase.CP ≠ current CP + 1, invalid sequence during the phase upshift. MST.Phase.CP ≠ CP0, invalid sequence during the phase downshift.
#5	---	NRT state	0xC30A0008	Bit 2..0 = 7 (-1)	MST timeout (130ms) occurs in CP0 to CP2, the slave shall switch to NRT state

5.2.4 Communication Version

The master transmits the communication version in the MDT0 of CP0. The communication version (see Table 9) defines functions used by the master in CP0 to CP2.

- The slave shall compare the received Communication version with its own.
- If no difference is detected, then the slave shall participate in CP0 and follows the requirements of the master.

- If a difference is detected, then the slave shall check at first the "basis frame structure of AT0" (bits 7-0) and secondly the requested functions and add-ons of CP0 to CP2 (bits 31-8) as described below.

a) Communication Version bits 7-0
   1) If the slave detects no difference within bits 7-0, then the slave shall participate in CP0 and follows the requirements of the master.
   2) If the slave detects a difference within bits 7-0, then the slave does not know the structure of AT0 and may not communicate with the master.
      - The slave shall not participate in the Type 19 communication in CP0, that means it shall not insert data (e.g. its device address) in the AT0.
      - The slave may signal this on its Type 19 LED (pattern #13). In this case the phase up-shifting to CP1 is not possible.
      - Continue with Error reaction.

b) Communication Version bits 31-8
   1) All add-ons shall be supported by the slave.
   2) If the slave supports all requested functions, then the slave shall set bit 15 = 1 in the topology index of AT0-CP0.
   3) If the slave does not support all requested functions, then the slave
      - may signal this on its Type 19 LED (pattern #13),
      - shall process the address allocation in the AT0-CP0 and
      - shall set bit 15 = 0 of the topology index in AT0-CP0.
      - In this case the phase up-shifting to CP1 is not possible.

c) Error reaction
   1) If the master switches with the announced communication version to CP1, then the slave shall set the S-0-0390 Diagnostic number to 0xC30F4021 and activates NRT state.
      - The master may identify the error and generates an error message shown on the display.
      - In this case the master shall activate CP0 again and shall announce a previous communication version, which is supported by all slaves. Otherwise the phase up-shifting is not possible.

5.2.5 Address allocation in the master and slave

The master and the slaves shall always support the address allocation.

The topology index determined by the address allocation is used to address the service channel in CP1 and CP2. Therefore a valid sub-device address is not necessary in CP0 to CP2.

The content of topology index in the AT0 of CP0 shall be as specified in Table 28.

Functional sequence with line, ring and interrupted ring:

- The master shall transmit the AT0 and set the content as specified in Table 27.
- The slave shall read and increment the content of sequence counter (SEQCNT) field in the AT0. The read sequence counter corresponds to the order of the slave in the topology, called topology index (TADR).
- Additionally the slave shall write its sub-device address into the corresponding topology index field. Slaves with sub-device address = 0 or sub-device address > 511 shall support CP0 in the same way as slaves with sub-device address 1 to 511.

- The address allocation shall be done by the slave in CP0 at every passed AT0.
- Devices with multiple slaves increment the sequence count by one for each slave.
- Each slave receives two sequence counters (port 1 and port 2), the higher sequence counter is discarded.
- The slave will always mask the bit 15 of the valid sequence counter when determining its own topology address.

Further functional sequence with line only (see Figure 10):

- The last slave in line increments the sequence counter once.
- The master will always mask the bit 15 of the received sequence counter.
- For monitoring the master shall divide the sequence counter by 2 to get the number of slaves in the topology.

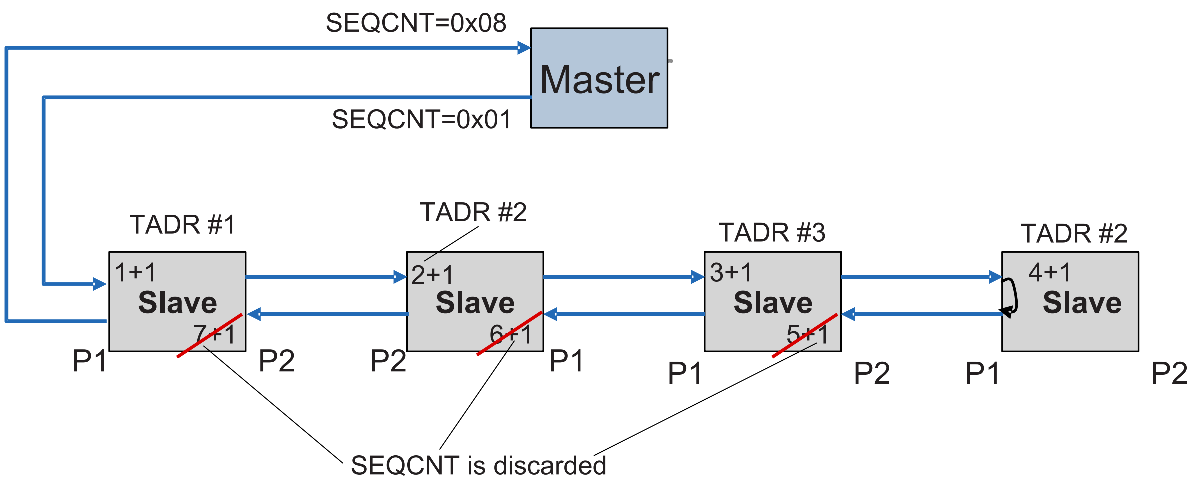

Figure 10 – Address allocation with line

Further functional sequence with ring only (see Figure 11):

- The master receives two sequence counters (port 1 and port 2), the higher sequence counter is discarded.
- For monitoring the master shall decrement the sequence counter by 1 to get the number of slaves in the topology.

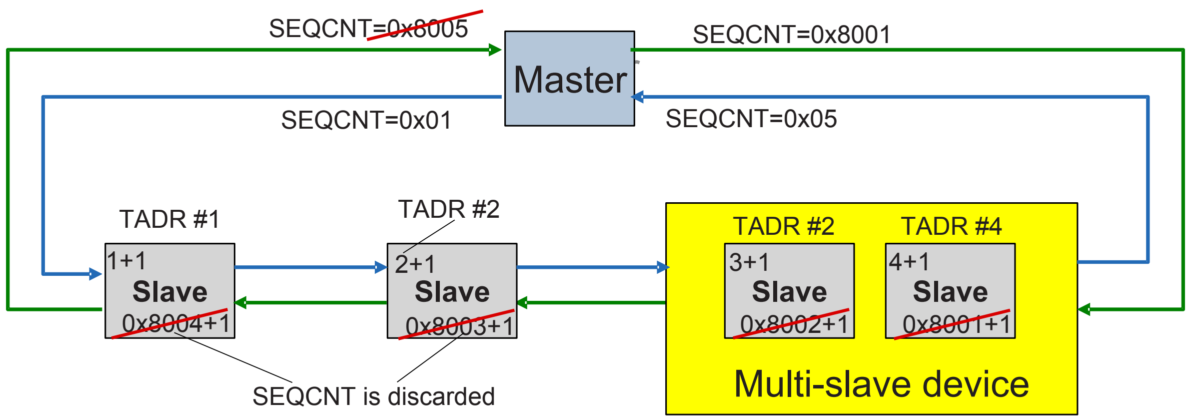

Figure 11 – Address allocation with ring

Further functional sequence with interrupted ring only (see Figure 12):

- The last slave in line increments the sequence counter once.
- The master shall transmit the sequence counter with the value 0x01 on port x. The received sequence counter on port Px shall be modified and transmitted on port Py.
- Modification of sequence counter on port Py:

$$SEQCNT\_Py = \frac{received\ SEQCNT\_Px}{2} + 0x8001$$

- The master shall transmit the modified sequence counter on Port Py. It is necessary to get an unique topology address, because the SVCs are addressed by the topology index in CP1 and CP2.
- The master will always mask the bit 15 of both received sequence counters.
- The master shall list the device addresses of the line with the greater received SEQCNT in a reversed order in its address table (e.g. for diagnosis purposes with device address and topology index).
- If the master detects that the ring can be closed, then the master may close the ring in CP1 to CP4.

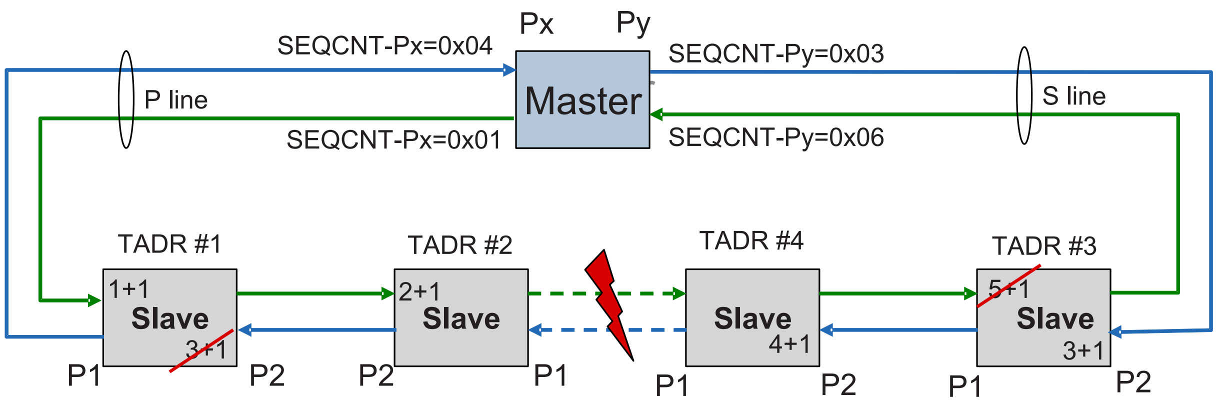

Figure 12 – Address allocation with interrupted ring

The master shall wait for its MDT0 and AT0 to be received. Depending on its configuration, the master may compare the detected device addresses with the device addresses that it is expecting to find, and then evaluate deviations (e.g., generate an error message).

If the procedure of the address allocation cannot be achieved within the time set by the master, the master shall remain in CP0 and generate a message. The scope of the message and at what point it has to be activated is a function of the master.

The master may generate the following 4 diagnostics in CP0:

- wrong device address: device address = 0 or greater than 511.
- same device address: master found several device addresses with the same value.
- not supported device address: device address is not supported by the master.
- unnecessary device address (waste slave): device address not necessary for the application.

5.3 Network topologies

5.3.1 Introduction

The physical network topology consists of full-duplex, point-to-point transmission lines and participants. The master and the slaves are parts of the Type 19 network and are its participants.

Each slave has two communication ports (port 1 and port 2). Port 1 (P1) and port 2 (P2) shall be interchangeable.

The physical network topology shall be either a ring structure or a line structure. A ring shall have two logical channels (primary and secondary) and a line shall have only one logical channel (primary or secondary).

The difference between ring and line structure is that the ring has a built-in redundancy against transmission media faults (for example, cable break) and should therefore be preferred.

Each master handles only one network. In line topology, the master needs one port only. In ring topology, the master shall support two ports.

NOTE A control unit may have one or more master interfaces depending on the configuration.

Type 19 communication interfaces shall be used to connect the slaves to the Type 19 network. At the physical layer, a Type 19 communication interface represents the connection of one or more slaves to the Type 19 network. Logically, one Type 19 communication interface with several slaves shall act the same as several Type 19 communication interfaces with one slave each. Cyclic communication may take place between all Type 19 devices within a Type 19 network.

The physical arrangement of slaves in the network is independent from the predefined device addresses of the slaves, as well as from the sequence of the real-time data fields in the MDT and AT.

Any slave is able to recognize the topology at any time, using the difference between primary and secondary telegrams. This is important if a slave is added to the communication at a later point in time (for example, addition of new machine parts, hot-plug). If a slave receives telegrams with the same Type 19 type on both ports (MDT0-P or MDT0-S) it recognizes a line. If it receives a MDT0-P on one port and a MDT0-S on the other port, it recognizes a ring.

5.3.2 Ring topology

The ring topology shall consist of a primary and a secondary channel. All slaves work in fast-forward mode (see Figure 13). Redundancy to protect against cable fault (for example disconnection, undesired cable break) is achieved through this ring. It is possible to open the ring and insert or remove slaves during operation (hot-plug). In case of a cable break at port 1 the slave shall activate loopback with forward at port 2 and vice versa.

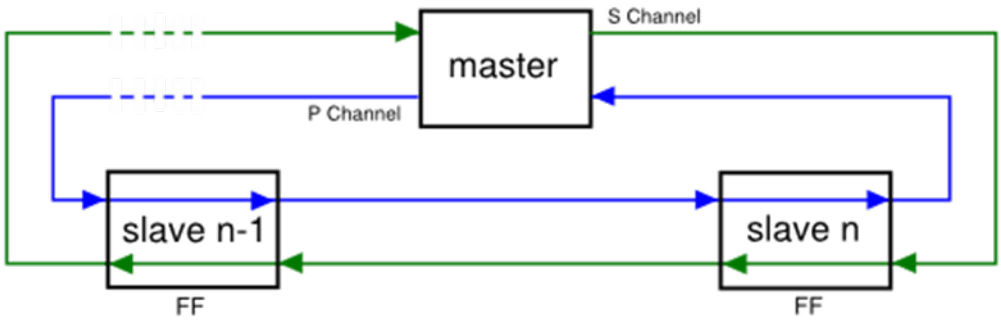

Figure 13 – Ring topology with P&S channel

5.3.3 Line topology

The line topology consists of either a primary or a secondary channel, depending upon configuration. The last physical slave performs the loopback with forward function. All other slaves work in fast-forward mode. It is possible to add or remove slaves at the end of the line during operation (hot-plug). In case of a cable break at one port the slave shall always activate loopback with forward at this port where it receives MST first (see Figure 14).

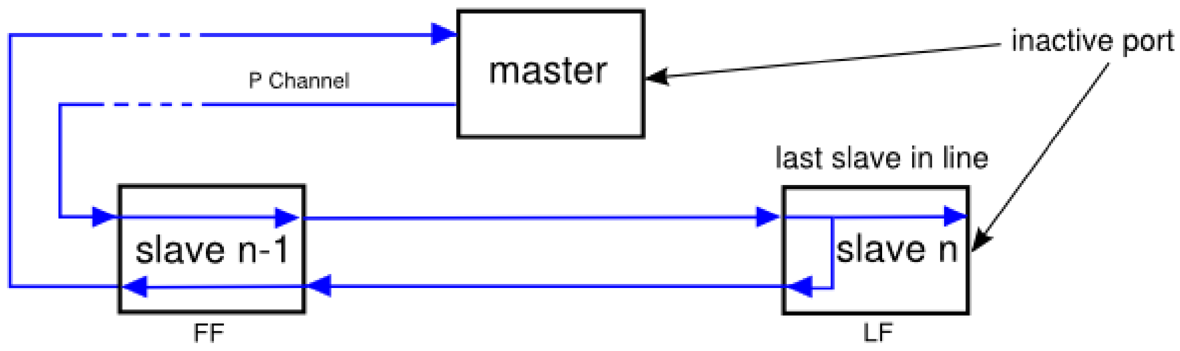

Figure 14 – Line topology with P channel (as example)

If no Ethernet device or no Type 19 device is connected or a ring break is present on the port, then this port is called inactive port. The inactive port

- shall receive any Ethernet-based telegrams.
- may support the insertion of a received non-Type 19 telegram into the Type 19 network.
- shall react on Type 19 telegrams and evaluate the MST.
- shall transmit UC telegrams as well as Type 19 telegrams.
- shall set the corresponding bits in device status to support ring recovery.
- shall use the Type 19 timing of the active port.

5.3.4 Topology conditions of a slave device

Each port of a slave shall be assigned to a processing unit and a multiplexer (see Figure 15). The functions in the slave shall depend on the topology and on the time slot within the communication cycle (RT channel or UC channel, see Figure 59).

The master has only a processing unit for each port (no multiplexing and no loopback).

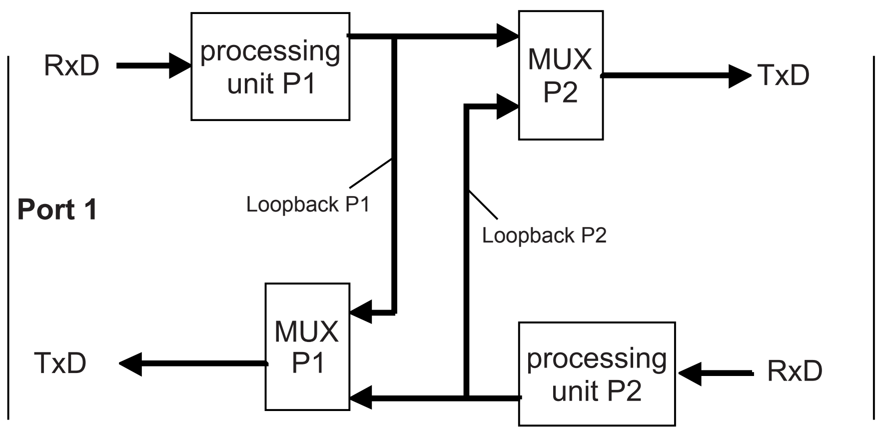

Figure 15 – Block diagram of a slave

- Case 1: During NRT state, a slave in NRT mode shall activate the loopback with forward on the port from where it received the first MST.

- Case 2: During CP0, a slave in loopback with forward shall disable loopback and enable fast-forward after it receives MST on both ports.
- Case 3: During CP0, a slave in fast-forward shall activate loopback on one port after it does not receive any MST on the other port within the MST timeout (130 ms).
- Case 4: During CP0, a slave in loopback with forward shall disable loopback and enable NRT mode after it does not receives MST on the active port within the MST timeout (130 ms).
- If the slave recognizes the phase switching (CPS = 1, CP = 1) it shall stop the automatically changing of the topology.

A slave can adjust 3 topology conditions in the RT channel (see Figure 16):

- fast-forward (normal condition)
- loopback P1 with forward (interrupted reception line on P2)
- loopback P2 with forward (interrupted reception line on P1)

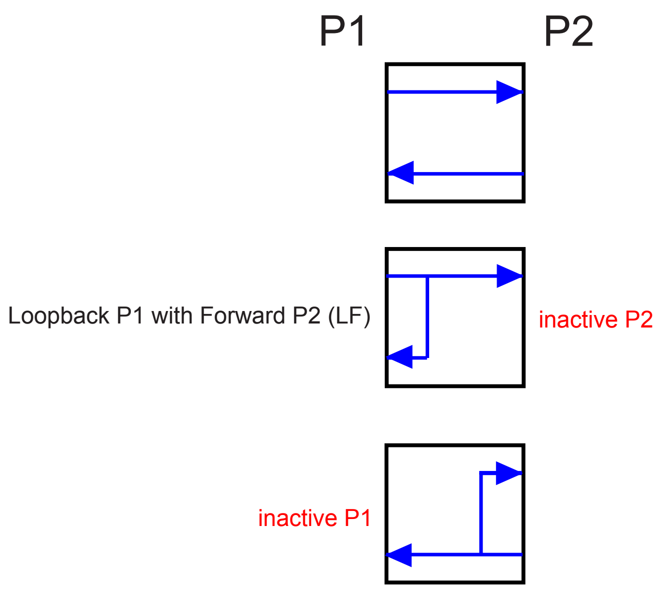

Figure 16 – Topology conditions of a slave

5.3.5 Topology conditions of a multi-slave device

A multi-slave device consists of several slaves as shown in Figure 17. The following functions shall be supported by the device and the slaves:

- S-0-1046 List of device addresses in device shall be supported by each slave.
- S-0-1037 Slave Jitter shall be supported by each slave and shall contain the same value.
- Handling of topology, sequence counter in CP0, Device Status (S-DEV) and Device Control (C-DEV) as described in clause 5.3.5.
- Deactivation of slaves within a multi-slave device:
  - If an application does not require all slaves in a multi-slave device, then the unused slaves shall be deactivated.
  - At least, one slave shall be always activated, because it shall handle the topology (e.g. redundancy etc.).
  - The functionality of the deactivation and activation of slaves within a multi-slave device is manufacturer specific.

Each of the slaves of a multi-slave device shall have a sub-device address that is written either during start-up or can be set by an external input unit such as a DIP switch or panel.

Each slave determines its position within the topology (topology address) in CP0. It is necessary to determine the topology index consistent in a multi-slave device. This determination shall be independent of how the two ports of the device are connected and which topology is present.

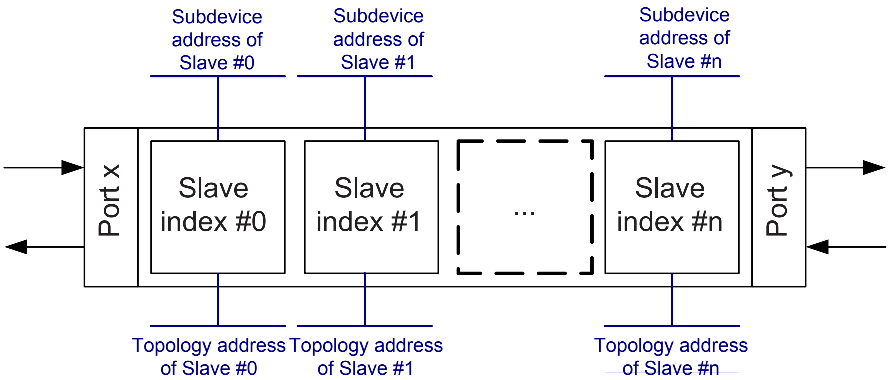

Figure 17 – Addressing of multi-slave device

It is specified that the cable from Port 1 and Port 2 can be interchanged without the master detects a change in the order of the topology. For this reason, a multi-slave device shall be insert the sub-device addresses in a different order in the AT0 in CP0, depending on the present topology.

It is also specified that the logical order corresponds to the multi-slave device shown in Figure 18. The slave right next to Port x is the slave #0 and has the smallest topology index in the multi-slave device. Then follows Slave #1, etc. up to the slave #n with the greatest topology address, which is located directly on port y.

The assignment of port x and port y to the real port 1 and port 2, is based on the received sequence counter of AT0 in CP0. This allows to replace the internal slave order of the multi-slave device. The assignment between slave and sub-device address is fixed, the topology index shall be determined. In addition, in AT0 of CP0 it is assumed that the received sequence counter on port 1 is called SEQCNT-P1 and the received sequence counter on port 2 is called SEQCNT-P2.

The following topology scenarios are possible:

- Ring and line topology: the multi-slave device is in a ring or not last in the line (see Figure 18)
- Line topology: the multi-slave device is last in the line (see Figure 19)

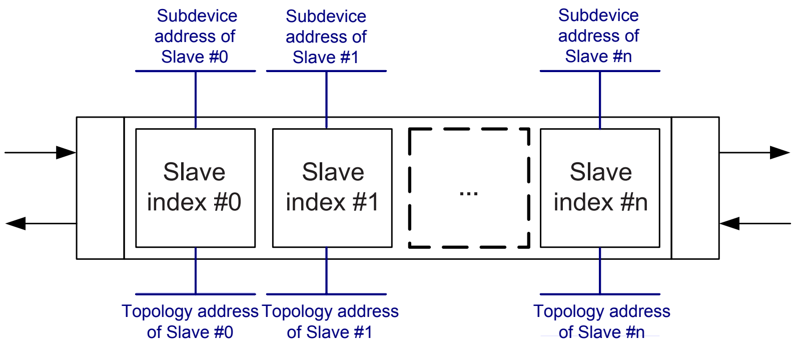

Figure 18 – Multi-slave device in ring topology or not last in line topology

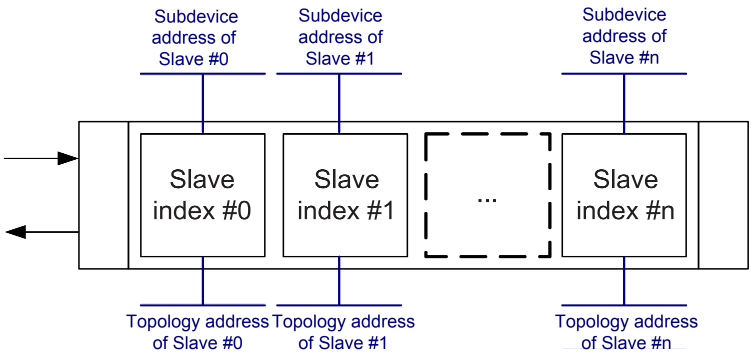

Figure 19 – Multi-slave device as last in line topology

Determination of the topology indices of a multi-slave device

Table 61, Table 62 and Table 63 contain the scenarios for line and ring with the assignment of the topology indices for slave #0 to slave #n.

Table 61 – Determination of the topology indices (1)

Type 19 telegram on P1 & P2	**P1.SEQCNT.bit15 ≠ P2.SEQCNT.bit15**	**P1.SEQCNT.bit15 = 0**	**P1.SEQCNT.bit15 = 1**
Topology	Ring	not last in line	not last in line
Sequence of topology indices	not inverted	not inverted	inverted

Table 62 – Determination of the topology indices (2)

Type 19 telegram on P1 only	**P1.SEQCNT.bit15 = 0**	**P1.SEQCNT.bit15 = 1**
Topology	last in line (P1)	last in line (P1)
Sequence of topology indices	not inverted	inverted

Table 63 – Determination of the topology indices (3)

Type 19 telegram on P2 only	P2.SEQCNT.bit15 = 0	P2.SEQCNT.bit15 = 1
Topology	last in line (P2)	last in line (P2)
Sequence of topology indices	not inverted	inverted

Due to the described assignment of the topology indices the behavior also arises in the topology status (S-DEV) and the topology control (C-DEV) of the slaves in a multi-slave device.

The slave with the lowest topology index and the slave with the greatest topology index (slave #0 and slave #n), shall evaluate the topology bits of the device control (C-DEV) and shall set the topology bits in the device status (S-DEV). All other slaves are only virtual in relation of the topology and are always in fast-forward and shall not evaluate the topology bits in the device control and shall not change the fast-forward in the device status.

Depending of the received P telegrams or S telegrams at the two ports, the topology status of both slaves (slave #0 and slave #n) are shown in Table 64.

Table 64 – Topology status of multi-slave device

Topology	Topology status (C-DEV)		
	Slave #0	Slave #1 to #n-1	Slave #n
see Figure 20	fast-forward	fast-forward	loopback with forward
see Figure 21	loopback with forward	fast-forward	fast-forward
see Figure 22	fast-forward	fast-forward	fast-forward

The permitted topology settings of a multi-slave device are shown in Table 65. In a ring topology it is forbidden that the master set the both slaves (close to the ports) to loopback with forward simultaneously.

Table 65 – Topology settings of multi-slave device

Topology	Permitted topology settings (C-DEV)		
	Slave #0	Slave #1 to #n-1	Slave #n
see Figure 20	fast-forward	fast-forward	loopback with forward or fast-forward, for example during hot-plug, ring recovery, etc.
see Figure 21	loopback with forward or fast-forward, for example during hot-plug, ring recovery, etc.	fast-forward	fast-forward
see Figure 22	fast-forward or loopback with forward, it shall not be activated simultaneously in slave #n	fast-forward	fast-forward or loopback with forward, it shall not be activated simultaneously in slave #0

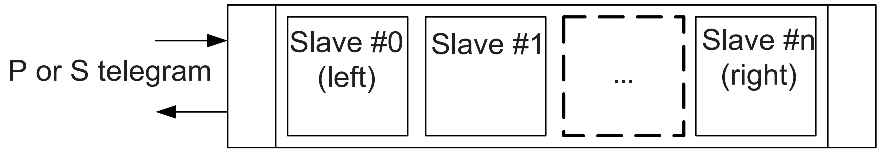

Figure 20 – Multi-slave device in line (left)

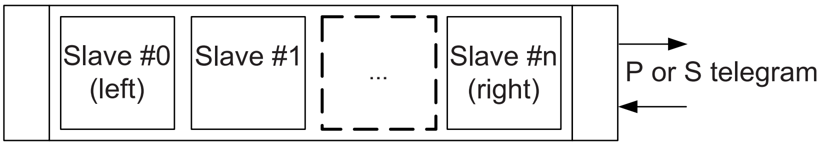

Figure 21 – Multi-slave device in line (right)

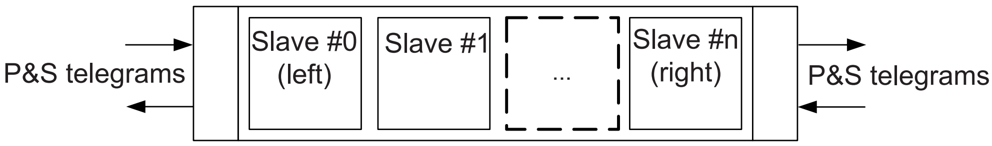

Figure 22 – Multi-slave device in ring

5.3.6 Topology state machine

The topology state machine of a slave shall consist of two states, the NRT state and the RT state (see Figure 23).

- NRT state
  - NRT mode is activated during NRT state. The slave shall check and evaluate the MST.
  - Functionality of a Type 19 device in NRT mode:
    - i) Standard Ethernet communication shall be active, if it is supported.
    - ii) At least store&forward shall be supported by the device, but cut-through may be supported also.
    - iii) The data from RxD (P1) shall be passed on with or without change to TxD (P2). The data from RxD (P2) shall be passed on with or without change to TxD (P1).
  - The NRT mode shall be activated during NRT state, HP0 and UC channel.
- RT state
  - consists of two sub-states, these are "fast-forward" and "loopback with forward".

NOTE the switching from RT Channel to UC channel and vice versa (at time t6 resp. t7) is not part of this state machine (see 7.1.10).

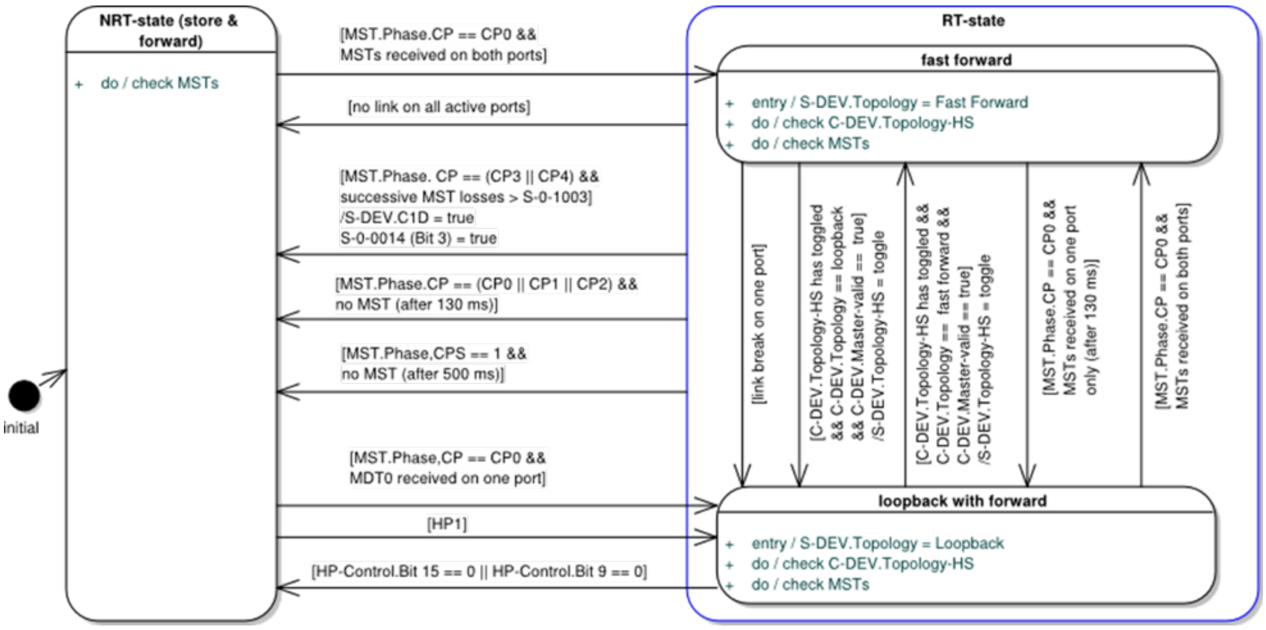

Figure 23 – Topology state machine of a slave

5.3.7 States of Topology state machine of slave

The states of the state machine are described in Table 66. The LED pattern shall be adjusted to the corresponding topology state.

Table 66 – States of Topology state machine of slave

State	Description
NRT state (NRT mode is active)	• Upon powering on, the master and each slave shall activate NRT state independently. • NRT mode is activated during NRT state. • The collision buffer shall be administrated as described in the topology state machine. • The master shall leave NRT state to CP0 upon request by its DL user. • The slave shall check the MST of the Type 19 telegrams. If the slave recognizes a – MST with CP = 0, then it shall activate CP0 and change the topology (see 5.3.6). – MST with CP = 4, then it shall activate HP0, if hot-plug is supported. • Before entering NRT state from a state different to CP0 or HP0 the slave shall generate a link down on both ports. • In NRT state the slave may activate pattern #1 of Type 19 LED.
RT state (fast-forward)	• The slave shall pass on the received data with or without change to the other port, delayed by tREP. • The slave shall set fast-forward in the device status (bit 13 and 12 = 0). • The slave shall check the topology HS in the device control (bit 14). • The slave shall check and evaluate the MST. • The master does not have fast-forward functionality.
RT state (Loopback with forward)	• The received data shall be passed on with or without change to both ports. • Loopback with forward may be activated either on P1 or P2 depending on the topology, but not on both ports simultaneously. • If the slave is the last physical one in the line topology, it shall thus activate loopback with forward only on the port from which it receives MSTs. • The slave shall set loopback with forward depending on the topology in the device status (bit 13 and 12 = 01 or 10). • The slave shall check the topology HS in the device control (bit 14). • The slave shall check and evaluate the MST. • The master does not have loopback with forward functionality.

5.3.8 Transitions of Topology state machine

The transitions of the state machine are described in Table 67, Table 68 and Table 69. In CP0 the slave shall activate the topology transition at time T7cp0. In CP1 to CP4 the slave may activate the topology transition as fast as possible. Telegram fragment may occur if traffic is not considered for topology transition.

Table 67 – Transitions of Topology state machine

Transition			Description
Source	**Target**	**Condition**	
NRT state	Fast-forward	CP is CP0 && MST received on both ports	During NRT state the slave shall activate fast-forward, as soon as MSTs with CP0 has been received at both ports.
NRT state	Loopback with forward	CP is CP0 && MST received on one port	During NRT state the slave shall activate loopback with forward, as soon as a MST with CP0 has been received at one port, but only as long as no MST has been received at the other port.
Loopback with forward	Fast forward	CP is CP0 && MST received on both ports	During CP0 a slave in loopback with forward shall disable loopback with forward and enable fast-forward as soon as it receives MST on both ports.
Fast-forward	Loopback with forward	CP is CP0 && MSTs received on one port only within 130 ms	During CP0, a slave in fast-forward shall activate loopback with forward on one port as soon as it does not receive MST on the other port within the MST timeout (130 ms).
NRT state	Loopback with forward	HP1	The hot-plug slave has received all HP0 parameters and activates loopback with forward.
RT state	NRT state	no MST within MST timeout (130 ms) && CP is (CP0, CP1, CP2, HP1, HP2)	If the slave does not receive MSTs within the MST timeout in CP0, CP1, CP2, HP1 or HP2, then it activates NRT state.
Loopback with forward	Fast forward	C-DEV.Topology-HS = toggle && C-DEV.Topology = fast-forward && C-DEV.Master-valid = 1	The master commands fast-forward. • If the slave still detects a Type 19 link on the inactive port, then the slave shall toggle S-DEV.Topology-HS, activate Fast-forward and shall set S.DEV.port status accordingly. The slave shall clear the warning 0xC30E4020 in S-0-0390 Diagnostic number. • If the slave does not detect a Type 19 link on the inactive port, then the slave shall toggle S-DEV.Topology-HS, remain in Loopback with forward and shall not change S.DEV.port status.
Fast-forward	Loopback with Forward	C-DEV.Topology-HS = toggle && C-DEV.Topology = loopback with forward && C-DEV.Master-valid = 1	The master commands loopback with forward, the slave toggles S-DEV.Topology-HS.
Loopback with forward	NRT state	(HP-Control.Bit 15 = 0) or (HP-Control.Bit 9 = 0)	Master cancels the hot-plug function.

Table 68 – Transitions of Topology state machine (transitions with warning)

Transition with warning			Description
Source	**Target**	**Condition**	
Fast-forward	Loopback with forward	link break on one port	A cable fault (for example: disconnection, undesired cable break) is detected at one port. If CP > CP0 the slave shall generate a warning (S-DEV.Bit 6 = 1), set the S-0-0390 Diagnostic number to 0xC30E4020 and activates loopback with forward.

Table 69 – Transitions of Topology state machine (transitions with error)

Transition with error			Description
Source	Target	Condition	
RT state	NRT state	no link on all active ports	If CP > CP0 the slave shall generate an error (S-DEV, bit 7 = 1), sets S-0-0390 Diagnostic number to 0xC30F4020 and activates NRT state.
RT state	NRT state	MST.Phase.CPS = 1 && no MST within MST-CPS timeout (500 ms)	If the MST-CPS timeout occurs during phase switching, the slave generates an error (S-DEV, bit 7 = 1), sets the S-0-0390 Diagnostic number to 0xC30F4017 and activates NRT state.
RT state	NRT state	MST losses > S-0-1003 Allowed MST losses in CP3&CP4 && CP is (CP3 or CP4)	If the MST losses in CP3 or CP4 exceeds the value of S-0-1003 Allowed MST losses in CP3&CP4, then the slave sets S-DEV.C1D-error = 1, S-0-0014.Bit 3 = 1, if supported and sets the S-0-0390 Diagnostic number to 0xC30F4001 and activates NRT state.

5.4 Redundancy of RT communication with ring topology

5.4.1 Introduction

The master shall send all telegrams with the same content on the P channel and on the S channel (see Figure 24). Likewise, the master shall receive the telegrams from the slaves twice and process the SVC and real-time data of one valid received telegram only (either P or S telegram). If the master does not receive the selected telegram, then the master shall process the SVC and real-time data of the other telegram.

Each slave shall receive both telegrams, work on the assigned data fields in P and S channel, and pass them on in their respective channels.

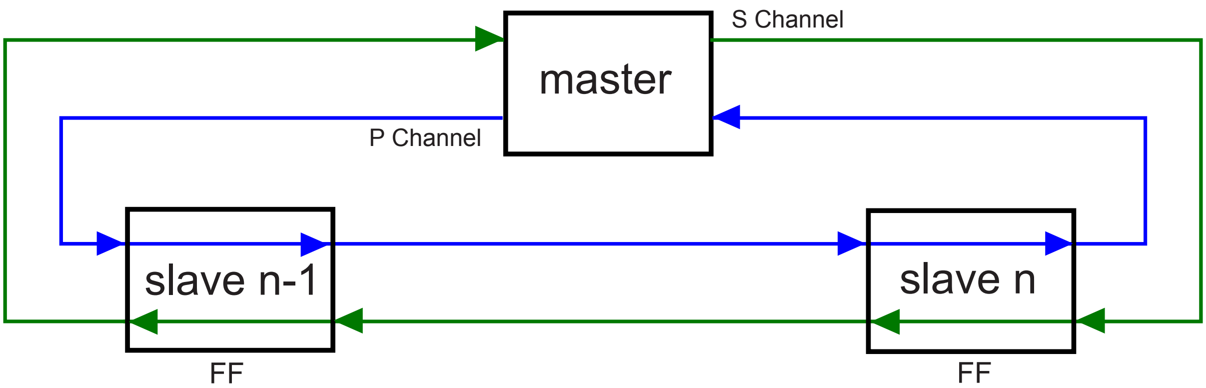

Figure 24 – Ring without break

5.4.2 Sequence with ring break

After a cable break, the slave shall switch from fast-forward to loopback with forward within less than minimal communication cycle time (tScyc, S-0-1002) and signal the changing of the topology in the device status (bits 14 – 12).

The slave activates loopback always at the undisturbed port, thereby the received RT telegrams are sent back to the master and forwarded on the disturbed port simultaneously.

In case of ring break the ring disintegrates into 1 or 2 lines (see Figure 25 and Figure 26).

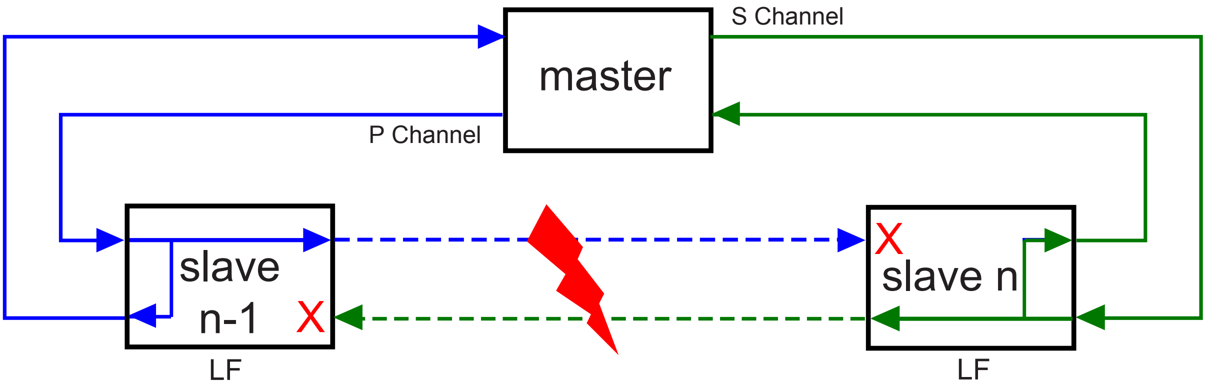

Figure 25 – Ring break

The master sends P telegrams and receives S telegrams at one port. At the other port the master sends S telegrams and receives P telegrams (see Figure 24).

When a slave changes the topology, this will be recognized by the master, because now the master sends and receives P telegrams at one port and sends and receives S telegrams at the other port (see Figure 25).

In case of a cable break between master and slave, the slave changes over the topology, this will be recognized by the master because now the master sends and receives for example P telegrams at one port and sends S telegrams and receives no telegrams at the other port (see Figure 26). In order to show the interrupted connection, the master has to analyze the device state of all slaves.

Recommendation: to serve the purpose for diagnosis, the master should save in a parameter all interrupted links (for example cable break between slave n-1 and slave n) with the sub-device addresses, the addresses of topology, date and time.

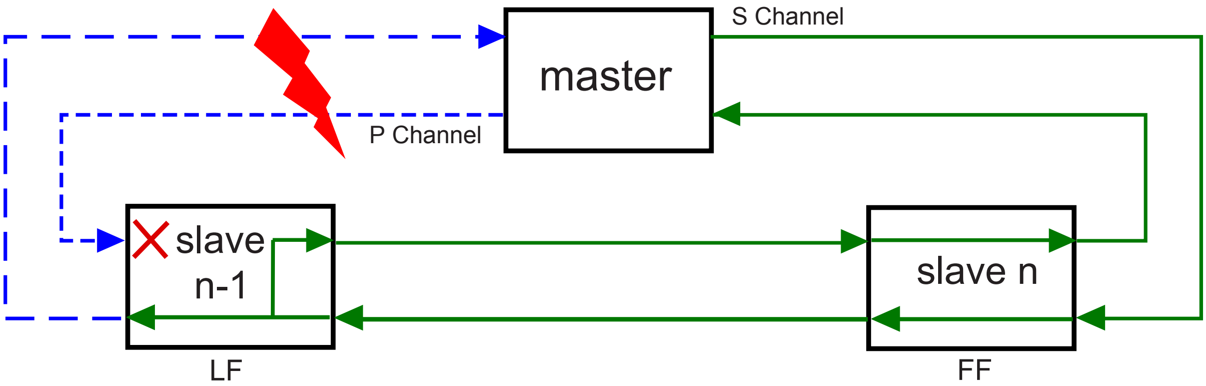

Figure 26 – Ring break on master

5.4.3 Recovery of ring topology

The slave monitors both ports and shows the state of the ports in the device status (S-DEV). With this information the master recognizes that the interruption has been removed. Before the master can close the ring it shall check that the P telegram or S telegram is received on the inactive port of the corresponding slave (see S-DEV, bit 11 and 10).

The slave shall only close the ring, if the master commands this in the device control (C-DEV). The P channel and S channel can be closed one after another or simultaneously (see 5.4.4 and 5.4.5).

When the master has closed the P channel and S channel, the establishment of the ring has been finished. After the ring recovery in CP4 and with running synchronization the master shall transmit the S-0-1015 Ring delay of the topology to all synchronizing slaves. After that, the master shall activate the S-0-1024 SYNC delay measuring procedure command to announce the slave, that it can synchronize on the two ports again.

To support the IP switch, the master shall transmit the time slot parameters of the UC channel with broadcast address in the MDT HP field. Directly after the start of CP3 and immediately after ring recovery the master shall transmit t6 and t7 with the defined scan cycles of each parameter. The contents of the MDT HP field are shown in Table 52.

5.4.4 Recovery of P channel

In Figure 27 the slave n recognizes, that it receives again P telegrams and shows this in the device status of the S channel.

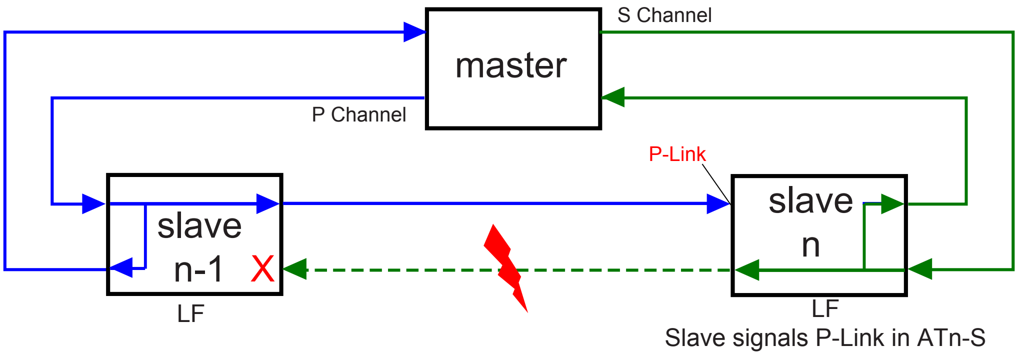

Figure 27 – Recovery of P channel (1)

Now it is possible for the master to change the topology of the slave n from loopback with forward to fast-forward (see Figure 28), in order to re-establish the P channel.

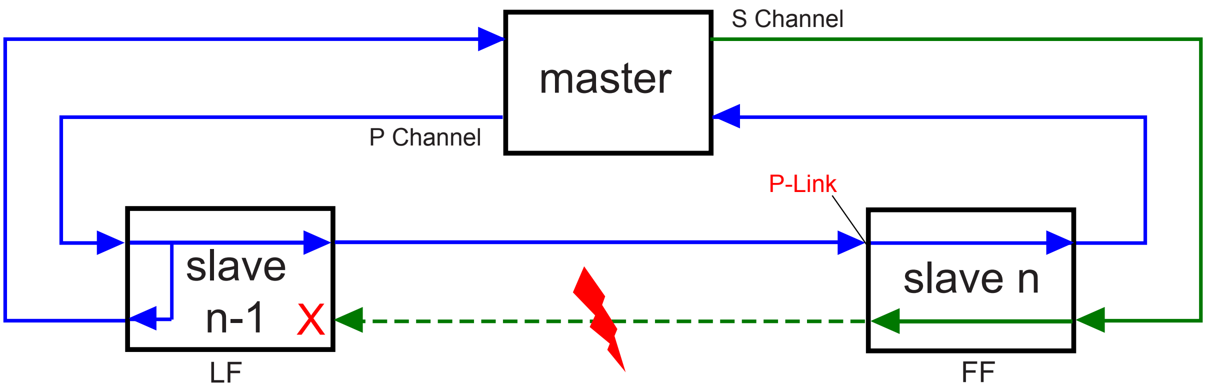

Figure 28 – Recovery of P channel (2)

5.4.5 Recovery of S channel

In Figure 29 the slave n-1 recognizes, that it receives S telegrams and shows it in the device status of the P channel.

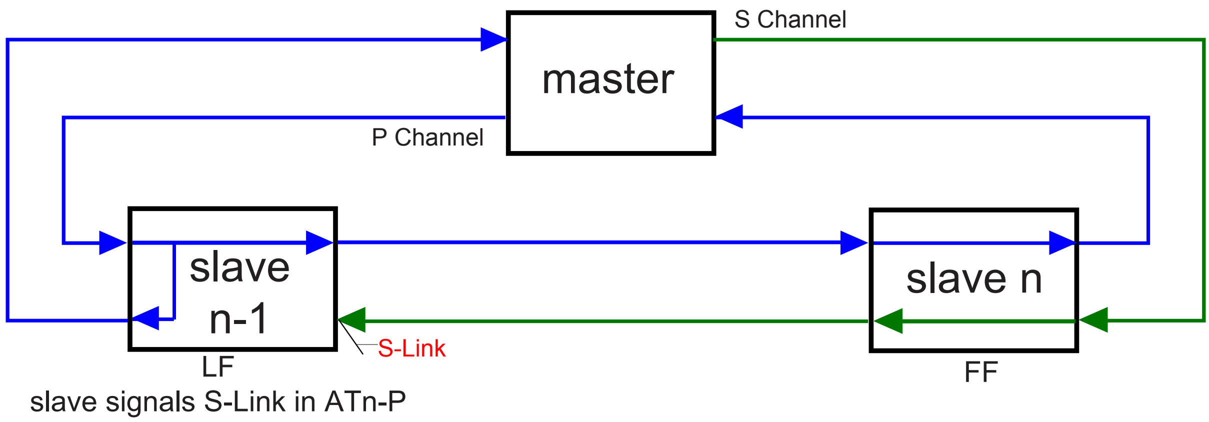

Figure 29 – Recovery of S channel (1)

Now the master can change the topology of the slave n-1 from loopback with forward to fast-forward (see Figure 30), in order to reestablish the S channel.

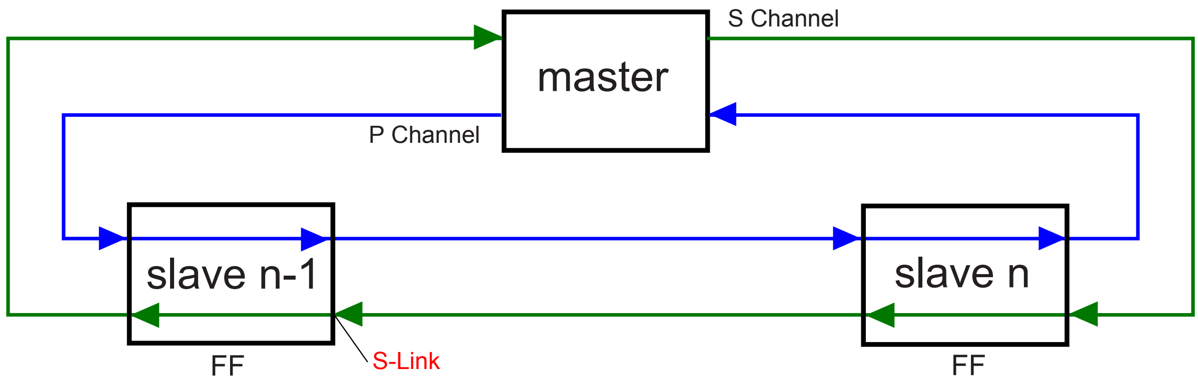

Figure 30 – Recovery of S channel (2)

5.5 Hot-plug procedure

5.5.1 Introduction

Upon an Enable_Hotplug (EHP) request by the DL user in the master device, a list of expected devices that may be hot-plugged (depending on the application) is passed to the DL. The DL user is informed by a Notify_Device_Status_Change (NDSC) if a new device has been hot-plugged.

The hot-plug mechanism provides the ability to bring devices with one or more slaves (multi-slave device) subsequently to CP4. The running Type 19 network with the participant remains in CP4. Hot-plugging is supported in CP4 with line topology only. With ring topology, a ring break has to be initiated first.

The last slave in a line topology shall continuously monitor its inactive port. If an additional device gets connected it receives all telegrams from the last slave.

The Master shall be prepared for the HP slaves, this means the telegram fields used by the hot-plug slaves shall be configured in the MDT and AT. In case of a broken ring topology, the master should activate the Hot-plug function on one channel only (P or S).

For a Type 19 slave there are two different ways to participate in the cyclic communication of CP4. The "communication phase run-up" with the sequence from CP0 to CP4 and the "Hot-Plug procedure" with the sequence from HP0 to HP2 and switching to CP4.

An additional device shall start in the NRT state ("store and forward") as described Non-real-time state. If the additional device supports the Type 19 protocol, it shall evaluate the hot-plug field in the MDT0. Using the hot-plug fields in MDT0 and AT0, the master and the hot-plug slave shall be able to communicate. After processing through the hot-plug procedure, the hot-plug slave shall become the last slave within the line topology.

The hot-plug procedure is divided in 3 phases (HP0, HP1 and HP2).

5.5.2 Hot-plug state machine

The hot-plug state machine is shown on the right side of Figure 31. The LED pattern shall be adjusted to the corresponding hot-plug states.

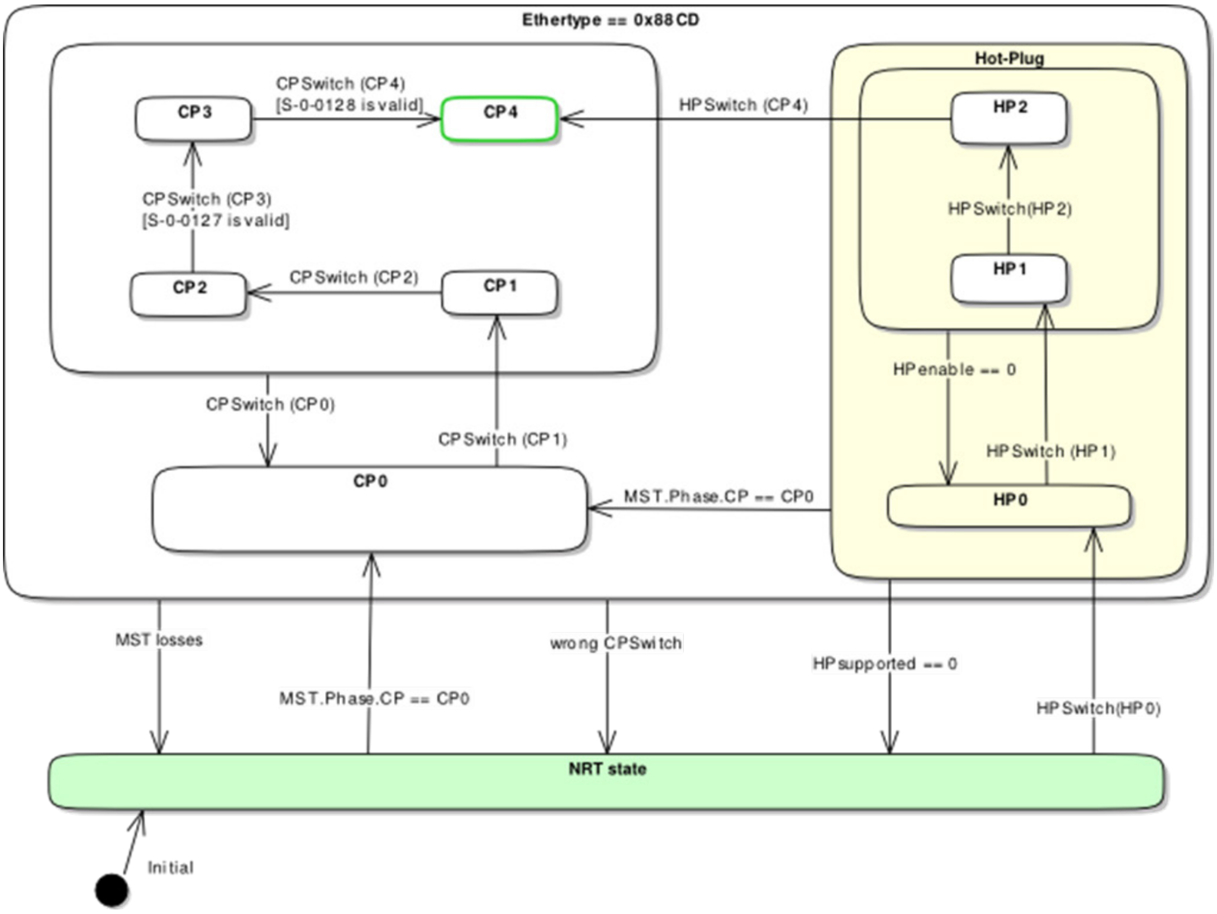

Figure 31 – Communication phase and hot-plug state machine

5.5.3 States of HP state machine

The states of the HP state machine are described in Table 70.

Table 70 – States of HP state machine

State	Description
NRT state	Upon powering on, the master and each slave shall activate NRT state independently.

State	Description
	NRT mode is activated during NRT state. The collision buffer shall be administrated as described in the topology state machine. The master shall leave NRT state to CP0 upon request by its DL user. The slave shall check the MST of the Type 19 telegrams. If the slave recognizes a • MST with CP = 0, then it shall activate CP0 and change the topology (see 5.3.6). • MST with CP = 4, then it shall activate HP0, if hot-plug is supported. Before entering NRT state from a state different to CP0 or HP0 the slave shall generate a link down on both ports. In NRT state the slave may activate pattern #1 of Type 19 LED.
HP0	In HP0 the HP slave is in NRT state and can not transmit Type 19 telegrams to the master. The HP slave • activates the parametrization level (PL). • handles the received telegrams in store&forward or cut-through. • evaluates additionally the MST and MDT-HP field. • generates pattern #7 of Type 19 LED. The master shall set • HP supported (HP control, bit 15=1), • HP enable (HP control, bit 9=1) and • starts the transmission of the HP0 parameters in the MDT-HP field by using the broadcast address. The master shall repeat the transmission of each HP0 parameter in successive communication cycles (like defined in HP1 for the slave), because the HP slave does not know yet the communication cycle time. The HP0 parameters are common to all hot-plugging slaves: • S-0-1002 Communication Cycle time (tScyc) • S-0-1017 UC transmission time • S-0-1027.0.1 Requested MTU Communication Version (MDT0 of CP0) • S-0-1010 Lengths of MDTs • S-0-1012 Lengths of ATs The HP slave shall read the HP0 parameters only, if the broadcast address (4095) is valid and the HP enable is set (HP control, bit 9=1) by the master. The contents of the MDT-HP field to transmit the HP0 parameters are shown in Table 71.
HP1	The master shall change the topology state of the last slave from loopback with forward to fast-forward. The HP slave shall generate pattern #8 of Type 19 LED. The master shall scan all HP slaves of the HP device (single-slave or multi-slave device) and transmit individual HP1 parameters to each HP slave. **Scanning of HP slave with slave index** Definition of slave index: In a single-slave device the slave index shall be 0. In a multi-slave device the slave index shall start with 0, 1, 2, etc. During the scanning of the HP slaves the master shall set the device address to 0. The Master shall start the scanning with slave index 0 and increments the index by 1 for the next slave etc. The scan cycles of each parameter are defined as follows: tscyc ≤ 2ms --> scan cycles = (2ms/tScyc) * 10 cycles tscyc > 2ms --> scan cycles = 10 cycles The HP slave shall match the slave index and inserts the corresponding device address in the AT-HP field. The HP slave shall react to a changing of slave index in the AT-HP field within the given amount of scan

State	Description
	cycles. If a HP slave is addressed with slave index which is not present in the HP device, then the HP device shall set the device address to 4093 (0xFFD) and match the slave index in the AT-HP field. The master shall stop the scanning of slaves if it receives 4093 as device address in the AT-HP field. If the master did not scan all present HP slaves, then the HP device sends an error message in the HP status (bit 8=1 and error code = 5). **Transmission of HP1 parameter** If the master has scanned all HP slaves, the master shall transmit the following slave specific HP1 parameters to all HP slaves of one HP device. The master shall use the device address of the HP slave. If the HP slave did not detect its own device address in the MDT-HP field, then the HP slave generates an error message in the HP status (bit 8=1 and error code = 4). List of HP1 parameter: • S-0-1013 SVC offset in MDT; • S-0-1014 SVC offset in AT; • S-0-1042 Topology index. The contents of the MDT-HP field to transmit the HP1 parameters are shown in Table 72. The reception of each HP1 parameter shall be acknowledged by the addressed HP slave (MDT-HP address = device address of HP slave) in the AT-HP field. Two cases are possible with S-0-1042 Topology index: Case 1: If the HP slave does not support S-0-1042 Topology index, then it shall ignore the reception and shall not generate any error. Case 2: If the HP slave supports S-0-1042 Topology index but the master did not transmit S-0-1042, then the slave shall not generate any error also. The acknowledgment of a HP slave is shown in Table 73.
HP2	The master shall fully configure the HP slave using SVC (service channel) as described in CP2. The HP slave shall write protect the HP0 and HP1 parameters and shall generate pattern #9 of Type 19 LED. After the transmission of the communication parameters as in CP2, the master shall activate the S-0-0127 CP3 transition check. The HP device shall process the procedure command as described in CP2. The master can transmit application parameters as in CP3 and shall produce valid real-time data (e.g. command values).

Table 71 – MDT hot-plug field in HP0

MDT-HP address	HP control	MDT-HP INFO	Description
4095 (broadcast)	0x8201	value of tScyc (Communication cycle time)	S-0-1002 Communication Cycle time (tScyc)
4095	0x8202	value of t6 (beginning of UC channel)	S-0-1017 UC transmission time, list element 0
4095	0x8203	value of t7 (end of UC channel)	S-0-1017 UC transmission time, list element 1
4095	0x8204	value of MTU	S-0-1027.0.1 Requested MTU
4095	0x8205	value of Communication version (MDT0 of CP0)	Communication version has no IDN
4095	0x8210 0x8211 0x8212 0x8213	value of MDT0 length value of MDT1 length value of MDT2 length value of MDT3 length	S-0-1010 Lengths of MDTs, list element 0 S-0-1010, list element 1 S-0-1010, list element 2 S-0-1010, list element 3
4095	0x8220 0x8221	value of AT0 length value of AT1 length	S-0-1012 Lengths of ATs, list element 0 S-0-1012, list element 1

	0x8222 0x8223	value of AT2 length value of AT3 length	S-0-1012, list element 2 S-0-1012, list element 3

Table 72 – MDT hot-plug field in HP1

MDT-HP address	HP control	MDT-HP INFO	Description
device address of HP slave	0x8280	value of MDT-SVC offset	S-0-1013 SVC offset in MDT
device address of HP slave	0x8281	value of AT-SVC offset	S-0-1014 SVC offset in AT
device address of HP slave	0x8282	value of Topology index	S-0-1042 Topology index if supported by the master

Table 73 – AT hot-plug field in HP1

AT HP address	HP status	AT HP INFO	Acknowledgment of
device address of slave	0x0080	don't care	MDT-SVC offset
device address of slave	0x0081	don't care	AT-SVC offset
device address of slave	0x0082	don't care	Topology index (only acknowledged if supported by master and slave, otherwise ignored by slave)

5.5.4 Transitions of HP state machine

The transitions of the state machine are described in Table 74.

Table 74 – Transitions of HP state machine

Transition		Condition	Description
Source	Target		
NRT-State	HP0	HPSwitch(HP0)	If a slave is connected during CP1 to CP4 and recognizes a MST of CP4 and the slave supports hot-plug, then it switches to hot-plug phase 0 (HP0) and evaluates the Hot-plug field of MDT0. If the last slave in a line topology signals a link in S-Dev.Topology & Port status at its inactive port, then the master recognizes an additional connected device (slave or Ethernet device) and shall activate the hot-plug procedure by setting the HP control, bit 15 to 1.
HP0	HP1	HPSwitch(HP1)	If the HP slave has received and checked all HP0 parameters without error, then the HP slave shall switch to HP1 and change the topology state to loopback with forward. This has to take place at the port on which the HP slave has received the HP0 parameters. If the last slave in line topology signals a Type 19 link in S-Dev.Topology & Port status at its inactive port, then the master shall switch to HP1.
HP1	HP2	HPSwitch(HP2)	If all HP slaves of one HP device have received and checked all HP1 parameters without error, then the master shall switch to HP2. For that the master shall set bit 8=1 in the HP control and the MHS=1 in the related SVC control. After that the HP slave shall set AHS=1 and SVC valid=1 in the SVC status of its assigned SVC in the AT. If an error occurs in the HP slave during switching to HP2 (e.g. faulty parameters), then it

Transition		Condition	Description
Source	Target		
			generates an error message in the HP control (bit 8=1 and error code = 2). In this case the master shall prepare the switching to HP0. Master and HP slave shall activate HP2 if no error occurs.
HP2	CP4	HPSwitch(CP4)	The master shall initiate the HP device to leave HP2 using S-0-0128 CP4 transition check to switching to CP4. The HP device shall process the procedure command as described in CP3. The HP slave activates CP4 when it generates a positive acknowledgment of the S-0-0128 CP4 transition check and the slave valid is set to 1 in the device status.
HP1..HP2	HP0	HPenable == 0	Case 1, HP1: If the master does not get a response from the scanned HP slave within this defined time, then the master shall stop the hot-plug procedure with an error message and shall prepare the switching to HP0. Case 2, HP1: If the transfer of one HP1 parameter from one HP slave of the HP device is acknowledged with error, then the master shall stop the hot-plug procedure with an error message and shall prepare the switching to HP0. The acknowledgment with possible errors of a HP slave is shown in Table 75. Case 3, HP2: If an error occurs in the HP slave during HP2 and the master is not able to detect the error, then the master shall prepare the switching to HP0. Prepare switching to HP0: Before the master switches to HP0, it shall activate loopback with forward at the last slave in line and waits of changed topology of last slave. After that, the master shall set the HP control, bit 9 to 0 and switches to HP0.
Hot-plug	NRT state	HPsupported==0	The HP slave shall switch from each HP phase to NRT state with one of the following conditions: If the HP slave does not receive a MST within a time of 130 ms. If the master does not support hot-plug by setting the HP control, bit 15 to 0.
Hot-plug	CP0	MST.Phase.CP == CP0	The HP slave shall switch to CP0 if the HP slave receives a MST with CP = CP0.

Table 75 – AT hot-plug field in HP1 (error)

AT-HP address	HP status	AT-HP INFO	Error of
device address of slave	0x0180	don't care	MDT-SVC offset
device address of slave	0x0181	don't care	AT-SVC offset
device address of slave	0x0182	don't care	Topology index (only if supported, otherwise ignored)

5.6 Status procedures

Upon a Get_Device_Status (GDS) request by the DL user in the master device, the status word of the specified device is returned to the DL user.

Upon a Set_Device_Status (GDS) request by the DL user in the master device, the control word of the specified device is set.

Upon a Get_Network_Status (GNS) request by the DL user in the master device, the status of the network is returned to the DL user.

6 Data transmission methods

6.1 Overview

Data transmission methods are the means by which a DLE performs its functions and affects the behavior of the DL-protocol. Methods are initiated, executed and terminated under the control of invoked services, as specified in the Type 19 DL-service.

6.2 Service channel (SVC)

6.2.1 SVC handling

Acyclic data is exchanged between a master and a slave device upon a Read (RD) request initiated by the DL user in a master device. To transmit this data, the SVC INFO field shall be reserved for the service channel in the MDT (see 4.5.7.3.3) and in the AT (see 4.6.7.3.3). Special bits in the MDT-SVC control and the AT-SVC status shall be used to control execution in the service channel. Therefore, the master shall be able to support a separate service channel for every used slave device.

With a SVC transmission, the following operations shall be possible:

- initialization of the Type 19 communication;
- transmission of all data block elements of a parameter;
- transmission of procedure commands;
- changing limit values on demand;
- changing control loop parameters on demand;
- obtaining detailed status messages from a slave device;
- diagnostic functions.

Any SVC transmission shall always be initiated and controlled by the master. The operations, “read data block element” or “write data block element”, shall be from the perspective of the master. All operations shall always relate to the last transmitted IDN.

The SVC transport of parameter or of a procedure command shall be handled via a predetermined handling and proceeding sequence (see Figure 32) for individual actions. The master shall follow strictly the outline of these diagrams.

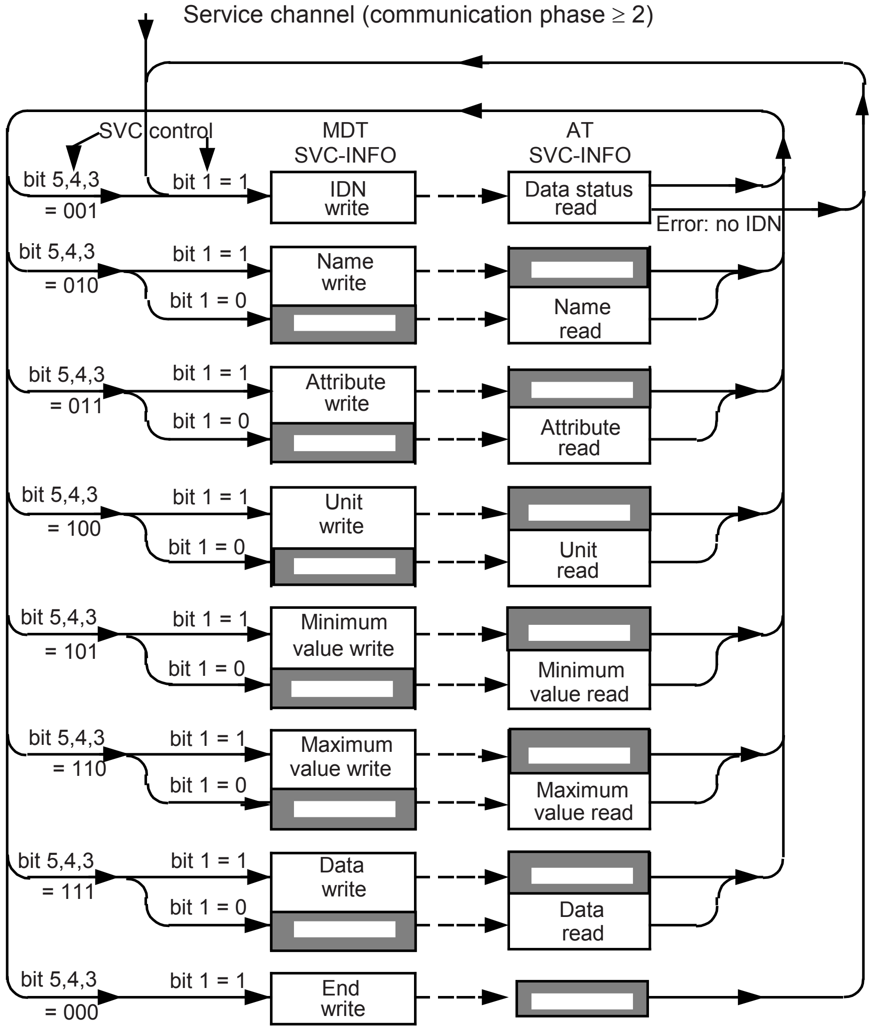

Figure 32 – Service channel handling diagram

6.2.2 Opening and closing SVC

- Opening SVC: The master shall open the SVC by transmitting the IDN of the parameter (SVC control, bits 5, 4, 3 = 001, element 1). Every access to the data block elements 2-7 refers to the last transmitted IDN. The slave shall respond by transmitting the data status or the procedure command acknowledgment of the received IDN.
- Closing SVC: The SVC of the previous IDN shall be closed by opening the SVC for a new IDN. Optionally the master may close the SVC by transmitting the data block element 0 (SVC control, bits 5, 4, 3 = 000, element 0).

6.2.3 Selection of data block element

During the next step, the master shall indicate which elements of the data block shall be processed. For this purpose, the master shall set bits 5, 4 and 3 accordingly in the SVC control.

6.2.4 Changing of data block element

Changing the data block element during transmission in progress shall be possible without an error message only if the following bits have the status given in Figure 33.

Table 76 – Condition for modifying data block elements

Information	SVC control bit	SVC status bit	bit value
Handshake bits equal	bit 0	bit 0	MHS = AHS
Busy		bit 1	0
SVC valid		bit 3	1

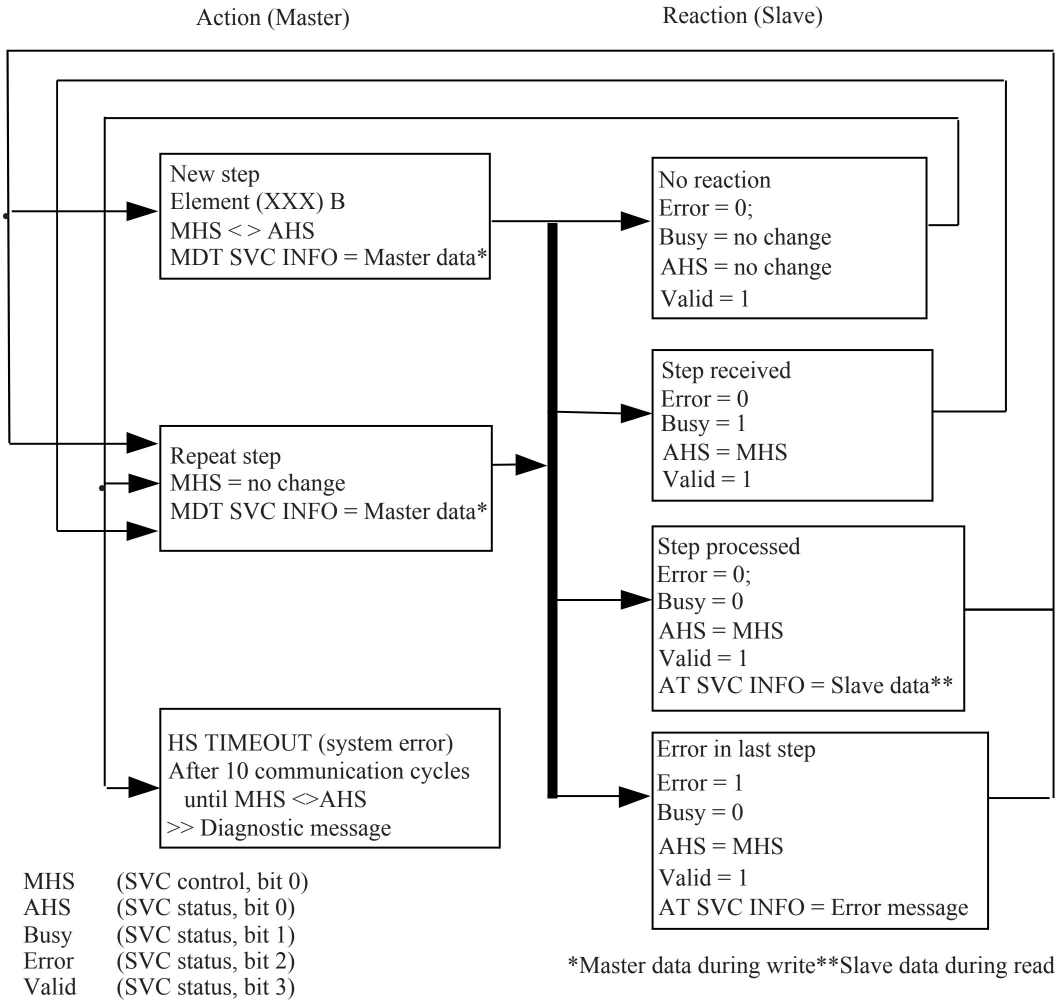

Figure 33 – Communication step proceeding diagram

6.2.5 Transmission steps

Depending on the length of the data block elements which need to be transmitted and of the length of the SVC INFO field, several steps shall be performed. Every step shall transport four octets of data.

Table 77 shows the necessary steps for the individual data block elements of a parameter.

Table 77 – List of data block element and step numbers

Data block element	Description	Requirement	Number of steps
1	IDN	Mandatory	1
2	Name	Optional	1 to 64
3	Attribute	Mandatory	1

Data block element	Description	Requirement	Number of steps	
4	Unit	Optional	1 to 16	
5	Minimum value	Optional	1or 2	
6	Maximum value	Optional	1or 2	
7	Operation data	Mandatory	Fixed length:	1 or 2
			Variable length:	1 to 16 384
0	**Closing** the service channel		1	

The error messages “data block element transmission too long” shall be executed by the slave if the current data block element is transmitted completely and the master indicates the transmission in progress (SVC control, bit 2 = 0).

The error messages “data block element transmission too short” shall be executed by the slave if the current data block element is not transmitted completely and the master indicates the last transmission (SVC control, bit 2 = 1).

The master shall indicate a transmission in progress (SVC control, bit 2 = 0) or a transmission in progress (bit 2 = 0) or the transmission of the last 4 octets (SVC control, bit 2 = 1). If the length of the data block element is ≤ 4 octets, just one step is required. In this case the master shall set the last transmission (SVC control, bit 2 = 1).

Access to an IDN with variable length without operation data (data block element = 4 octets):

- The master does not know the length of the data block element and sets transmission in progress (SVC control, bit 2 = 0).
- Therefore the slave generates the error message “data block element transmission too long”.
- The master may read the same data block element with last transmission (SVC control, bit 2 = 1) to get the 4 octets lengths indication.

6.2.6 SVC valid

Master and slave shall read SVC information in each cycle, either on port1 or on port2 depending on topology. Should it fail on either port, it shall automatically read on the other port.

Master and slave shall always write identical SVC information on both ports.

The master shall evaluate the slave’s SVC answer only if it reads “SVC valid” = 1 (SVC status, bit 3). It shall not evaluate it if “SVC valid” = 0 (invalid).

The SVC valid is set to 1 by the slave in CP1.

6.2.7 Handshake bits

During SVC transmissions, the transport of every step shall be secured by two service transport handshake bits. These shall be the bits 0 in the SVC control (MHS) and in the SVC status (AHS).

For every new step during the transmission, the master shall toggle the MHS-bit. The slave shall recognize by the toggled MHS-bit that a new step needs to be executed. After the slave has received the required step and secured it for processing, it shall proceed to set its AHS-bit equal to the MHS-bit. By comparing the MHS-bit with the AHS-bit, the master and the slaves shall always be able to recognize the actual transport status during SVC transmission. see Table 78.

Table 78 – SVC channel evaluation

Perspective	Condition	Description
Master	AHS bit = MHS bit (SVC valid = 1)	The step was received by the slave and secured, slave starts processing. The master shall wait for processing acknowledgment (busy = 0, bit 1 in the SVC status)
	AHS bit ≠ MHS bit or SVC valid = 0	The steps were not yet received or secured by the slave. The master shall repeat the last step
Slave	MHS bit = AHS bit	The master does not require a new step, slave repeats the last step
	Master MHS-bit ≠ slave AHS-bit	The master requests a new step

The service transport handshake bits shall enable the slaves and the master to insert “wait cycles” during the transmission, for example:

- if more than one cycle will be required for receiving or transmitting a step;
- if a new step has not been recognized due to an error during the transmission;
- if the master does not issue any new steps at this time.

During every “wait cycle”, the master or the slave shall transmit the data of the previous communication cycle into the SVC INFO field.

The slave shall acknowledge the proper reception of a step by matching its AHS-bit within a maximum of 10 communication cycles.

6.2.8 Read/Write

The master shall indicate in bit 1 of the SVC control whether the element will be read or written to. While writing, the MDT SVC INFO field shall be filled with the appropriate data for the slave (contents of the AT SVC INFO field are invalid).

If reading is selected, the slave shall insert the appropriate data in the AT SVC INFO field (contents of the MDT SVC INFO field are invalid).

6.2.9 Busy bit

The slave shall be able to control any SVC transmission through the busy bit. The busy bit shall indicate that the slave is processing or just finishing the requested step at this time. The master shall not be allowed to start the next step until the slave sends the processing acknowledgement (busy bit = 0). The busy bit shall allow the slave to prevent the master from forcing the steps on the slave too quickly.

A maximum time if the busy bit = 1 is not specified. The maximum acceptable dwell time and the reaction in the master is related to the application.

6.2.10 Service channel initialization

The service channel shall be initialized during CP1 and be functional for the remainder of the communication phases.

In CP1, each service channel shall start with the following status:

- the master shall set the MHS-bit to 1 in the MDT SVC control word,

- the slave shall set “SVC valid” and AHS to 1 in the AT SVC status word; if it was requested by the master in CP1,
- all other bits in SVC control or SVC status shall be set to 0,
- all bits in the SVC INFO fields are invalid.

Starting with CP2, the MDT SVC (control and INFO) and the AT SVC (status and INFO) become valid.

During phase switching the state of MHS and AHS shall be saved in the master and the slaves. The saved state shall be restored in the next CP.

6.2.11 Reaction to SVC handshake timeout

A handshake (HS) timeout shall occur if any addressed slave does not acknowledge its AHS-bit in the Status word after 10 communication cycles in CP2 to CP4. During CP1, a slave shall be registered as not present if the AHS-bit has not been set to a logical 1 within the maximum identification time of the master (see Table 79).

Table 79 – Reaction to handshake timeout

CP	Reaction in master	Reaction in the slave
2-4	Display of an error message. The master responds with an error handling procedure that may be stored in the control unit and may then switch back to CP0	

6.2.12 Reaction to error messages in the service channel

A valid error message for the master is present in the service channel if the slave sets bit 2 in the SVC status to 1 and the AHS-bit of the slave equals the MHS-bit of the SVC control (see Table 80).

Table 80 – Reaction to error message

CP	Reaction in master	Reaction in the slave
2-4	Displaying an error message	The step currently being processed is interrupted, the busy bit (bit 1 – SVC status) is set to 0.

6.2.13 Service channel error messages

Should an error occur in the transport mechanism of the service channel (for example, if the lengths of the operation data differ between the master and the slave, or vice versa, or if the IDN is undefined), the slave shall announce it by setting the error bit (bit 2) in the status word and by writing an error code into the service INFO field of its AT.

The slave shall be allowed to report an error message only if a new processing step is issued by the master, which is in any of following cases:

- MHS-bit ≠ AHS-bit (step not yet secured);
- busy bit = 1 (step still in process).

If the slave recognizes an error, it shall ignore the actual step, interrupt and acknowledge by:

- setting the AHS-bit equal to the MHS-bit (if not already acknowledged in a previous cycle);

- setting the error bit to 1 (SVC status, bit 2);
- setting the busy bit to 0;
- setting the SVC valid to 1;
- sending the error codes in the AT SVC INFO field (see 4.6.7.3.3).

If the master intends to start a transmission of an element after an error message, the slave shall update the SVC status and AT SVC INFO.

All possible SVC error messages are shown in Table 81.

Table 81 – Error messages

Error code	Description
0x0nnn	General error
0x0000	No error in the service channel
0x0001	Service channel not open
0x0009	Invalid access to closing the service channel
0x1nnn	Element 1 (Identification number)
0x1001	IDN not available
0x1009	Invalid access to element 1
0x2nnn	Element 2 (Name)
0x2001	Name not available
0x2002	Name transmission too short
0x2003	Name transmission too long
0x2004	Name cannot be changed (read only)
0x2005	Name is write-protected at this time
0x3nnn	Element 3 (Attribute)
0x3002	Attribute transmission too short
0x3003	Attribute transmission too long
0x3004	Attribute cannot be changed (read only)
0x3005	Attribute is write-protected at this time
0x4nnn	Element 4 (Unit)
0x4001	Unit not available
0x4002	Unit transmission too short
0x4003	Unit transmission too long
0x4004	Unit cannot be changed (read only)
0x4005	Unit is write-protected at this time
0x5nnn	Element 5 (Minimum value)
0x5001	Minimum input value not available
0x5002	Minimum value transmission too short
0x5003	Minimum value transmission too long
0x5004	Minimum value cannot be changed (read only)
0x5005	Minimum value is write-protected at this time
0x6nnn	Element 6 (Maximum value)
0x6001	Maximum input value not available
0x6002	Maximum value transmission too short
0x6003	Maximum value transmission too long

Error code	Description
0x6004	Maximum value cannot be changed (read only)
0x6005	Maximum value is write-protected at this time
0x7nnn	Element 7 (Operation data)
0x7002	Operation data transmission too short
0x7003	Operation data transmission too long
0x7004	Operation data cannot be changed (read only)
0x7005	Operation data is write-protected at this communication phase
0x7006	Operation data is smaller than the minimum input value
0x7007	Operation data is greater than the maximum input value
0x7008	Invalid operation data: Configured IDN will not be supported, invalid bit number or bit combination
0x7009	Operation data write protected by a password
0x700A	Operation data is write protected, it is configured cyclically. (IDN is configured in the MDT or AT. Therefore writing via the service channel is not allowed).
0x700B	Invalid indirect addressing: (for example, data container, list handling)
0x700C	Operation data is write protected, due to other settings. (for example, operation mode, sub-device is enabled etc.)
0x700D	Invalid floating point number
0x700E	Operation data is write protected at parameterization level
0x700F	Operation data is write protected at operating level
0x7010	Procedure command already active
0x7011	Procedure command not interruptible
0x7012	Procedure command at this time not executable (for example, in this phase the procedure command cannot be activated).
0x7013	Procedure command not executable (invalid or false parameters)
0x7014	The received current length of list parameter does not match to expectation
0x71nn	Segmentwise SVC access for parameters with variable length
0x7101	IDN in S-0-0394 not valid
0x7102	Empty list in S-0-0397 not allowed for write access
0x7103	Maximum length of the list in S -0-0394 is exceeded by take-over of the list segment.
0x7104	Read access only: The length of the list segment as of the list index exceeds the current length of the list in S -0-0394.
0x7105	IDN in S-0-0394 is write protected
0x7106	Operation data in list segment is smaller than the minimum input value
0x7107	Operation data in list segment is greater than the maximum input value
0x7108	Invalid list index in S-0-0395
0x7109	Parameter in IDN S-0-0394 does not have variable length
0x710A	IDN S-0-0397 not permitted as operation data in S-0-0394
0x8nnn	(reserved for master internal error codes)
0xAnnn	Reserved
0xBnnn	Reserved
0xCnnn	Reserved
0xDnnn	Error codes are not generated and transmitted via SVC
0xD000	no error

Error code	Description
0xD001	service channel (temporarily) not available
0xD002	service channel engaged by an application
0xD003	service channel busy, slave is processing previous request
0xD004	Type 19 slave not reachable
0xD005	service channel transaction aborted
0xD006	writing this element is not supported by the service channel
0xEnnn	(reserved for master internal error codes)
0xFnnn	(reserved for master internal error codes)
All other codes shall be reserved.	

6.2.14 Procedure command functions via the service channel

6.2.14.1 General

The procedure command functions shall be transmittable through the service channel. A procedure command is considered as a special type of non-cyclic data which invokes fixed functional processes in the slaves and the master. These processes may take up some time. Hence, a procedure command shall only cause a functional process to start. After a procedure command has started its function, the service channel shall become available again immediately for the transmission of non-cyclic data or for more procedure commands.

Contrary to non-cyclic data transmission, whose proceeding shall be finished with the last transmitted step, the end of a procedure command during a lengthy procedure command execution shall be indicated by the procedure command change bit (bit 5 in the device status). The master shall also be able to interrupt a procedure command during its execution.

Every procedure command shall be assigned to an IDN. Not all data block elements of the parameter are defined, however, and other data block elements have a predetermined form. Some procedure commands are not interruptible.

6.2.14.2 Procedure command control and acknowledgment

A procedure command function shall always prompt a procedure command control from the master to the slave and a procedure command acknowledgment from a slave to the master. The procedure command control shall be data block element 7 of the parameter (data block element 7 is always represented as a bit string for procedure commands). see Table 82.

Procedure command control shall allow procedure commands to be:

- set;
- enabled for execution;
- interrupted during execution;
- canceled.

The slave shall acknowledge the transmission of a procedure command from the master via the service channel with its AHS-bit, the busy bit and the SVC valid in its SVC status.

Table 82 – Structure of Procedure command control

Bit no,	Value	Description
15-2		(reserved)
1-0		procedure command control (PCC)
	00	procedure command is not activated or canceled
	01	procedure command is set and interrupted
	10	procedure command is canceled
	11	procedure command is set and enabled

When starting the initializing (CP0), all procedure commands inside the master shall be canceled and then the procedure command control shall be updated appropriately internally in the master.

The procedure command acknowledgment shall be part of the data status (see Table 83).

In order to receive a procedure command acknowledgment, the master shall write the IDN of the procedure command via the service channel.

When acknowledging a procedure command, the slave shall indicate the current status of the procedure command as given in Table 83.

If the master activates a procedure command, it can take several communication cycles until the slave generates the corresponding procedure command acknowledgment. Therefore it is recommended that the master scans the procedure command acknowledgment as shown in Table 83.

Table 83 – Procedure command acknowledgment (data status)

Bit no,	Value	Description
15-9		(reserved)
8		Data valid
	0	Operation data is valid
	1	Operation data is invalid
7-4		(reserved)
3-0		Procedure command acknowledgment (PCA)
	1111	error: procedure command execution impossible (procedure command change bit is set)
	0111	procedure command is activated but not yet executed
	0101	procedure command execution is interrupted
	0011	procedure command has been executed correctly (procedure command change bit is set)
	0001	procedure command is set
	0000	procedure command has been canceled
		All other codings are reserved

With the beginning of initialization (CP0), all procedure commands within the slave shall be disabled and then the procedure command acknowledgment shall be updated appropriately internally in the slave.

6.2.14.3 Procedure command change bit

In order to inform the master of the end of a procedure command being executed in the slave, the procedure command change bit is defined in the device status (bit 5).

Only the following changes in the Procedure Command Acknowledgment (PCA) shall set the procedure command change bit:

- procedure command executed correctly (positive acknowledgment);
- error, procedure command execution impossible (negative acknowledgment).

The procedure command change bit (PCB) shall not indicate any other change of the procedure command acknowledgment (for example, an interrupt).

The master shall read the data status by writing the IDN of the procedure command and check the procedure command acknowledgment contained therein. This indicates whether the procedure command was executed positively or negatively.

At negative procedure command acknowledgment, it is recommended that the master reads the diagnosis (if desired) before the procedure command is canceled.

If a procedure command is canceled by the master, all the effects of the procedure command on the procedure command change bit in the slave shall be canceled as well. If the master has activated several procedure commands concurrently, all resulting procedure command acknowledgments shall be checked after setting the procedure command change bit in order to determine which procedure command caused the change.

As a rule, the master shall cancel a procedure command after it has been processed, irrespective of whether it was acknowledged positively or negatively.

A procedure command shall be canceled by setting bit 0 in the procedure command control to 0. This shall be independent from the actual procedure command execution state. If the slave recognizes that a procedure command is canceled it shall set the procedure command acknowledgment to 0 and set the Busy to 0 simultaneously.

The state machine in Figure 34 describes the allowed state changes for procedure commands.

For Procedure Command Control (PCC), only the values from 0x00 to 0x03 are allowed. If the value is invalid, the slave shall generate the error message “invalid operation data” (0x7008) in the SVC INFO.

A state change to “procedure command not set” (PCA = 0x00), shall only be possible by canceling the procedure command.

If more than one procedure command execution is active and the “Procedure Command Change Bit” (PCB) is set by more than one procedure command, this bit is reset in the device status when all procedure commands have been canceled which had set the bit.

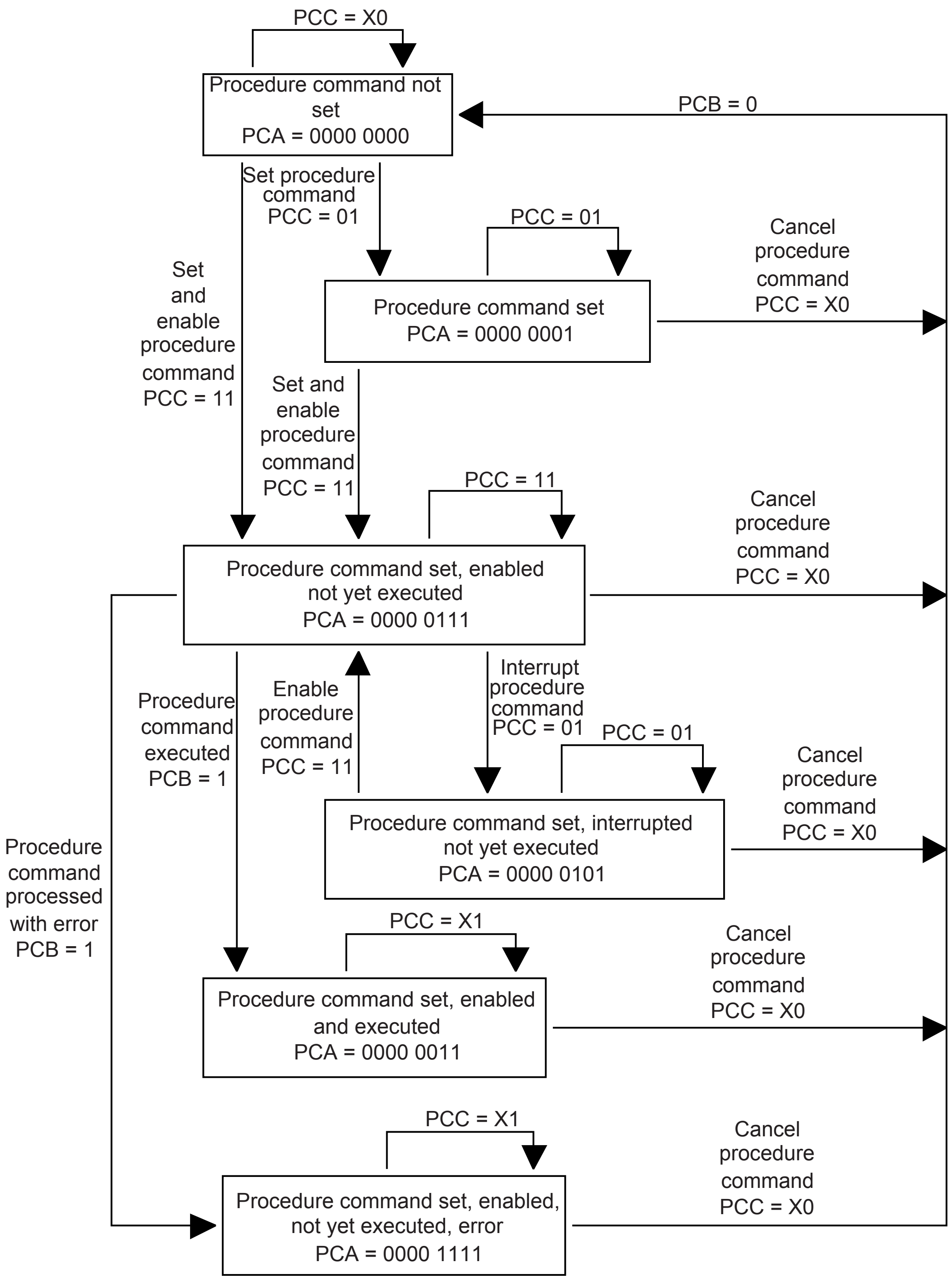

Figure 34 – State machine for procedure command execution

Figure 35 shows the sequence of procedure command handling that shall be met by the master.

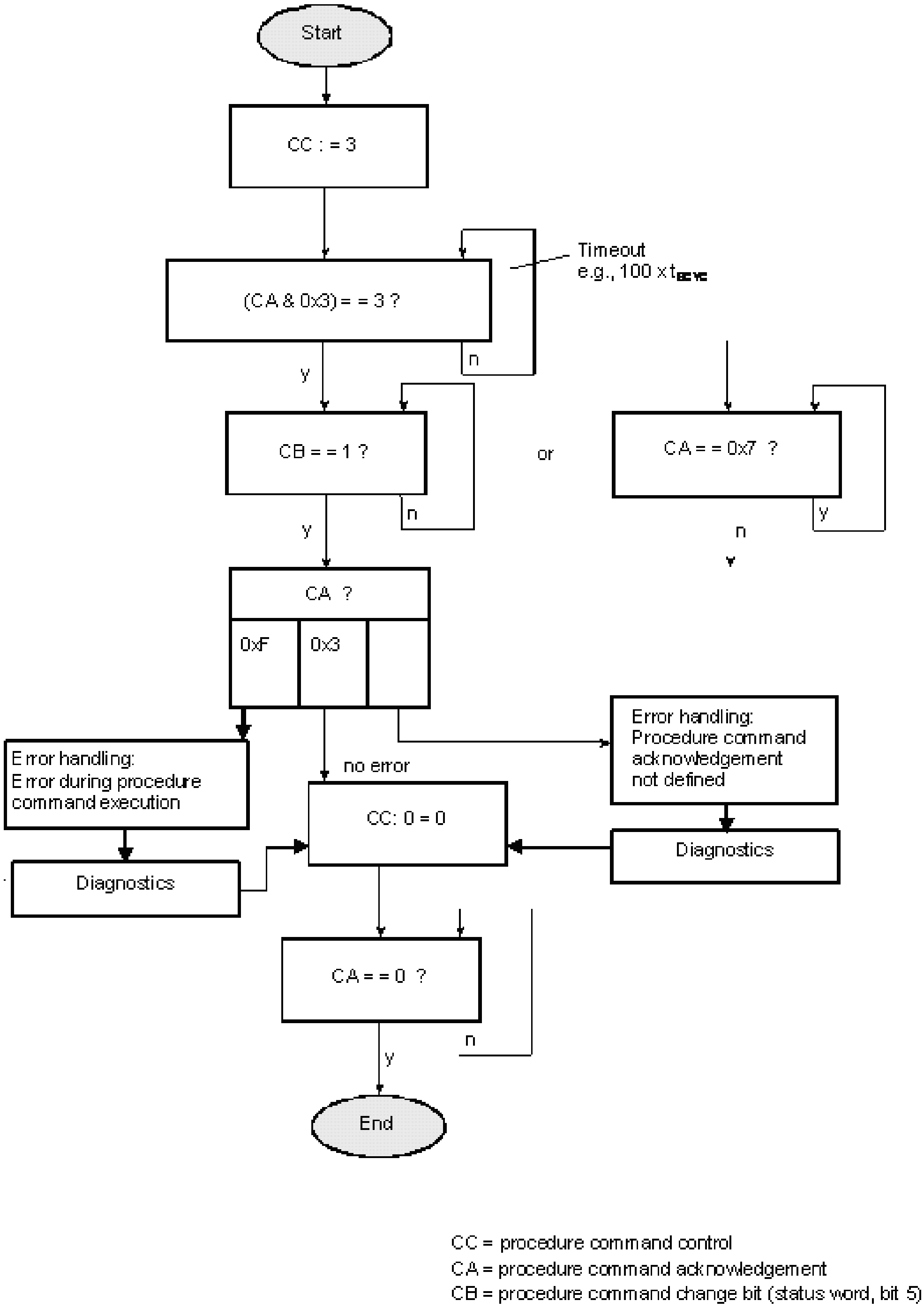

Figure 35 – Interaction of procedure command control and acknowledgement

6.2.14.4 Procedure command execution

In the following Figure 36, Figure 37 and Figure 38, the interactions between the master and the slave are represented, including procedure command executions with or without interruption and procedure command executions with error messages.

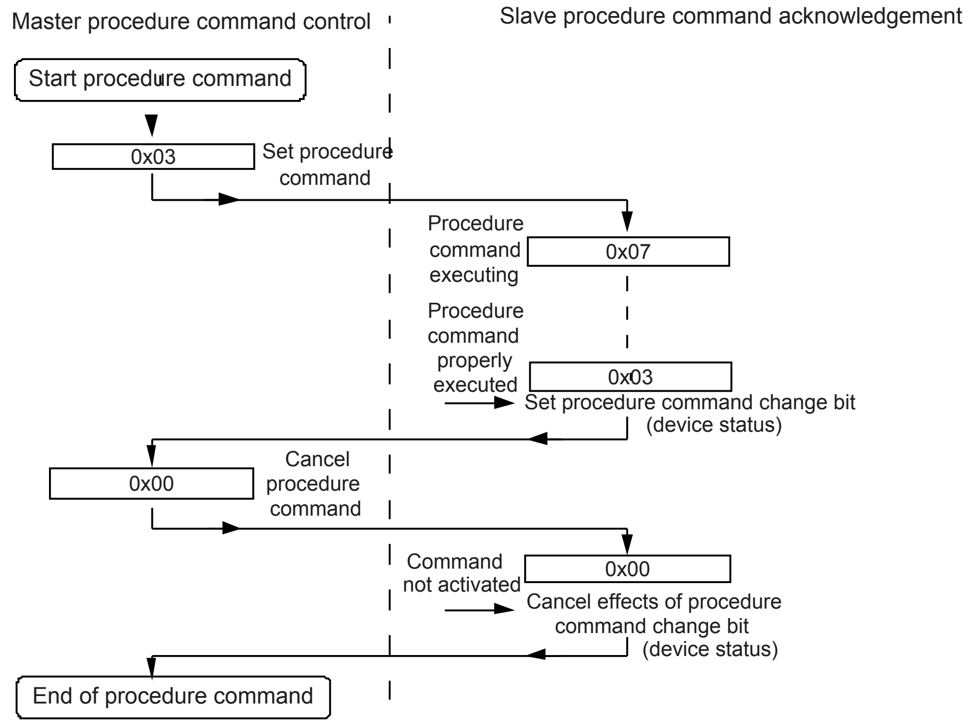

Figure 36 – Procedure command execution without interrupt

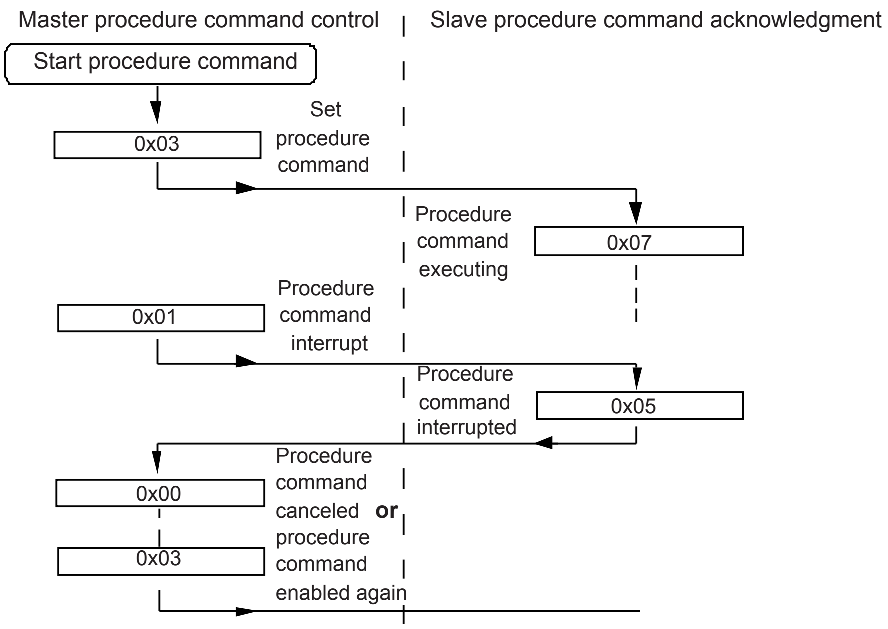

Figure 37 – Procedure command execution with interrupt

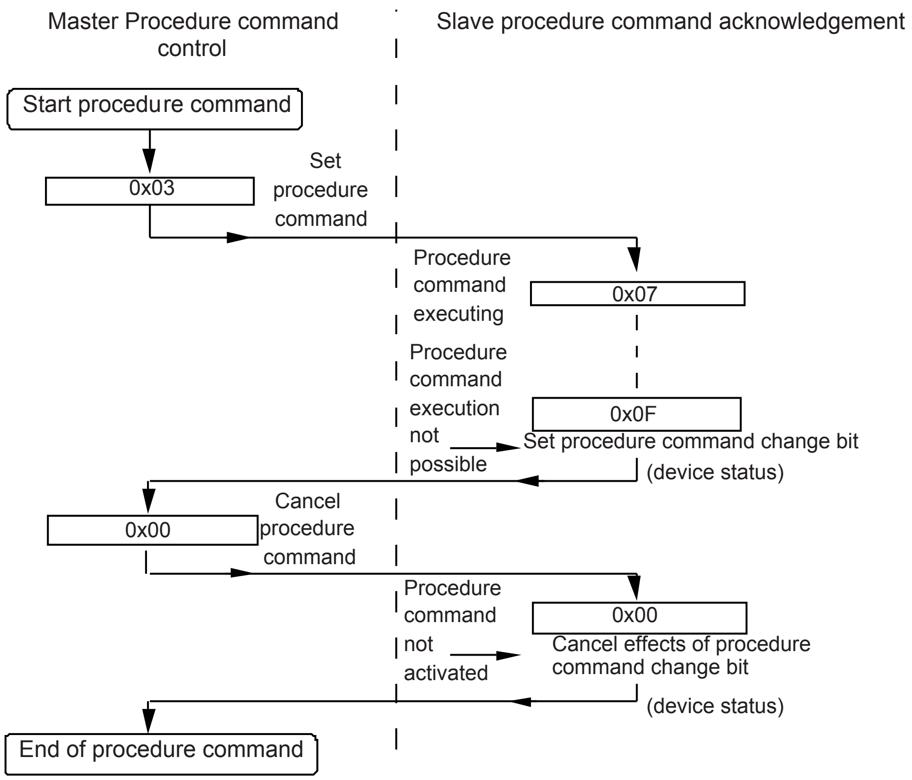

Figure 38 – Procedure command execution with error message

6.3 RT Channel

6.3.1 Introduction

Cyclic data is exchanged between all devices in a Type 19 network in communication phase CP4 according to the configuration given by the Initiate_cyclic_communication request (see 5.2).

6.3.2 Read_Cyclic (RDC)

Cyclic data is read by a DL user using the Read_Cyclic (RDC) request.

6.3.3 Write_Cyclic (WRC)

Cyclic data is written by a DL user using the Write_Cyclic (WRC) request. The cyclic data is transmitted in the next communication cycle of the Type 19 network.

6.3.4 Notify_Cyclic_Data (NCD)

Upon reception of a DLPDU of Type MDT0-MST the DL generates a Notify_Cyclic_Data (NCD) indication for the DL user.

6.4 Transmission and activation of Type 19 time

The Type 19 time is 8 octets long and is transmitted in fragments of 2 octets in the Extended field of MDT0. The fragmentation is controlled by 2 control bits (bits 31-30) in the Extended field and the Cycle CNT of MST. The Cycle CNT uses the bits 6-4 in MDT phase and bit 5 in MDT type. Each time fragment is transferred Initiate_cyclic_communication reqd in two consecutive communication cycles, therefore the transfer is tolerant of one telegram loss. The following sequence shall be handled by the master:

- Cycle CNT = 0 and 1: bits 0-15 of Type 19 time are transmitted in time fragment

- Cycle CNT = 2 and 3: bits 16-31 of Type 19 time are transmitted in time fragment
- Cycle CNT = 4 and 5: bits 32-47 of Type 19 time are transmitted in time fragment
- Cycle CNT = 6 and 7: bits 48-63 of Type 19 time are transmitted in time fragment

The slave shall assemble all received time fragments to the Type 19 time, taking into account the 2 control bits and the Cycle CNT. If one time fragment is invalid or loss, then the slave shall discard all received time fragments. The slave shall wait to the next transmission of the Type 19 time.

Control bit - Time fragment valid

- If the Time fragment valid bit is set to 1 then the data in the field "Time fragments" are valid.
- The slave can use the time fragment only, if this bit is set to 1.
- The master shall set this bit to 1 only, when the Cycle CNT of MST is 0.
- The master may set the valid bit to 0 at any time.

Control bit - Activate time

- In a continuous transfer of time fragments this bit is toggled when the master has transmitted a new time
- The master shall toggle this bit only, when the Cycle CNT of MST is 0.
- The slave shall activate the Type 19 time if this bit has toggled.

Calculation of Time forecast in the master

The transfer and activation of the Type 19 time takes a certain delay time, therefore, the master sends a predicted time to the slave. The master stores this predicted time at start of communication cycle only if the Cycle CNT of MST is 0. The stored time is transmitted to the slaves (see Figure 39).

In the slaves the activation of the Type 19 time occurs at the synchronization reference time (TSref). For this reason the master shall add the time delay to TSref. The predicted time results in the following formula:

$$\text{TIME forecast} = \text{TIME current} + 8 \cdot t_{Scyc} + \text{Time delay}$$

NOTE Time delay is the time between the start of the communication cycle in the master and the time TSref in the slave.

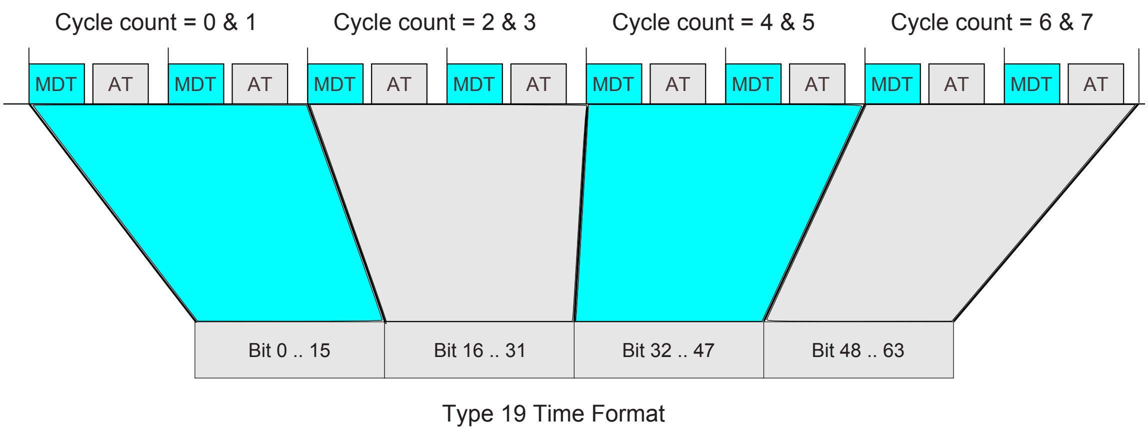

Figure 39 – Type 19 Time Transmission

6.5 Multiplexing of real-time data with data containers

6.5.1 General

The multiplexing of parameters offers additional functions in the communication.

This means:

a) In the same place of the real-time data field different parameters may be transferred;
b) It is not necessary to change the telegram configuration by a new phase initialization;
c) The access of discrete list elements works by this procedure.

Multiplexing uses:

a) Parameters in form of containers, for transmitting different parameters;
b) Parameters in form of lists, for listing all parameters to be transmitted;
c) Parameters in form of pointers, to address the parameter which will be transmitted;
d) Parameters in form of indices, to address the list element which will be transmitted in case a list parameter is addressed.

For multiplexing there are two methods available:

a) Standard (standard data container)
b) Extended (extended data container)

This function group includes the following IDNs:

- S-0-0360 MDT data container A1
- S-0-0361 MDT data container B1
- S-0-0362 MDT data container A list index
- S-0-0363 MDT data container B list index
- S-0-0364 AT data container A1
- S-0-0365 AT data container B1
- S-0-0366 AT data container A list index
- S-0-0367 AT data container B list index
- S-0-0368 Data container A pointer
- S-0-0369 Data container B pointer
- S-0-0370 MDT data container A&B configuration list
- S-0-0371 AT data container A&B configuration list
- S-0-0444 IDN-list of configurable data in the AT data container
- S-0-0445 IDN-list of configurable data in the MDT data container
- S-0-0450 MDT data container A2
- S-0-0451 MDT data container A3
- S-0-0452 MDT data container A4
- S-0-0453 MDT data container A5
- S-0-0454 MDT data container A6
- S-0-0455 MDT data container A7
- S-0-0456 MDT data container A8
- S-0-0457 MDT data container A9

- S-0-0458 MDT data container A10
- S-0-0459 MDT data container B2
- S-0-0480 AT data container A2
- S-0-0481 AT data container A3
- S-0-0482 AT data container A4
- S-0-0483 AT data container A5
- S-0-0484 AT data container A6
- S-0-0485 AT data container A7
- S-0-0486 AT data container A8
- S-0-0487 AT data container A9
- S-0-0488 AT data container A10
- S-0-0489 AT data container B2
- S-0-0490 MDT data container A2 configuration list
- S-0-0491 MDT data container A3 configuration list
- S-0-0492 MDT data container A4 configuration list
- S-0-0493 MDT data container A5 configuration list
- S-0-0494 MDT data container A6 configuration list
- S-0-0495 MDT data container A7 configuration list
- S-0-0496 MDT data container A8 configuration list
- S-0-0497 MDT data container A9 configuration list
- S-0-0498 MDT data container A10 configuration list
- S-0-0500 AT data container A2 configuration list
- S-0-0501 AT data container A3 configuration list
- S-0-0502 AT data container A4 configuration list
- S-0-0503 AT data container A5 configuration list
- S-0-0504 AT data container A6 configuration list
- S-0-0505 AT data container A7 configuration list
- S-0-0506 AT data container A8 configuration list
- S-0-0507 AT data container A9 configuration list
- S-0-0508 AT data container A10 configuration list

6.5.2 Functionality of standard data container

6.5.2.1 General

The standard data containers offer multiplexed switching between different real-time data in MDT and AT with a separate addressing mechanism.

In order to use this mechanism the standard data containers shall be configured in MDT and AT.

Via standard data containers it is possible to

a) exchange more application data in MDT and AT in spite of limited length of the connections;
b) access discrete list elements by means of the list index parameters of MDT and AT;

c) transfer multiplexed application data in every communication cycle with a cycle time of tscyc * number of multiplex parameters by incrementing the addressing.

There are 2 data containers with 4 octets and 2 data containers with 8 octets length defined for MDT and AT.

6.5.2.2 Data containers (standard)

Several standard data container are defined for the MDT and AT, serving as placeholders. The contents of the data containers can be dynamically changed by the master as necessary.

The master writes parameters in the slave by using “MDT data containers”.

The master reads parameters from the slave by using “AT data containers”.

The specified “standard data containers“ are listed below:

- S-0-0360 MDT data container A1
- S-0-0457 MDT data container A9
- S-0-0361 MDT data container B1
- S-0-0459 MDT data container B2
- S-0-0364 AT data container A1
- S-0-0487 AT data container A9
- S-0-0365 AT data container B1
- S-0-0489 AT data container B2

For both the „MDT data containers“ and „AT data containers“ the combinations in Table 84 are allowed.

Table 84 – List of valid standard data container combinations

Short name	Length (octets)
A1	4
A9	8
A1 + B1	4 + 4
A9 + B1	8 + 4
A1 + B2	4 + 8
A9 + B2	8 + 8

As can be seen:

a) Maximum one "data container A" (A1 or A9) is allowed;
b) Maximum one "data container B" (B1 or B2) is allowed but only in addition to a "data container A".

The “MDT data container” combination is independent of the “AT data container” combination. Data containers shall be configured in CP2 only.

If a transmitted parameter is shorter than its data container, the parameter shall be placed to the lower part of the data container. In this case the higher part remains free respectively not valid.

6.5.2.3 Configuration of standard data container

a) Configuration lists
   1) The S-0-0370 MDT data container A&B configuration list shall contain all configured parameters of the MDT data containers.
   2) The S-0-0371 AT data container A&B configuration list shall contain all configured parameters of the AT data containers.
   3) These two configuration lists are writable in CP2 only.

b) IDN-lists of configurable parameters
   1) All configurable parameters for the MDT data container may be stored in the S-0-0445 IDN-list of configurable data in the MDT data container.
   2) All configurable parameters for the AT data container may be stored in the S-0-0444 IDN-list of configurable data in the AT data container.

The following Parameters are defined:

- S-0-0370 MDT data container A&B configuration list
- S-0-0371 AT data container A&B configuration list
- S-0-0445 IDN-list of configurable data in the MDT data container
- S-0-0444 IDN-list of configurable data in the AT data container

6.5.2.4 Addressing of standard data container

The data container pointer is the offset, within the data container configuration list (S-0-0370 MDT data container A&B configuration list and S-0-0371 AT data container A&B configuration list) from the start of the configuration list to the desired IDN. The master places the desired parameter in the MDT data container, while the slave places the desired parameter in the AT data container.

- Data container A pointer (S-0-0368) and
- Data container B pointer (S-0-0369) are specified.

Each data container pointer contains two 8-bit pointers.

- One 8-bit pointer addresses the IDNs in the MDT data container A&B configuration list (S-0-0370). The parameter of the selected IDN shall be placed in the MDT data container.
- The other 8-bit pointer addresses the IDNs in the AT data container A&B configuration list (S-0-0371). The parameter of the selected IDN shall be placed in the AT data container.

The data container pointers (S-0-0368&S-0-0369) shall be configured in the MDT since the master commands the slave the interpretation of the data containers (see Figure 40). Thereby, a switching of the parameters in the data container during a communication cycle is possible.

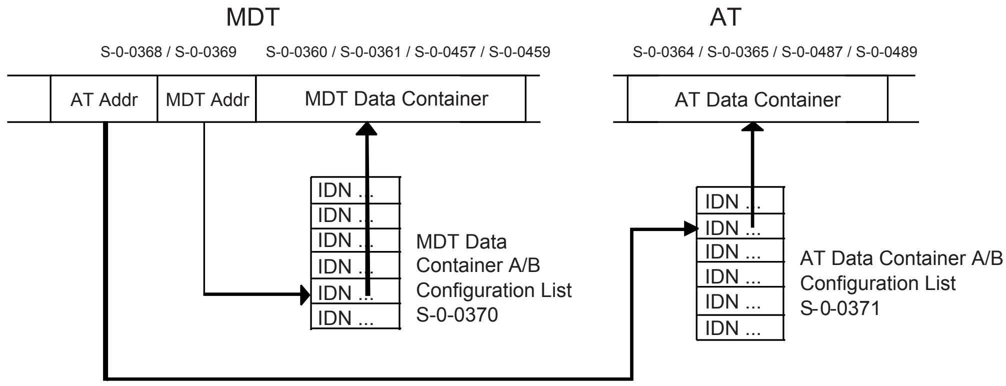

Figure 40 – Data container configuration without acknowledge (slave)

6.5.2.5 Acknowledgment of standard data containers

If the master requires an acknowledgment of the data container transmission, then it has 2 possibilities.

- To configure the identical data container pointer as well in the AT.
- To read the identical data container pointer via the SVC.

The slave shall generate the acknowledgment by copying the data container pointer of MDT to the AT.

The slave shall acknowledge the 8-bit pointer in the AT with the value 255 (not valid) if

- the pointer is situated outside of the configuration lists for the MDT or AT data container or
- the parameter is greater than the data container.

In this case the master and the slave shall ignore the corresponding data container (see Figure 41).

The master shall compare the data container pointer (S-0-0368&S-0-0369) of MDT and AT. If the result is equal, then the slave accepted the data in the MDT data container or wrote the requested data into the AT data container.

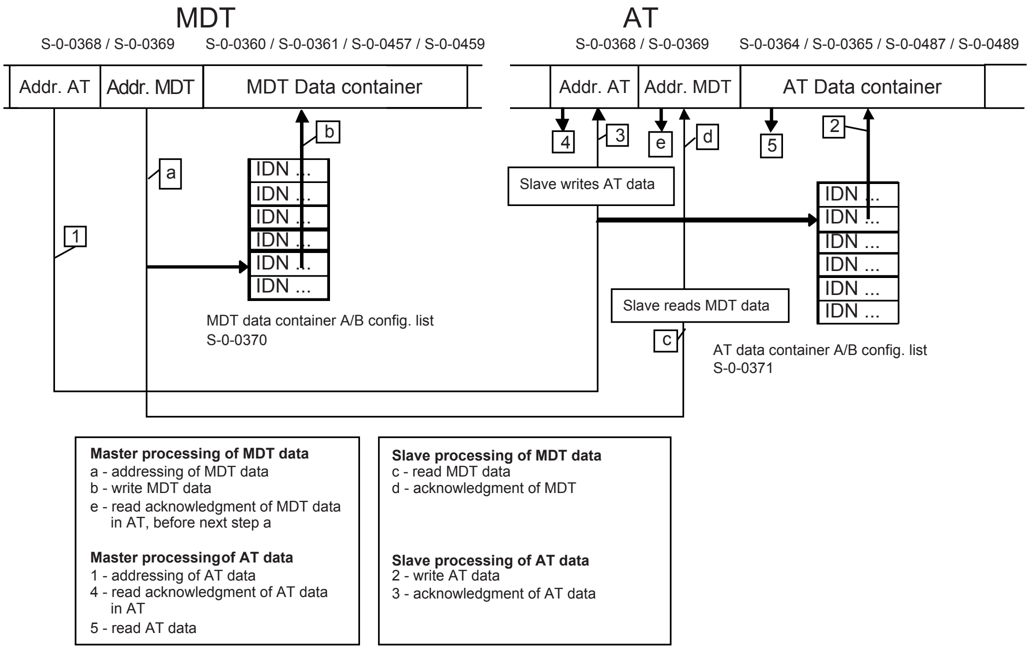

Figure 41 – Data container configuration with acknowledge (slave)

6.5.2.6 Addressing with list index (parameter lists)

If at least one parameter with „variable length“ (i.e. list parameter) has been programmed inside the “data container A/B configuration list“, this data would be too long for a data container. In this case

a) the corresponding list element will be addressed via the list index, therefore
b) the length of the list parameter shall not be changed.

There are 4 data container list indices specified, one for each data container:

a) IDN S-0-0362 MDT data container A list index;
b) IDN S-0-0363 MDT data container B list index;
c) IDN S-0-0366 AT data container A list index;
d) IDN S-0-0367 AT data container B list index.

Every data container list index consists of a 16 bit address.

Data container list indices shall be configured in the MDT since the master commands the slave the interpretation of the data containers. Thereby, a switching of the list elements in the data container during a communication cycle is possible.

Every data container list index consists of a 16 bit address.

Data container list indices shall be configured in the MDT since the master commands the slave the interpretation of the data containers. Thereby, a switching of the list elements in the data container during a communication cycle is possible.

If the master requires an acknowledgment of the data container transmission, then it has two possibilities.

a) To configure the identical data container list index as well in the AT;
b) To read the identical data container list index via the SVC.

The slave shall generate the acknowledgment by copying the data container list index of MDT to the AT.

The slave shall acknowledge the data container list index in the AT with the value 65 535 (not valid) if the data container list index is situated outside of the length of the list parameter. In this case the master and slave shall ignore the corresponding data container. Optionally the slave may acknowledge the corresponding data container pointer in the AT with the value 255 (not valid).

The master shall compare the data container list index of MDT and AT. If the result is equal, then the slave accepted the data in the MDT data container or wrote the requested data into the AT data container.

Figure 42 shows the processing of list elements via data containers with list index.

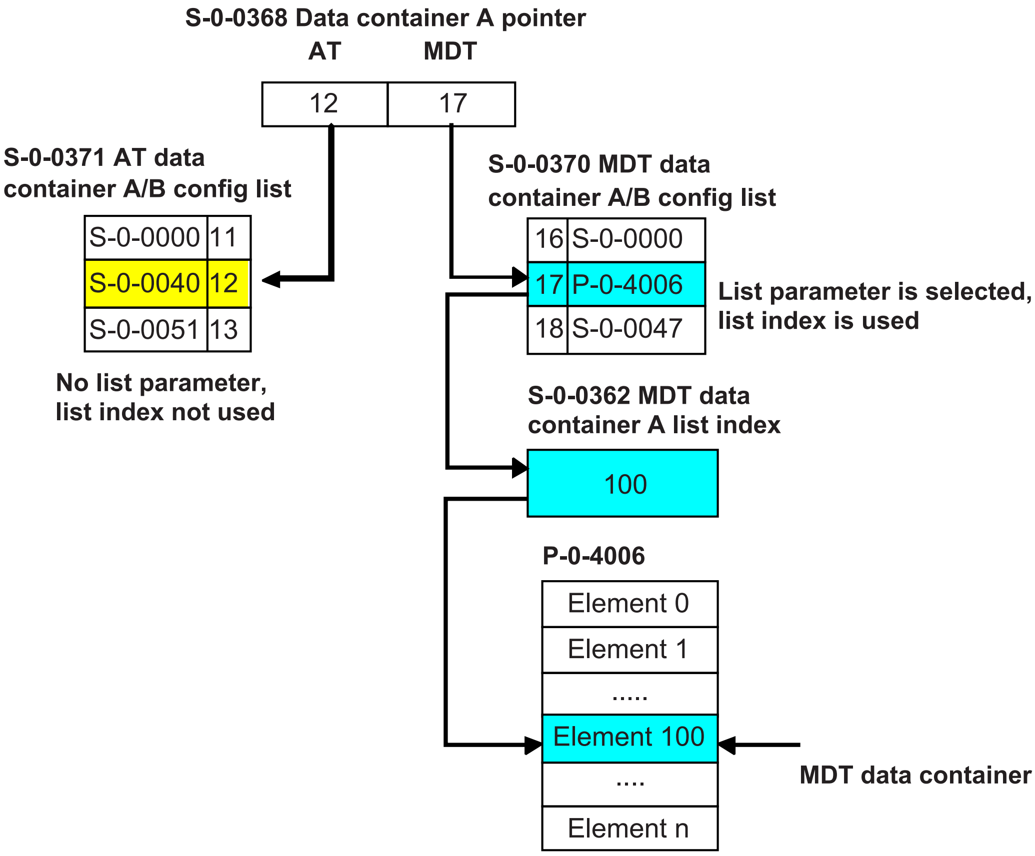

Figure 42 – Processing of list index in the MDT data

6.5.3 Functionality of extended data container (preferred function)

6.5.3.1 General

The extended data container offers multiplexed switching between different application data in MDT and AT with a common addressing mechanism.

In order to use this mechanism the data containers should be configured in MDT and AT.

Via extended data containers it is possible to

a) exchange more data in MDT and AT in spite of limited length of the connections;

b) access discrete list elements by means of list index parameters of MDT and AT;

c) transfer multiplexed application data in every communication cycle with a cycle time of tscyc * number of multiplex levels by incrementing the addressing.

There are 8 data containers with 4 octets and 2 data containers with 8 octets length defined for MDT and AT.

6.5.3.2 Data containers (extended)

There are 10 data containers defined for the MDT and another 10 for the AT, serving as placeholders. The contents of the data containers can be dynamically changed by the master as necessary.

The master writes parameters in the slave by using “MDT data containers”.

The master reads parameters from the slave by using “AT data containers”.

The specified “extended data containers“ are listed below:

a) MDT data container
   1) IDN S-0-0360 MDT data container A1
   2) IDN S-0-0450 MDT data container A2
   3) IDN S-0-0451 MDT data container A3
   4) IDN S-0-0452 MDT data container A4
   5) IDN S-0-0453 MDT data container A5
   6) IDN S-0-0454 MDT data container A6
   7) IDN S-0-0455 MDT data container A7
   8) IDN S-0-0456 MDT data container A8
   9) IDN S-0-0457 MDT data container A9
   10) IDN S-0-0458 MDT data container A10

b) AT data container
   1) IDN S-0-0364 AT data container A1
   2) IDN S-0-0480 AT data container A2
   3) IDN S-0-0481 AT data container A3
   4) IDN S-0-0482 AT data container A4
   5) IDN S-0-0483 AT data container A5
   6) IDN S-0-0484 AT data container A6
   7) IDN S-0-0485 AT data container A7
   8) IDN S-0-0486 AT data container A8
   9) IDN S-0-0487 AT data container A9
   10) IDN S-0-0488 AT data container A10

Any combination of "MDT data container" may be selected for the MDT. Any combination of "AT data container" may be selected for the AT.

Data containers shall be configured in CP2 only.

If a transmitted parameter is shorter than its data container, the parameter shall be placed to the lower part of the data container. In this case the higher part remains free respectively not valid.

6.5.3.3 Configuration of extended data container

a) Configuration lists (general)
   1) Each MDT data container and AT data container corresponds exclusively with its data container configuration list.
   2) The configuration lists can be written in CP2 only.
b) Configuration lists (MDT)
   1) All used MDT data container configuration lists shall have the same length.
   2) Unused list elements shall be programmed with IDN S-0-0000.
   3) If parameters of "variable length" (i.e. list parameter) are programmed in a MDT data container configuration list, then these list parameters shall have the same length.
c) Configuration lists (AT)
   1) All used AT data container configuration lists shall have the same length.
   2) Unused list elements shall be programmed with IDN S-0-0000.
   3) If parameters of "variable length" (i.e. list parameter) are programmed in a AT data container configuration list, then these list parameters shall have the same length.
d) Configurable parameters
   1) All configurable parameters for the MDT data container are optionally stored in the IDN lists of configurable data (IDN S-0-0445).
   2) All configurable parameters for the AT data container are optionally stored in the IDN lists of configurable data (IDN S-0-0444).

During procedure command "CP3 transition check" (IDN S-0-0127) the slave may check the mentioned restrictions about same length of list parameters. In case of a negative result the procedure command generates the error code in the diagnostic number (IDN S-0-0390).

All configuration lists of extended data container are listed below:

a) MDT data container configuration lists
   1) IDN S-0-0370 MDT data container A/B configuration list
   2) IDN S-0-0490 MDT data container A2 configuration list
   3) IDN S-0-0491 MDT data container A3 configuration list
   4) IDN S-0-0492 MDT data container A4 configuration list
   5) IDN S-0-0493 MDT data container A5 configuration list
   6) IDN S-0-0494 MDT data container A6 configuration list
   7) IDN S-0-0495 MDT data container A7 configuration list
   8) IDN S-0-0496 MDT data container A8 configuration list
   9) IDN S-0-0497 MDT data container A9 configuration list
   10) IDN S-0-0498 MDT data container A10 configuration list
b) AT data container configuration lists
   1) IDN S-0-0371 AT data container A/B configuration list
   2) IDN S-0-0500 AT data container A2 configuration list
   3) IDN S-0-0501 AT data container A3 configuration list
   4) IDN S-0-0502 AT data container A4 configuration list
   5) IDN S-0-0503 AT data container A5 configuration list
   6) IDN S-0-0504 AT data container A6 configuration list
   7) IDN S-0-0505 AT data container A7 configuration list
   8) IDN S-0-0506 AT data container A8 configuration list

9) IDN S-0-0507 AT data container A9 configuration list

10) IDN S-0-0508 AT data container A10 configuration list

c) IDN lists of configurable data

1) IDN S-0-0445 IDN list of configurable data in the MDT data container

2) IDN S-0-0444 IDN list of configurable data in the AT data container

6.5.3.4 Addressing of extended data container

The data container pointer is the offset, within all used data container configuration lists from the start of the configuration list to the desired IDN. The master places the desired parameter in the MDT data container, while the slave places the desired parameter in the AT data container.

Only the Data container A pointer (IDN S-0-0368) is required and it applies to any data container.

The data container A pointer contain two 8-bit pointers.

- One 8-bit pointer addresses the IDNs in the MDT data container configuration lists (IDN S-0-0370, IDN S-0-0490 to IDN S-0-0498). The parameter of the selected IDN shall be placed in the MDT data container.
- The other 8-bit pointer addresses the IDNs in the AT data container configuration lists (IDN S-0-0371, IDN S-0-0500 to IDN S-0-0508). The parameter of the selected IDN shall be placed in the AT data container.

The data container A pointer (IDN S-0-0368) shall be configured in the MDT since the master commands the slave the interpretation of the data containers.

Figure 43 shows an example of configuration lists with a number of 32 levels.

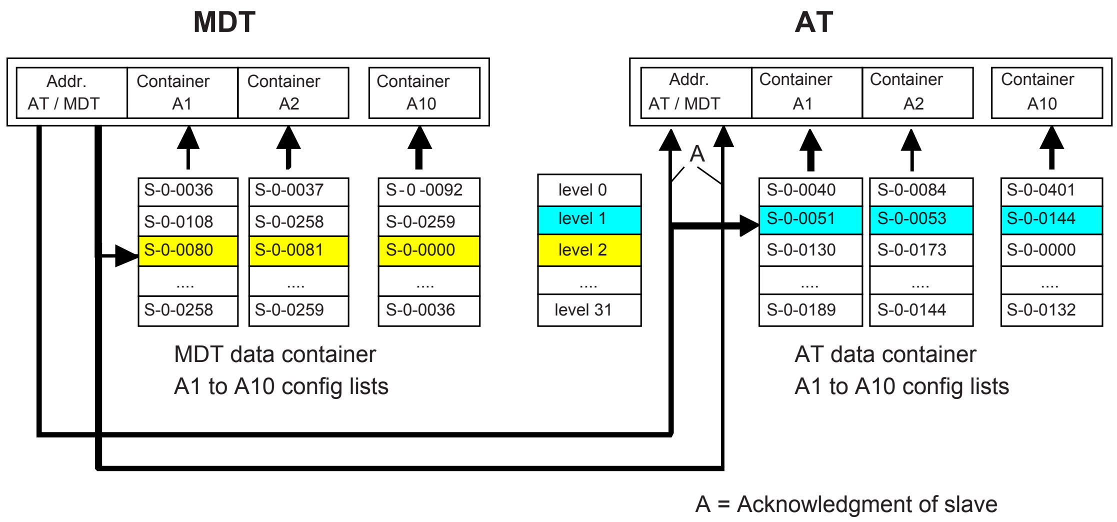

Figure 43 – Structure of extended data container

6.5.3.5 Acknowledgment of extended data container

If the master requires an acknowledgment of the data container transmission, then it has 2 possibilities.

- To configure the data container A pointer as well in the AT;
- To read the data container A pointer via the SVC.

The slave shall generate the acknowledgment by copying the data container A pointer of MDT to the AT.

The slave shall acknowledge the 8-bit pointer in the AT with the value 255 (not valid) if

- the pointer is situated outside of the configuration lists for the MDT or AT data container or
- the parameter is greater than the data container.

In this case the master and the slave shall ignore all data containers for which the 8-bit pointer is responsible.

The master shall compare the data container A pointer (IDN S-0-0368) of MDT and AT. If the result is equal, then the slave accepted the data in the MDT data container or wrote the requested data into the AT data container.

6.5.3.6 Addressing with list index

If at least one parameter with „variable length“ (i.e. list parameter) has been programmed inside the “data container configuration list“, this data would be too long for a data container. In this case

- the corresponding list element will be addressed via the list index, therefore
- the length of the list parameter shall not be changed and
- the length of all selected list parameter shall have the same length.

There are 2 data container list indices specified:

- the MDT data container A list index (IDN S-0-0362) for all MDT data containers
- the AT data container A list index (IDN S-0-0366) for all AT data containers

Every data container list index consists of a 16 bit address.

The data container A list index shall be configured in the MDT since the master commands the slave the interpretation of the data containers. Thereby, a switching of the list elements in the data container during a communication cycle is possible.

If the master requires an acknowledgment of the data container transmission, then it has 2 possibilities.

- To configure the identical data container list index as well in the AT.
- To read the identical data container list index via the SVC.

The slave shall generate the acknowledgment by copying the data container list index of MDT to the AT.

The slave shall acknowledge the data container list index in the AT with the value 65535 (not valid) if the data container list index is situated outside of the length of one of the selected list parameters. In this case the master and slave shall ignore all corresponding data containers. Optionally the slave may acknowledge the corresponding data container pointer in the AT with the value 255 (not valid).

The master shall compare the data container list index of MDT and AT. If the result is equal, then the slave accepted the data in the MDT data container or wrote the requested data into the AT data container.

6.5.4 Data container diagnostic

The parameters in the data containers are checked during initialization as well as during operation in CP4. If a slave detects an error in the data container it generates the corresponding diagnostic message.

- Data not configurable: IDN cannot be configured in MDT or AT data container. It has to ensured that the IDNs in the configuration lists can be transmitted as cyclic data. The slave checks it with procedure command IDN S-0-0127.
- Invalid addressing: In CP4 the slave checks whether the addressing is outside of the configuration lists. In the case of error the appropriate addressing in AT is set to 255. The contents of the data containers are invalid. Therefore all configuration lists shall be programmed with the same length. Unused list elements have to be programmed with IDN S-0-0000.

In CP4 the slave checks whether the list index is outside of the list parameter. In in the case of error the appropriate list index is set to 65535. Optionally the appropriate addressing in AT is set to 255. Therefore all list parameters shall be programmed with the same length. Unused list elements have to be set to 0x0.

- If the addressing of the MDT data container is invalid, then the slave may set S-0-0390 Diagnostic number to 0xC30E0008.
- If the addressing of the AT data container is invalid, then the slave may set S-0-0390 Diagnostic number to 0xC30E0009.

6.6 Handling of Real-time bits

6.6.1 General

The function group Real-Time bits (FG RTB) contains 3 options to exchange real-time bits (for example signals or events) between Type 19 nodes.

- Option 1: Two real-time bits are defined in the connection control. These are part of the application data.
- Option 2: One RTB word container as producer and one as consumer.
- Option 3: One RTB list container as producer and one as consumer.

Real-time bits in the consuming connection are distinguished from real-time bits in the producing connection. All logical assignments shall be IDNs of binary operation data (for example, level of switching signals, bits etc.).

The master shall assign IDNs only, which are supported by the slave in the "IDN list of configurable real-time bits".

Any real-time bits activated through these assignments maintain their meaning until the master overwrites or erases them with S-0-0000 or until another IDN changes the logical assignment.

When there is a write access over the service channel to the operation data of an IDN which is assigned to a real-time bit, the slave generates the error message „operation data is write protected, it is configured cyclically (error code 0x700A)“ via the service channel.

This function group includes the following IDNs:

- S-0-0026 IDN allocation of producer RTB word container
- S-0-0027 IDN allocation of consumer RTB word container
- S-0-0144 Producer RTB word container
- S-0-0145 Consumer RTB word container

- S-0-0328 Bit allocation of producer RTB word container
- S-0-0329 Bit allocation of consumer RTB word container
- S-0-0398 IDN list of configurable real-time bits as producer
- S-0-0399 IDN list of configurable real-time bits as consumer
- S-0-1050.x.20 IDN Allocation of real-time bit
- S-0-1050.x.21 Bit allocation of real-time bit
- S-0-1080.x.02 Producer RTB list container
- S-0-1080.x.03 IDN allocation of producer RTB list container
- S-0-1080.x.04 Bit allocation of producer RTB list container
- S-0-1081.x.02 Consumer RTB list container
- S-0-1081.x.03 IDN allocation of consumer RTB list container
- S-0-1081.x.04 Bit allocation of consumer RTB list container

And the following Control and Status Bits

- C-CON/Real-time bit 1
- C-CON/Real-time bit 2

6.6.2 Real-time bits (RTB)

Two real-time bits are defined in the connection control (C-CON), which may be used with special assignments. Assignments are transmitted on demand via the service channel. The real-time bits are signals which indicate some selected status or event (for example, level of switching signals, bits etc.) in the master or the slaves. This status or event from the producer to the consumers is represented in real-time.

The consumer shall evaluate the real-time bits in the connection control only, if the monitoring of the connection mechanism was correct.

Real-time bits are assigned a logic meaning by means of the following assignments:

- The master uses assignment S-0-1050.x.20 and S-0-1050.x.21 to inform the producer and consumer which logical value is assigned to real-time bit 1 or real-time bit 2 in the connection control.
- For real-time bit 1 the master shall configure list element 0 of S-0-1050.x.20 and S-0-1050.x.21.
- For real-time bit 2 the master shall configure list element 1 of S-0-1050.x.20 and S-0-1050.x.21.
- The real-time bits shall be always write protected in a producing connection.

For using the real-time bits the following parameters are available:

- C-CON/Real-time bit 1
- C-CON/Real-time bit 2
- S-0-0398 IDN list of configurable real-time bits as producer
- S-0-0399 IDN list of configurable real-time bits as consumer
- S-0-1050.x.20 IDN Allocation of real-time bit
- S-0-1050.x.21 Bit allocation of real-time bit

6.6.3 RTB word container

Real-time bits (for example signals or events) can be exchanged between Type 19 nodes by means of the RTB word container (see S-0-0144 and S-0-0145). For this purpose, the RTB word container needs to be integrated in a producing and/or consuming connection. Bits in the RTB word container are definable by means of the IDN allocation of RTB word container (see S-0-0026 and S-0-0027) and of the Bit allocation of RTB word container (see S-0-0328 and S-0-0329).

The sequence of the IDNs in the IDN allocation determines the bit numbering scheme in the RTB word container. The first IDN (list element 0) of the IDN allocation defines bit 0, the last IDN (list element 15) defines bit 15 of the RTB word container. The Bit allocation defines for each allocated IDN which bit is used. If the Bit allocation is not supported by the slave, the bit 0 of the allocated IDN is configured automatically.

The example in Table 85 shows the allocation of IDNs and bits of the producer RTB container.

The RTB word container has a length of 2 octets and shall be always write protected in a producing connection.

For using the RTB word container the following parameters are available:

RTB word container in producing connection

- S-0-0026 IDN allocation of producer RTB word container
- S-0-0144 Producer RTB word container
- S-0-0328 Bit allocation of producer RTB word container
- S-0-0398 IDN list of configurable real-time bits as producer

RTB word container in consuming connection

- S-0-0027 IDN allocation of consumer RTB word container
- S-0-0145 Consumer RTB word container
- S-0-0329 Bit allocation of consumer RTB word container
- S-0-0399 IDN list of configurable real-time bits as consumer

6.6.4 RTB list container

Real-time bits (for example signals or events) can be exchanged between Type 19 nodes by means of the RTB list container (see S-0-1080.x.02 and S-0-1081.x.02). For this purpose, the RTB list container needs to be integrated in a producing and/or consuming connection. Bits in the RTB list container are definable by means of the IDN allocation of RTB list container (see S-0-1080.x.03 and S-0-1081.x.03) and of the Bit allocation of RTB list container (see S-0-1080.x.04 and S-0-1081.x.04).

The sequence of the IDNs in the IDN allocation determines the bit numbering scheme in the RTB list container. The first IDN (list element 0) of the IDN allocation defines bit 0, the last IDN (list element n) defines the bit (n) of the RTB list container. The Bit allocation defines for each allocated IDN which bit is used.

The example in Table 85 shows the allocation of IDNs and bits of a producer RTB container.

The RTB list container shall have a variable length with an even number of octets and shall be always write protected in a producing connection.

For using the RTB list container the following parameters are available:

RTB list container in producing connection

- S-0-0398 IDN list of configurable real-time bits as producer
- S-0-1080.x.02 Producer RTB list container
- S-0-1080.x.03 IDN allocation of producer RTB list container
- S-0-1080.x.04 Bit allocation of producer RTB list container

RTB list container in consuming connection

- S-0-0399 IDN list of configurable real-time bits as consumer
- S-0-1081.x.02 Consumer RTB list container
- S-0-1081.x.03 IDN allocation of consumer RTB list container
- S-0-1081.x.04 Bit allocation of consumer RTB list container

Table 85 – Example of IDN and bit allocation of RTB container

Bit number of RTB container	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	etc.
IDN allocation	S-0-0403	S-0-0013	S-0-0000	S-0-0013	S-0-0013	S-0-0012	S-0-0330	S-0-0403	S-0-0012	…
Bit allocation	0	5	x	9	0	4	0	1	1	…

NOTE RTB container Bit 0: Bit 0 of S-0-0403 is assigned. RTB container Bit 1: Bit 5 of S-0-0013 is assigned. RTB container Bit 2: is not used. RTB container Bit 3: Bit 9 of S-0-0013 is assigned. etc.

6.7 SMP

6.7.1 Definitions

The Type 19 Messaging Protocol uses the S-0-1101.x.01 SMP Container Data to transmit SMP messages or SMP fragments. The length of the SMP container may be between 4 and 258 octets without any restriction. The SMP container shall be configured in a connection. Therefore, the connection length shall be equal or greater than the SMP container. Two formats are defined for the SMP message resp. SMP fragment, these are the regular format and extended format. These two formats are selected with the EXTF bit in the Session Control Header. The length of SMP container is defined by the current length of S-0-1101.x.01.

Figure 44 shows the two structures of the SMP container.

- Regular format contains the session control header (2 octets) and the application data. The size of the application data may be configured between 2 octets and 256 octets. The size of application data is not changeable.
- Extended format contains the session control header and can be filled only partially with application data. The extended size of the transmitted application data is indicated in the last octet of the SMP container. Extended sizes of application data are changeable between 0 octet and 255 octets in every transmission. If the extended size is 0, then the consumer shall evaluate the session control header only.

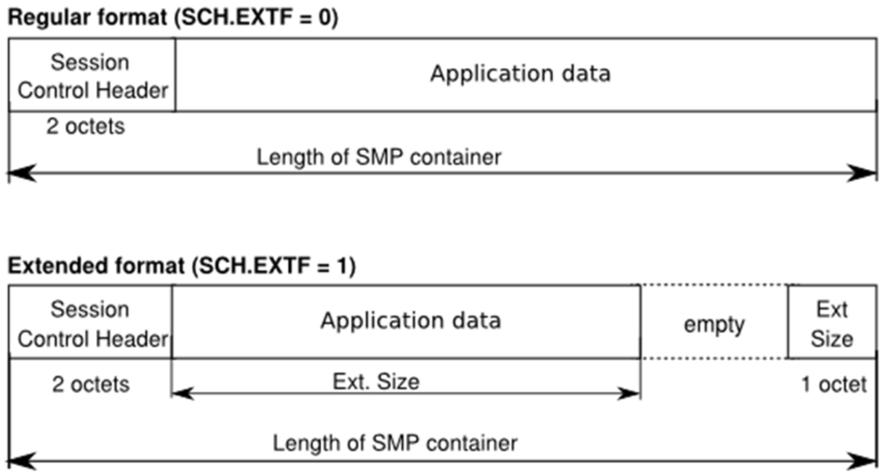

Figure 44 – Transport container

NOTE Length of SMP container = current length of S-0-1101.x.01 SMP Container Data (4…258 octets)

6.7.2 Structure of the Session Control Header (SCH)

The Session Control Header controls

- the fragmentation of SMP messages which are larger than the SMP containers used,
- the prioritization of SMP messages, and
- multiplexing several logic sessions through a single SMP container.

Its structure is as shown in Table 86.

Table 86 – Structure of the Session Control Header

Bit number	Value	Description
15-8	—	Session identifier (SID) Up to 255 sessions can be transmitted via a SMP container. The Session Identifier indicates the session of a SMP message for which the application data are intended.
	0-254	Valid values for the SID (0x00–0xFE)
	255	The value 255 (0xFF) is reserved for services of the transport layer
7	—	Last of Sequence (LOS): The received SMP container transported the last SMP fragment of a session sequence. The bits LOS & FOS controls the fragmentation of SMP messages that exceed the size of the SMP container.
	0	SMP fragment of a session
	1	last SMP fragment of a session
6	—	First of Sequence (FOS): The received SMP container

Bit number	Value	Description
		transported the first SMP fragment of a session sequence. All SMP fragments so far received on this priority level of this SMP container and not yet completed by a LOS are discarded. The bits LOS & FOS controls the fragmentation of SMP messages that exceed the size of the SMP container.
	0	SMP fragment of a session
	1	first SMP fragment of a session
5-4	—	Sequence Counter (SC) The sequence counter is managed separately for each session. It is incremented for every SMP fragment or SMP message transmitted within a session. The SC uncovers transmission failures that occurred within a fragmented SMP message.
	0-3	Sequence Counter value
3-2	—	Priority (PRI) Corresponds to the priority of the session through which this SMP message is sent. In accordance with this bit field, the transport layer can give high-priority SMP messages (for example cyclical data) preferential treatment in contrast to less time-critical SMP messages (for example non-cyclical configuration services).
	00	Priority 0 (highest level)
	01	Priority 1
	10	Priority 2
	11	Priority 3 (lowest level)
1	—	Extended format (EXTF) This bit indicates that the SMP container is not filled completely. In this case, the number of actually contained application data is entered in the last octet of the SMP container.
	0	Regular format: SMP container is filled completely
	1	Extended format: SMP container not filled completely. The consumer shall evaluate the extended size (last octet of SMP container).
0	Toggle	New Data Toggle (NDT) This bit indicates that the contents of the SMP container (i.e. the Session Control Header or the application

Bit number	Value	Description
		data) have been changed. The consumer needs to evaluate the SMP container only if the status of this bit has been toggled.
	0/1	Current value

6.7.3 Evaluation sequence of session control header by the consumer

The consumer shall evaluate the SMP message in a defined sequence as follows:

- The consumer shall check the New data toggle (NDT).
  - If it has toggled (NDT != int.NDT), then a new SMP fragment has been received. The consumer shall continue to evaluate the SMP fragment and shall change its expectation. Therefore, it shall toggle its internal bit "int.NDT".
- The consumer shall check the configuration of the received SID in S-0-1101.x.02 List of session identifiers.
  - If the SID is configured, then the consumer shall continue to evaluate the SMP fragment.
  - Otherwise, the consumer cancels the evaluation.
- The consumer shall check the configuration of the received PRI in S-0-1101.x.03 List of session priorities.
  - If the PRI is configured and corresponds to the SID, then the consumer shall continue to evaluate the SMP fragment.
  - Otherwise, the consumer cancels the evaluation.
- The consumer shall check the FOS.
  - If the FOS is set to 1, then the consumer shall reset the SMP fragment buffer of the received priority.
  - The consumer stores the received SMP fragment in the corresponding buffer of the priority.
  - The consumer shall set the received SC to its internal SC to generate an expectation for the next SMP fragment.
- If the FOS is set to 0, and the received SMP fragment is not the first, then the SC shall be checked with the internal SC.
  - If the check of SC is valid, then the consumer stores the received SMP fragment in the corresponding buffer of the priority.
  - If the check of SC is invalid, then consumer cancels the evaluation and the SMP fragment is discarded.
- If the FOS is set to 0, and the received SMP fragment is the first, then the consumer cancels the evaluation and the SMP fragment is discarded.
- If the SMP fragment has been stored and the LOS is 1, then the consumer shall assemble all stored SMP fragments of the buffer to a SMP message and pass it on to the application.

6.7.4 Multiplexing of two sessions (example)

Figure 45 shows an example for the transmission of two sessions through a transport container of a length of 10 octets. In the first session (SID 0x7), 8 octets of "application data A" are transmitted. The second session (SID 0x3) is used to transmit "application data B" with a length of 24 octets. The "application data B" are divided into three SMP fragments of 8 octets each. The example shows how the fragmentation is controlled by means of the FOS and LOS bits. The consumer stores the received SMP fragments in a fragment buffer until a Last-of-Sequence-Fragment (LOS = 1) is received. Independent of the currently transmitted session, the NDT bit toggles with each new SMP message. If no application data is ready for transmission in the producer, the NDT will not be toggled in the next SMP message. In this case, the consumer will not evaluate the other bits of the session control header and the application data of the SMP container. The SC is managed separately for each session and is incremented with every SMP message pertaining to a session. This assures that a low priority session can be interrupted by a session of higher priority. After the high-priority transmission is completed, the interrupted session will be resumed.

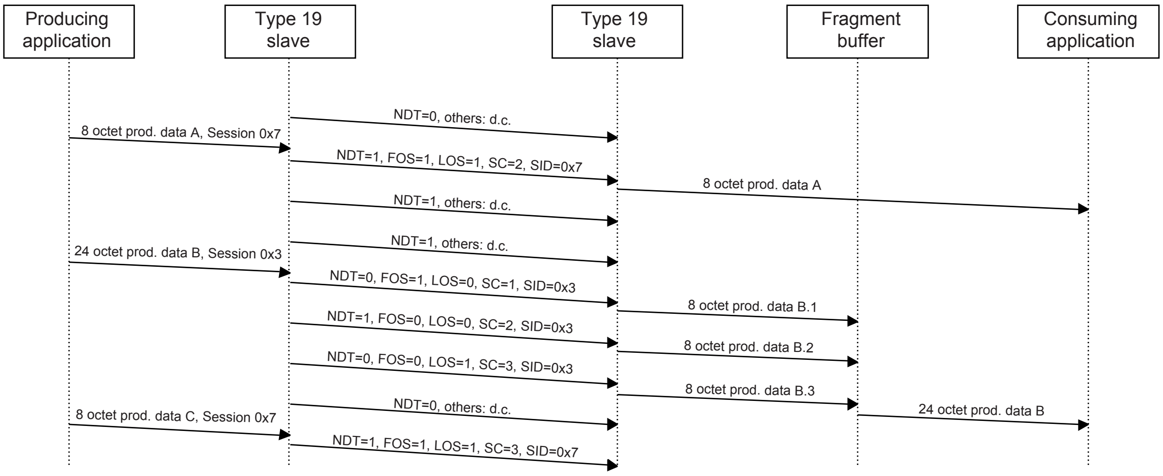

Figure 45 – UML Sequence Diagram: Multiplexing of two sessions (Example)

6.7.5 Priorization

The priority (PRI) field of the session control header controls the transmission order of SMP fragments. SMP fragments with a higher priority (lower value in the PRI field) shall be transmitted first. SMP fragments on the same priority level shall be transmitted in the same order as they were generated by SMP's session layer.

This prioritization scheme has the following characteristics:

- If a SMP message is to be sent while a transmission with lower priority is in progress, the low-priority SMP message will be interrupted. The higher-priority SMP message will be be sent first. After it is completed, the low-priority SMP message will be resumed.
- If an SMP message is to be sent while a transmission with equal or higher priority is in progress, the new SMP message is queued until all pending SMP messages of higher or equal priority have been transmitted.

One possible implementation would be a set of four FIFO queues (one for each priority level) that store the outgoing SMP fragments. When a new message is sent via SMP, the session layer will split the application data into SMP fragments. These SMP fragments are pushed to the end of the FIFO queue associated with the priority level of the sending session. Whenever a new SMP fragment can be sent, the SMP network layer will take the SMP fragment from the highest priority non-empty queue and copy it into the SMP container.

6.7.6 Diagnosis of SMP

Diagnosis information is collected centrally in parameter IDN S-0-1100 SMP Diagnosis. This diagnosis contains the following structure elements:

a) IDN S-0-1100.0.1 Diagnostic counter sent SMP fragments
b) IDN S-0-1100.0.2 Diagnostic counter received SMP fragments
c) IDN S-0-1100.0.3 Diagnostic counter discarded SMP fragments

6.7.7 Definition of SMP containers

Parameter IDN S-0-1101 SMP Transport container is defined for the transport of SMP fragments through the SMP container. The structure of this container is defined by the following structure elements:

a) IDN S-0-1101.x.01 SMP container data
b) IDN S-0-1101.x.02 List of the session identifiers
c) IDN S-0-1101.x.03 List of the session priorities

The structure element IDN S-0-1101.x.1 is configured in a SMP container; thus, it contains the application data transmitted in this SMP container.

The other structure elements describe the sessions currently active for this SMP container.

The lists in IDN S-0-1101.x.2 and IDN S-0-1101.x.3 shall have the same actual length. List elements with identical index describe one session.

6.7.8 Example

In the following example, 3 sessions are active in the SMP container 7 (S-0-1101.7.01):

- Session 0x04, priority 0
- Session 0x05, priority 3
- Session 0xF3, priority 0

Table 87 – Lists in S–0–1101.7.x

Element	S-0-1101.7.2 List of session identifiers	S-0-1101.7.3 List of session priorities
Actual	6	6
Max	20	20
0	0x04	0
1	0x05	3
2	0xF3	0

6.8 Oversampling

6.8.1 Description

Oversampling is a method used to receive or send more information (samples) about a signal (inputs or outputs) between two (producer) cycles. N samples are transmitted together in one cycle, therefore requiring at least N times as much telegram space compared to the "normal" signal. The method can be used for inputs as well as for outputs.

6.8.2 General

Oversampling is described in this function group. There are two structured IDNs within this function group (see Figure 46).

- Configuration (S-0-1150): Holds all the Oversampling configurations.
- Capabilities(S-0-1151): Describes all of the parameters of this function group.

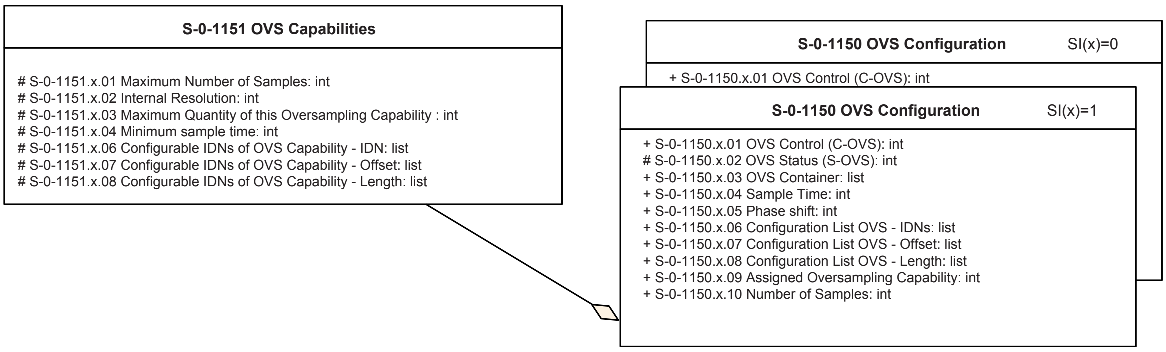

Figure 46 – Oversampling overview

6.8.3 Constraints

The following constraints shall be considered:

- The oversampling domain describes the set of IDNs which belong together, defining one oversampling machine.
- There is exact one oversampling clock for an oversampling domain.
- The sampling time is an integer fraction of the producer cycle time (for example Pcyc = 1 ms, sampling time = 50 µs).
- Sampling points are equidistant with respect to time.

6.8.4 Oversampling Input

One or several signals are sampled in a producer cycle (Pcyc) with a defined faster sample clock. The sampled values are combined in a so called oversampling container and transmitted in the following producer cycle. The last sampling point transmitted in one producer cycle is the sample of the respective T4pc.

Figure 47 shows an example with a tPcyc of 2 * tScyc and Oversampling factor of 8.

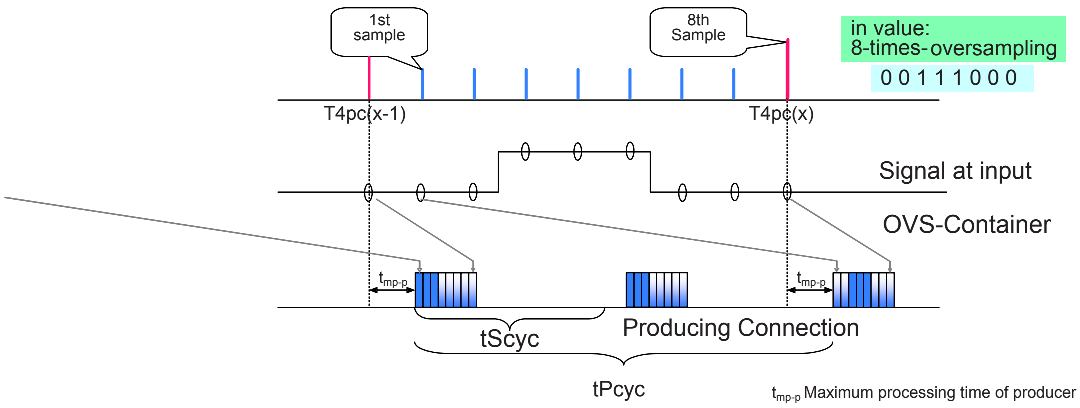

Figure 47 – Oversampling timing input (producer)

6.8.5 Oversampling Output

One or several signals are output within a producer cycle (Pcyc) with a defined faster sample clock. The values which are received get valid in the next producer cycle.

Figure 48 shows an example with a tPcyc of 2 * tScyc and Oversampling factor of 8. The first sampling point gets valid at T4pc.

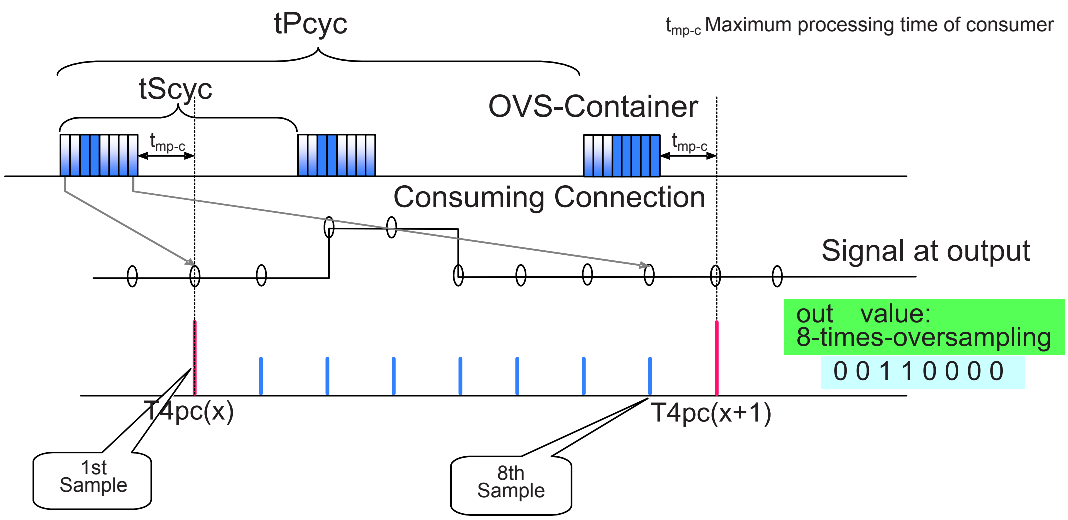

Figure 48 – Oversampling timing output (consumer)

6.8.6 Oversampling Identification

IDN S-0-1151 with its structure elements (SEs) defines the capabilities of the function group oversampling.

An oversampling domain is assigned to a structure instance. The available domains are shown in S-0-1152 Amount of OVS Domains.

- S-0-1151.x.01 Maximum number of samples
- S-0-1151.x.02 Internal resolution
- S-0-1151.x.03 Maximum quantity of this oversampling capability
- S-0-1151.x.04 Minimum sample time
- S-0-1151.x.06 Configurable IDNs of OVS capability
- S-0-1151.x.07 Configurable IDNs of OVS Capability - Offset
- S-0-1151.x.08 Configurable IDNs of OVS Capability - Length

All structure elements of parameter S-0-1151 are read only.

6.8.7 Oversampling Configuration

Each structure instance (SI) of IDN S-0-1150 contains the configuration of the corresponding oversampling domain.

S-0-1150.x.09 Assigned Oversampling Capability assigns the instance of the dedicated oversampling capability S-0-1151 OVS Capabilities.

The structure elements are:

- S-0-1150.x.01 OVS Control (C-OVS)
- S-0-1150.x.02 OVS Status (S-OVS)
- S-0-1150.x.03 OVS Container
- S-0-1150.x.04 Sample time

- S-0-1150.x.05 Phase shift
- S-0-1150.x.06 Configuration List OVS - IDNs
- S-0-1150.x.07 Configuration List OVS - Offset
- S-0-1150.x.08 Configuration List OVS - Length
- S-0-1150.x.09 Assigned Oversampling Capability
- S-0-1150.x.10 Number of Samples

6.8.8 Application example

- Sampling of fast inputs (<tPcyc)
- Creating fast signal forms (<tPcyc) at an output

Relationship between S-0-1150.x.04 Sample time and S-0-1150.x.10 Number of Samples

The configuration of both IDNs can be performed in the following ways:

- If S-0-1150.x.10 Number of Samples is written, the operation data of S-0-1150.x.04 Sample time is calculated by the slave according to: S-0-1150.x.04 Sample time = S-0-1050.x.10 Producer Cycle Time / S-0-1150.x.10 Number of Samples.
- If S-0-1150.x.04 Sample time is written, the operation data of S-0-1150.x.10 Number of Samples is calculated by the slave according to: S-0-1150.x.10 Number of Samples = S-0-1050.x.10 Producer Cycle Time / S-0-1150.x.04 Sample time. If the result is not an integer in nanoseconds it is rounded.
- If the corresponding OVS container has not been configured in a connection (for example for OVS via SVC), no S-0-1050.x.10 is available, which can be used for the calculation of S-0-1150.x.04. In this case the configuration of the oversampling machine has to be done via S-0-1150.x.04 Sample time.

6.8.9 Oversampling State Machine

Figure 49 shows the state machine that shall be applied.

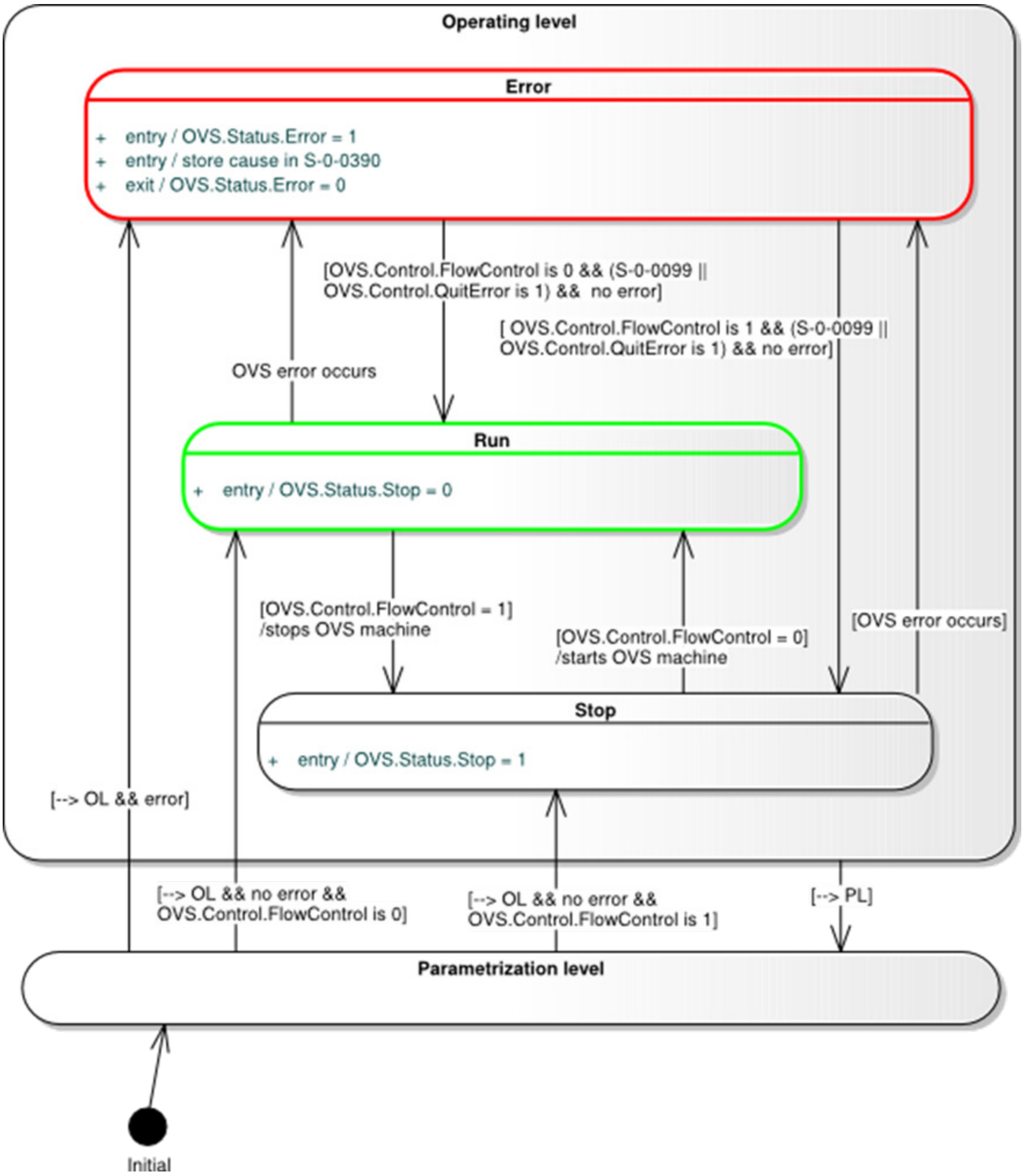

Figure 49 – Oversampling state machine

The states of the state machine are described in Table 88.

Table 88 – States of the oversampling state machine

State	Description
Run	In this state the OVS.Status.stop is set to 0. The oversampling state machine is ready-to-operate and remains in this state until it is stopped or an error occurs.
Stop	In the state stop the OVS mechanism is stopped and the OVS.Status.stop is set to 1. The input container contains the last valid samples (freeze). The output signals are frozen.
Error	If an error occurs the container values are invalid. The OVS.Status.Error is set to 1 in order to show this error. A diagnostic message 0xC30F4018. shall be generated using the appropriate diagnostic mechanisms (S-0-0390).

The transitions of the state machine are described in Table 89.

Table 89 – Transitions of the oversampling state machine

Transition		Description
Source	Target	
Run	Stop	If the OVS.Control.FlowControl is 1, then the oversampling machine stops and switches to "Stop" state.
Run	Error	With the occurrence of an error the oversampling machine stops and switches to the "Error" state.
Stop	Run	If the OVS.Control.FlowControl is 0, then the oversampling machine starts and switches to "Run" state.
Stop	Error	With the occurrence of an error the oversampling machine switches to the "Error" state.
Error	Run	If OVS.Control.QuitError is 1 or the procedure command S-0-0099 is executed and OVS.Control.FlowControl is 0 and no error is present, then the oversampling machine starts again and switches to "Run" state.
Error	Stop	If OVS.Control.QuitError is 1 or the procedure command S-0-0099 is executed and OVS.Control.FlowControl is 1 and no error is present, then the oversampling machine switches to the "Stop" state.

7 Telegram timing and DLPDU handling

7.1 Communication mechanisms

7.1.1 Cycle time

The communication cycle time, tScyc, shall have one of the following values:

tScyc = 31,25 μs, 62,5 μs, 125 μs, 250 μs up to 65 ms (in 250 μs increments)

This cycle time may have some jitter. The jitter describes the deviations from the tScyc value in the distance between two MSTs.

Therefore, the actual time interval between the end of a MST and the end of the following MST shall have a

- minimum value of j x tScyc × 0,999 9 × JtScyc (j = 1, 2, 3, ...) and a
- maximum value of j x tScyc × 1,000 1 + JtScyc (j = 1, 2, 3, ...).

NOTE j is an ordinary integer and not related to the abbreviations.

The factors 0,999 9 and 1,000 1 take into account the deviation of the communication cycle time S-0-1002 (tScyc), compared to the accuracy of the usual crystal oscillators (±50 *μHz/Hz*). The jitter shall not accumulate over several periods (i.e., the average value shall be zero).

7.1.2 Medium access

7.1.2.1 Definitions of the communication timing

The timing calculations of a Type 19 network are based on 100 Mbit/s transmission rate.

During the initialization, the master shall inquire for time parameters from the slaves, see Communication phase 2 (CP2). With this information, the master shall calculate collision-free transmission time-slots of the Type 19 telegrams within the RT channel.

The master shall transmit to each slave the S-0-1006 AT0 transmission starting time (t1), as well as the beginning and end times of the UC channel, t6 and t7 (S-0-1017 UC transmission time) respectively. These starting times for the transmitting time-slots for the telegrams are defined below, whereas the jitters and delay times have been incorporated in this timing.

RT channel ≈ MDT block + AT block

For a collision-free communication during the RT channel the following parameters shall be taken into account when calculating the related timing by the master (see Table 90).

Table 90 – Parameter for timing calculation

Parameter	Description
TTref[ns]	In the following text, all timings refer to the "end of MST" called TTref (see Figure 50). It is defined as the last edge time of the MST-CRC.
tTH[ns]	The telegram header (tTH) defines the delay time from beginning of preamble to end of MST (TTref) and is a constant time of 2240 ns (28 octets).
MST jitter[ns]	Depending on the performance of the hardware, the master sends the MST (MDT block) with a jitter. The MST jitter is part of the S-0-1023 SYNC jitter.
t1 jitter[ns]	Depending on the performance of the hardware, the master sends the AT0 (AT block) with a jitter.
Slave jitter[ns]	The interface (hardware) of the slave produces the telegram jitter defined in S-0-1037 Slave Jitter.
Inter frame gap (IFG)[octets]	The IFG defines the distance between two Ethernet frames. The Ethernet specification requires 12 octets as a minimum. The inter frame gap is dependent on the number of participants in the topology. The formula of IFG shall be used by the master to calculate the S-0-1036 Inter Frame Gap for the given application.
tIFG[ns]	corresponds to the inter frame gap of octets converted into time (ns).
IFG jitter[ns]	corresponds to tIFG – 960 ns (12 octets). The IFG jitter is part of the S-0-1023 SYNC jitter.
MDT block[ns]	Period of all used MDTs with the corresponding inter frame gaps.
AT block[ns]	Period of all used ATs with the corresponding inter frame gaps.

The timing parameters of the communication shall be set by the master in CP2 and activated in the master and the slaves in CP3 and CP4.

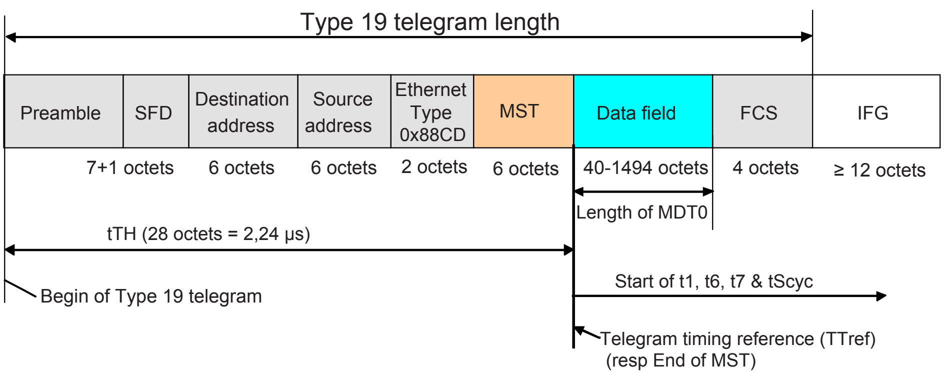

Figure 50 – Telegram timing reference

7.1.3 Calculation of the Type 19 telegram length

The calculation of a Type 19 telegram length includes the following telegram fields shown in Figure 51:

- Preamble + SFD (7 + 1 octets)
- Destination address (6 octets)
- Source address (6 octets)
- Ethernet type (2 octets)
- Type 19 header (6 octets)
- Frame check sequence (4 octets)

--> The total header of a Type 19 telegram contain 32 octets.

- Data field (40 to 1494 octets)

Minimum and maximum telegram lengths and periods:

- The minimum telegram length of a Type 19 telegram contains 72 octets (32 octets + 40 octets) resulting in a telegram period of 5,8 μs.
- The maximum telegram length of a Type 19 telegram contains 1526 octets (32 octets + 1494 octets) resulting in a telegram period of 122,1 μs.

An individual SIII telegram period is calculated as follows:

- Type 19 telegram period (μs) = Type 19 telegram length (octets) * 8 (bit) * 0,01 (μs)

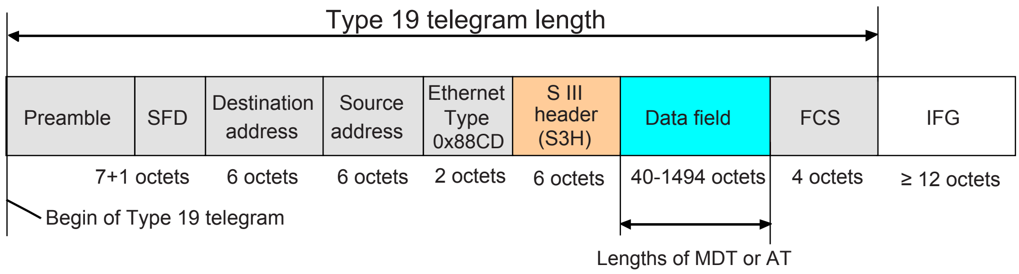

Figure 51 – Calculation of telegram length

7.1.4 Timing calculation of RT channel

At first the master shall calculate the time of the RT channel, then the master can calculate the time-slot of the UC channel.

The following calculations shall be done by the master:

- MDT block = sum of Type 19 telegram periods of all used MDTs + number of MDTs * tIFG (see S-0-1036 Inter Frame Gap).
- AT block = sum of Type 19 telegram periods of all used ATs + number of ATs * tIFG (see S-0-1036 Inter Frame Gap).
- with the given communication cycle time and the MDT block and AT block the master calculate the times t1min and t1max.
- the master shall determine the time t1 under consideration of t1min, t1max and tScyc.
- S-0-1006 AT0 transmission starting time (t1): this is the nominal time interval between the end of MST and the beginning of AT0. This parameter shall be determined by the master and stored in the associated slaves.

- S-0-1017 NRT transmission time: beginning of UC channel (t6) and end of UC channel (t7). Within the UC channel there shall not be any special time-slots. Every participant shall be able to send its Ethernet frames during this time-slot.
- S-0-1023 SYNC jitter: is calculated by the master using the MST jitter and IFG jitter. This parameter is transmitted to the slaves in CP2.
- Formula of S-0-1036 Inter Frame Gap:

$$S-0-1036 \geq \frac{27*(S-0-1037)_{MAX}*\sqrt{2*N}}{8000}*\frac{1octet}{0,08\mu s}+12octets$$

NOTE N is the number of participants in the topology.

7.1.5 Calculation of S-0-1006 AT0 transmission starting time (t1)

The master shall calculate the minimum time of t1 (t1min) and the maximum time of t1 (t1max) under consideration of the S-0-1002 Communication Cycle time (tScyc).

- t1min = MDT block + (t1 jitter) - tTH
- t1max = tScyc - AT block - tTH - t1 jitter - (SYNC jitter)
- if t1max < t1min, then the master shall use a longer tScyc or reduce the amount of data in the Type 19 telegrams.
- t1min ≤ S-0-1006 ≤ t1max

Figure 52 shows the timing of t1 which is possible in CP3 and CP4.

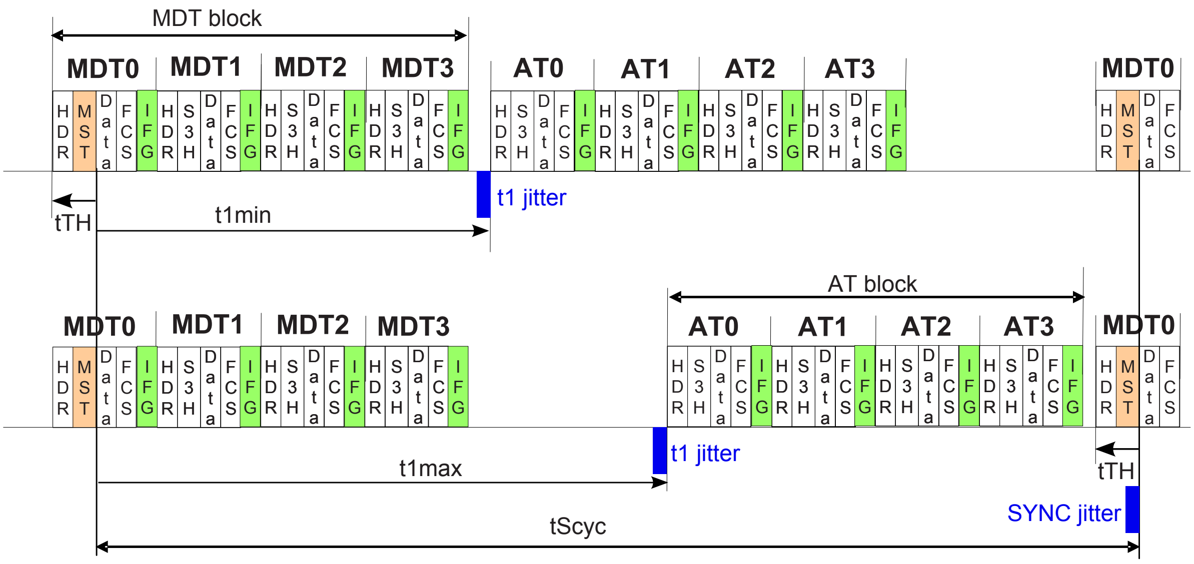

Figure 52 – Calculation of t1

NOTE

- t1 jitter = transmitting jitter of AT0
- MST jitter = transmitting jitter of MDT0
- Sync jitter = S-0-1023 SYNC jitter
- HDR = Ethernet header
- MST = Type 19 header of MDT0
- S3H = Type 19 header of MDT1 to MDT3 and AT0 to AT3
- Data = Payload
- FCS = Frame check sequence

- IFG = S-0-1036 Inter Frame Gap
- tTH = Type 19 telegram header
- MDT block = Period of all transmitted MDTs
- AT block = Period of all transmitted ATs

7.1.6 Timing calculation of UC channel

The master shall calculate the time S-0-1017 NRT transmission time (t6 and t7) for the method 1 or method 2 under consideration of the parameters t1 and tScyc.

The duration of the UC channel should be set as great as possible.

Calculation for method 1 (m1):

- The time t6 (beginning of UC channel) shall be calculated as follows: t6.m1 = t1 + (t1 jitter) + AT block (t1 jitter = t1_transmit_jitter + IFG jitter)
- The time t7 (end of UC channel) shall be calculated as follows: t7.m1 = tScyc – (SYNC jitter) - tTH
- The master shall set the time t6 as small as possible --> t6 ≥ t6.m1
- The master shall set the time t7 as great as possible --> t7 ≤ t7.m1

Calculation for method 2 (m2)

- The time t6 (beginning of UC channel) shall be calculated as follows: t6.m2 = MDT block - tTH (+ Jitter)
- The time t7 (end of UC channel) shall be calculated as follows: t7.m2 = t1 – (t1 jitter)
- The master shall set the time t6 as small as possible --> t6 ≥ t6.m2
- The master shall set the time t7 as great as possible --> t7 ≤ t7.m2

Limitations of UC channel in CP3 and CP4

- The minimum duration of the UC channel = (t7 - t6) ≥ 5,8 μs + tIFG (0,96 μs)

Figure 53 shows the calculation of t6 and t7 for method 1 and 2 which are used in CP3 and CP4.

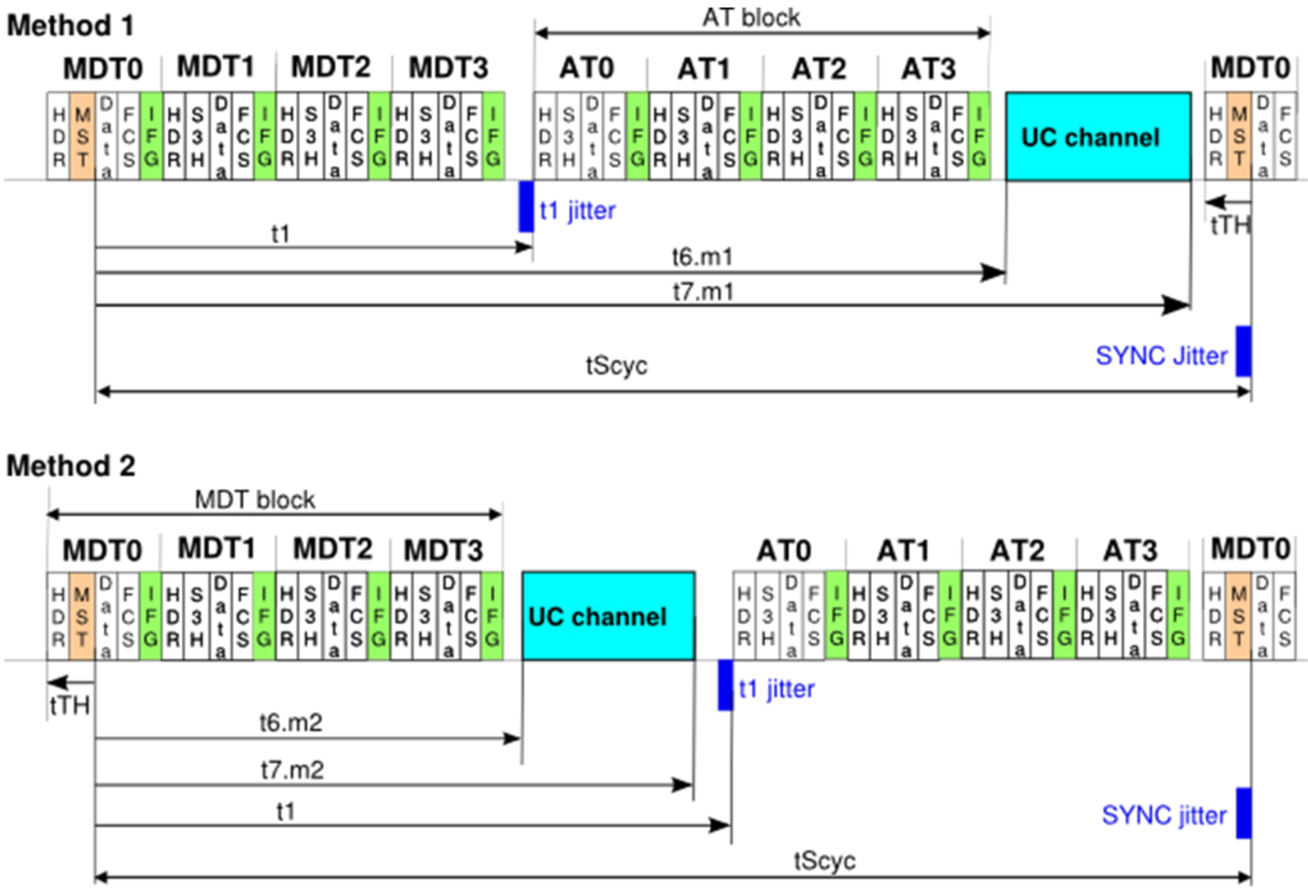

Figure 53 – Determination of UC channel

7.1.7 Telegram timing in CP0

The communication cycle time shall be preset by the master with 1 ms ≤ tScyc-cp0 ≤ 65 ms. The telegram timing of CP0 is shown in Figure 54. No transmission time is specified for AT0, but it shall be transferred after MDT0 and before the UC channel is activated (time t6cp0 is reached).

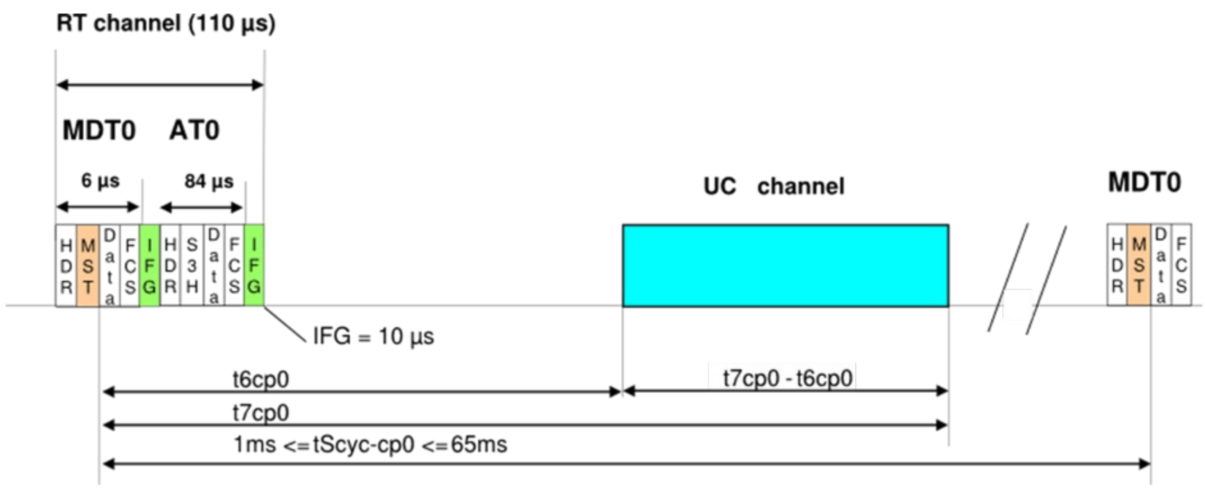

Figure 54 – Timing diagram of CP0

The default values of the timing in CP0 are defined in Table 91.

7.1.8 Telegram timing in CP1 and CP2

Four telegram set-ups are possible in CP1 and CP2. Two set-ups have a minimum communication cycle time of 1 ms and can address up to 255 slaves (case 1 and case 3); the others have a minimum communication cycle time of 2 ms and can address up to 511 slaves (case 2 and case 4). An inter frame gap of 10 μs are used for all telegram set-ups.

- Case 1 (up to 255 slaves): The communication cycle time shall be preset by the master with 1 ms ≤ tScyc-cp1/2 ≤ 65 ms. The telegram sequence during CP1 and CP2 are shown in Figure 55. No transmission times are defined for MDT1, AT0 and AT1; but they shall all be transferred in that order before the time t6cp1/2 is elapsed (beginning of UC channel in CP1&2). The master shall send the next MDT0 after the time t7cp1/2. The Master may announce the transmission of the timing parameters of CP1/2 in the communication version. If the slave supports this functionality, then it shall acknowledge it by setting bit 15 = 1 of the topology index in AT0 of CP0. If the master didn't transmit the timing parameters of CP1/2 in the MDT0 of CP0 or the slave didn't evaluate or cannot accept these parameters, then it shall not acknowledge it by setting bit 15 = 0 of the topology index in AT0 of CP0. Therefore, the default values of CP1/2 shall be activated in the master and the slaves (see Table 91).

Table 91 – Default values of CP1/2 (case 1)

Parameter	Default value
t6cp1/2	650 μs
t7cp1/2	950 μs

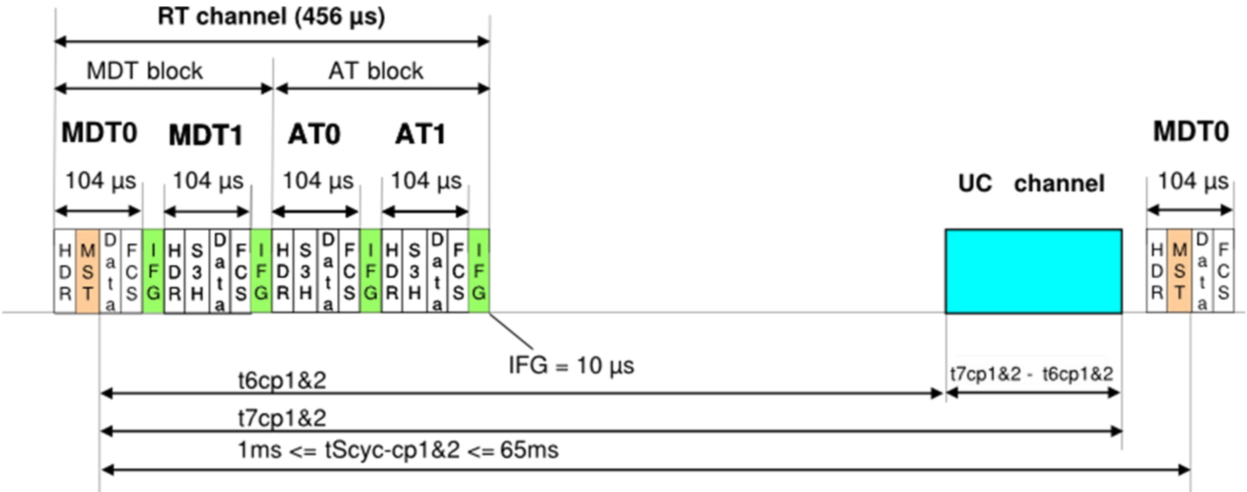

Figure 55 – Timing diagram of CP1 and CP2 with 2 MDT, 2AT and UC channel

- Case 2 (up to 511 slaves): The communication cycle time shall be preset by the master with 2 ms ≤ tScyc-cp1/2 ≤ 65 ms. The telegram sequence during CP1 and CP2 are shown in Figure 56. No transmission times are defined for MDT1, MDT2, MDT3, AT0, AT1, AT2 and AT3; but they shall all be transferred in that order before the time t6cp1/2 is elapsed (beginning of UC channel). The master shall send the next MDT0 after the time t7cp1/2. The Master may announce the transmission of the timing parameters of CP1/2 in the communication version. If the slave supports this functionality, then it shall acknowledge it by setting bit 15 = 1 of the topology index in AT0 of CP0. If the master didn't transmit the timing parameters of CP1/2 in the MDT0 of CP0 or the slave didn't evaluate or cannot accept these parameters, then it shall not acknowledge it by setting bit 15 = 0 of the topology index in AT0 of CP0. Therefore, the default values of CP1/2 shall be activated in the master and the slaves (see Table 92).

Table 92 – Default values of CP1/2 (case 2)

Parameter	Default value
t6cp1/2	1050 µs
t7cp1/2	1950 µs

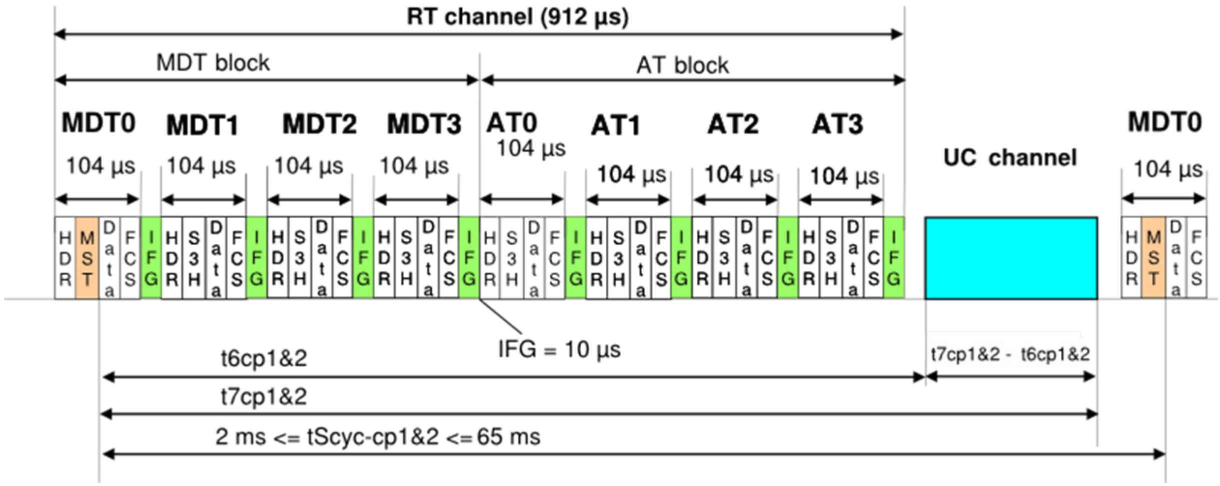

Figure 56 – Timing diagram of CP1 and CP2 with 4 MDT, 4 AT and UC channel

- Case 3 (up to 255 slaves): The communication cycle time shall be preset by the master with 1 ms ≤ tScyc-cp1/2 ≤ 65 ms. The telegram sequence during CP1 and CP2 are shown in Figure 57. The transmission time of AT0 (t1cp1/2) is defined. No transmission times are defined for MDT1 and AT1. MDT1 shall be transferred in that order before the time t6cp1/2 is elapsed (beginning of UC channel in CP1&2). AT1 shall be transferred in that order before the transmission of the next MDT0 is started. Therefore, the master shall send the next MDT0 after the end of the AT1. The Master may announce the transmission of the timing parameters of CP1/2 in the communication version. If the slave supports this functionality, then it shall acknowledge it by setting bit 15 = 1 of the topology index in AT0 of CP0. If the master didn't transmit the timing parameters of CP1/2 in the MDT0 of CP0 or the slave didn't evaluate or cannot accept these parameters, then it shall not acknowledge it by setting bit 15 = 0 of the topology index in AT0 of CP0. Therefore, the default values of CP1/2 shall be activated in the master and the slaves (see Table 91).

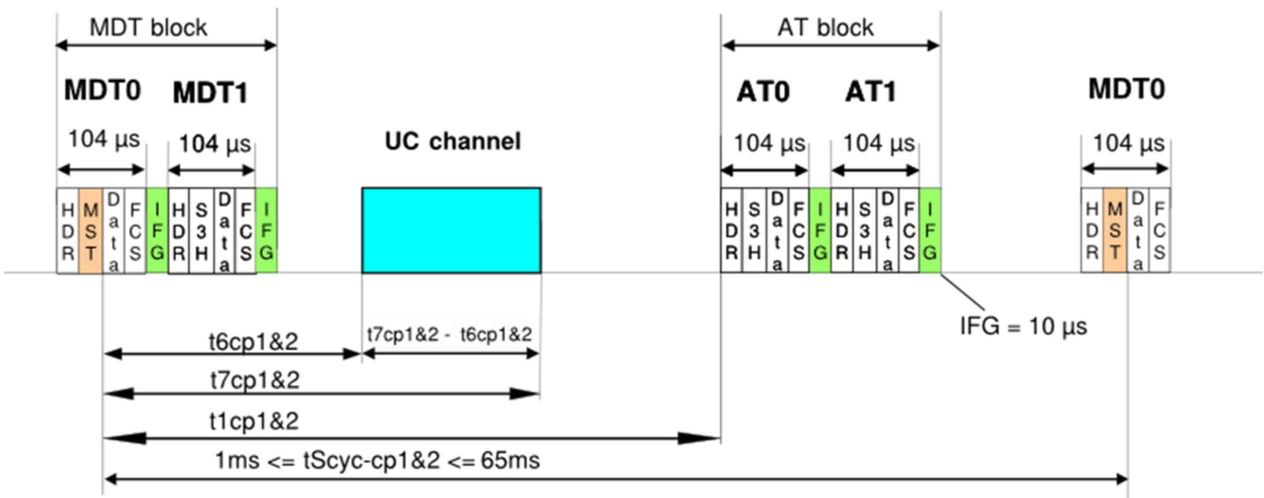

Figure 57 – Timing diagram of CP1 and CP2 with 2 MDT, UC channel and 2 AT

- Case 4 (up to 511 slaves): The communication cycle time shall be preset by the master with 2 ms ≤ tScyc-cp1/2 ≤ 65 ms. The telegram sequence during CP1 and CP2 are shown in Figure 58. The transmission time of AT0 (t1cp1/2) is defined. No transmission times are defined for MDT1, MDT2, MDT3; but they shall all be transferred in that order before the time t6cp1/2 is elapsed (beginning of UC channel). No transmission times are defined for AT1, AT2, AT3; but they shall all be transferred in that order before the transmission of the next MDT0 is started. Therefore, the master shall send the next MDT0 after the end of the AT1. The Master shall announce the transmission of the timing parameters of CP1/2 in the communication version. If the slave supports this functionality, then it shall acknowledge it by setting bit 15 = 1 of the topology index in AT0 of CP0. If the master didn't transmit the timing parameters of CP1/2 in the MDT0 of CP0 or the slave didn't evaluate or cannot accept these parameters, then it shall not acknowledge it by setting bit 15 = 0 of the topology index in AT0 of CP0. Therefore, the default values of CP1/2 (case 2) shall be activated in the master and the slaves (see Table 92).

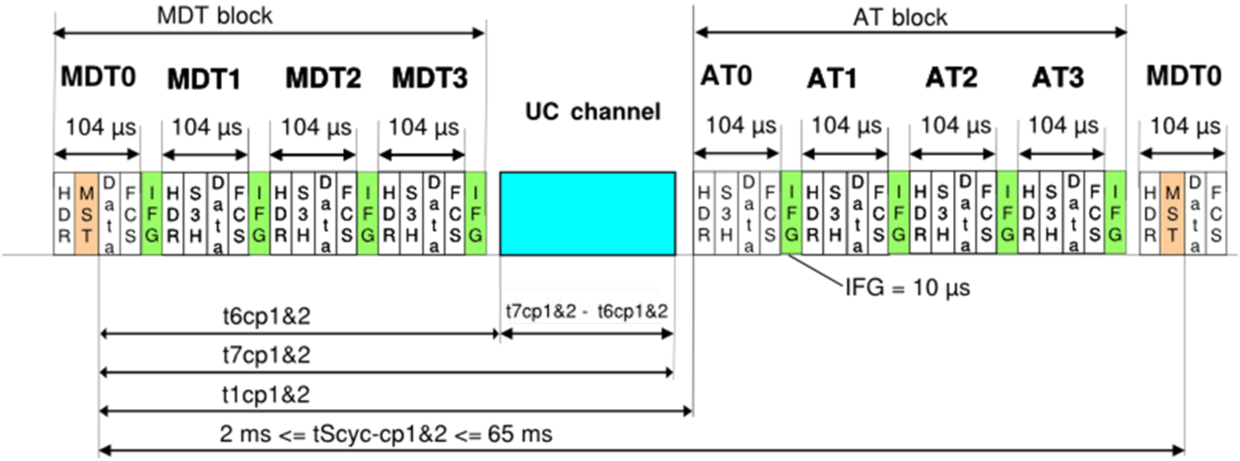

Figure 58 – Timing diagram of CP1 and CP2 with 4 MDT, UC channel and 4 AT

7.1.9 Telegram timing in CP3 and CP4

The sequence of transmitted Type 19 telegrams and UC telegrams shall be repeated every communication cycle. The time slots for the RT channel and the UC channel and the transmission time of the AT shall be transmitted during initialization and are therefore known by each slave. Figure 59 shows the two possible arrangements of the RT channel and the UC channel. The master shall always use one method out of these two, the choice of which depends on configuration.

NOTE Some control units calculate the new command values only after having received all feedback values. In method 1 there is more time available for the calculation of the command values. Method 2 is more appropriate for position control since the control unit can calculate the new command values while still receiving the feedback values.

A synchronous collision-free media access control shall be used in the RT channel. Telegrams shall be exchanged in fixed communication cycles. The master shall start the communication cycle strictly equidistant with the communication cycle time tScyc, by transmitting the MDT0. The next communication cycle shall start with the transmitting of the next MDT0. The communication cycle is defined from the end of the MST of communication cycle (n) to the end of the MST of the next communication cycle (n+1).

The MDTs (MDT0 to MDT3) shall be transmitted to all slaves. The MDT0 shall contain the synchronization information and the status of the communication in the MST field.

The ATs (AT0 to AT3) shall be transmitted by the master with the configured telegram length but the data field is filled with zeros. Each slave shall insert its real-time data into its allocated data field within the ATs. The sequence of the slave data fields within the ATs shall be independent of the physical order of the topology as well as the predefined sub-device address. The master shall be the final recipient of the ATs.

The data field length and content meaning of the MST and AT header shall remain constant and thus have the same length at each communication cycle.

Every Type 19 device may send its UC telegrams during the UC channel.

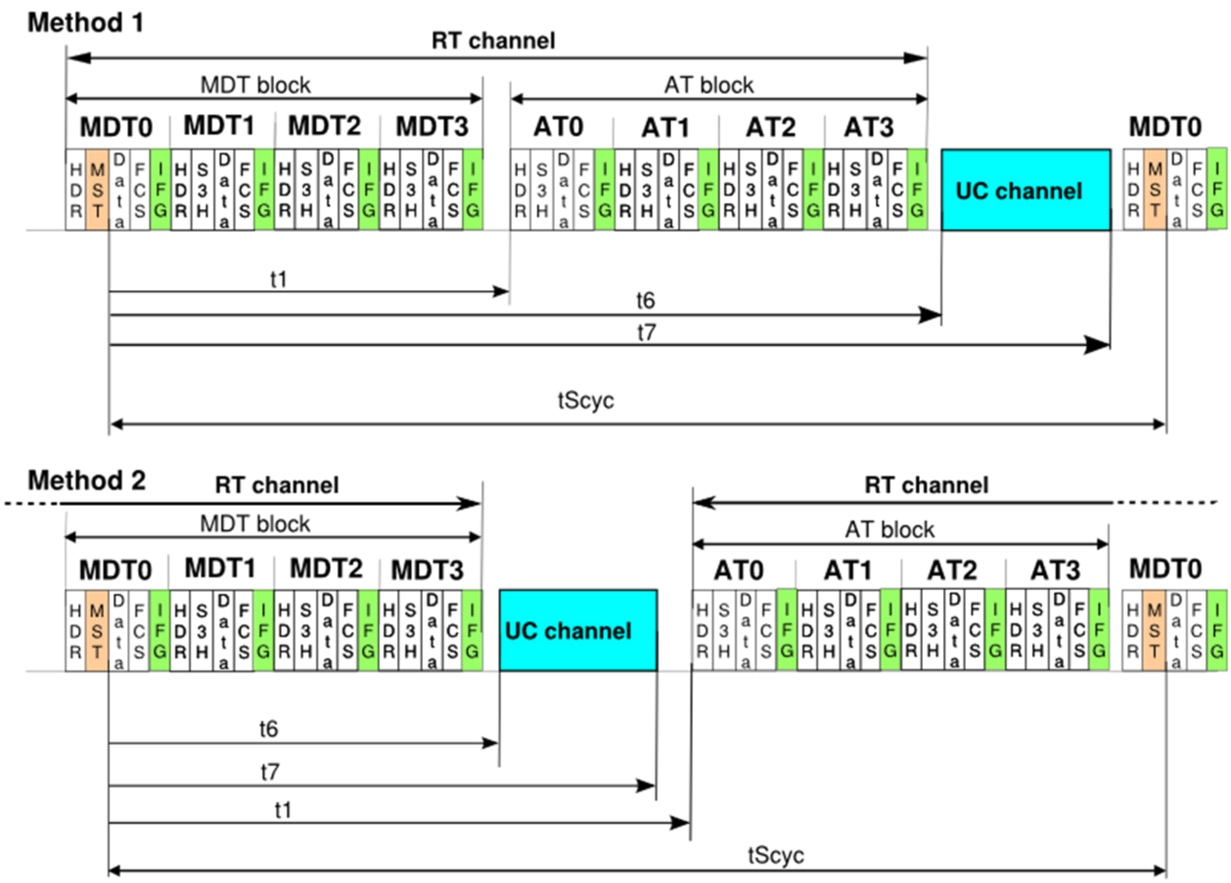

Figure 59 – Telegram sequence

7.1.10 NRT communication mechanisms

7.1.10.1 Introduction

Before the Type 19 real-time communication is initialized, communication with the slaves shall use standard IP mechanisms if it supports this function. As soon as initialization has started or is done, IP communication shall be possible only in the UC channel.

During the UC channel (time slot between t_6 and t_7) the participants shall send any kind of UC telegrams on one port depending on the destination MAC address. If the Type 19 network devices have not yet learned the right port, they shall send the UC telegrams on both ports.

The time t6 marks the beginning and t7 marks the end of the UC channel relative to the end of MST (Type 19 Header of MDT0 resp. the time TTref).

In CP0, CP1, CP2 the time slot between t6 and t7 is only available if the slave receives a valid MST. Otherwise the communication is not possible in this communication cycle.

Each transmission of an UC telegram shall be canceled at time t7.

UC telegrams may be forwarded either immediately or later depending on communication load.

Type 19 telegrams shall be dropped when the time TTref (end of Type 19 header) is received between t6 and t7.

Loopback with forward shall never be active when the UC channel is active, even if a Type 19 device detects a communication interruption and becomes the last one in a line configuration.

In CP0, CP1 and CP2 the timing of the UC channel is defined with default values. In CP3 and CP4 the UC channel can be positioned before or after the ATs. These two positions are shown in Figure 60.

The method to calculate a valid time slot defined by t6 and t7 (S-0-1017), as well as the meaning of t1, tTH, etc. is described in 7.1.2. The UC channel may be deactivated by setting the time t6 to 0. In this case the collision buffer shall be deactivated in the master and the slaves also. This shall prevent that UC telegrams are stored in the collision buffer.

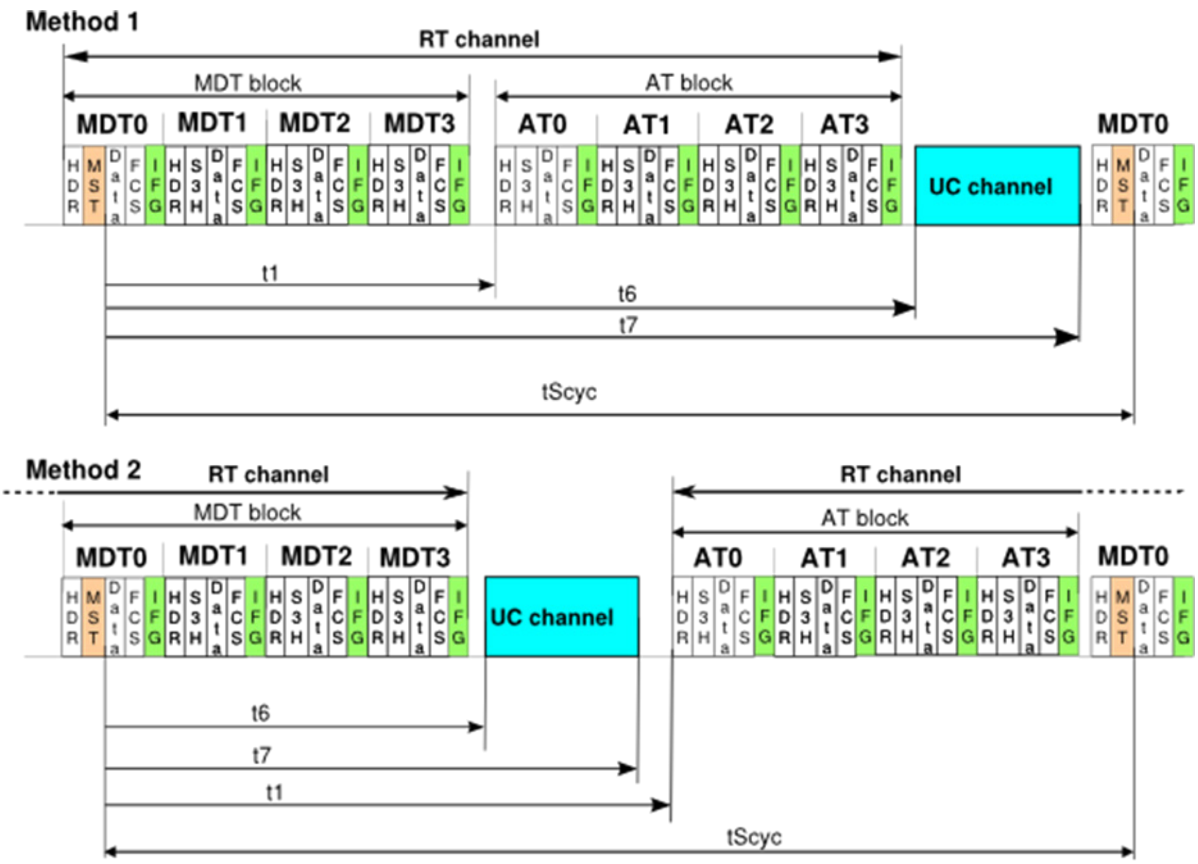

Figure 60 – The two defined positions of the UC channel.

- Sending UC telegrams:
  - – First transmit: The first UC telegram shall be sent during the UC channel at the earliest 80 ns after the time T6 (beginning of UC channel). --> First transmit = T6 + 80ns (see Figure 61)
  - – Last transmit: The last UC telegram shall be sent during the UC channel at the latest before the time T7 (end of UC channel) by considering the following conditions. -->

Last transmit = T7 - period of IP telegram - switching from UC to RT channel (tHW) - IFG (see Figure 61)

- The period of the UC telegram (with preamble, DA, SA, type/length, payload and FCS),
- the time needed within the Type 19 device for switching from UC channel to RT channel (tHW) and
- the IFG with 12 octets (tIFG = 960 ns).

– After T7 (end of UC channel) the Type 19 device shall be able to process RT frames (Type 19 telegrams) immediately.

- Receiving UC telegrams: The receiving of UC telegrams shall be possible between T6 and T7 in a Type 19 device.

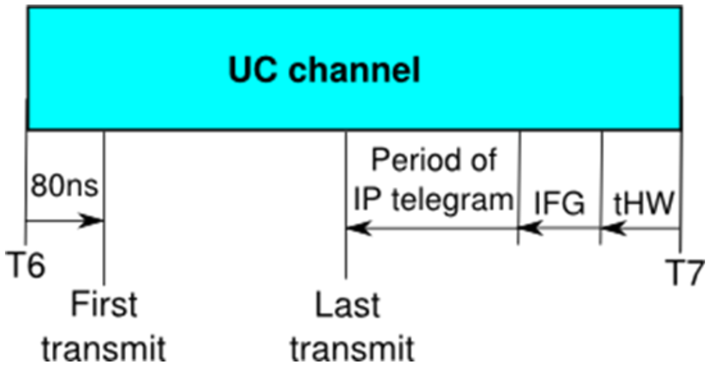

Figure 61 – First and last transmit during UC channel

7.1.10.2 Slaves within a line or a ring

If a slave receives a telegram while the UC channel is active and it is transmitting an Ethernet frame, the currently transmitted Ethernet frame shall not be interrupted and the incoming telegram shall be stored if possible.

Slaves shall always send their own Ethernet frames on one port (P1 or P2) depending on the destination mac address, provided that all following conditions are met before doing so:

- it is not already forwarding another telegram; otherwise the slave shall wait until this telegram has been fully forwarded;
- the remaining UC channel duration is long enough to fully transmit its own Ethernet frame;
- its memory has enough free capacity for storing at least one new incoming Ethernet frame with maximum length.

If the Slaves have not yet learned the right port, they shall send the frame on both ports.

7.1.10.3 Slave in the last position within a line

Although the last slave in a line has its loopback with forward active, it shall check for any incoming Ethernet frame on its inactive port. It shall forward Ethernet frames if its UC channel is active, provided that the remaining duration of this UC channel is long enough to fully transmit this Ethernet frame. It shall forward all incoming “non-Type 19” Ethernet frames as soon as the UC channel is active again, provided that the remaining duration of this new UC channel is long enough.

The last slave in a line topology shall use the MST at its active port also for its inactive port to support the timing of the UC channel.

7.1.10.4 Forwarding of packets

Every Type 19 device that supports IP communication can send and receive packets on the primary and secondary port of its interface. Packets received on one port of a devices interface, which are not for the local device, need to be forwarded to and sent on the other port. This forwarding can either be done with the method store and forward or with the method cut through forwarding, explained in the next section.

Since the other port can already be busy, the interface has to buffer packets until the port is free. This buffer is called the Collision Buffer (CB).

The left side of Figure 62 shows a device with an activated collision buffer as well as the receive and send buffers of P and S port.

The right side shows a device with a deactivated collision buffer.

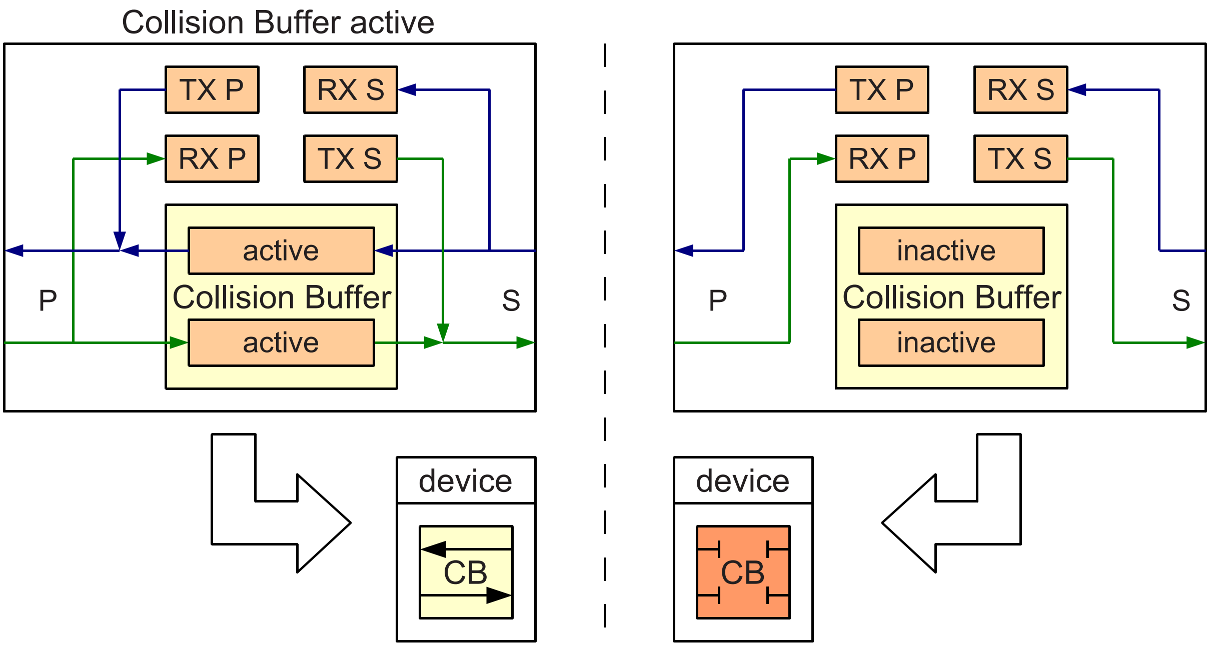

Figure 62 – Activated and deactivated collision buffer

7.1.10.4.1 Store and forward

Store and forward defines that each packet shall be received and buffered completely.

- The frame check sequence (FCS) of packets is checked.
- If the FCS is invalid, the packet will be discarded.
- Otherwise the packet will be forwarded to the next station towards its destination.

With this method of forwarding, the packet will be delayed at each station for a time about equal to its transmission time. Figure 63 shows the time response between an input port and an output port for a forwarded packet.

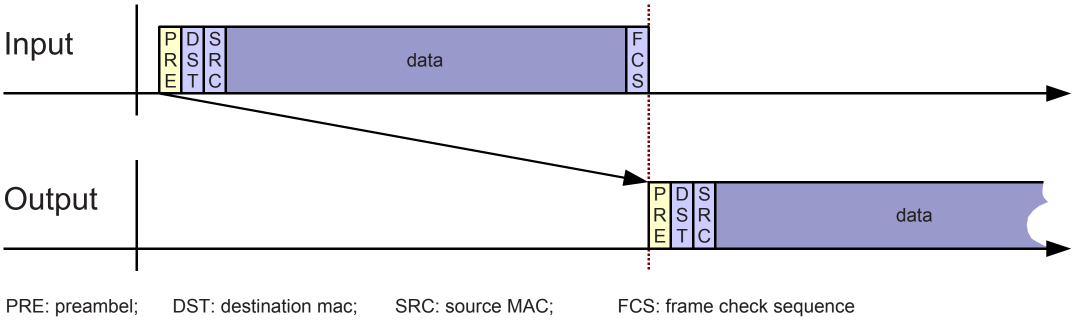

Figure 63 – Time response of store and forward method

7.1.10.4.2 Cut through forwarding

In opposite to store and forward, cut through forwarding does not wait until a packet is received completely. Cut through forwarding rather starts forwarding packets as soon as it knows the packet's destination.

- This means, that on the one hand each packet is delayed only for a time about equal to the transmission time of the preamble and destination mac address (approx. 1,2 µs at 100 Mbit/s).
- On the other hand packets with an invalid FCS will not be discarded.

Figure 64 shows the time response between an input port and an output port for a packet forwarded with cut through forwarding.

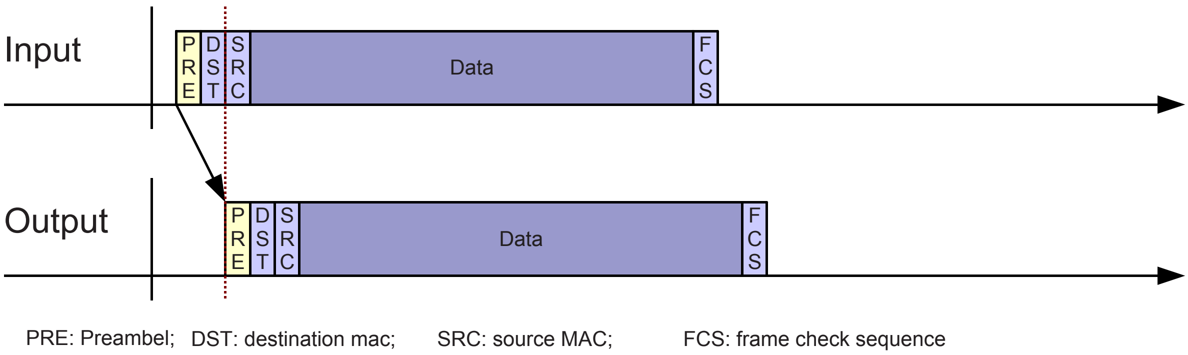

Figure 64 – Cut through forwarding

7.1.10.5 Timing at the end of the UC channel

All IP packet transmissions shall be finished at the end of the UC channel (t7). Therefore the start of each packet transmission needs to be early enough. This means that the start of transmission tTS shall be at least the packet's transmission period tTP before t7 but not before t6.

Expressed as a formula this means: t6 < tTS < (t7 - tTP)

7.1.10.5.1 Last transmission of CP0,CP1 and CP2

With the default value (576) of MTU in CP0 to CP2 the last start of transmission shall be at least ≥ 50 µs before end of UC channel.

7.1.10.5.2 Transmission period of an Ethernet frame

Figure 65 shows a complete ethernet packet followed by the formula to calculate the transmission period of an Ethernet frame.

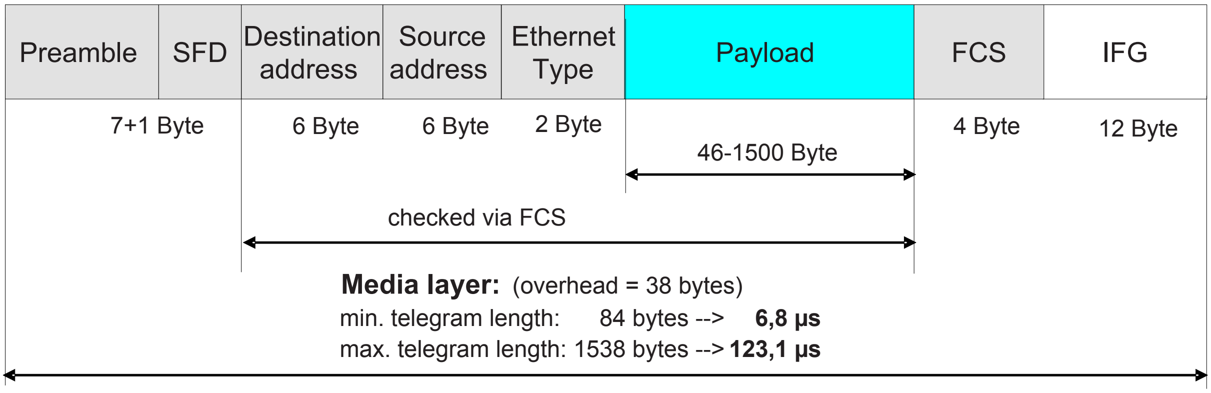

Figure 65 – Ethernet frame with payload

According to Figure 65 the period of an Ethernet frame shall be calculated like shown in the following formula.

- Preamble: = 8 octets
- Destination address: = 6 octets
- Source address: = 6 octets
- Ethernet type: = 2 octets
- FCS: (frame check sequence) = 4 octets
- IFG: (inter frame gap) ≥ 12 octets (depending on number of nodes in the topology, see S-0-1036 Inter Frame Gap)
- Payload: this is a variable that shall contain the amount of octets for the desired payload for which tTP shall be calculated

Calculation of transmission period:

- Length [octets] = 26 octets + Payload + IFG
- Transmission period (tTP) [µs] = length [octets] * 8 [bit] * 0,01 [µs]
- Transmission period min = 6,8 µs (with 12 octets IFG)
- Transmission period max = 123,1 µs (with 12 octets IFG)

7.1.10.5.3 Maximum payload based upon the UC channel length

The duration of the UC channel (tNRT = t7 - t6) can be set to arbitrary values down to 7µs. An Ethernet frame can only be transmitted during the UC channel if it fits into it.

This means, that the UC channel must be longer than the transmission period: tNRT > tTP. The amount of octets that fit into a given duration of time is calculated with this formula:

amount of octets = period of UC channel [µs] * 12,5 [octets/µs] - 38 octets

Example:

If the UC channel has a duration of 250 µs, the maximum possible payload would be: 250 [µs] * 12,5 [octets/µs] - 38 = 3087 octets

NOTE Ethernet specifies that the maximum payload of one packet on Ethernet is 1500 octets.

7.1.10.6 MTU and issues with packets sizes

7.1.10.6.1 General

Ethernet allows a payload of up to 1500 octets per packet. The UC channel of a Type 19 network can be configured to be shorter than ΔtTP(1500octets). This means, that the maximum payload of packets in the UC channel needs to be limited. This limit is usually called the maximum transmission unit (MTU) which describes the maximum payload, in octets, for a single packet, that may be transmitted over a network without fragmentation. The upper limit of the MTU can be calculated using the formula of the previous section:

$$MTU_{UL}(\Delta_{NRT}) = \min\{1500; a(\Delta(_{NRT})\}$$

Ethernet specifies that the minimum value of the maximum transmission unit (MTU) is 46 octets. If the calculated upper limit of the MTU, MTU_{UL} is lower than 46 octets, the UC channel shall be deactivated.

The MTU affects two places within a Type 19 device:

- network device interface of the devices network stack
- the Type 19 communication hardware

7.1.10.6.2 MTU and network stack interfaces

Every network interface of a network stack usually has a MTU attribute. This is necessary, so that higher layer protocols like IPv4 can find out what maximum packet size they may deliver to the interface. Therefore the network stack interface's MTU shall be set and updated according to S-0-1027.0.1.

7.1.10.6.3 MTU and Type 19 networks

A special MTU can be requested by a communication master using a slaves S-0-1027.0.1 Requested MTU. In this IDN, the master can set the requested MTUreq. The current MTUcur is dependent on S-0-1017 NRT transmission time and calculated by the slave at phase change and on write access to S-0-1027.0.1 Requested MTU. A master can read a slaves current MTUcur out of S-0-1027.0.2 Effective MTU.

7.1.10.6.4 Additional notes

- For one Type 19 line or ring, every device should be configured to the same MTU. If not, it is likely that packets will be lost. There will although, be no issues while the line or ring is in NRT state.
- Path MTU discovery (PMTUD) as specified in RFC1191 shall be supported by all devices, especially devices that route packets between different networks.
- The default value for the requested MTU is 576 octets.

7.1.10.7 Topology issues

Every Type 19 device needs to care about these issues regarding the topology of a Type 19 network:

- Collision buffers: Every Type 19 device has a collision buffer that forwards packets from one port to the other port. In order to avoid broadcasts running around forever in Type 19 ring topologies, exactly one of the collision buffers needs to be deactivated.
- Port/MAC table: Every device (masters, slaves and other components like IP plugs) that injects IP packets into a Type 19 ring sends each IP packet on both the primary and secondary port. This means that each packet is duplicated when it is injected into a Type

19 ring. To prevent this packet duplication, every device needs to know, on which port every single packet needs to be sent.

7.1.10.8 Port/MAC table of devices

7.1.10.8.1 General

Section 4.5.2.9 of the Type 19 communication specification defines, that packets shall only be sent on one port during the UC channel. Therefore each Type 19 device needs a table, in which it stores known port/MAC relations. The structure of this table as well as its handling is described in the following sections.

7.1.10.8.2 Structure

Every Type 19 device shall maintain a table (see Table 93) in which it keeps track of which device it reaches on which port. This table is called the port/MAC table and consists of at least two columns, one for a port entry, the other for a MAC entry.

Table 93 – Structure of port/MAC table

Port	MAC
…	…
…	…

7.1.10.8.3 Insertion of rows

Every time a device receives a IP packet from another device, it shall update its Port/MAC table accordingly. E.g. if a device with an empty port/MAC table receives a packet with source address 0d:ea:d0:0b:ee:f0 on port P2, the updated table shall look as shown in Table 94.

Table 94 – Insertion of entry

Port	MAC
P2	0d:ea:d0:0b:ee:f0
…	…

If the same device receives another Ethernet frame with the same source address 0d:ea:d0:0b:ee:f0 but this time on port P1, it shall update its port/MAC table as shown in Table 95.

Table 95 – Update of entries

Port	MAC
P1	0d:ea:d0:0b:ee:f0
…	…

7.1.10.8.4 Deletion of rows

There are some cases, in which entries of the port/MAC table need to be deleted:

- Every device shall clear its port/MAC table completely when switching from NRT state to CP0.
- Every master device shall clear its port/MAC table completely when a change of the physical topology has been detected. Every master device shall therefore adjust the bit 11 in the device control of all slaves accordingly.

- Every slave device shall clear its port/MAC table completely when bit 11 in its device control has been toggled.
- If the port/MAC table has no more spare rows, obsolete rows shall be removed. The algorithm that decides which rows are obsolete is relinquished to the programmers creativity.

7.1.10.9 Collision Buffer

7.1.10.9.1 General

In order to avoid having a ring, where Ethernet broadcast packets run around in circles, exactly one collision buffer has to be inactive when a physical ring topology is set up. Since the master of a Type 19 ring shall always be aware of the topology, the master shall set its collision buffer according to the current topology. There are two exceptions in which a master will detect a wrong topology. In this exceptions one slave shall deactivate its collision buffer.

7.1.10.9.2 Slave collision buffer

At power on and during the boot sequence, every slave shall switch off its collision buffer. As soon as the slave has finished its boot sequence, the collision buffer shall be set according to Table 96:

Table 96 – Slave collision buffer

Collision Buffer Slave	Condition
Inactive	(NRT state or HP0) and (receiving P- and S-telegrams) and (CP > 0)
active	all other cases

That is the slaves collision buffer shall only be inactive if all of these three conditions are true:

- the slave is in NRT state or HP0
- the slave receives Type 19 P- and S-telegrams
- the communication phase of the received Type 19 telegrams is greater than 0.

This is necessary for a master to differentiate between a recovered broken ring and a broken ring with Type 19 slaves in between. If the slaves would not deactivate their collision buffer, a master would assume that the actual topology is a unhealed ring. The Figure 66 and Figure 67 below visualize these two issues.

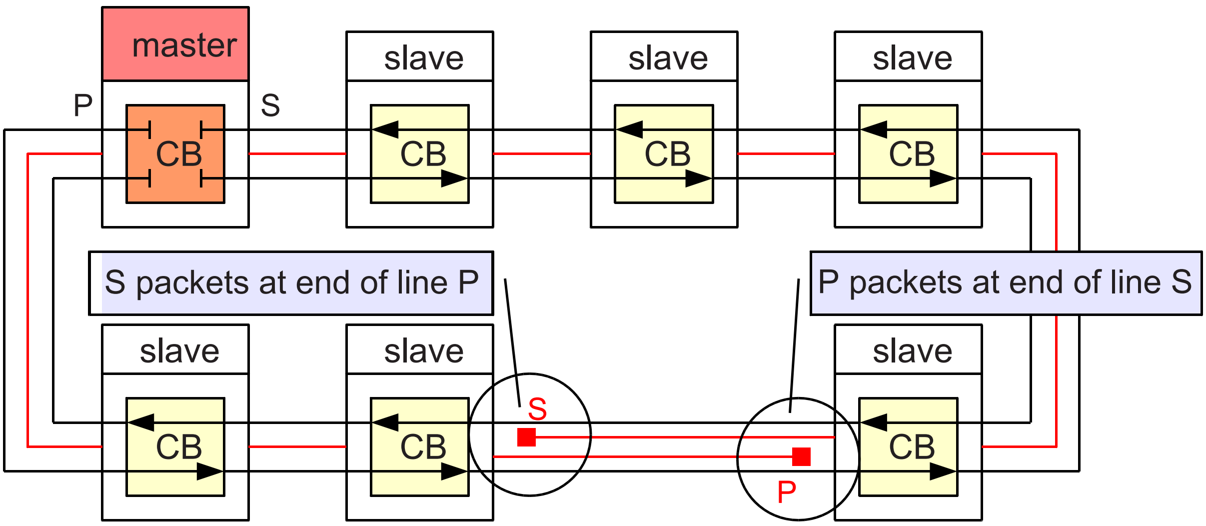

Figure 66 – Unhealed broken ring

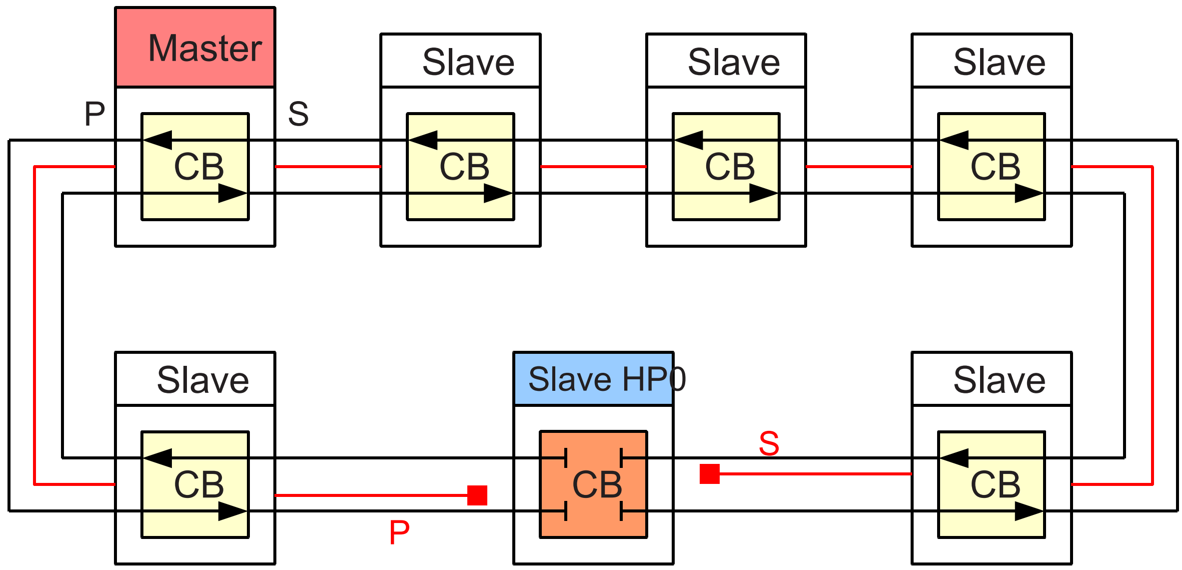

Figure 67 – Broken ring with Type 19 slave in between

7.1.10.9.3 Master collision buffer

The state of the master collision buffer is dependent of the communication phase. The following sections describe how the master collision buffer shall be set for the different phases.

7.1.10.9.4 NRT state

Every master shall keep the last known state of its collision buffer. If there is no known state, for example after power on, it is recommended, that every master switches to CP0 to determine the physical topology until the AT becomes stable for 100 cycles.

7.1.10.9.5 Communication Phase 0

Every master shall detect the physical topology by looking at the received packets on its ports (see Table 97).

- A physical ring is present, if the master receives S-packets on its P-port and P-packets on its S-Port. In this case the master shall deactivate its collision buffer.
- In all other cases, its collision buffer shall be activated.

Table 97 – Physical topology Master (CP0)

Comment	Packets received by Master on P-channel	Packets received by Master on S-channel	Collision Buffer Master
ring topology	S	P	inactive
no ring topology	*	*	active

7.1.10.9.6 Communication Phases 1 - 4

In the following cases the master collision buffer shall be deactivated. In all other cases it shall be activated. Note, that »UC Port of last Slave on P-channel« refers to the device status bits 10 and 11 of the last Type 19 slave on P-channel in line topology. »UC Port of last Slave on S-channel« refers to the device status bits 10 and 11 of the last Type 19 slave on S-channel in line topology (Table 98).

Table 98 – Physical topology Master (CP 1-4)

Comment	Status of inactive port of last slave on P-channel	Packets received by Master on P-channel	Packets received by Master on S-channel	Status of inactive port of last slave on S-channel
broken ring between master S-Port and last slave of P-channel	S-Telegram on inactive port	P	P	no last slave on S-channel
ring topology	no last slave on P-channel	S	P	no last slave on S-channel
broken ring; HP1/2 on P-channel	P-Telegram on inactive port	P	S	P-Telegram on inactive port
physically closed but unhealed Type 19 ring	S-Telegram on inactive port	P	S	P-Telegram on inactive port
broken ring between master P-Port and last slave of S-channel	no last slave on P-channel	S	S	P-Telegram on inactive port
broken ring; HP1/2 on S-channel	S-Telegram on inactive port	P	S	S-Telegram on inactive port

7.1.11 Internet Protocol Services (IPS)

7.1.11.1 Introduction

The Type 19 Internet protocol services specify different services for NRT communication participants in a Type 19 network. They use the following protocols:

- UDP
- TCP
- TFTP

NOTE Internet protocol services assume a well configured IP network.

The services are categorized according their functionality as follows:

- S/IP Connection Services: Connection services, which are required for TCP based S/IP data exchange
- Explore & IP Configuration Services: Services for:
  – Detection of devices in Type 19 networks
  – IP configuration of devices to enable point to point communication
- Identification Services: Services for the identification of S/IP devices
- Parameter Access: Read and write access to Type 19 parameters (IDNs)
- Device Management Services:
  – Restart devices
  – TFTP Firmware Management
- Other Services:
  – Identification of supported services
  – Supervision of services based on a watchdog mechanism

By the help of these services, the following use cases become realizable:

- Identification and configuration of IP devices within Type 19 networks

- Desk test of a device
- Maintenance or diagnosis of devices within an industrial plant

7.1.11.2 General definitions

7.1.11.2.1 Data types

Table 99 shows the data types that are used within the S/IP PDUs.

Table 99 – Definition of data types

Type	Definition	Comment
Bool	1 octet	Coding: 0(false), !=0 (true)
byte, uint8	1 octet unsigned integer	
int8	1 octet signed integer	
int16	2 octet signed integers	Coded in little endian
int32	4 octet signed integers	Coded in little endian
uint16	2 octet unsigned integers	Coded in little endian
uint32	4 octet unsigned integers	Coded in little endian
string	utf-8 coded	Fix-Len: • There are fixed number of reserved bytes. • The unused bytes shall be zero. • There is no terminating 0, if the coding completely fills up the reserved number of bytes! Variable-Len: • A leading explicit length contains the number of bytes. • There is no terminating 0.

The Type 19 specific IDN data (min, max, operation data) is exchanged as byte arrays.

7.1.11.2.2 Alignment

There are no spare octets due to an alignment.

7.1.11.2.3 Services

The IPS are different services, which can be accessed using different transport protocols. Due to specific limitations of the used protocols, not every transport protocol can be used for each service.

7.1.11.2.4 Node Identifier

Since the nodes in a Type 19 network may have more than one network interface, a node may be detected multiple times. In order to identify a node in network uniquely, another identifier, the so called node identifier, is introduced. The node identifier consists of six bytes. For slave devices the value of the node identifier shall be equal to the value of S-0-1019 MAC Address.

For nodes that support the IPS but do not have any Type 19 slave interface, the MAC address of the first Ethernet adapter shall be used as node identifier. Nodes without Ethernet interface cannot offer the IPS.

7.1.11.3 Transport Protocols

7.1.11.3.1 Introduction

Internet protocol services (IPS) use TCP, UDP and TFTP as transport protocols. Depending on the characteristics of a particular service, one or more of these transport protocols can be used for the transmission.

All these protocols run on top of the Internet Protocol (IP). The IPS do not support any fragmentation on the IP layer.

Table 100 gives an overview of the differences between TCP, UDP and TFTP based IPS transmission.

Table 100 – Overview on IP-based protocols

Restriction/ Addition	TCP	UDP	TFTP
Data exchange	Based on TCP sockets	Based on UDP datagrams	Based on UDP datagram
Limitation of data size	No limitation	Depends on MTU size	TFTP does not support the transfer of files which are larger than 32MB, required minimum MTU size is 576 Byte
Reliable transmission	yes	no	yes
Flow control	yes	no	yes
Multicast communication	no	yes	no
Requires additional S/IP connections	yes	no	no

7.1.11.3.2 UDP based S/IP Protocol Handling

7.1.11.3.2.1 General

UDP based S/IP requests are generally transmitted to UDP server port 35021 (0x88cd). The client shall use a socket bound to any port except 35021.

Depending on the used service S/IP uses broadcast or unicast UDP telegrams. In general unicasts are used to address single S/IP nodes and broadcasts are used to address multiple nodes.

7.1.11.3.2.2 Limitations of PDU size

The maximum size of UDP based S/IP transmission is restricted to the configured MTU size.

The following restrictions to the response packet can be derived by considering a regular UDP packet. Every S/IP telegram contains a fixed part which is 50 octets in size, i.e.

- 14 octets Ethernet II Header
- 20 octets IP header without optional fields
- 8 octets UDP header
- 8 octets S/IP header (Transaction ID, Message Type)

Hence minimum size of response PDU is 50 octets. If the total size of a S/IP response exceeds the configured MTU size, the server must respond by sending an exception (see 7.1.11.4.3).

NOTE In order to be able to use TFTP over UDP the MTU size has to be at least 576 octets, see also RFC-879.

7.1.11.3.2.3 Reliability of Transmission

In general UDP does not guarantee the exchange of messages. Request or response messages may be lost during the communication.

In order to overcome this problem, the client shall implement a timeout for each request. After the timeout, the client may send the same request again with the same transaction ID or new transaction ID.

The server will process each incoming request and send the response back to the client. The server shall not implement a functionality to ensure that duplicated requests will be executed only once.

7.1.11.3.2.4 Unicast Services

S/IP services using UDP unicasts can be used if server and client are in the same or in different subnets. The response of the server response is a UDP unicast too.

7.1.11.3.2.5 Broadcast Services

S/IP uses UDP broadcasts to

- reach many devices with one request
- enable communication to non-well configured devices. For example: a device may be connected to the same physical network, but has an unknown or invalid IP configuration. The IP address may be of an invalid or different subnet.

UDP broadcast telegrams can be responded by the server with UDP unicast or UDP broadcast messages. If the IP subnet of the client is different from the IP subnet of the server, the server shall send the response as a broadcast message. Otherwise the response shall be sent as a unicast message to avoid CPU load on other devices.

7.1.11.3.3 TCP based S/IP Protocol Handling

7.1.11.3.3.1 General

S/IP requests are generally transmitted to TCP server port 35021 (0x88cd). The client can use any port assigned by its TCP stack.

If TCP based S/IP is used, an S/IP connection has to be established before further S/IP services can be requested.

7.1.11.3.3.2 S/IP Connections

The S/IP connection provides the following features:

- The S/IP connection is established by the client.
- The S/IP connection is timeout controlled by the client and the server.
- The S/IP connection properties including timeouts, version information and supported message types are negotiated during connection start-up.
- The server may offer several connections. The client may connect to one server several times by using several TCP sockets.

- While shutting down the connection, all the resources used internally should be released in the client and the server.

The S/IP connection has to be established after the initiation of the TCP socket connection.

The services required for the connection initiation are described in S/IP Connection Services.

If the server is not able to serve a TCP based S/IP connection the server shall send an exception to the client (see 7.1.11.4.3).

S/IP Connections are controlled by the following two timeouts.

- Lease timeout: The lease timeout specifies the amount of time after which the server shutdowns an idle connection to the client. This timeout exists to save resources on the server in case of losing the link to the client without a proper shutdown of the TCP connection. The client should use the S/IP Ping Service to prevent the lease timeout. Every new request from client shall restart the lease timeout at the server.
- Busy timeout: The busy timeout specifies the amount of time within which the server should send a response to the client's request. If the server is processing the request and is not able to send the response within the busy timeout, it shall send busy responses to the client with the cycle of the busy timeout. If the client receives neither the busy response nor the response to pending request within the busy timeout, it shall shutdown the connection to the server.

Every response from server shall restart the busy timeout at the client. Figure 68 shows an example for busy response messages.

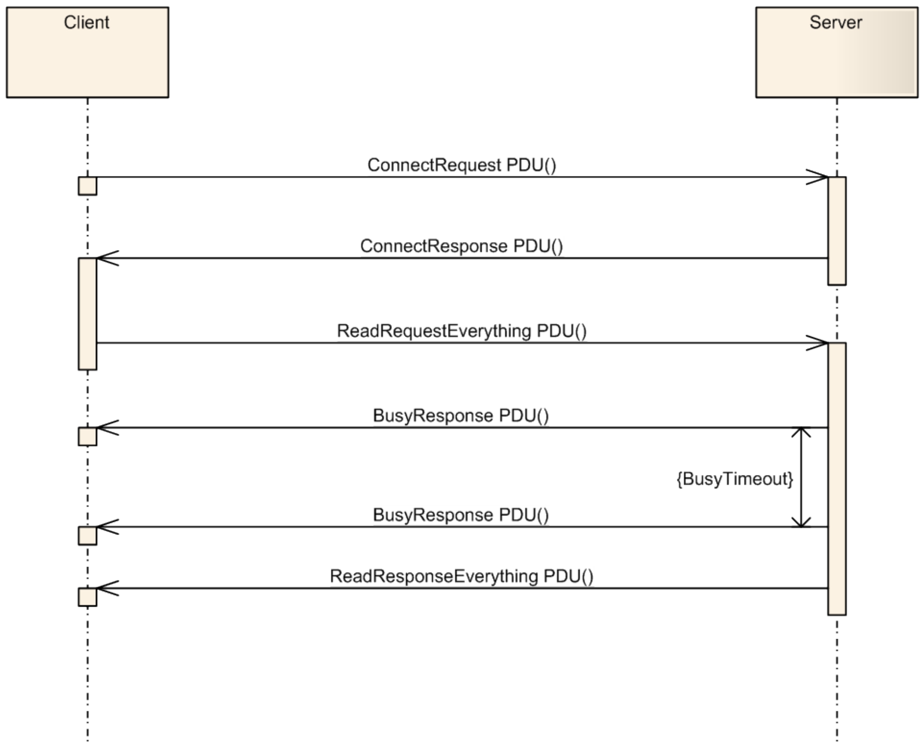

Figure 68 – S/IP busy response

The busy response restarts the busy timeout of an open request. The busy response looks as follows:

struct BusyResponse // MessageType: 68

{

}

Figure 69 shows a state machine of an S/IP connection from the client's point of view, including all states and transitions for busy- and lease-timeout, which needs to be implemented by S/IP client.

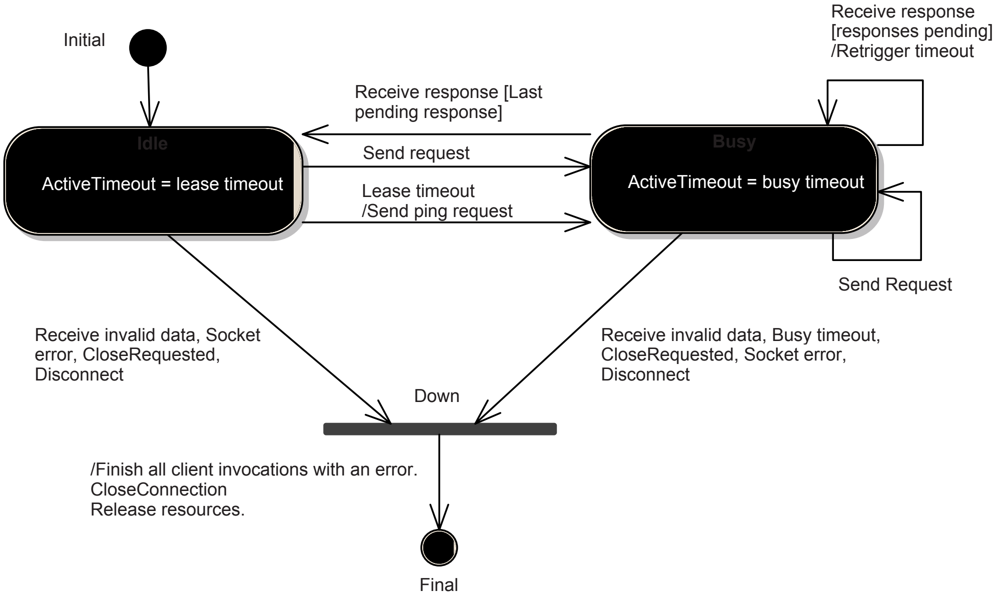

Figure 69 – Client connection

Figure 70 shows a state machine of an S/IP connection from the server's point of view, including all states and transitions for busy- and lease-timeout which needs to be implemented by S/IP server.

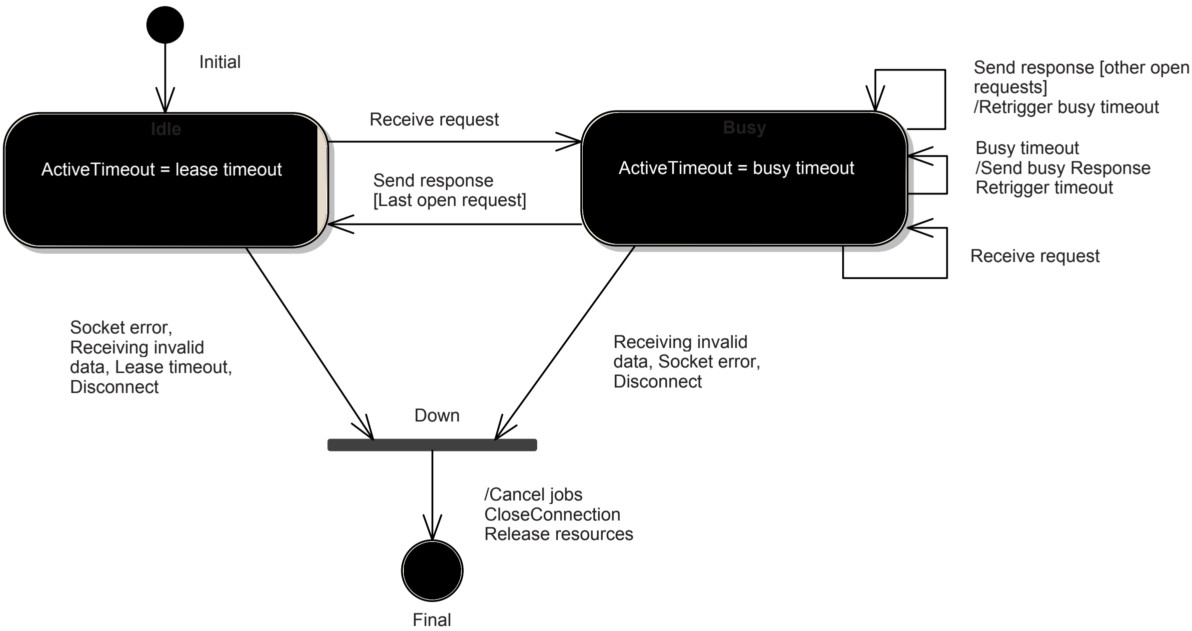

Figure 70 – Server connection

7.1.11.3.4 TFTP

TFTP as specified in RFC 1350 shall be the standard protocol for file transfer.

The Type 19 device shall implement a TFTP server.

- The file name activates an associated action on the Type 19 device.
- The content shall be checked by the Type 19 device. If the Type 19 device is in an improper operation state, it shall return the corresponding TFTP error message. E.g. access violation including a meaningful error message.
- The string error-message of the TFTP protocol shall be used to transmit meaningful error messages in readable form.
- It shall be possible to upload firmware via TFTP.

It is allowed to trigger functions or commands in the device via TFTP using special file names. Functions, file names and file contents are vendor specific and not defined by Type 19.

NOTE TFTP does not support the transfer of files which are larger than 32 MB.

7.1.11.4 Communication

7.1.11.4.1 General

Each S/IP service is defined by a request and a response message. The client can send several requests before it receives any response from the server. Furthermore the server can process incoming requests in any order.

Figure 71 shows an example of three requests answered by the server in a different order.

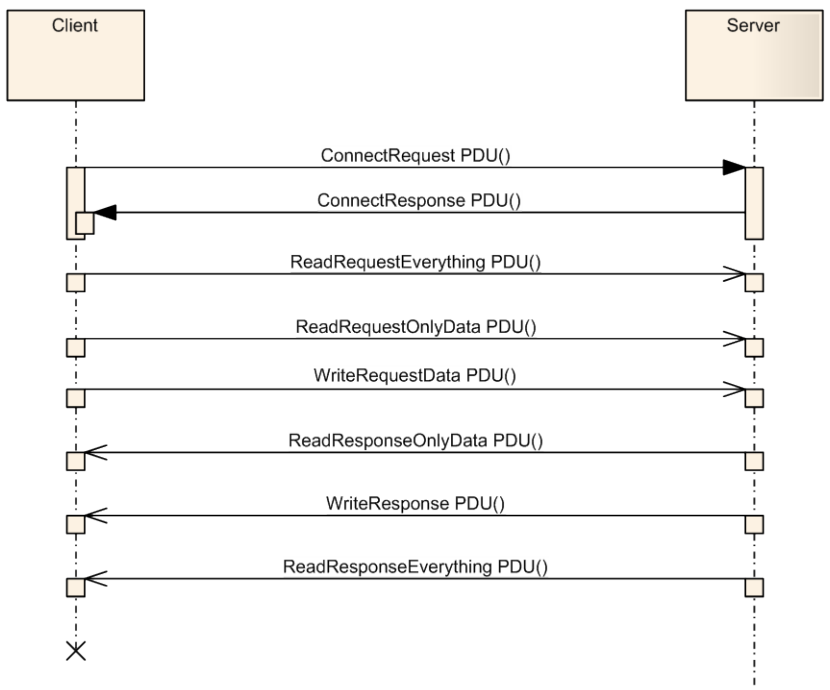

Figure 71 – S/IP asynchronous request

If the server receives a request with an unknown message type, the request shall be ignored and an exception shall be returned to the client. Broadcast messages with an unknown message type shall not be responded with an error exception (see 7.1.11.4.3).

Every request and response is composed of a S/IP Header followed by a S/IP service data unit (SDU), shown in Figure 72.

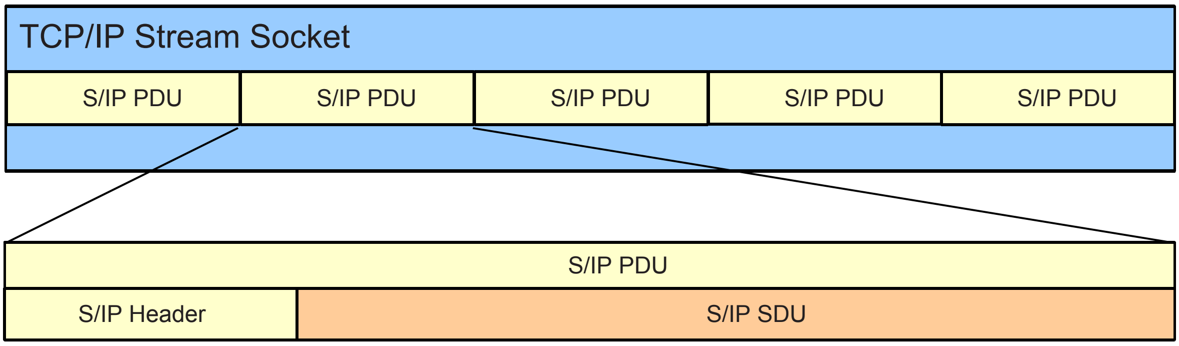

Figure 72 – S/IP PDU

7.1.11.4.2 S/IP Header

7.1.11.4.2.1 General

The S/IP header contains a transaction identifier and the message type of the PDU. The structure looks as follows:

```
struct header {

  uint32 TransactionID;

  uint32 MessageType;

}
```

7.1.11.4.2.2 Transaction ID

In order to distinguish different request-response pairs in the client, the S/IP header contains a transaction identifier. This transaction identifier is an identification number which is set by the requesting client for each request. The server uses the same transaction identifier (as of request) for the corresponding response. The client uses this transaction identifier of a response to assign the response to a corresponding pending request.

7.1.11.4.2.3 Message Types

The message type identifies the requested service of the Type 19 devices.

Table 101 shows the services that have been standardized by S/IP.

Table 101 – Message Types

S/IP Services	Message Type Request	Message Type Response	TCP	UDP
SupportedUDPServices	61	62	-	X
Connect	63	64	X	-
Ping	65	66	X	X
ReadEverything	69	70	X	X
ReadOnlyData	71	72	X	X

S/IP Services	Message Type Request	Message Type Response	TCP	UDP
ReadDescription	73	74	X	X
WriteName	75	76	X	X
WriteAttribute	77	78	X	X
WriteUnit	79	80	X	X
WriteMinMax	81	82	X	X
WriteData	83	84	X	X
WriteDataBits	85	86	X	X
ReadDataStatus	87	88	X	X
Nameplate	89	90	X	X
NameplateBroadcast	99	-	-	X
Browse	91	100	-	X
Identify	93	94	-	X
SetIp	95	96	-	X
Watchdog	101	102	X	X
Interfaces	103	104	X	X
Slaves	105	106	X	X
BrowseOnSlaveInterfaces	107	108	X	X
ReadSegment	109	110	X	X
Exception	-	67	X	X
Busy	-	68	X	X
Reset	97	-	-	X
Reserved	92	-	-	-

Manufacturer-specific MessageTypes set the most significant bit of the MessageType to 1 (see Table 102). The following 15 bits contain the actual manufacturer-specific MessageType (0..32767). The least significant 16 bits contain the Vendor Code managed by SI (S-0-1300.x.03 Vendor Code).

Table 102 – User-specific Message Types

(User specific) Message Type	1uuuuuuuuuuuuuuuvvvvvvvvvvvvvvvv (binary)
u: User specific Message Type	(0..32767)
v: Type 19	VendorCode

7.1.11.4.3 Exception Handling

7.1.11.4.3.1 General

If a service is requested, which cannot be handled successfully by the server, the server returns an exception message to the client. Transaction identifier of the exception message will be same as the transaction identifier of the request.

A server shall not respond with an exception to a UDP broadcast request, which does not contain its node identifier.

Figure 73 shows an exception for a request of a non-existent IDN.

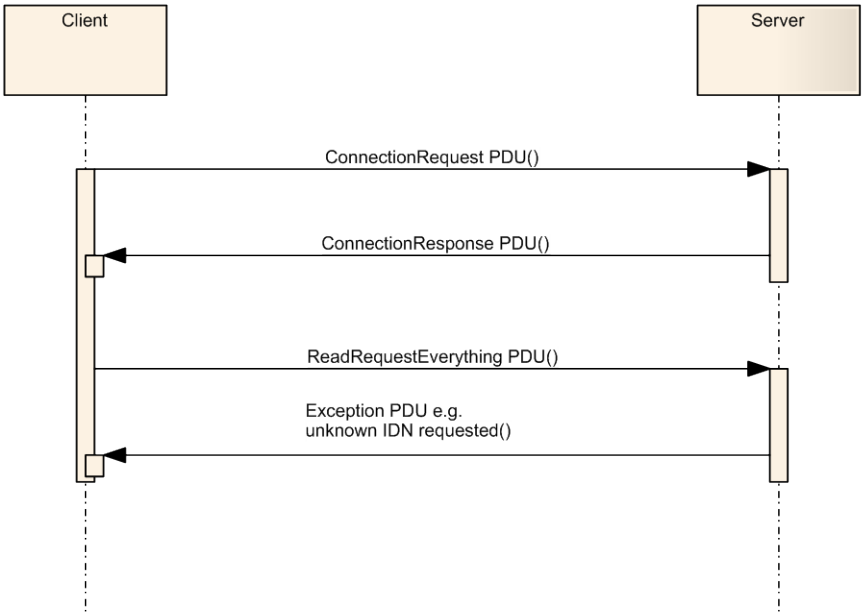

Figure 73 – S/IP error response

7.1.11.4.3.2 Exception structure

The exception message contains a common error code and an optional service specific error code.

```
struct Exception               // MessageType: 67

{

  uint16 CommonErrorCode;      // see enum below

  uint32 SpecificErrorCode;    // service specific error codes,

                               // e.g. error codes of service channel

}
```

7.1.11.4.3.3 Common error codes

```
enum CommonErrorCode

{

  CONNECTION_ERROR = 1,     // connection cannot be established

  TIMEOUT = 2,              // connection lost, timeout

  UNKNOWN_MESSAGE_TYPE = 3, // service not supported

  SERVICE_SPECIFIC = 4,     // service specific error

                            // --> see SpecificErrorCode for details

  PDU_TOO_LARGE = 5,        // request or response does not fit

                            // to the UDP datagram
```

```
                    // (limitation of PDU size)

  PDU_PROTOCOL_MISMATCH = 6 // malformed PDU e.g. received UDP datagram does not

                    // correspond to the expected PDU size

}
```

NOTE For service specific error code definitions, refer to the corresponding S/IP services.

Table 103 lists the common error codes together with the context, in which they are used.

Table 103 – Common error codes

Error code	Context
CONNECTION_ERROR	If the server is not able to serve a TCP based S/IP connection. see TCP based communication initialization for further details.
TIMEOUT	If a timeout exceeds (see Timeouts for further details) or a TCP connection gets lost. Network activities are controlled by local timeout handling. If the server doesn't respond in time, this error code is used to indicate the error to the user on client-side.
UNKNOWN_MESSAGE_TYPE	If the server receives an unknown message type, it shall send an exception with this error code to the client. In case of a TCP based S/IP request the server returns the exception to the client and shall close the TCP stream socket connection.
SERVICE_SPECIFIC	Services are able to have their own error code. Further information is available in the SpecificErrorCode of the Exception structure. see also specific error code descriptions of the service invoked.
PDU_TOO_LARGE	This is an UDP specific error. see Limitations of PDU size for further details.
PDU_PROTOCOL_MISMATCH	This is an UDP specific error. E. g. the length of the received datagram is not conforming to the expected PDU size of the service. This error indicates an incompatible implementation.

7.1.11.5 Services

7.1.11.5.1 General

The following sections describe the services provided by S/IP.

The information, which services are supported by a device, can be obtained by one of the following mechanisms:

- as a part of the connect response if TCP based S/IP connections are used (see 7.1.11.5.2).
- the SupportedUDPServices service can be used to get a list of all UDP services supported by a device (see 7.1.11.5.7.1).

7.1.11.5.2 S/IP Connection Service

7.1.11.5.2.1 General

Are used to establish and maintain an S/IP connection using TCP based S/IP.

7.1.11.5.2.2 Connect Service

This service can be used via the following transport protocol: TCP.

7.1.11.5.2.2.1 Connect Request

In order to initiate an S/IP connection, the client sends a ConnectRequest PDU to the server. This request contains the desired S/IP version number and desired timeout values for the connection.

```
struct Connect                // MessageType: 63
{
 uint32 version;              // S/IP protocol version
                              // version=1 shall be used for this
                              // protocol version
 uint32 busyTimeout;          // requested busy-timeout in milliseconds,
                              // the server should use
 uint32 leaseTimeout;         // requested lease-timeout in milliseconds,
                              // the server should use
}
```

7.1.11.5.2.2.2 Connect Response

If the server is able to serve the connection, a ConnectResponse PDU shall be returned to the client. In the ConnectionResponsePDU, the server provides its protocol version and the busy timeout and lease timeout values that are active for this connection to the client. If the server cannot serve the desired version of the client, it shall respond with an available version number that is lower or equal than the desired version number.

Furthermore, the ConnectResponse PDU contains a list of services supported by the server.

```
struct ConnectResponse          // MessageType: 64
{
  uint32  version;              // S/IP protocol version
                                // version=1 is used for this
                                // protocol version
  uint32 busyTimeout;           // busy-timeout in milliseconds,
                                // the server is using
  uint32 leaseTimeout;          // lease-timeout in milliseconds,
                                // the server is using
                                // typically 10000 - 30000 msec
```

```
  uint32 noMessageTypes;        // number of Request MessageTypes

  uint32 messagetypes[];        // The supported Request MessageTypes

                                // of the server on this TCP connection.

                                // The client must only use these

                                // message types in a request.

}
```

If the server is not able to serve the connection an Exception PDU shall be returned.

If the timeout values in the ConnectResponse PDU differs from the values of the ConnectRequest PDU, the client is responsible to adjust the timeout values to consider the delay times and reaction times of the application.

7.1.11.5.2.3 Ping Service

This service can be used via the following transport protocols: TCP, UDP (unicast).

If the server does not receive any message from the client before the lease timeout, it may close the connection to the client. The client may re-trigger the lease timeout in the server using the ping service.

7.1.11.5.2.3.1 Ping Request

```
struct Ping         // MessageType: 65

{

}
```

7.1.11.5.2.3.2 Ping Response

```
struct PingResponse // MessageType: 66  (pong)

{

}
```

7.1.11.5.3 Explore and IP Configuration Service

7.1.11.5.3.1 General

The following services are used to browse for devices within Type 19 networks and to configure their IP settings.

7.1.11.5.3.2 Browse service

This service can be used via the following transport protocol: UDP (broadcast).

The Browse service is used to detect devices within the same subnet.

Figure 74 shows the following sequence of the Browse services.

- A software tool (client) sends a browse request as global UDP broadcast to the local network. Within this browse request it is possible to set filters so that not all local Type 19 devices (servers) need to send a response. Possible filters are:
  – Master only: Only devices those are communication masters on the local network shall respond.
  – Address range: Only devices those fit the desired Device address range shall respond.
- All the Type 19 devices (servers) which receives the browse request and meets the filter condition shall send a browse response. This response contains the information like device's current network configuration, FSP types and the name that easily identifies that device.

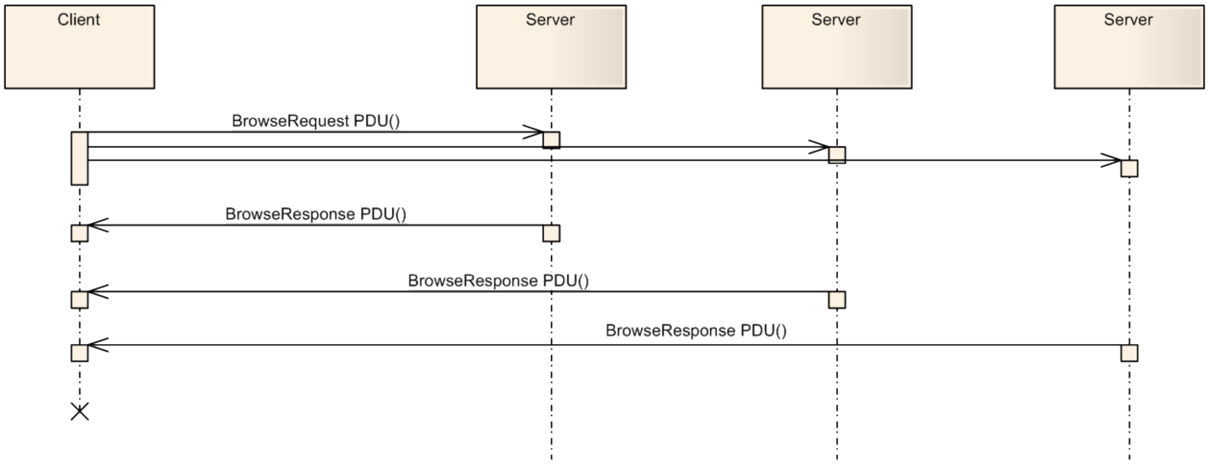

Figure 74 – UDP Browsing

7.1.11.5.3.2.1 Browse Request

```
struct Browse                  // MessageType = 91

{

                               // ipaddress of the sending node.

                               // The ip-address must correspond

                               // to the interface of the sender

  byte  ipaddress[4];          // [0] msb, [3] lsb

  bool  masterOnly;            // if != 0, only devices who have

                               // a Type 19 master interface

                               // shall reply


  uint16 lowerType19Address;   // Only devices with a Type 19 address

                               // between lower and upper bound shall

                               // reply.  If both values are 0, all

                               // devices shall reply. This filter
```

```
                                  // setting is not relevant for
                                  // master devices.
  uint16 upperType19Address;
}
```

7.1.11.5.3.2.2 Browse Response

The BrowseResponse depends on the interface on which the BrowseRequest has been received. The interface specific items are MAC Address, DHCP Features, DHCP Mode, IP Address, Subnet and Gateway.

If the implementing service doesn't know the receiving interface, it should send a response for each interface over all interfaces.

Example: Consider a device having 2 interfaces, interface 1 and interface 2. In this case service should

- Send the response containing the information for interface 1 over interface 1 and interface 2 and
- Send the response containing the information for interface 2 over interface 1 and interface 2.

The client (the one which sent the browse request) should check the Node Identifier to detect multiple responses from a node.

```
struct BrowseResponse             // MessageType: 100
{
  uint32  version;                // S/IP protocol version
                                  // version=1 is used for this
                                  // protocol version
  byte node_identifier[6];        // node identifier
                                  // 00:11:22:33:44:55 [0] = 00, [5] = 55
  byte macAddress[6];             // mac address of the interface from which
                                  // the response has been sent.
                                  // 00:11:22:33:44:55 [0] = 00, [5] = 55
  byte DHCPFeatures;              // bit[0]: dhcp client available
                                  // bit[1]: dhcp server available
                                  // bit[2]: dhcp relay agent available
  byte DHCPMode;                  // 0: no dhcp a static IP address is used
                                  // 1: dhcp client active
                                  // 2: dhcp server active
```

```
                                // 3: dhcp relay agent active
  byte ipaddress[4];            // [0] msb, [3] lsb
  byte subnet[4];               // [0] msb, [3] lsb
  byte gateway[4];              // ip address of responsible gateway
  uint32 dplength;              // length of display name in octets
  byte displayname[dplength];   // display name to identify the device
                                // in an browser
                                // length should not exceed 64 characters
  uint32 hnlength;              // length of hostname in octets
  byte hostname[hnlength];      // name to identify the node S-0-1039
}
```

If a server receives a request with an invalid PDU size, it shall ignore the message and shall not send an error exception.

The element "displayname" may contain information to uniquely identify the found device. It is supposed to be directly displayed as single information in a user interface. The displayname should not exceed 64 characters. It should be language neutral.

Examples:

- <device type> <serial number> --> "LXM62PS (12345678-0000)"
- <application type> <device identification number> <sudevice address> --> "AT:X-Achse ID:8a AD:12"

7.1.11.5.3.3 SetIP service

This service can be used via the following transport protocol: UDP (broadcast).

The SetIp service is used to set an IP configuration of an interface.

If DHCP client is enabled ipaddress, subnet and gateway are ignored.

Figure 75 shows the sequence for the network configuration service.

- A client sends a set IP configuration request as global UDP broadcast to the local network. The device to be configured is addressed using the node identifier. The request contains a complete IP configuration for the device. If the transmitted gateway address does not fit the transmitted network configuration, no default gateway shall be set (0.0.0.0). The new configuration can be set persistent or temporarily (until reboot). This is signaled by the field persistent in the request structure.
- The addressed Type 19 device shall either
  - – Respond with a set IP response message and activate the new network configuration. The order, in which both actions are executed, is not defined. That means, the set IP response may be send with the old or new IP configuration.
  - – Respond with an error message as UDP broadcast. This may be done if:
    - Device is in operational mode controlled by a Type 19 master or

- Device does not support setting a temporary network configuration.

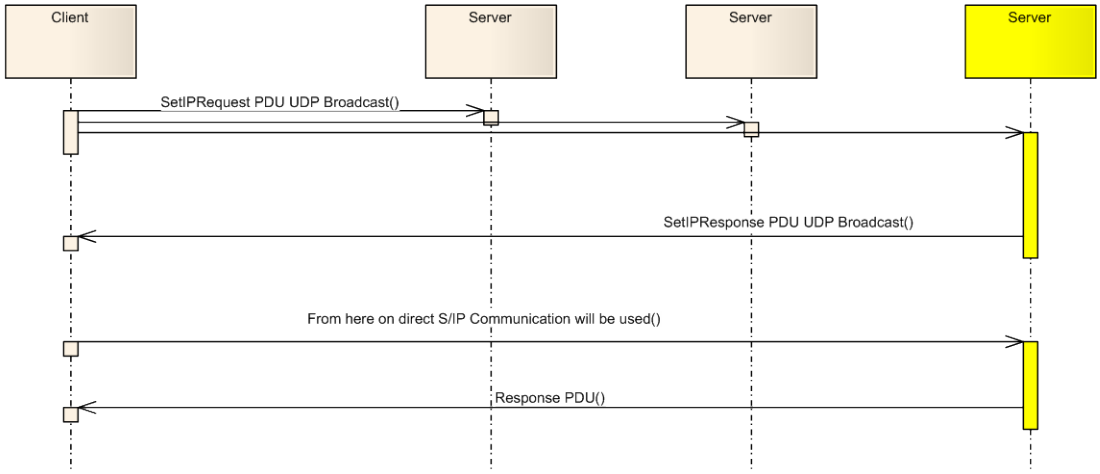

Figure 75 – Sequence of setting a new network configuration on one device using UDP

NOTE 1 Set the gateway element of the request to "0.0.0.0" to disable the gateway functionality.

NOTE 2 If the service is used for the slave interface, the settings should be reflected by the parameters: ipaddress (see S-0-1020 IP address), subnet (see S-0-1021 Subnet Mask), gateway (see S-0-1022 Gateway address).

7.1.11.5.3.3.1 SetIp Request

```
struct SetIp               // MessageType = 95

{                          // node identifier of the device

  byte node_identifier[6]; // 00:11:22:33:44:55 [0] = 00, [5] = 55

  byte macAddress[6];      // mac address of the interface.

                           // 00:11:22:33:44:55 [0] = 00, [5] = 55

  byte DHCPMode;           // 0: DHCP disabled

                           // 1: enable dhcp client

                           // 2: enable dhcp server

                           // 3: enable dhcp relay agent

  byte ipaddress[4];       // [0] msb, [3] lsb

                           // new subnet mask

  byte subnet[4];          // [0] msb, [3] lsb

                           // default gateway to forward ip messages

  byte gateway[4];         // [0] msb, [3] lsb

  bool persistent;         // if != 0, store settings persistent

  uint32 hnlength;         // length of hostname in octets
```

```
  byte hostname[hnlength]; // name to identify the node (see S-0-1039)
}
```

7.1.11.5.3.3.2 SetIp Response

```
struct SetIpResponse       // MessageType = 96
{
}
```

In case of an error an Exception SDU containing an error code will be sent.

```
enum SetIpErrorCode
{
  INVALID_DEVICE_STATE = 1      //  invalid device state
                                // (IP  configuration cannot be activated)
  TEMP_IP_NOT_SUPPORTED  = 2    // temporary IP configuration is
                                // not supported
  INVALID_SERVICE_DATA  = 3     // The given data are incorrect
  SET_IP_NOT_SUPPORTED  = 4     // setting of static IP_ address
                                // is not supported
  INVALID_DHCP_MODE  = 5        // device does not support DHCP_MODE
}
```

7.1.11.5.3.4 Interfaces service

This service can be used via the following transport protocols: TCP, UDP (unicast).

This service gets all the interfaces of a node.

7.1.11.5.3.4.1 Interfaces Request

```
struct Interfaces                 // MessageType = 103
{
}
```

7.1.11.5.3.4.2 Interfaces Response

```
struct InterfacesResponse       // MessageType = 104
{
  byte node_identifier[6];      // 00:11:22:33:44:55 [0] = 00, [5] = 55
  uint32 dplength;              // length of display name in octets
```

```
  byte displayname[dplength];   // display name to identify the device
                                // in an browser, see BrowseRespone
                                // length should not exceed 64 characters
  uint32 noInterfaces;          // number of interfaces
  struct
  {
    byte macAddress[6];    // mac address of the interface 00:11:22:33:44:55 [0] = 00,
[5] = 55
    byte ipAddress[4];     // [0] msb, [3] lsb ipaddress of the slave interface
    byte interfaceType;    // 0: an Ethernet interface, e. g. engineering port
                           // 1: a Type 19 Slave interface
                           // 2: a Type 19 Master interface
  } interfaces[noInterfaces];
}
```

7.1.11.5.3.5 Slaves service

This service can be used via the following transport protocols: TCP, UDP (unicast).

This service gets all slaves connected to a master interface. Only Type 19 masters have to implement this service.

7.1.11.5.3.5.1 Slaves Request

```
struct Slaves                       // MessageType = 105
{
  byte macAddress[6];               // mac address of the master interface
                                    // 00:11:22:33:44:55 [0] = 00, [5] = 55
}
```

7.1.11.5.3.5.2 Slaves Response

```
struct SlavesResponse         // MessageType = 106
{
  uint32 noSlaves;            // number of slaves
  struct
  {
    byte ipaddress[4];        // [0] msb, [3] lsb ipaddress of the slave
```

```
    byte node_identifier[6];  // 00:11:22:33:44:55 [0] = 00, [5] = 55
  } slaves[noSlaves];
}
```

In case of an error an Exception SDU containing an error code will be sent.

```
enum SlavesErrorCode
{
  INVALID_MAC_ADDRESS = 1       // Interface MAC address is not valid
                                // for Slaves service
}
```

7.1.11.5.3.6 BrowseOnSlaveInterface Interface

This service can be used via the following transport protocols: TCP, UDP (unicast).

This service requests the master to browse on another slave interface for masters. Only Type 19 masters have to implement this service. This service allows a slave of ring1 to get information from its master (ring1) from another ring (ring2). The node, master of ring1, must be a slave of ring2. Browse responses are collected by the server for a duration of timeToBrowse. After timeToBrowse the server will report the collected responses to the client.

7.1.11.5.3.6.1 BrowseOnSlaveInterface Request

```
struct BrowseOnSlaveInterface       // MessageType = 107
{
  byte macAddress[6];               // macAddress of the slave interface
                                    // to browse for masters
  int32 timeToBrowse;               // time to wait for responses in
                                    // milliseconds
}
```

7.1.11.5.3.6.2 BrowseOnSlaveInterface Response

```
struct BrowseOnSlaveInterfaceResponse // MessageType = 108
{
  uint32 noMasters;                     // number of masters
  struct
  {
    byte node_identifier[6];    // master node identifier
                                // 00:11:22:33:44:55 [0] = 00, [5] = 55
```

```
    byte ipaddress[4];          // master ip address
                                // [0] msb, [3] lsb ipaddress
                                // of the slave
  } masters[noMasters];
}
```

In case of an error an Exception SDU containing an error code will be sent.

```
enum BrowseOnSlaveInterfaceErrorCode
{
  INVALID_MAC_ADDRESS = 1       // Interface MAC address is not valid
                                // for Slaves service
}
```

7.1.11.5.4 Identification Services

7.1.11.5.4.1 General

The following services allow the identification of S/IP devices.

7.1.11.5.4.2 Identify service

This service can be used via the following transport protocol: UDP (broadcast).

The Identify service is used for the identification of a device, by instructing the device to identify itself, for example blinking.

Figure 76 shows the sequence of the UDP identify service.

- A client sends an identify request as global UDP broadcast to the local network. The device to be identified is addressed using the node identifier in the identify request.
- The addressed Type 19 device shall identify itself, the device signals for 4 seconds. After that, signaling is stopped automatically. The signaling LED behavior is defined in the section Type 19_LED. Identified device shall also send an identify response.

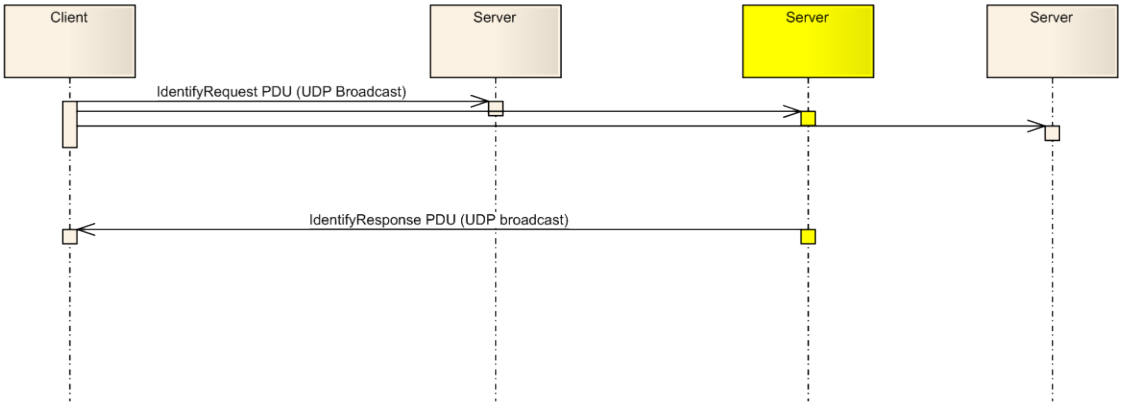

Figure 76 – UDP Identification

7.1.11.5.4.2.1 Identify Request

```
struct Identify             // MessageType = 93

{

                            // node identifier of the device

  byte node_identifier[6]; // 00:11:22:33:44:55 [0] = 00, [5] = 55

}
```

7.1.11.5.4.2.2 Identify Response

Only the addressed device sends a response.

```
struct IdentifyResponse    // MessageType = 94

{

}
```

7.1.11.5.4.3 Nameplate service

This service can be used via the following transport protocols: TCP, UDP (unicast).

The nameplate service requests the electronic nameplate of the node. The nameplate information is returned as a list of nameplate entries. Each entry contains a leading id and a string value. The ids are defined in Table 104. If not explicitly defined, the display format of the string value is given by the attribute of the referred IDN.

7.1.11.5.4.3.1 Nameplate Request

```
struct Nameplate                    // MessageType = 89

{

}
```

7.1.11.5.4.3.2 Nameplate Response

```
struct NameplateResponse            // MessageType = 90
```

```
{
  uint32 noEntries;              // number of nameplate entries
  struct
  {
     byte id;                    // id of the nameplate entry
     uint32 length;              // length of value in octets
     byte[length] value;         // value of the nameplate entry:
                                 // string Variable-Len
  } nameplate_entries[noEntries];
}
```

Table 104 – Nameplate IDs

ID	Name	Comments
1	reserved	
2	Vendor name	refers to S-0-1300.x.02 Vendor Name
3	Type 19 Vendor code	refers to S-0-1300.x.03 Vendor Code
4	Device name	refers to S-0-1300.x.04 Device Name
5	Vendor device ID	refers to S-0-1300.x.05 Vendor Device ID
6	(reserved)	
7	Function Revision	refers to S-0-1300.x.07 Function Revision
8	HW Revision	refers to S-0-1300.x.08 Hardware Revision
9	FW Revision	refers to S-0-1300.x.09 Software Revision
10	Firmware loader revision	refers to S-0-1300.x.10 Firmware Loader Revision
11	(reserved)	
12	Serial number	refers to S-0-1300.x.12 Serial Number
13	Manufacturing Date	refers to S-0-1300.x.13 Manufacturing Date
14 …19	(reserved)	
20	Operational Hours	refers to S-0-1300.x.20 Operational Hours
21 … 31	(reserved)	
32	Hostname	refers to S-0-1039 Hostname
33	Quantity of slaves	refers to the number of list elements of S-0-1046 List of sub-device addresses in device the display format shall unsigned decimal
34 … 127	(reserved)	
128 … 255	Manufacturer-specific	

7.1.11.5.4.4 NameplateBroadcast service

This service can be used via the following transport protocol: UDP (broadcast).

The NameplateBroadcast service is used to gather information about a device.

The NameplateResponse returns a list of IDN parameters. It may be used for additional info about devices without defined communication settings.

```
struct NameplateBroadcast  // MessageType = 99

{                          // node identifier of the device

  byte node_identifier[6]; // 00:11:22:33:44:55 [0] = 00, [5] = 55

}
```

The response is defined in Nameplate Service.

7.1.11.5.5 Parameter Access

7.1.11.5.5.1 General

In most cases a Type 19 device will have only one interface and hence only one IP address. To access the parameters (reading and writing parameters) from different slaves of a device, slaves need to be addressed in the request. This addressing will be done using slave index. The slave-index is an internal numeration of all slaves in a device.

The number of slaves in one device corresponds to the number of elements in the global parameter S-0-1046. Absence of parameter S-0-1046 indicates that the device has exactly one slave. In this case zero shall be used for slave index. Global parameters are available with any slave index.

A device may define additional parameters using the Type 19 parameter structure. These parameters are accessible by the SlaveExtension. If not used, the SlaveExtension should be zero.

7.1.11.5.5.2 Parameter data

The coding of the parameter data including the min and max value corresponds to the coding of the service channel with some exceptions described below:

- Coding is done in little endian
- The length of the value is given with the attribute size of (1,2,4,8) in octets.
- Parameter with string as data uses lists of 1 byte elements. The length of the parameter data represents the coding length of the string in octets, not the number of characters. There is no terminating 0 byte. Strings are coded in utf-8.
- Lists:
  - The list header which contains the maximum length and the current length is not part of the S/IP data. Only list element value data's are part of the transmitted information.
  - The current length of the list will be returned by the datalength. The number of elements in the list may be evaluated using the element size information of the attribute and the datalength.
  - Use ReadEverything or ReadDescription to get the maximum available list size in octets.

7.1.11.5.5.3 IDNs

IDNs are passed in the services to address a specific parameter. IDNs are also part of the parameter value, if the parameter display format is "IDN". see the IDN specification how to code and decode the IDN.

7.1.11.5.5.4 Parameter service specific error codes

The following enumeration defines the service specific error codes for S/IP. see also the SVC error codes

```
enum SipErrorCode

{

  // 0x0 - 0x7FFFFFFF                     // reserved for SVC Error Codes,

                                          //  e.g. 0x1001 No IDN

  // > 0x80000000                         // Addition Error Codes

  SLAVE_INDEX_INVALID      = 0x80000001,

  PARAMETER_INVALID        = 0x80000002, // can be used to avoid invalid access;

                                          // e.g. write access to procedure commands

  PARAMETER_NOT_ACCESSIBLE = 0x80000003, // Parameter cannot be accessed

                                          // e.g. if parameter database is not ready

}
```

7.1.11.5.5.5 ReadEverything Service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.5.1 ReadEverything Request

```
struct ReadEverything          // MessageType: 69

{

  uint16 SlaveIndex;

  uint16 SlaveExtension;

  uint32 IDN;

}
```

7.1.11.5.5.5.2 ReadEverything Response

```
struct ReadEverythingResponse  // MessageType: 70

{

  uint16   validelements;  // bitmask of the following elements,

                           // which are valid in this response

                           // invalid length elements shall be set to 0

                           // bitmask | description

                           // --------------------------------------------------
```

```
  uint16   data_status;    // 0x01    | data valid & command
                           //         | acknowledge ref. IDN numbers
  uint16   namelength;     // 0x02    | length of datablock element 2 in octets
  uint32   attribute;      // 0x04    | datablock element 3
  uint16   unitlength;     // 0x08    | length of datablock element 4 in octets
  byte[8]  min;            // 0x10    | datablock element 5
  byte[8]  max;            // 0x20    | datablock element 6
  uint32   maxlistlength;  // --      | maximum length of datablock element 7
                           //         | in octets. Valid if parameter is a list
  uint32   datalength;     // 0x40    | length of datablock element 7 in octets
  byte[namelength] name;   // 0x02    | name of parameter; string Variable-Len
  byte[unitlength] unit;   // 0x08    | unit of parameter; string Variable-Len
  byte[datalength] data;   // 0x40    | datablock element 7
}
```

The response shall contain all implemented elements of the IDN.

7.1.11.5.5.6 ReadyOnlyData service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.6.1 ReadyOnlyData Request

```
struct ReadOnlyData            // MessageType: 71
{
  uint16 SlaveIndex;
  uint16 SlaveExtension;
  uint32 IDN;
}
```

7.1.11.5.5.6.2 ReadyOnlyData Response

```
struct ReadOnlyDataResponse    // MessageType: 72
{
  uint32 attribute;
  uint32 length;               // length of data counted in octets
  byte[length] data;
```

```
}
```

7.1.11.5.5.7 ReadDescription service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.7.1 ReadDescription Request

```
struct ReadDescription         // MessageType: 73

{

  uint16 SlaveIndex;

  uint16 SlaveExtension;

  uint32 IDN;

}
```

7.1.11.5.5.7.2 ReadDescription Response

```
struct ReadDescriptionResponse // MessageType: 74

{

  uint16   validelements;  // bitmask of the following elements,

                           // which are valid in this response

                           // invalid length elements shall be set to 0

                            // bitmask | description

                           // -------------------------------------------------

  uint16   namelength;     // 0x02    | length of datablock element 2 in octets

  uint32   attribute;      // 0x04    | datablock element 3

  uint16   unitlength;     // 0x08    | length of datablock element 4 in octets

  byte[8]  min;            // 0x10    | datablock element 5

  byte[8]  max;            // 0x20    | datablock element 6

  uint32   maxlistlength;  // --      | maximum length of datablock element 7

                           //         | in octets. Valid if parameter is a list

  byte[namelength] name;   // 0x02    | name of parameter; string Variable-Len

  byte[unitlength] unit;   // 0x08    | unit of parameter; string Variable-Len

}
```

NOTE The response has to contain all implemented elements of the IDN.

7.1.11.5.5.8 ReadDataStatus service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.8.1 ReadDataStatus Request

```
struct ReadDataStatus         // MessageType: 87
{
  uint16 SlaveIndex;
  uint16 SlaveExtension;
  uint32 IDN;
}
```

7.1.11.5.5.8.2 ReadDataStatus Response

```
struct ReadDataStatusResponse // MessageType: 88
{
  uint16   data_status;        // data valid & command acknowledge
                               // refer to the IDN numbers
}
```

7.1.11.5.5.9 ReadSegment service

This service can be used via the following transport protocols: TCP, UDP (unicast).

This service reads operation data elements. The response length depends on the data type.

7.1.11.5.5.9.1 ReadSegment Request

```
struct ReadSegment                   // MessageType: 109
{
  uint16 SlaveIndex;
  uint16 SlaveExtension;
  uint32 IDN;
  uint16 StartIndex;                 // 0 based index of the first
                                     // list element
  uint16 NumberOfRequestedElements;  // requested number of list elements
}
```

7.1.11.5.5.9.2 ReadSegment Response

```
struct ReadSegmentResponse           // MessageType: 110
```

```
{
  uint32 attribute;
  uint16 NumberOfAvailableElements;  // Number of listelements
  uint32 length;                     // length of data counted
                                     // in octets
  byte[length] data;                 // parameter data
}
```

The device may return fewer elements than requested. Response having fewer elements indicates that the MTU size can be too small or the end of the list is reached.

If the start index of the request is out of range, the server shall reply with the data length of zero.

IDNs with a fix length shall be treated as list IDNs with length 1.

7.1.11.5.5.10 WriteName service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.10.1 WriteName Request

```
struct WriteName               // MessageTypes: 75
{
  uint16           SlaveIndex;
  uint16           SlaveExtension;
  uint32           IDN;
  uint16           namelength;
  byte[namelength] name;       // string Variable-len */;
}
```

7.1.11.5.5.10.2 WriteName Response

```
struct WriteNameResponse       // MessageTypes: 76
{
                               // empty response; data was written correctly
                               // in case of an error, an exception will be
                               // sent instead
}
```

7.1.11.5.5.11 WriteAttribute service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.11.1 WriteAttribute Request

```
struct WriteAttribute          // MessageType: 77

{

  uint16  SlaveIndex;

  uint16  SlaveExtension;

  uint32  IDN;

  uint32  attribute;

}
```

7.1.11.5.5.11.2 WriteAttribute Response

```
struct WriteAttributeResponse  // MessageTypes: 78

{

                               // empty response; data was written correctly

                               // in case of an error, an exception will be

                               // sent instead

}
```

7.1.11.5.5.12 WriteUnit service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.12.1 WriteUnit Request

```
struct WriteUnit               // MessageType: 79

{

  uint16           SlaveIndex;

  uint16           SlaveExtension;

  uint32           IDN;

  uint16           unitlength;

  byte[unitlength] unit;       // string with var-len

}
```

7.1.11.5.5.12.2 WriteUnit Response

```
struct WriteUnitResponse     // MessageType: 80
```

```
{
                             // empty response; data was written correctly
                             // in case of an error, an exception will be
                             // sent instead
}
```

7.1.11.5.5.13 WriteMinMax service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.13.1 WriteMinMax Request

```
struct WriteMinMax           // MessageType: 81
{
  uint16  SlaveIndex;
  uint16  SlaveExtension;
  uint32  IDN;
  byte[8] min;               // datablock element 5
  byte[8] max;               // datablock element 6
}
```

7.1.11.5.5.13.2 WriteMinMax Response

```
struct WriteMinMaxResponse   // MessageType: 82
{
                             // empty response; data was written correctly
                             // in case of an error, an exception will be
                             // sent instead
}
```

7.1.11.5.5.14 WriteData service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.14.1 WriteData Request

```
struct WriteData             // MessageType: 83
{
  uint16  SlaveIndex;
  uint16  SlaveExtension;
```

```
  uint32  IDN;
  uint32  length;            // length of data counted in octets
  byte[length] data;
}
```

7.1.11.5.5.14.2 WriteData Response

```
struct WriteDataResponse     // MessageType: 84
{
                             // empty response; data was written correctly
                             // in case of an error, an exception will be
                             // sent instead
}
```

7.1.11.5.5.15 WriteDataBits service

This service can be used via the following transport protocols: TCP, UDP (unicast).

7.1.11.5.5.15.1 WriteDataBits Request

```
struct WriteDataBits         // MessageType: 85
{
  uint16  SlaveIndex;
  uint16  SlaveExtension;
  uint32  IDN;
  uint32  length;            // length of data and mask counted in octets
  byte[length] data;         // value of bits to be modified
  byte[length] dataMask;     // bits to be modified
}
```

If the length sent by the client is less than the size of the parameter, the server shall extend the MSB of mask and value with 0 octets.

NOTE WriteDataBits is supported only for non-list parameters.

7.1.11.5.5.15.2 WriteDataBits Response

```
struct WriteDataBitsResponse // MessageType: 86
{
                             // empty response; data was written
                             // correctly in case of an error, an
```

```
                          // exception will be sent instead

}
```

7.1.11.5.6 Device Management Services

7.1.11.5.6.1 Reset service

This service can be used via the following transport protocol: UDP (broadcast).

The Reset service is used to reset one or more devices in a network.

Since the reset service is going to interrupt the communication, the reset has to be delayed by the server. The devices which shall participate in the reset are identified by the node_identifiers. As soon as a reset request has been received at a device, it shall start an internal timer and after the timeout time has elapsed, reset will be performed. Because of possible data losses, it is recommended that the client repeats the request up to five times with a gap of ~100 ms between the requests. If a server receives a reset request twice, the timer shall be restarted. If there is any situation which does not allow the execution of the reset request, for example safety conditions, the reset request shall be ignored by the device.

The maximum number of devices which can be addressed by a reset request is restricted to 85. This is because of restriction on UDP packet size. If there are more than 85 devices which should be restarted, it is possible to send different reset requests consecutively.

This service shall perform a cold reset. The main purpose of this service is activating the depending update files, transferred for example via TFTP.

NOTE Devices will not send any response to this request.

7.1.11.5.6.1.1 Reset Request

```
struct Reset                                    // MessageType = 97

{

  uint32 timeout;                               // waiting time in milliseconds

                                                //  until  reset  is  performed  by  the
device

  uint16 no_node_identifiers;                     // number of node_identifiers inside
range

                                                // from 0 to 85. If 0, no device shall

                                                // perform a reset,

                                                // else the amount of node_identifiers

  byte node_identifier[no_node_identifiers][6]; // Array of node_identifiers:

                                                // 00:11:22:33:44:55 [0] = 00, [5] = 55

}
```

Figure 77 shows the usage of UDP ResetRequest. A client tool is used to send two reset requests. The first request intends to reset the first and the third device. After the request has been sent, the user wants to reset the second device, too. Therefore, the client tool sends a second request which addresses the second device only. After the timeout time has elapsed, the devices perform the hardware reset.

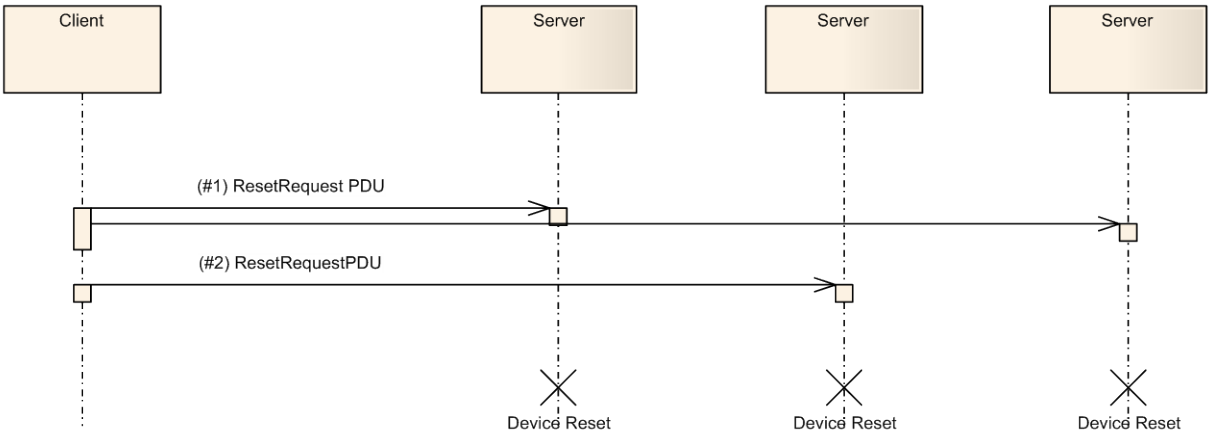

Figure 77 – Usage UDP reset request

NOTE Figure 77 does not show the recommended repetition of the sent requests.

7.1.11.5.6.2 TFTP Firmware Management / Update

This service can be used via the following transport protocol: TFTP.

- Firmware management uses the TFTP protocol.
- Any error shall be responded using the TFTP error messages, for example wrong firmware version including a describing error message. TFTP error messages shall not be used for a successful operation.

An overall activation of a new firmware shall be invoked using the reset service.

7.1.11.5.7 Other Services

7.1.11.5.7.1 SupportedUDPServices Services

This service can be used via the following transport protocol: UDP (unicast).

This service returns a list of all supported "UDP based S/IP" services (unicast and broadcast).

7.1.11.5.7.1.1 SupportedUDPServices Request

```
struct SupportedUDPServices         // MessageType: 61

{

}
```

7.1.11.5.7.1.2 SupportedUDPServices Response

```
struct SupportedUDPServicesResponse // MessageType: 62

{

  uint32 noMessageTypes;            // number of Request MessageTypes

  uint32 messagetypes[];            // The supported request
```

```
                                   // MessageTypes of the server
                                   // The client shall only use
                                   // these message types in a request.
}
```

7.1.11.5.7.2 Watchdog Services

This service can be used via the following transport protocols: TCP, UDP (unicast).

This service implements a keep alive mechanism for an activity. This can be used to ensure an "in time termination" of specific activities of the device, started by the client.

While the specific activity is being executed by the device, the client shall keep on triggering the watchdog to keep the activity running. With this mechanism, device will come to know that the client is still alive and observing the activity being executed. If the device does not receive watchdog request before the timeout, due to any communication problems, the device will stop the specific activity automatically. Refer to Figure 78 for details.

Example: This service should be used to terminate an axis movement initiated by S/IP client.

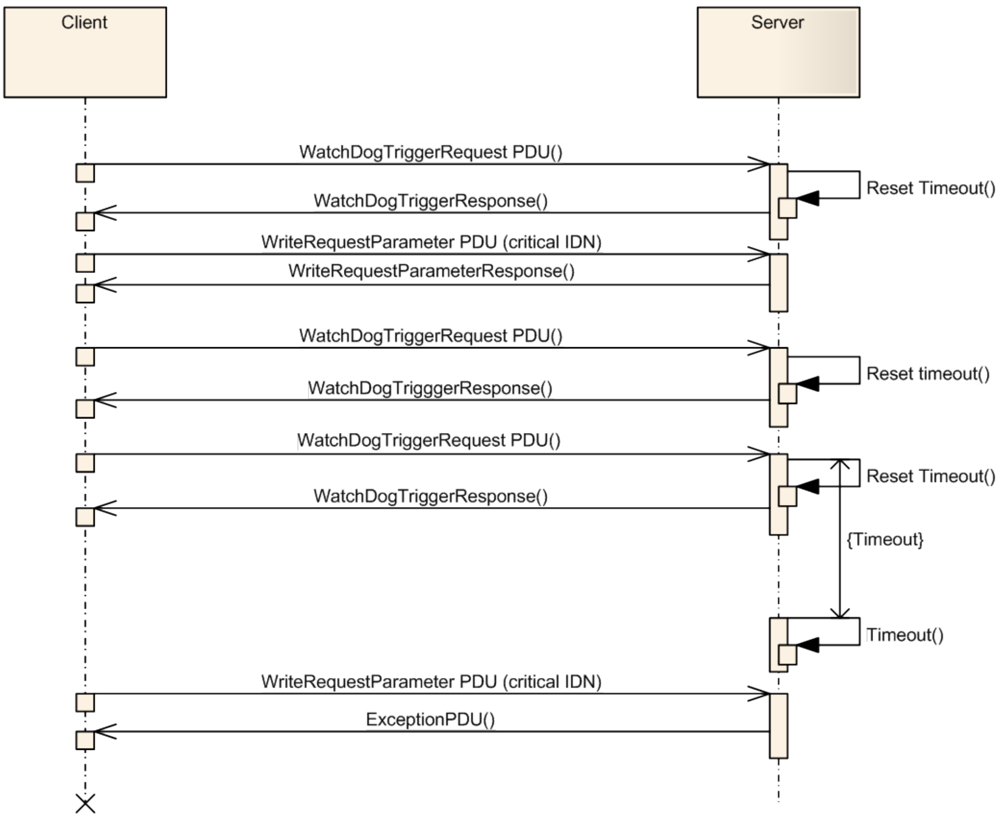

Figure 78 – Sequence for watchdog trigger service and client application timeout

This service message shall always be transmitted if an axis is moved using parameter access over S/IP. If this service message is not transmitted by the server for a dedicated time, a watchdog shall trigger and put the device into a safe state.

Since S/IP is only there to transport the data, functional safety cannot be provided by S/IP when moving axes. The definition of a safe state as well as the timeout values are dependent on the device and cannot be specified by S/IP.

As an example, critical IDNs like S-0-0134 on drives shall only be writable, as long as the watchdog has not expired. After the watchdog has expired which is the initial state of the watchdog, all critical IDNs shall be reset to fall back values. Every device needs to know its critical IDNs by itself. If the watchdog has expired and a critical IDN should be written, the SVC error message 0x700c, „Operation data is write protected, due to other settings. (for example, parameter, operation mode, drive enable, drive on etc.)“, shall be responded.

The request message received by a device resets the devices watchdog. The device will reply to the request with a response message including the expiration period of the watchdog in milliseconds, refers to Watchdog trigger service.

7.1.11.5.7.2.1 Watchdog Request

```
struct WatchdogTrigger         // MessageType: 101

{

}
```

7.1.11.5.7.2.2 Watchdog Response

```
struct WatchdogTriggerResponse // MessageType: 102

{

  uint32 timeout;              // in milliseconds

}
```

7.1.11.6 Classification

Type 19 defines several IPS classes that may be implemented by the devices.

If a device supports a class, all services of that class shall be implemented. Table 105, Table 106, Table 107, Table 108, Table 109 and Table 110 show the mandatory services for master and slaves. Table 105 shows the defined Internet protocol classes.

Table 105 – IPS classes

Class IDs	Class name	Comments
1	TCP Basic	
2	UDP Basic	
3	Device Management	
4	Explore & IP Configuration Services	
5	Type 19 Parameter access	

At least one of the classes TCP Basic or UDP Basic (see Table 106 and Table 107) shall be implemented.

Table 106 – Class TCP Basic

Service	If UDP is supported	If TCP is supported
Connect	-	X
Ping	-	X
Busy	-	X

Nameplate	-	X

Table 107 – Class UDP Basic

Service	If UDP is supported	If TCP is supported
Nameplate	X	-
SupportedUdpServices	X	-

Table 108 – Class Device Management

Service	If UDP is supported	If TCP is supported
TFTP Firmware Update	X	X

Table 109 – Explore & IP Configuration Services

Service	If UDP is supported	If TCP is supported
Browse	X	-
SetIpConfiguration	X	-
Identify	X	-

Table 110 – Class Type 19 Parameter Access

Service	If UDP is supported	If TCP is supported
ReadEverything	X	X
ReadOnlyData	X	X
ReadDescription	X	X
ReadDataStatus		X
ReadSegment	X	
WriteData	X	X
WriteDataBits	X	X

7.2 Synchronization

7.2.1 Network synchronization

7.2.1.1 General

The synchronization shall be used for synchronizable slaves which supports SCP_Sync only and consists of several functions.

- Compensation of the physical delay times (tring).
- Determination of the synchronization reference time (TSref) related to ring delay.
- Trigger of synchronization with end of MST (TTref).
- Determination of the optimal synchronization time (TSync) related to the processing times in the slaves.

The master shall measure the physical delay times (tring) in CP0, CP1 or CP2 as specified in this clause. The master shall transmit the S-0-1015 Ring delay to all synchronizable slaves. The master may measure also the physical delay times in CP4 for monitoring and diagnosis purposes.

Then the slaves shall calculate its internal synchronization timing (S-0-1016 Slave delay (P&S)). This synchronization time and all times derived from it in the slave shall be activated with the S-0-1024 SYNC delay measuring procedure command.

Synchronization shall be generated once per communication cycle by a valid MST only (see Figure 79). The MST is protected via an additional CRC. The end of the MST is the synchronization trigger (TTref) in every communication cycle (see Figure 80).

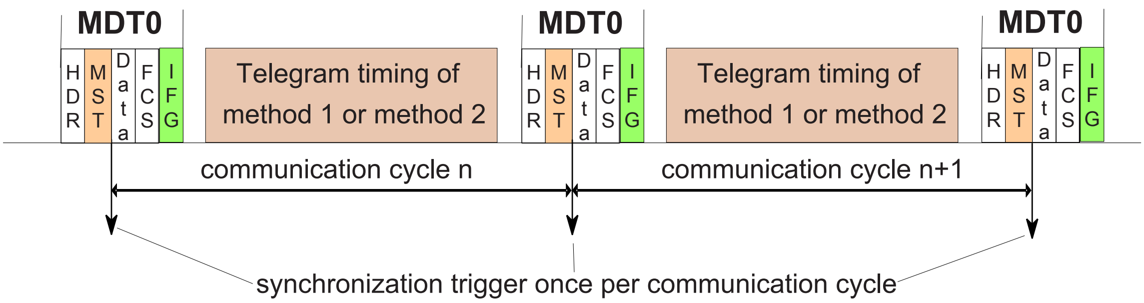

Figure 79 – Synchronization timing

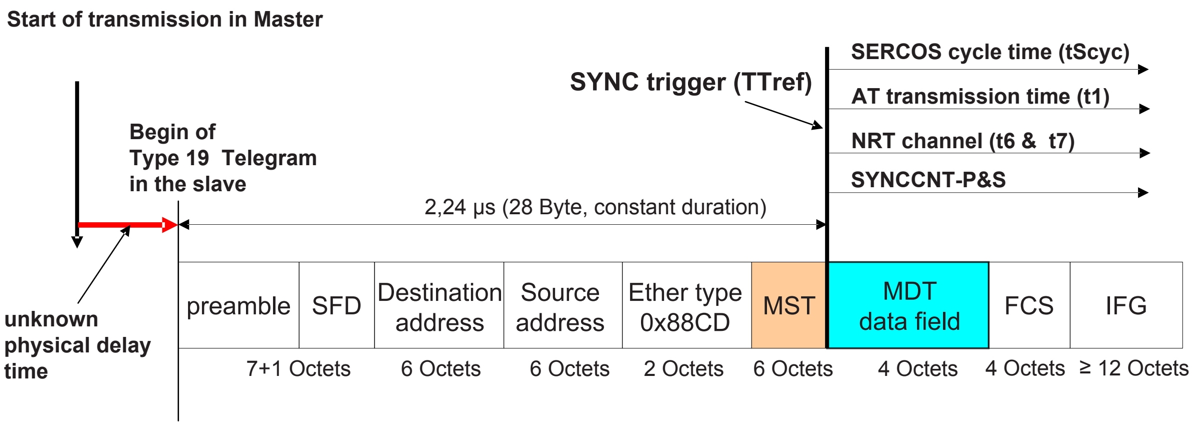

Figure 80 – Synchronization trigger

7.2.1.2 Ring delay acquisition procedure in the master

7.2.1.2.1 General

The master shall send MDTs cyclically as soon as it leaves NRT state for initializing the communication (CP0 to CP4). It shall then start sending MDT0 frames at the beginning of each communication cycle, within a jitter tolerance (MST jitter). The S-0-1023 SYNC jitter includes the MST jitter.

The ring delay acquisition procedure is the same for ring and for line topology. The master shall continuously monitor the results and compare them with the corresponding values that have been obtained during the last previously achieved ring delay acquisition procedure. It shall be able to issue warnings if any measured physical delay time differs from a previously obtained value by more than application dependent value and tolerance.

The master shall complete the ring delay acquisition procedure before it transmits the parameter S-0-1015 Ring delay to the slaves.

In addition, depending upon application, the master may activate it at any time during CP3 or CP4, i.e. for stability verification purposes, or when it detects topology changes (removing slaves, hot plugging slaves, ring recovery, etc.).

For the ring delay acquisition procedure the master performs the following features:

- measure the physical delay time (tRing) of the given topology,
- compute the parameter S-0-1015 Ring delay,
- transfer the parameter S-0-1015 Ring delay to each synchronizing slave,
- activate the S-0-1024 SYNC delay measuring procedure command in each synchronizing slave.

The following timing parameters shall be characteristics of the network:

- trep – time by which the received signal shall be delayed by a Type 19 interface in fast forward and loopback with forward (input to output, approx. 600 ns);
- tcable – time by which the transmitted signal is actually delayed by a cable (CAT5 max. 5,56 ns per meter; glass fiber max. 5 ns per meter);
- tRing – average of the physical delay times measured by the master:
  $\text{tRing} = \sum \text{tcable} + \sum \text{trep}$
- IFG jitter – dependent of the participants in the topology, the master shall determine the IFG jitter (see S-0-1036 Inter Frame Gap);
- Extra delay – additional delay defined or calculated by the master (for example in case of hot-plugging slaves, delay times of the master hardware, etc).

7.2.1.2.2 Determination of ring delay with topology line and ring

In topology line and ring the master shall use the same formula. In line is either P channel or S channel available. Both channels (P and S) are available in ring. Timing in master and slave, see Figure 81.

- The master shall measure at least 64 times the physical delays of the P channel (primary) and / or S channel (secondary) and determines the average (tRing) of all measurements.
- The master shall determine the IFG jitter dependent of the participants in the topology.
- The master may add an extra delay.
- The Master calculates the ring delay for a given line or ring topology using the following formula: S-0-1015 Ring delay = tRing + IFG jitter + extra delay.

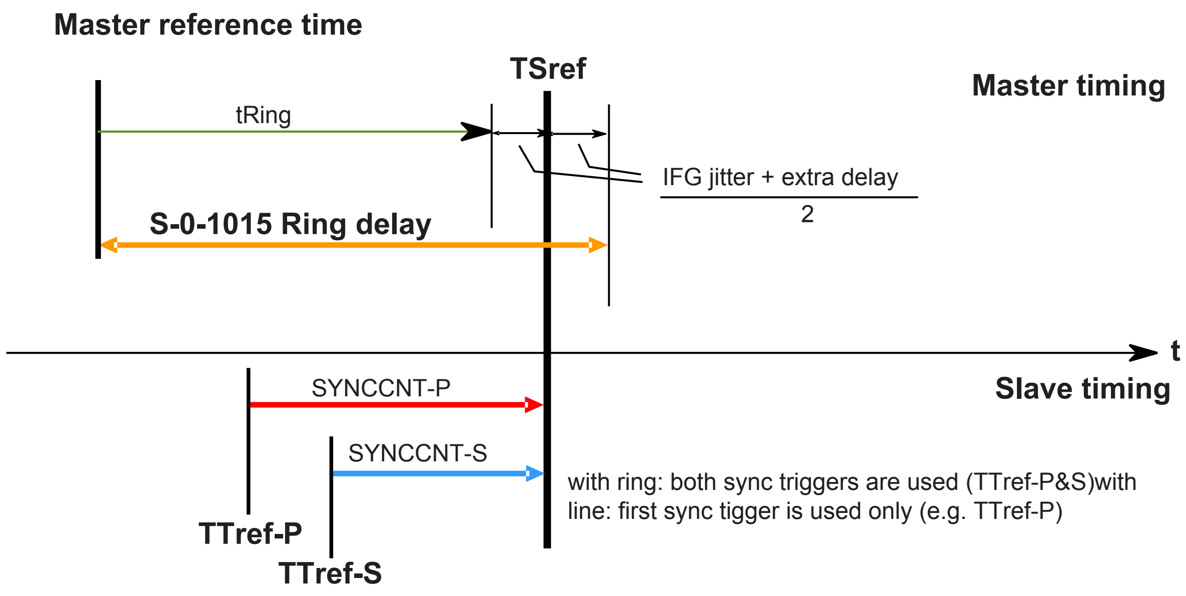

Figure 81 – Timing of TSref with ring and line

7.2.1.2.3 Determination of ring delay with interrupted ring topology

This calculation is only necessary in CP2 if the master shifts up with an interrupted ring from CP0 to CP4 and closes the ring in CP4. In this case the master shall determine two S-0-1015 Ring delay separately. One for the P channel, the other for the S channel. This shall be done to keep the TSref time synchronous of the P channel and S channel.

- The master shall measure at least 64 times the physical delays of the P channel and S channel and determines the average (tRing_P/S) separately.
- The master shall determine the IFG jitter dependent of the participants of the P channel and S channel separately (IFG jitter_P/S).
- The Master shall calculate the extra delay_P/S to keep the TSref time synchronous of the P channel and S channel of an interrupted ring. The following formula is used:

  tRing_P+(IFGjitter_P+extra delay_P)/2 = tRing_S+(IFG jitter_S + extra delay_S)/2
- With this calculation the times of TSref are generated at the same time for the P channel and S channel (see Figure 82).

After the ring recovery in CP3 and CP4:

- The master shall measure at least 64 times the physical delays of the P channel (primary) and / or S channel (secondary) and determines the average (tRing) of all measurements. The measured tRing can be checked using the following formula:

  tRing = (tRing_P + tRing_S)/2
- The master shall determine the IFG jitter dependent of the participants in the topology.
- The master shall add an extra delay taking into account one of the following formulas.
  - (IFG jitter + extra delay)/2 = tRing_P+(IFGjitter_P+extra delay_P)/2- tRing
  - (IFG jitter + extra delay)/2 = tRing_S+(IFGjitter_S+extra delay_S)/2- tRing
- The Master calculates the ring delay for the given ring to keep the TSref time synchronous to the previous P channel and S channel. The time TSref shall be constant during ring recovery.
- The master shall transmit the S-0-1015 Ring delay to all synchronizing slaves. After that, the master shall activate the S-0-1024 SYNC delay measuring procedure command to announce the slave, that it shall synchronize on both ports.

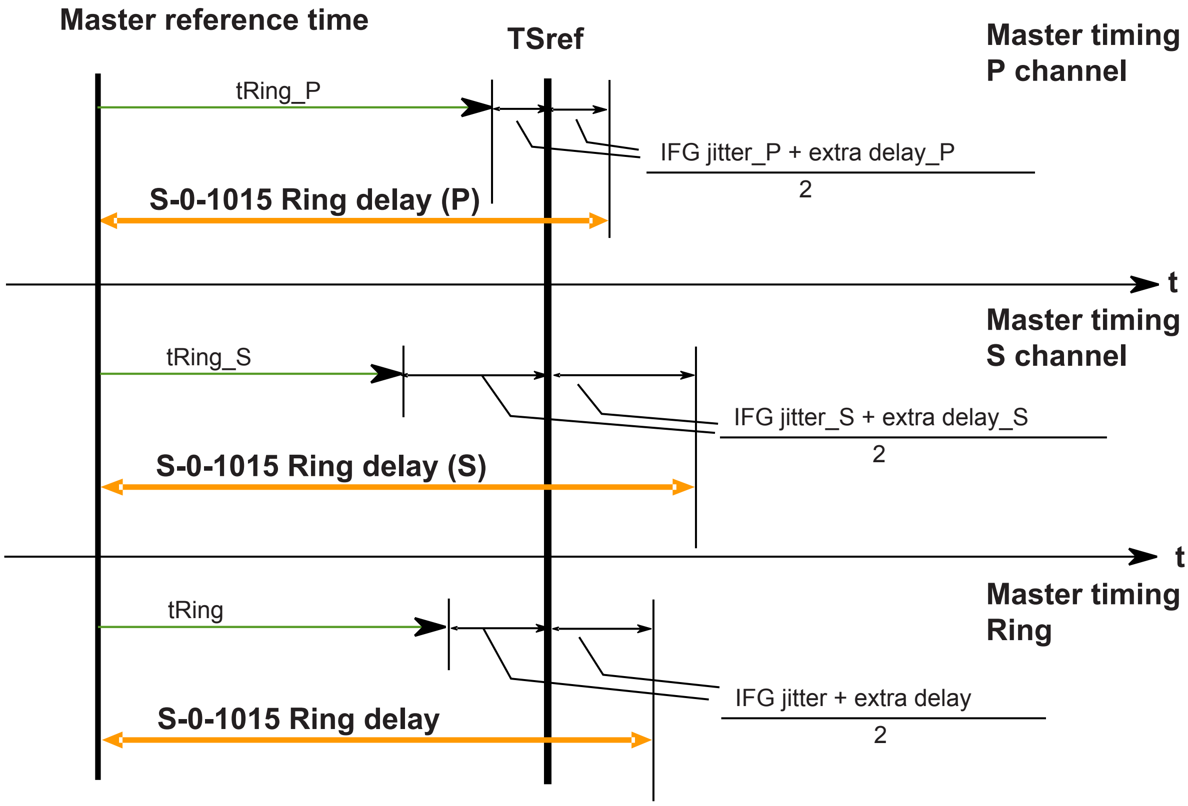

Figure 82 – Timing of TSref with interrupted ring

7.2.1.3 Acquisition of synchronization reference time in the slave

With the activation of the S-0-1024 SYNC delay measuring procedure command the slave starts the ring delay acquisition procedure.

As shown in Figure 83, each slave shall be fitted with the following features:

- S-0-1016 Slave delay (P&S) contains two SYNC counter (SYNCCNT-P/S) one for each port, whereas each of them shall start counting from zero, if a MST is received on the corresponding port.
- An adder, which continuously builds the sum of both SYNC counter values.
- A comparator, which continuously compares the adder state value (sum of both SYNC counter values) with the value of the parameter S-0-1015 Ring delay.
- The sum related comparator signalizes that the sum of both SYNC counter values has reached a value greater or equal to the ring delay value. In this case both SYNC counter are stored in the parameter S-0-1016 Slave delay (P&S).
- The slave acknowledges the S-0-1024 SYNC delay measuring procedure command positively.
- Each slave determines 2 SYNC counter values dependent on the physical order in the topology (SYNCCNT-P + SYNCCNT-S = S-0-1015 Ring delay).
- The master cancels the S-0-1024 SYNC delay measuring procedure command.
- With ring topology:
  - these 2 SYNC counters are used for the synchronization in the slave.
  - the slave shall synchronize on both ports

- With line topology:
  - the SYNCCNT with the lower value is set to 0 by the slave and is not used for the synchronization.
  - the slave shall synchronize on one port only.

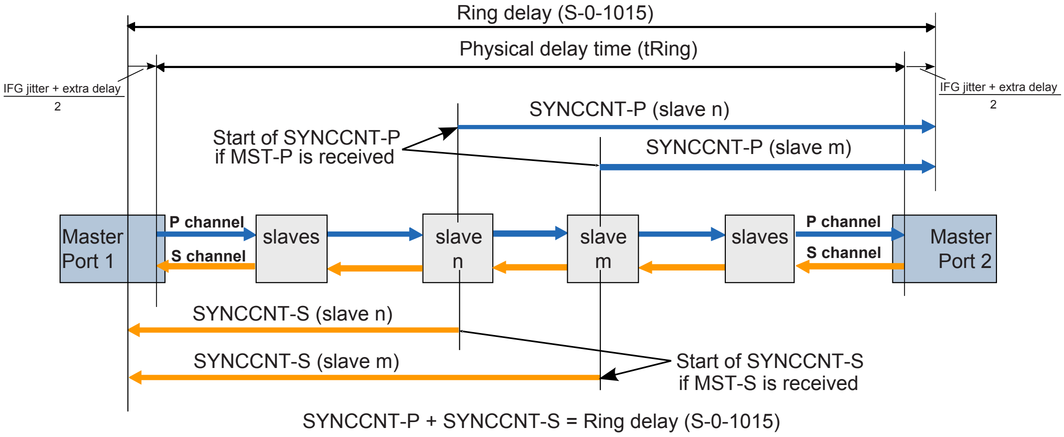

Figure 83 – Determination of the SYNC delay time

7.2.1.4 Synchronization with ring topology

As shown in Figure 76 and in Figure 80, each slave shall generate a synchronization trigger once per communication cycle. The slave shall evaluate only the MST on port 1 and port 2 for synchronization. The SIII header within MDT1 to MDT3 and AT0 to AT3 shall not be used for synchronization.

As shown in Figure 84, each slave shall generate its synchronization reference time (TSref) as soon as any one of the following conditions is met:

- The MST-P (primary channel) has been received at one port, the slave has activated the synchronization delay time (SYNCCNT-P) and this delay has been expired, or
- the MST-S (secondary channel) has been received at the other port, the slave has activated the synchronization delay time (SYNCCNT-S), and this delay has been expired.

The slave shall generate one synchronization reference time (TSref) only within one communication cycle, as soon as the first of these conditions is met. The slave may ignore the second synchronization trigger which occurs later in the communication cycle.

7.2.1.5 Synchronization with line topology

For synchronization purpose the slave shall evaluate the MST on one port (port 1 or port 2) only. The SIII header within MDT1 to MDT3 and AT0 to AT3 shall not be used for synchronization.

The slave shall use the port (port 1 or port 2) on which it receives first the MST (this port is closer to the master) and generates the synchronization trigger. The port with the later received MST shall not be used for the synchronization trigger.

As shown in Figure 84, each slave shall generate its synchronization reference time (TSref) as soon as the earlier MST (P or S) has been received at one port, the slave has activated the corresponding synchronization delay time (SYNCCNT-P or S) and this delay has been expired.

The slave shall generate one synchronization reference time (TSref) only within one communication cycle. The slave shall ignore the second synchronization trigger which occurs later in the communication cycle.

7.2.1.6 Slave behavior by missing synchronization signals

If a slave doesn't receive a MST on port 1 nor on port 2 within one communication cycle, then the slave shall

- generate an internal synchronization trigger during this communication cycle.
- increment the internal MST error counter once per communication cycle.

If a slave generates an internal synchronization trigger because of missing MSTs during several successive communication cycles, then the slave shall behave as specified in S-0-1003 Allowed MST losses in CP3&CP4.

7.2.1.7 Definitions of Synchronization reference time and S-0-1007 Synchronization time (Tsync)

Each slave receives two MSTs in line or ring topology and gets two synchronization triggers. But the slave selects only the first received synchronization trigger to generate the synchronization reference time (TSref) (see Figure 84). If the application needs a better performance of the synchronization timing, then the synchronization reference time (TSref) can be smoothed by a digital phase-locked loop (DPLL).

The synchronization reference time (TSref) is based on communication and is identical in all synchronized slaves in a given network. But a network can have several synchronization groups with different S-0-1007 Synchronization time (Tsync) depending of the application. All synchronized slave within one synchronization group shall have the same S-0-1007 Synchronization time (Tsync).

With S-0-1007 Synchronization time (Tsync) the master can optimally adjust the processing cycles of the application in the control unit and in the slaves to the telegram timing of the communication cycle. Therefore, the processing cycles of the application are decoupled from communication cycle, and the dead time can be minimized in a closed-loop control circuit which contains the communication.

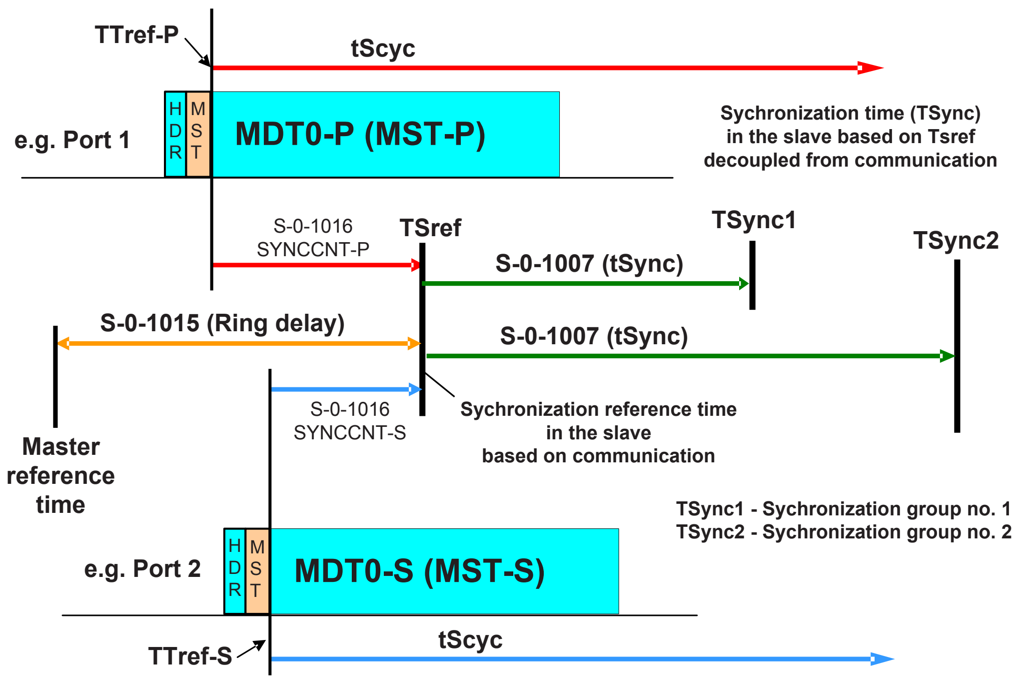

Figure 84 – Definition of TSref

7.2.2 Synchronization of producer cycles

7.2.2.1 General

Type 19 slave or master devices are able to be producer of connections. Each connection has its own S-0-1050.x.10 Producer Cycle Time (tPcyc), which can be the communication cycle time (tScyc S-0-1002) or a multiple of it. If there was no synchronization, each producer would produce application data independently.

Type 19 networks offers a possibility to synchronize these connections. This means all synchronized connections provide application data cyclically at a common time. The base for this time is the Type 19 header of MDT0 (MST). Having received the synchronization MST all connections prepare to capture application data. The connections need to wait for a period of time, called the Feedback acquisition capture point (S-0-1007 Synchronization time (Tsync)) until they capture the real time data. Connections that shall be synchronized need to be configured with the same T4pc value. If several connections synchronize at the same T4pc, this T4pc is also called TSync.

For being able to have different producer cycle times in connections that shall be synchronized, TSync doesn't necessarily appear within each communication cycle. This means that not every MST needs to be a synchronization MST (see Figure 85).

- MaximumScycCnt = (LCM(tPcyc1,tPcyc2,tPcyc3,...) / tScyc) − 1

The period between two TSyncs is called the synchronization cycle time (tSync-cycle). This tSync-cycle needs to be determined in regard to the used producer cycle times. The least common multiple (LCM) of all used producer cycle times are used to define tSync-cycle. The calculation looks as follows:

The tSync-cycle is (Maximum ScycCnt +1) * tScyc.

In order to have a better understanding for TSync and tSync-cycle, two cases with different tPcyc values are considered in 7.2.2.2 and 7.2.2.3:

7.2.2.2 Producer cycle time are integer multiple of communication cycle time

In this example each producer has different producer cycle times. There are two connections with producer cycle times defined as follows:

- tPcyc1 = 2x tScyc
- tPcyc2 = 3x tScyc

The tSync-cycle is calculated as follows:

- tSync − cycle = LCM(tPcyc1,tPcyc2) = 6 * tScyc

The MST is used as synchronization base. In the producer cycle after TSync both producers produce application data synchronously. The second MST received within tSync-cycle is not used for synchronization purposes. Having produced application data synchronously, each producer continues producing application data in frequency of its producer cycle time. The next synchronization takes place after tSync-cycle.

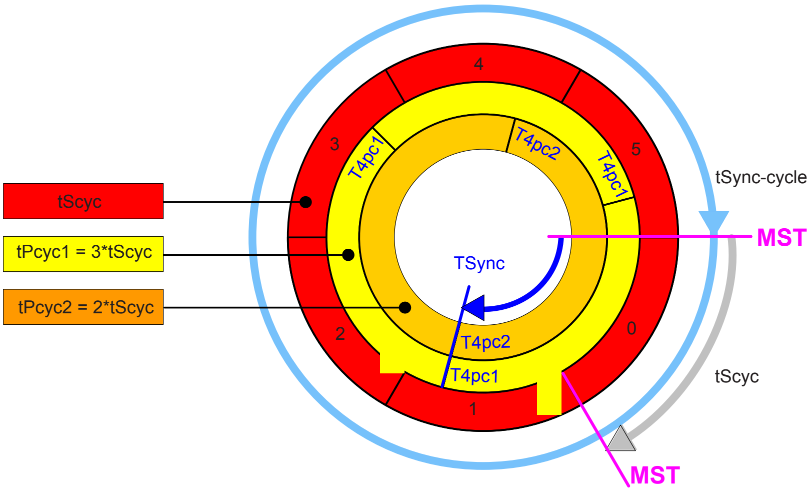

Figure 85 – Timing with different cycle times

7.2.2.3 Producer cycle time is equal to communication cycle time

In this example there are two producers with the same producer cycle time (tScyc = tPcyc1 = tPcyc2). In this case each MST is used for synchronization purposes (see Figure 86). The tSync-cycle is calculated as follows:

- tSync − cycle = LCM(tPcyc1,tPcyc2) = 1 * tScyc

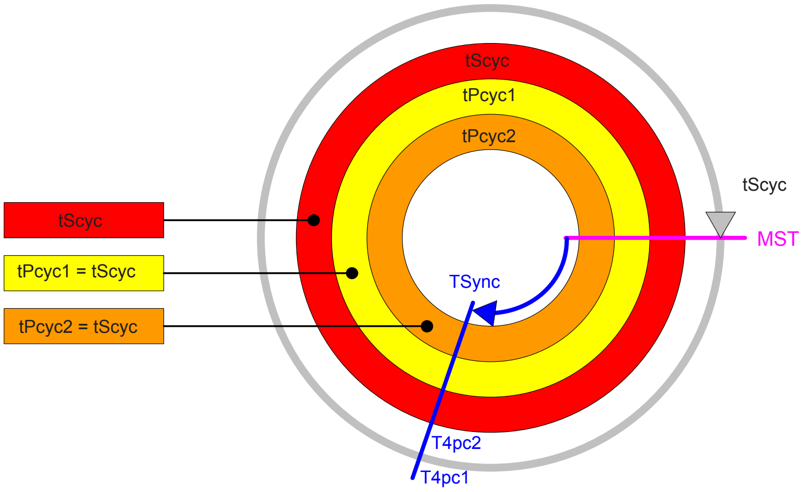

Figure 86 – Timing with the same cycle times

7.3 Processing methods of connection data

7.3.1 General

Three methods of connection processing are specified for Type 19 networks. These are

- Synchronous processing of application data
- Cyclic processing of application data
- Non-synchronous processing of application data

7.3.2 Synchronous processing of application data in the slave

The timing of the synchronous processing is related to the S-0-1007 Synchronization time (Tsync) and at every T4pc defined by the S-0-1050.x.10 Producer Cycle Time. The timing of the telegram processing is related to the end of MST (TTref). The delay of the telegram timing and synchronous timing is defined in S-0-1016 Slave delay (P&S). The timing delay (Time-del) is the difference of SYNCCNT-P and SYNCCNT-S (Time_del = SYNCCNT-P - SYNCCNT-S). The timing of synchronous processing of application data is shown in Figure 87.

- Synchronous processing of application data in the producer: The application data shall be produced in every producer cycle (tPcyc). The processing time of the producer data (tmp-P) are specified in S-0-1060.x.07 Maximum processing time resp. S-0-1005 Minimum feedback processing time (t5). The processing time (tmp-P) shall start at time T4pc resp. TSync.
  Depending on the functionality, the slave shall either use the S-0-1005 Maximum Producer processing Time (t5) for all connections or the S-0-1060.x.07 Maximum processing time for each connection separately.
  – Case 1 of producing application data: If the difference between t1 and T4pc is greater than or equal to the processing time (tmp-P), taking into account the timing delay (Time_del), then the application data are transmitted in the AT in the same communication cycle. tmp-P or t5 ≤ t1 - Time_del - T4pc.
  – Case 2 of producer data: If the processing time (tmp-P) takes longer than the time t1, then the application data are transmitted in the AT of the next communication cycle. tmp-P or t5 > t1 - Time_del - T4pc.
- Synchronous processing of application data in the consumer: The application data shall be consumed in every producer cycle (tPcyc). The processing time of the consuming (tmp-C) are specified in S-0-1060.x.07 Maximum processing time resp. S-0-1047 Maximum Consumer Activation Time (t11). The processing time (tmp-C) should end before the time T4pc resp. TSync. The end of MDT block is determined in the telegram timing calculation.

– Case 1 of consuming application data: If the difference between End of MDT block and T4pc is greater than or equal to the processing time (tmp-C), taking into account the timing delay (Time_del), then the received application data are activated at time T4pc of the same producer cycle. tmp-C or t11 ≤ T4pc + Time_del - End of MDT block.

– Case 2 of consuming application data: If the processing time (tmp-C) takes longer than the time T4pc, then the received application data are activated at time T4pc of the next producer cycle. tmp-C or t11 > T4pc + Time_del - End of MDT block.

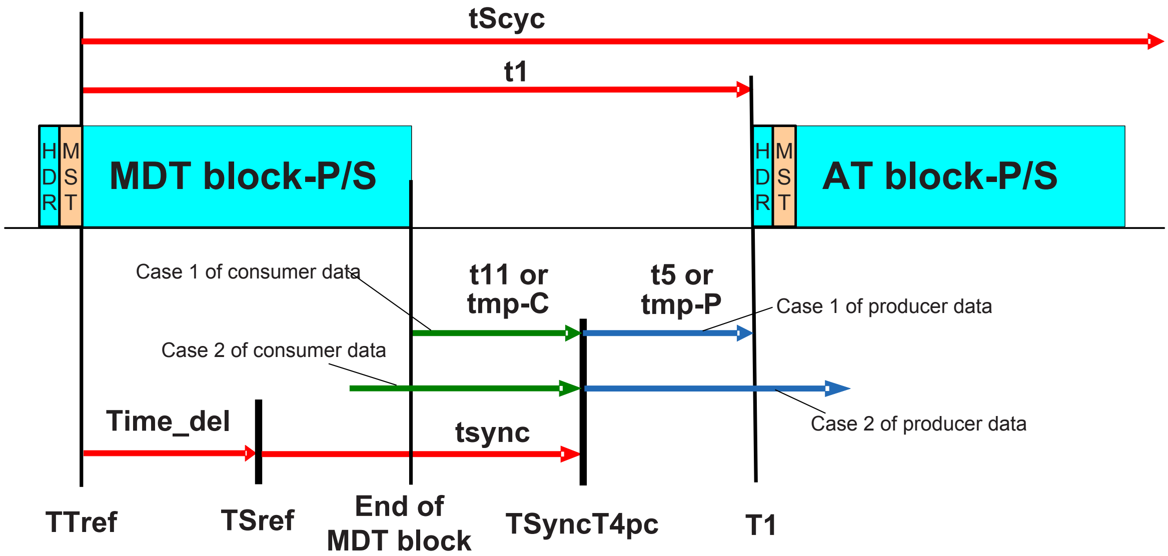

Figure 87 – Synchronous application data processing

7.3.3 Cyclic processing of application data in the slave

The cyclic processing is based to the beginning and ending of the corresponding telegram and shall be activated in every S-0-1050.x.10 Producer Cycle Time. The beginning and end of the Type 19 telegram is determined in the telegram timing calculation. The timing of the telegram processing is related to the end of MST (TTref). The delay between the P telegram and S telegram is defined in S-0-1016 Slave delay (P&S). The timing delay (Time-del) is the difference of SYNCCNT-P and SYNCCNT-S (Time_del = SYNCCNT-P - SYNCCNT-S). The timing of cyclic processing of application data is shown in Figure 88.

- Cyclic processing of application data in the producer: The application data shall be produced in every producer cycle (tPcyc). The processing time of the producing (tmp-P) are specified in S-0-1060.x.07 Maximum processing time resp. S-0-1005 Minimum feedback processing time (t5).
  – Case 1 of producing application data: If the processing time (tmp-P or t5) is finished before the Type 19 telegram is started, then the application data are transmitted in this Type 19 telegram of this communication cycle.
  – Case 2 of producing application data: If the processing time (tmp-P or t5) is not finished before the Type 19 telegram is started, then the application data are transmitted in the Type 19 telegram of the next communication cycle.
- Cyclic processing of application data in the consumer: The application data shall be consumed in every producer cycle (tPcyc). The processing time of the consuming (tmp-C) are specified in S-0-1060.x.07 Maximum processing time resp. S-0-1047 Maximum Consumer Activation Time (t11).
  – Case 1 of consuming application data: If the processing time (tmp-C or t11) starts after the end of the Type 19 telegram, then the received application data in this Type 19 telegram are activated in the same producer cycle.

– Case 2 of consuming application data: If the processing time (tmp-C ot t11) starts before the end of the Type 19 telegram, then the received application data in this Type 19 telegram are activated in the next producer cycle, that means the application data of a previous producer cycle are activated in a later producer cycle.

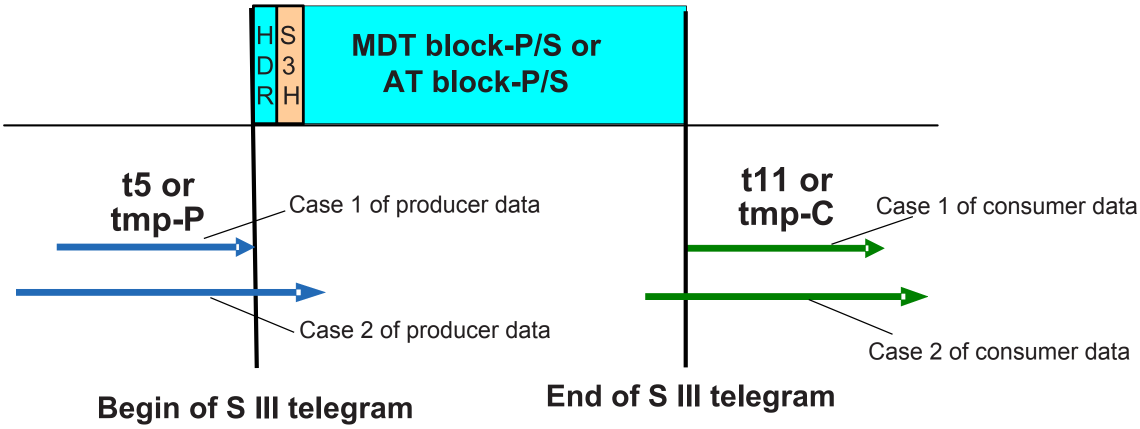

Figure 88 – Cyclic application data processing

7.3.4 Non-synchronous processing of application data in the slave

The non-synchronous processing does not use any timing parameters. The beginning and end of the Type 19 telegram is not defined by parameters. The delay between the P telegram and S telegram is not defined. The diagram of non-synchronous processing of application data is shown in Figure 89.

Non-synchronous processing of application data in the producer: The application data can be produced to any time, that means a producer cycle time is not defined. The producing of the application data is random and the start of the processing time and the processing time itself is not defined also.

Non-synchronous processing of application data in the consumer: The application data can be consumed at any time, that means a producer cycle time is not defined. The consuming of the application data is random and the processing time of the application data is not defined also. It makes sense to start the consuming of the application data only, if the corresponding Type 19 telegram was received.

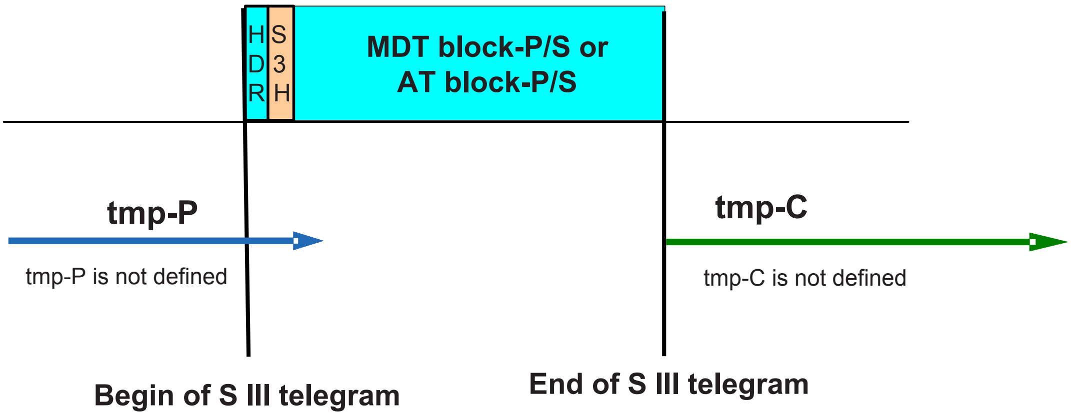

Figure 89 – Non-synchronous application data processing

8 Communication Error handling and monitoring

8.1 Invalid telegrams

The contents of invalid MAC telegrams shall not be passed to sublayers. The occurrence of invalid MAC telegrams may be communicated to network management. Invalid MAC telegrams may be ignored or discarded.

An invalid telegram shall be defined as one that meets at least one of the following conditions:

- Alignment error: It is not an integral number of octets in length.
- FCS error: The bits of the incoming telegram (exclusive of the FCS field itself) do not generate a FCS value identical to the one received.
- CRC error: The bits of the incoming telegram (exclusive of the CRC field itself of Type 19 header) do not generate a CRC value identical to the one received.
- Type 19 telegram length error: The telegram was not received with the expected length.
- Type 19 telegram loss: The telegram was not received in the expected time slot (for example MST not received within MST window).

8.2 Response to MDT and AT telegram failure

If a telegram failure occurs, the master and the slave shall respond as follows:

- the synchronization of the interface shall be maintained;
- several counters (internal) shall be incremented for missing telegrams.

The application profile may specify additional response (for example on the basis of the last correct command values, a power drive system shall calculate internal command values to replace the data of the missing telegram, as specified in FSP_Drive).

The parameter S-0-1003 (Allowed MST losses in CP3&CP4) defines the maximum number of communication cycles in which a slave may not receive its MST in CP3 and CP4. The slave shall switch from CP3 and CP4 back to NRT state if it did not receive its MSTs within more than the user-defined quantity in S-0-1003 and shall set

- the error C1D in the device status and
- the S-0-0390 Diagnostic number to 0xC30F4001.

After half the amount of cycles specified in S-0-1003, the slave shall set

- the communication warning bit in its device status (bit 15) and

- the S-0-0390 Diagnostic number to 0xC30E4001.

8.3 Error counters in the slave

In order to make available the amount of errors recognized in the slaves two IDN are defined and shall be implemented in the slaves:

- S-0-1028 Error counter MST-P&S: This IDN shall be used to show the total amount of MST (MDT0) losses occurred.
- S-0-1035 Error counter Port1 and Port2: This IDN shall be used to show the total amount of telegram losses (Type 19 telegrams and “non-Type 19” telegrams) occurred.

8.3.1 Error effects on communication phases

8.3.1.1 Ascending communication phases

The sequence of communication phases shall be maintained in ascending order (0, 1-4). If this sequence is not maintained, the slave shall return to NRT state.

The slave shall set

- the error C1D in the device status and
- the S-0-0390 Diagnostic number to 0xC30F4004.

8.3.1.2 Descending communication phases

A change of the CPs in descending order shall only be accomplished through CP0. The switching of communication phases from CP0 to CP4 shall be accomplished in accordance with 5.2.

If the master switches from a higher CP to a lower CP other than CP0, the slave shall then immediately return to NRT state and wait for the MDT0 of CP0 from the master.

The slave shall set

- the error C1D in the device status and
- the S-0-0390 Diagnostic number to 0xC30F4005.

8.4 Status codes of Type 19 communication profile (SCP)

This article defines the status codes for the Type 19 communication profile (SCP), which are used for the language-neutral presentation of diagnostic information of Type 19 slave devices. The defined diagnosis classes and status codes of the SCP are listed in Table 111 and Table 112.

Table 111 – SCP specific status codes

Bit 31 – 20 Interpret. & Source (hex)	Bit 19-16 Class (hex)	Bit 15-0 Status code (hex)	Description
C30	A	0000	Communication phase 0
C30	A	0001	Communication phase 1
C30	A	0002	Communication phase 2
C30	A	0003	Communication phase 3
C30	A	0004	this status code shall not be used
C30	A	0005	this status code shall not be used
C30	A	0006	this status code shall not be used
C30	A	0007	this status code shall not be used
C30	A	0008	NRT state
C30	A	0009	this status code shall not be used
C30	A	0030	Hot-plug phase 0 (HP0)
C30	A	0031	Hot-plug phase 1 (HP1)
C30	A	0032	Hot-plug phase 2 (HP2)
C30	C	0100	S-0-0127 CP3 transition check (Transition from CP2 to CP3)
C30	C	0101	Invalid parameters (Data block elements required in CP3 are missing or invalid)
C30	C	0104	Configured IDN for MDT not configurable
C30	C	0105	Maximum length for MDT exceeded
C30	C	0106	Configured IDNs for AT not configurable
C30	C	0107	Maximum length for AT exceeded
C30	C	0108	Timing parameter > Type 19 cycle time (tScyc) - t1, t6, t7, etc.
C30	C	0109	Telegram offset unsuitable (e.g. telegram offset outside of telegram, etc.)
C30	C	0110	this status code shall not be used
C30	C	0111	this status code shall not be used
C30	C	0112	this status code shall not be used
C30	C	0113	this status code shall not be used
C30	C	0114	this status code shall not be used
C30	C	0115	this status code shall not be used
C30	C	0116	this status code shall not be used
C30	C	0139	this status code shall not be used
C30	C	0170	Configured IDNs for connection not configurable
C30	C	0171	Maximum length for connections exceeded
C30	C	0172	S-0-1024 SYNC delay measuring procedure command not performed (The master didn't activate this procedure command)
C30	C	0173	Quantity of connections is not configurable
C30	C	0174	Connection configuration is not possible
C30	C	0175	Producer cycle time (tPcyc) of a connection is wrong
C30	C	0176	SCP classes not correct configured (Configured SCP Classes or its combination in S-0-1000.0.1 List of Active SCP Classes & Version is not supported by the slave)
C30	C	5200	S-0-0128 CP4 transition check (Transition from CP3 to CP4)

Bit 31 – 20 Interpret. & Source (hex)	Bit 19-16 Class (hex)	Bit 15-0 Status code (hex)	Description
C30	C	5300	S-0-1024 SYNC delay measuring procedure command
C30	C	5301	S-0-1024 SYNC delay measuring procedure command failed (S-0-1015 Ring delay = 0)
C30	C	5302	S-0-1024 SYNC delay measuring procedure command error (Measuring is interrupted or disturbed or S-0-1015 Ring delay is too small)
C30	C	0	reserved
C30	E	4001	Warning of MST losses (Amount of MST losses are half of S-0-1003 in CP3 & CP4)
C30	E	4002	RTD-failure shutdown
C30	E	4003	invalid communication phase recognized
C30	E	4004	sequence of CP during phase upshift is not correct
C30	E	4005	sequence of CP during phase downshift is not correct
C30	E	4006	The master changes the current-CP with MST.Phase.CPS = 0,
C30	E	4007	Consumer connection failed
C30	E	4008	Invalid addressing of MDT data container A
C30	E	4009	Invalid addressing of AT data container A
C30	E	4010	reserved
C30	E	4019	CPS=1 and master changes the CP to an invalid value (see 5.2.3.4)
C30	E	4020	Topology status changes from fast-forward (FF) to loop-back with forward (L&F) - removed when changing from L&F to FF (occurs when changing from FF to L&F, removed when changing from L&F to FF)
C30	E	0	reserved
C30	F	4001	Error of MST losses (Amount of MST losses in CP3 & CP4 exceeded. The slave switches to NRT mode. see 5.2.3.4)
C30	F	4002	connection losses (Amount of data losses > S-0-1050.x.11 in CP4. see 4.7)
C30	F	4003	reserved for Type 16
C30	F	4004	reserved for Type 16
C30	F	4005	reserved for Type 16
C30	F	4006	reserved (this status code shall not be used)
C30	F	4017	CPS-MST timeout (500 ms) occurs during phase switch (see 5.2.3.4)
C30	F	4018	OVS producer data are invalid
C30	F	4019	CPS=0 and master changes the CP to an invalid value (see 5.2.3.4)
C30	F	4020	Topology status changes from Loop-back with Forward to NRT mode (occurs if cable is broken on active port)
C30	F	4021	Slave doesn't support the announced Communication Version for CP1 and CP2
C30	F	4022	reserved
C30	F	4023	reserved
C30	F	0	reserved

8.5 Priority of diagnosis classes

Type 19 defines the following diagnosis classes, which are used to categorize diagnostic information according to their context and impact on the error status of the device and its components.

Each diagnosis class has a specific priority, which is used for the prioritization purpose.

In general, the assignment of diagnosis classes to particular diagnostic information is manufacturer-specific.

Table 112 – Overview on diagnosis classes

Diagnosis class	Priority	Description
Operational state	4 (lowest)	Diagnostic information, representing an event or state, which implicates no threat to the device or component, for example “Torque control“ or “compatible replacement of IO module”.
Procedure command specific state	3	Diagnostic information, which is generated during the execution of a procedure command and clearly can be assigned to this command, for example “S-0-0420 Activate parametrization level procedure command (PL) executed” or “reference mark not detected”
Warning (C2D)	2	Diagnostic information, representing an event or state, which implicates a threat to the device or component, for example “motor over-temperature warning” or “under voltage of certain device components”.
Error (C1D)	1 (highest)	Diagnostic information, representing an event or state, which implicates a failure of the system, “loss of motor encoder reference” or “malfunction of the local bus”.
Operational state	4 (lowest)	Diagnostic information, representing an event or state, which implicates no threat to the device or component, for example “Torque control“ or “compatible replacement of IO module”.

But in the following cases, Type 19 dictates the diagnosis classes, which have to be assigned to diagnostic information:

- If diagnostic information is specific to a procedures command, it has to be assigned to the diagnosis class “procedure command specific state”.
- If diagnostic information in general has an impact on the error status of a component, Type 19 assigns the diagnostic information fix to the corresponding diagnosis class by means of the description of the status codes.

Annex A
(normative)

IDN – Identification numbers

A.1 IDN specification

A.1.1 Introduction

All parameters shall be assigned to IDNs.

Every parameter consists of elements. Elements are used to supply additional information, which is required to allow the display and input of data and the use of universal routines by means of the control terminal. This additional information is necessary for handling arbitrary slave-related data. With this information, anonymous parameter can be interpreted by the user interface. The parameter structure shall be as shown in Table A.1. In a parameter, elements 1, 3, and 7 are mandatory and shall always be present. Elements 2, 4, 5, and 6 are optional and may be supported depending on configuration. Elements 5 and 6 are mandatory for cycle time parameters only. The appropriate elements of the parameters shall be selected via the service channel control bits.

Scope of parameter (global / local)

This aspect of the scope of a parameter is only relevant if a device consists of multiple sub-devices. There are two different scopes of parameters, global and local.

- Global parameters are unique within one device. Changes on a global parameter in one sub-device, causes changes in all other sub-devices in the device. For example, the IP-address is unique in one device. Global parameter shall exist in every sub-device.
- Local parameters are unique in one sub-device. Local parameter may exist only in one sub-device.

Table A.1 – Data block structure

Element No.	Description	Requirement
1	IDN	mandatory
2	Name	optional
3	Attribute	mandatory
4	Unit	optional
5	Minimum value	optional
6	Maximum value	optional
7	Operation data	mandatory
NOTE Elements 5 and 6 are mandatory for cycle time parameters (S-0-1050.x.10, S-0-1002).		

A.1.2 Element 1: structure of IDN

If written and read via the service channels, the appropriate data shall be addressed by means of the IDNs. Beyond that, operation data within the configurable part of the data records of the AT and MDT shall be as defined by means of the IDNs.

IDN numbering shall have a range of 2^{32}, which shall be subdivided as follow:

- Two ranges shall be available for standard IDNs and product-specific IDNs. Product-specific IDNs are out of the scope of standardization.
- Every range shall be subdivided into eight parameter sets.
- Each set shall thus have up to 4 095 data block numbers or function groups.
- Each IDN may have up to 256 structure instances and up to 256 structure elements.

IDNs shall be transferred in telegrams as 32-bit binary numbers.

Table A.2 describes the structure of IDNs.

Table A.2 – Parameter structure

Bit Number	Value	Description
31-24	—	Structure instance (SI)
	0-255	Number of structure instance (SI)
23-16	—	Structure element (SE)
	0-127 (bit 23 = 0)	Standard SE (bit 15 = 0) determined by Type 19 (bit 15 = 0)
	128-255 (bit 23 = 1)	Product specific SE determined by manufacturer (bit 15 = 0)
15	—	Standard or product specific data (S or P)
	0	Standard IDN (S-0-nnnn), SE (0-127), SI and data block number determined by Type 19
	1	Product-specific IDN (P-0-nnnn), Bits 31 to 0 determined by manufacturer
14-12	—	Parameter sets
	0-7	Parameter set 0 – 7 (Type 19 specifies IDNs with parameter set 0 only.)
11-0	—	Data block or Function group
	0-4095	Data block number (if SI = SE = 0); Function group (if SI or SE is not 0)

The notation of an IDN shall be as follows:

- If no SI or SE exists, it shall be: S/P - Parameter Set - Data block number (for example S-0-1002)
- If SI and SE exist, it shall be: S/P - Parameter Set - Function group . SI . SE (for example S-0-1300.0.2)

A.1.3 Element 2: structure of name

The name shall have two length specifications of two octets each and a character string of maximum 60 UTF8 characters (up to 240 octets). Octets 1 and 2 of the name shall specify the current text length in octets. Octets 3 and 4 of the name shall indicate the maximum text length in octets in a slave.

Figure A.1 shows the IDN name structure.

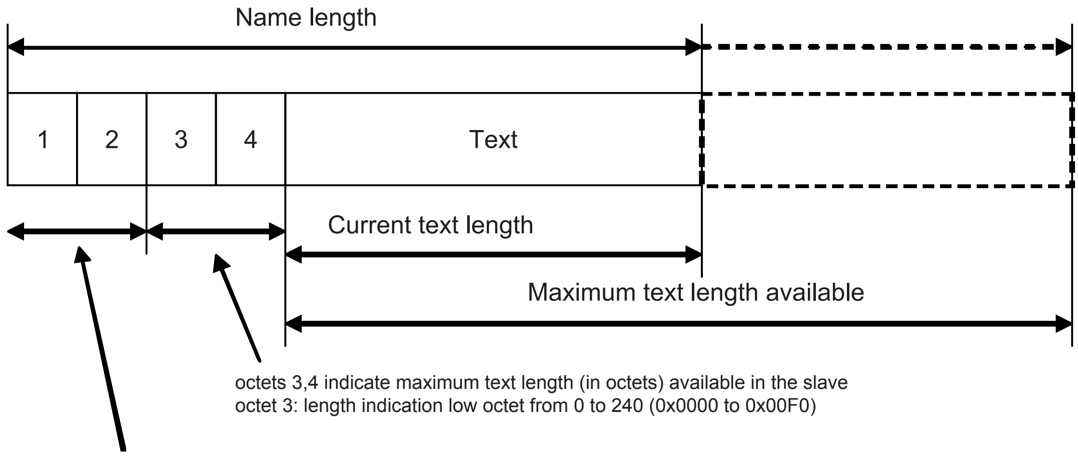

Figure A.1 – IDN name structure

If the current text has a length of 0, only the two length indications shall be transmitted. Octets 1 and 2 shall then contain the value 0.

Reading: In order to complete a read command in the service channel, the master shall require octets 1 and 2. Octets 3 and 4 shall only be read by the master to prevent writing text which is too long.

Writing: When writing a name, the master shall set octets 1 and 2 according to the current text length. The text shall not be longer than specified in octets 3 and 4. During writing the slave shall ignore octets 3 and 4 and insert its available length during reading.

A.1.4 Element 3: structure of attribute

Every parameter shall have an attribute which allows for an intelligible representation of various operation data by means of universal routines. The attribute shall contain all information which is needed to display operation data intelligibly. The attribute makes it possible to convert the transferred operation data into intelligible display data and vice versa. The conversion shall have no impact on the data itself. If data needs to be scaled, specific scaling parameters shall be supplied. Every scaling modification needs a change in the attributes of the affected data. It is recommended to write protect the attribute. See Table A.3.

Table A.3 – Element 3 of IDNs

Bit no,	Value	Description
31	—	(reserved)
30	—	Write protected in CP4
	0	Operation data is writeable
	1	Operation data is write protected
29	—	Write protected in CP3
	0	Operation data is writeable
	1	Operation data is write protected
28	—	Write protected in CP2
	0	Operation data is writeable
	1	Operation data is write protected
27-24	—	Decimal point: This is an additional display information.

Bit no,	Value	Description
		Places after the decimal point indicate the position of the decimal point for the display and input of appropriate operation data. Decimal point is used for displaying of signed and unsigned decimal. For all other display formats the decimal point shall be = 0
	0000 to 1111	No place to 15 places after decimal point (maximum)
23	—	(reserved)
22-20	—	Data type and display format. Data type and display format are used to convert the operation data and the minimum and maximum input value to the correct display format
	000	Data type: Binary number Display format: Binary
	001	Data type: Unsigned integer Display format: Unsigned decimal
	010	Data type: Integer Display format: Signed decimal
	011	Data type: Unsigned integer Display format: Hexadecimal
	100	Data type: Extended character set Display format: Text (ASCII)
	101	Data type: Unsigned integer Display format: IDN
	110	Data type: Floating-point number Display format: Signed decimal with exponent (float) Single or double precision, according to ANSI/IEEE 752-1995
	111	Data type: Type 19 time Display format: according to IEC 61588 4 octets seconds & 4 octets nano seconds, starts with 1.1.1970 computed in UTC
19	—	Parameter is:
	0	Not a procedure command
	1	a procedure command
18-16	—	Data length: Data length is required so that the master is able to complete service channel data transfers correctly
	000	(reserved)
	001	Operation data is two octets long
	010	Operation data is four octets long
	011	Operation data is eight octets long
	100	Variable length with one-octet data strings
	101	Variable length with two-octet data strings
	110	Variable length with four-octet data strings
	111	Variable length with eight-octet data strings
15-0	—	Conversion factor: the conversion factor is an unsigned integer used to convert numeric data to display format. The conversion factor shall be set to a value of 1 when it is not needed for data display (for example: for binary number, character string or floating-point number etc.)

The display formats and data length shall have any of the valid combinations (“yes” marked) in Table A.4.

Table A.4 – Valid combinations of the display formats

Data length	Binary	Unsigned decimal	Signed decimal	Hex	Text	IDN	Float	Time
2 octet	yes	yes	yes	yes				
4 octet	yes	yes	yes	yes		Yes	yes	
8 octet	yes	yes	yes	yes			yes	yes
1 octet list	yes	yes	yes	yes	yes			
2 octet list	yes	yes	yes	yes				
4 octet list	yes	yes	yes	yes		yes	yes	
8 octet list	yes	yes	yes	yes			yes	yes

A.1.5 Element 4: structure of unit

The unit shall have two length specifications of two octets each and a character string of maximum 12 UTF8 characters (up to 48 octets). Octets 1 and 2 of the unit shall specify the current text length in octets. Octets 3 and 4 of the unit shall indicate the maximum text length in octets in a slave. A parameter shall not have any unit if the data type is either a binary number or a character string. see Figure A.2.

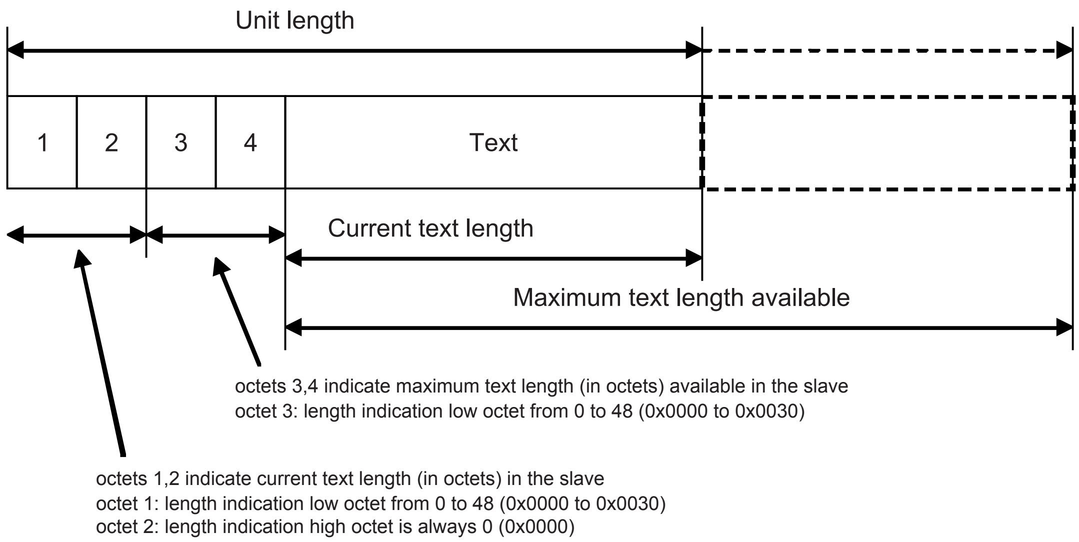

Figure A.2 – Unit structure

If the current text has the length 0, only the two length indications shall be transmitted. Octets 1 and 2 shall then contain the value 0.

Reading: In order to complete a read command in the service channel, octets 1 and 2 shall be required by the master. Octets 3 and 4 shall only be read by the master to prevent writing text which is too long.

Writing: When writing a unit, the master shall set octets 1 and 2 according to the current text length. The text shall not be longer than specified in octets 3 and 4. During writing the slave shall ignore octets 3 and 4 and insert its available length during reading.

A.1.6 Element 5: structure of minimum value

The minimum input value shall be the smallest numerical value for the operation data which the slave is able to process and have the same length as operation data.

If the slave ignores the write request of the operation data, only in this case the slave shall return the corresponding error code and set the SVC error bit in the SVC status.

The following data types do not have minimum input value:

- binary number
- character string
- IDN.

The minimum input value shall be displayed like the operation data.

It is recommended to write protect the minimum input value.

A.1.7 Element 6: structure of maximum value

The maximum input value shall be the largest numerical value for the operation data which the slave is able to process and has the same length as operation data.

If the slave ignores the write request of the operation data, only in this case the slave shall return the corresponding error code and set the SVC error bit in the SVC status.

If the operation data is a binary number, then the supported bits are set in the maximum input value. The master therefore recognizes which bits are supported by the slave in this parameter.

The following data types do not have maximum input value:

- character string and
- IDN.

The maximum input value shall be displayed like the operation data.

It is recommended to write protect the maximum input value.

A.1.8 Element 7: structure of operation data

When the operation data of an IDN is written, which is a bit field, the slave shall mask the written operation data by using the mask of supported bits (which the slave may show by the maximum value of the IDN) and then process the written data.

The operation data shall have any one of following lengths:

- fixed length with two octets;
- fixed length with four octets;
- fixed length with eight octets;
- variable length with list elements of 1, 2, 4 and 8 octets up to 65 532 octets.

Variable length

Length specifications for the variable length only shall be coded in the initial four octets for hexadecimal digits. Structure of operation data with variable length shall be as shown in Figure A.3.

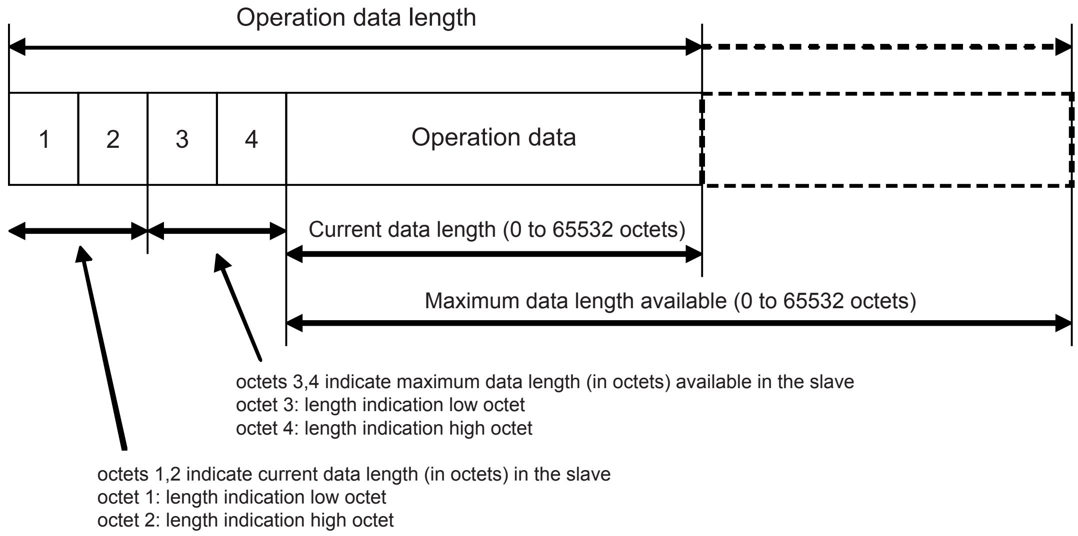

Figure A.3 – Structure of IDN operation data with variable length

Operation data with variable length shall consist of length indicators in the initial four octets, followed by the programmed operation data.

Files or tables shall be loaded from the control unit to the slaves or vice versa by means of the transfer of operation data with variable length (for example the IDN-list of all operation data in a slave).

If the operation data has the length 0, only the two length indications shall be transmitted. Octets 1 and 2 shall then contain the value 0.

Reading: In order to complete a read command in the service channel correctly, the master shall require octets 1 and 2. Octets 3 and 4 shall only be read by the master to prevent writing operation data which is too long.

Writing: When writing operation data, the master shall set octets 1 and 2 according to the current data length. The data shall not be longer than specified in octets 3 and 4. During writing, the slave shall ignore octets 3 and 4 and insert its available length during reading.

Sorting of the IDNs in IDN lists:

IDNs shall be sorted in the IDN lists in one of the following ways:

- Unsorted IDN lists (for example configuration lists)
- Sorted by Function group or data block number (FG)
  - Function group or data block number
  - Structure instance,
  - Structure element.
- Sorted by Structure instance (SI)
  - Structure instance,
  - Function group or data block number
  - Structure element.

Table A.5 shows the structure of an IDN list and the sorting of the IDNs as an example.

Table A.5 – Example of the structure of an IDN-list

<table>
<tr><th>1. 2.</th><th>3. 4.</th><th colspan="7">← Length of list →</th></tr>
<tr><td>1C 00</td><td>1C 00</td><td colspan="7"></td></tr>
<tr><td></td><td></td><td>S-0-1000</td><td>S-0-1300.
0.2</td><td>S-0-1300.
0.4</td><td>S-0-1502.
0.1</td><td>S-0-1502.
0.2</td><td>S-0-1502.
1.1</td><td>S-0-1502.
1.2</td></tr>
<tr><td></td><td></td><td>List element #0</td><td>List element #1</td><td>List element #2</td><td>List element #3</td><td>List element #4</td><td>List element #5</td><td>List element #6</td></tr>
<tr><td></td><td colspan="8">Octets 3 and 4 indicates the maximum length for operation data available in the slave; example length = 28 octets (0x001C)</td></tr>
<tr><td colspan="9">Octets 1 and 2 indicates the current length for programmed operation data; example length = 28 octets (0x001C)</td></tr>
</table>

A.1.9 Structure of Data status

The content of "data status" shall be related to the entire parameter. "Data status" shall contain conditions which change dynamically. When opening the service channel via an IDN, the current data status shall be transferred automatically to the master. This enables the control unit to respond to procedure command acknowledgments during transmission of a procedure command. The data status (procedure command acknowledgment) shall be reset by the device during every renewed initialization.

Bits 3-0 shall only be present for procedure commands (procedure command acknowledgment).

Changes in the procedure command acknowledgment by:

- bit 3-0: procedure command executed correctly (0111-->0011 = positive acknowledgment), or
- bit 3-0: procedure command execution is impossible (0111-->1111 = negative acknowledgment)

shall lead to setting the procedure command change bit in the device status.

Bit 8 shall be set by the device if the parameter is recognized as invalid, such as if the data memory is checked for data loss and a checksum error is set.

The structure of the data status is shown in Table A.6.

Table A.6 – Data status structure

Bits	Value	Meaning
15-9	—	(reserved)
8	—	Validity of operation data
	0	Operation data is valid
	1	Operation data is invalid
7-4	—	(reserved)
3-0	—	Procedure command acknowledgment
	0000	Procedure command not activated
	0001	Procedure command is set and the execution is interrupted
	0011	Procedure command executed correctly (positive acknowledgment), procedure command change bit in device status is set to 1
	0111	Procedure command not yet executed (processing)
	1111	Error, procedure command execution is impossible (negative acknowledgment), procedure command change bit in device status is set to 1
		All other codings are reserved

A.2 Identification numbers in numerical orders

Table A.7 lists the IDNs which are related to communication, and that devices of this type shall support. Their detailed description appears in Clause A.3.

Application-specific data content is specified in other relevant standards, for example IEC 61800-7-20x.

Table A.7 – List of relevant communication-related IDNs

IDN number	IDN name
S-0-0014	Interface status
S-0-0021	IDN-list of invalid operation data for CP2
S-0-0022	IDN-list of invalid operation data for CP3
S-0-0026	IDN allocation of producer RTB word container
S-0-0027	IDN allocation of consumer RTB word container
S-0-0127	CP3 transition check
S-0-0128	CP4 transition check
S-0-0144	Producer RTB word container
S-0-0145	Consumer RTB word container
S-0-0187	IDN-list of configurable data as producer
S-0-0188	IDN-list of configurable data as consumer
S-0-0328	Bit allocation of producer RTB word container
S-0-0329	Bit allocation of consumer RTB word container
S-0-0360	MDT data container A1
S-0-0361	MDT data container B1
S-0-0362	MDT data container A list index
S-0-0363	MDT data container B list index
S-0-0364	AT data container A1

IDN number	IDN name
S-0-0365	AT data container B1
S-0-0366	AT data container A list index
S-0-0367	AT data container B list index
S-0-0368	Data container A pointer
S-0-0369	Data container B pointer
S-0-0370	MDT data container A&B configuration list
S-0-0371	AT data container A&B configuration list
S-0-0394	List IDN
S-0-0396	Number of list elements
S-0-0397	List segment
S-0-0398	IDN list of configurable real-time/status bits
S-0-0399	IDN list of configurable real-time/control bits
S-0-0444	IDN-list of configurable data in the AT data container
S-0-0445	IDN-list of configurable data in the MDT data container
S-0-0450	MDT data container A2
S-0-0451	MDT data container A3
S-0-0452	MDT data container A4
S-0-0453	MDT data container A5
S-0-0454	MDT data container A6
S-0-0455	MDT data container A7
S-0-0456	MDT data container A8
S-0-0457	MDT data container A9
S-0-0458	MDT data container A10
S-0-0459	MDT data container B2
S-0-0480	AT data container A2
S-0-0481	AT data container A3
S-0-0482	AT data container A4
S-0-0483	AT data container A5
S-0-0484	AT data container A6
S-0-0485	AT data container A7
S-0-0486	AT data container A8
S-0-0487	AT data container A9
S-0-0488	AT data container A10
S-0-0489	AT data container B2
S-0-0490	MDT data container A2 configuration list
S-0-0491	MDT data container A3 configuration list
S-0-0492	MDT data container A4 configuration list
S-0-0493	MDT data container A5 configuration list
S-0-0494	MDT data container A6 configuration list
S-0-0495	MDT data container A7 configuration list
S-0-0496	MDT data container A8 configuration list
S-0-0497	MDT data container A9 configuration list
S-0-0498	MDT data container A10 configuration list

IDN number	IDN name
S-0-0500	AT data container A2 configuration list
S-0-0501	AT data container A3 configuration list
S-0-0502	AT data container A4 configuration list
S-0-0503	AT data container A5 configuration list
S-0-0504	AT data container A6 configuration list
S-0-0505	AT data container A7 configuration list
S-0-0506	AT data container A8 configuration list
S-0-0507	AT data container A9 configuration list
S-0-0508	AT data container A10 configuration list
S-0-1000.0.1	Active SCP Classes
S-0-1000	SCP Type & Version
S-0-1002	Communication cycle time (tScyc)
S-0-1003	Allowed MST losses in CP3/CP4
S-0-1005	Minimum feedback processing time (t5)
S-0-1006	AT transmission starting time (t1)
S-0-1007	Synchronization time (Tsync)
S-0-1008	Command value valid time (t3)
S-0-1009	Device Control (C-DEV) offset in MDT
S-0-1010	Lengths of MDTs
S-0-1011	Device Status (S-DEV) offset in AT
S-0-1012	Lengths of ATs
S-0-1013	SVC offset in MDT
S-0-1014	SVC offset in AT
S-0-1015	Ring delay
S-0-1016	Slave delay (P&S)
S-0-1017	UC channel transmission time
S-0-1019	MAC address
S-0-1020.0.1	Current IP address
S-0-1020	IP address
S-0-1021.0.1	Current subnet mask
S-0-1021	Subnet mask
S-0-1022.0.1	Current gateway address
S-0-1022	Gateway address
S-0-1023	SYNC jitter
S-0-1024	SYNC delay measuring procedure command
S-0-1026	Version of communication hardware
S-0-1027.0.1	Requested MTU
S-0-1027.0.2	Effective MTU
S-0-1028	Error counter MST-P&S
S-0-1031	Test pin assignment Port 1 & Port 2
S-0-1035	Error counter Port 1 & Port 2
S-0-1040	Sub-device address
S-0-1041	AT Command value valid time (t9)

IDN number	IDN name
S-0-1044	Device control (C-DEV)
S-0-1045	Device status (S-DEV)
S-0-1046	List of sub-device addresses in device
S-0-1047	Maximum consumer activation time (t11)
S-0-1048	Activate network settings
S-0-1050.x.1	Connection setup
S-0-1050.x.2	Connection Number
S-0-1050.x.3	Telegram assignment
S-0-1050.x.4	Max. Length of Connection
S-0-1050.x.5	Current length of Connection
S-0-1050.x.6	Configuration List
S-0-1050.x.7	Assigned connection capability
S-0-1050.x.8	Connection control (C-CON)
S-0-1050.x.10	Producer cycle time
S-0-1050.x.11	Allowed Data Losses
S-0-1050.x.12	Error Counter Data Losses
S-0-1050.x.20	IDN Allocation of real-time bit
S-0-1050.x.21	Bit allocation of real-time bit
S-0-1051	Image of connection setups
S-0-1060.x.01	Default configuration
S-0-1060.x.02	Configuration mask
S-0-1060.x.03	Maximum quantity of this connection capability
S-0-1060.x.04	Max. connection length of connection capability
S-0-1060.x.06	Configurable IDNs of connection capability
S-0-1060.x.07	Maximum processing time
S-0-1060.x.10	Minimum producer cycle time
S-0-1061	Maximum TSref-Counter
S-0-1080.x.02	Producer RTB list container
S-0-1080.x.03	IDN allocation of producer RTB list container
S-0-1080.x.04	Bit allocation of producer RTB list container
S-0-1081.x.02	Consumer RTB list container
S-0-1081.x.03	IDN allocation of consumer RTB list container
S-0-1081.x.04	Bit allocation of consumer RTB list container
S-0-1099.0.01	Test-IDN Control for SCP conformity purpose
S-0-1099.0.02	Test-IDN Container for SCP conformity purpose
S-0-1100.0.01	Diagnostic counter sent SMP fragments
S-0-1100.0.02	Diagnostic counter received SMP fragments
S-0-1100.0.03	Diagnostic counter discarded SMP fragments
S-0-1101.x.01	SMP Container Data
S-0-1101.x.02	List of session identifiers
S-0-1101.x.03	List of session priorities
S-0-1150.x.01	OVS Control (C-OVS)
S-0-1150.x.02	OVS Status (S-OVS)

IDN number	IDN name
S-0-1150.x.03	OVS Container
S-0-1150.x.04	Sample time
S-0-1150.x.05	Phase shift
S-0-1150.x.06	Configuration List OVS - IDNs
S-0-1150.x.07	Configuration List OVS - Offset
S-0-1150.x.08	Configuration List OVS - Length
S-0-1150.x.09	Assigned Oversampling Capability
S-0-1150.x.10	Number of samples
S-0-1151.x.01	Maximum number of samples
S-0-1151.x.02	Internal resolution
S-0-1151.x.03	Maximum quantity of this oversampling capability
S-0-1151.x.04	Minimum sample time
S-0-1151.x.06	Configurable IDNs of OVS capability
S-0-1151.x.07	Configurable IDNs of OVS capability - Offset
S-0-1151.x.08	Configurable IDNs of OVS capability - Length
S-0-1152	Amount of OVS Domains
NOTE All other IDN numbers are reserved.	

A.3 Detailed specification of communication-related IDNs

A.3.1 IDN S-0-0014 Interface status

A.3.1.1 Attributes

Table A.8 shows the possible attributes for this IDN.

Table A.8 – Attributes for IDN S-0-0014

Attribute	Value
Name	Interface status
Version	—
Length	2
Display Format	binary
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.1.2 Description

If an error is set in the interface status, then the error C1D is set in device status (S-0-1045 Device Status (S-DEV)).

The setting of bits 2–0 does not signify an error. If there are no communication errors present, the current communication phase is contained in the interface status. If an error has occurred, the error and the current CP will be stored simultaneously.

The error bits of the interface status are reset to '0' by the sub-device only if no errors of interface status exists and after the procedure command S-0-0099 Reset class 1 diagnostic has been received by the sub-device via the service channel.

Table A.9 shows the structure of interface status.

Table A.9 – Structure of interface status

Bit No	Value	Meaning
15	0 = no error 1 = error	Manufacturer-specific error
14		Topology and Communication
13		Phase switching with invalid conditions
12		CPS-MST timeout during phase switching
11		(reserved)
10		(reserved)
9		(reserved)
8		(reserved)
7		(reserved)
6		reserved
5		invalid phase (phase > 4)
4		(reserved)
3		MST failure (S-0-1003 Allowed MST losses in CP3 & CP4)
2-0	000 … 100	CP0 … CP4
	101 … 110	(reserved)
	111	NRT Mode

A.3.2 IDN S-0-0021 IDN-list of invalid operation data for CP2

A.3.2.1 Attributes

Table A.10 shows the possible attributes for this IDN.

Table A.10 – Attributes for IDN S-0-0021

Attribute	Value
Name	IDN-list of invalid operation data for CP2
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1

Attribute	Value
Unit	—

A.3.2.2 Description

IDNs which are considered invalid by the slave when performing the CP3 transition check are stored in this IDN-list.

- Case 1: procedure command S-0-0127 is performed correctly; the IDN-list (S-0-0021) contains no IDNs.
- Case 2: procedure command S-0-0127 results in an error; the IDN-list (S-0-0021) contains all IDNs of invalid operation data.

The presence of this parameter is only necessary if this slave have the ability for errors during phase switching (S-0-0127).IDN S-0-0022 IDN-list of invalid operation data for CP3.

A.3.2.3 Attributes

Table A.11 shows the possible attributes for this IDN.

Table A.11 – Attributes for IDN S-0-0022

Attribute	Value
Name	IDN-list of invalid operation data for CP3
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.2.4 Description

IDNs which are considered invalid by the slave when performing the CP4 transition check are stored in this IDN-list.

- Case 1: procedure command S-0-0128 is performed correctly; the IDN-list (S-0-0022) contains no IDNs.
- Case 2: procedure command S-0-0128 results in an error; the IDN-list (S-0-0022) contains all IDNs of invalid operation data.

The presence of this parameter is only necessary if this slave have the ability for errors during phase switching (S-0-0128).

A.3.3 IDN S-0-0026 IDN allocation of producer RTB word container

A.3.3.1 Attributes

Table A.12 shows the possible attributes for this IDN.

Table A.12 – Attributes for IDN S-0-0026

Attribute	Value
Name	IDN allocation of producer RTB word container
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.3.2 Description

Bits in the producer RTB word container (S-0-0144) are definable by means of the configuration list of the producer RTB word container represented in this IDN. The sequence of the IDNs in the configuration list determines the bit numbering scheme in the producer RTB word container. The first IDN of the configuration list defines bit 0, the last IDN defines bit 15 of the producer RTB word container. If IDN S-0-0328 is not supported by the slave, bit 0 of all the configured IDNs are used, else S-0-0328 defines for each IDN which bit is used. Maximum 16 IDNs can be taken into this list, therefore this list shall have a fixed length of 64 octets.

A.3.4 IDN S-0-0027 IDN allocation of consumer RTB word container

A.3.4.1 Attributes

Table A.13 shows the possible attributes for this IDN.

Table A.13 – Attributes for IDN S-0-0027

Attribute	Value
Name	IDN allocation of consumer RTB word container
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.4.2 Description

Bits in the consumer RTB word container (S-0-0145) are definable by means of the configuration list of the consumer RTB word container represented in this IDN. The sequence of the IDNs in the configuration list determines the bit numbering scheme in the consumer

RTB word container. The first IDN of the configuration list defines bit 0, the last IDN defines bit 15 of the consumer RTB word container. If IDN S-0-0329 is not supported by the slave, bit 0 of all the configured IDNs are used, else S-0-0329 defines for each IDN which bit is used. Maximum 16 IDNs can be taken into this list, therefore this list shall have a fixed length 64 octets.

A.3.5 IDN S-0-0127 CP3 transition check

A.3.5.1 Attributes

Table A.14 shows the possible attributes for this IDN.

Table A.14 – Attributes for IDN S-0-0127

Attribute	Value
Name	CP3 transition check
Version	—
Length	2
Display Format	Binary
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.5.2 Description

The master uses this procedure command to instruct the slave to check that all necessary parameters have been transferred for CP3. Otherwise, this procedure command results in an error (see S-0-0021). After the procedure command is performed correctly, the master has to cancel the procedure command. The master can then activate CP3 in the MST.

A.3.6 IDN S-0-0128 CP4 transition check

A.3.6.1 Attributes

Table A.15 shows the possible attributes for this IDN.

Table A.15 – Attributes for IDN S-0-0128

Attribute	Value
Name	CP4 transition check
Version	—
Length	2
Display Format	binary
Min input value	—
Max input value	—

Attribute	Value
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.6.2 Description

The master uses this procedure command to instruct the slave to check that all necessary parameters have been transferred for CP4. Otherwise, this procedure command results in an error (see S-0-0022). After the procedure command is performed correctly, the master has to cancel the procedure command. The master can then activate CP4 in the MST.

A.3.7 IDN S-0-0144 Producer RTB word container

A.3.7.1 Attributes

Table A.16 shows the possible attributes for this IDN.

Table A.16 – Attributes for IDN S-0-0144

Attribute	Value
Name	Producer RTB word container
Version	—
Length	2
Display Format	Binary
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.7.2 Description

Signals can be transmitted in real-time from the producer to the consumer by means of the producer RTB word container. For this purpose, the producer RTB word container needs to be integrated in one or more producer connections.

Bits in the producer RTB word container are defined by means of the configuration lists of the producer RTB word container (see S-0-0026 and S-0-0328).

A.3.8 IDN S-0-0145 Consumer RTB word container

A.3.8.1 Attributes

Table A.17 shows the possible attributes for this IDN.

Table A.17 – Attributes for IDN S-0-0027

Attribute	Value
Name	Consumer RTB word container
Version	—
Length	2
Display Format	Binary
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.8.2 Description

Signals can be transmitted in real-time from the producer to the consumer by means of the consumer RTB word container. For this purpose, the consumer RTB word container needs to be integrated in one consumer connection.

Bits in the consumer RTB word container are definable by means of the configuration lists of the consumer RTB word container (see S-0-0027 and S-0-0329).

A.3.9 IDN S-0-0187 IDN-list of configurable data as producer

A.3.9.1 Attributes

Table A.18 shows the possible attributes for this IDN.

Table A.18 – Attributes for IDN S-0-0187

Attribute	Value
Name	IDN-list of configurable data as producer
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.9.2 Description

This IDN list contains all IDNs of operation data of the producer (for example: feedback values, inputs) which can be processed by the sub-device cyclically in the AT.

If S-0-1060 exists you will find more detailed information of the connection related parameter.

A.3.10 IDN S-0-0188 IDN-list of configurable data as consumer

A.3.10.1 Attributes

Table A.19 shows the possible attributes for this IDN.

Table A.19 – Attributes for IDN S-0-0188

Attribute	Value
Name	IDN-list of configurable data as consumer
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.10.2 Description

This IDN list contains all IDNs of operation data (command values) which can be processed by the sub-device cyclically.

If S-0-1060 exists you will find more detailed information of the connection related parameter.

A.3.11 IDN S-0-0328 Bit allocation of producer RTB word container

A.3.11.1 Attributes

Table A.20 shows the possible attributes for this IDN.

Table A.20 – Attributes for IDN S-0-0328

Attribute	Value
Name	Bit allocation of producer RTB word container
Version	—
Length	2, variable
Display Format	Unsigned decimal (bit number)
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.11.2 Description

In this configuration list the bit numbers of the operation data are programmed, which are copied into the producer RTB word container (S-0-0144). The sequence of the bit numbers in the configuration list sets the numerical order in the producer RTB word container. The first bit number in the configuration list sets bit 0, the last bit number sets bit 15 into the producer RTB word container. Maximum one 16 bit number can be taken into this list, therefore this list shall have a fixed length of 32 octets (see also S-0-0026).

A.3.12 IDN S-0-0329 Bit allocation of consumer RTB word container

A.3.12.1 Attributes

Table A.21 shows the possible attributes for this IDN.

Table A.21 – Attributes for IDN S-0-0329

Attribute	Value
Name	Bit allocation of consumer RTB word container
Version	—
Length	2, variable
Display Format	Unsigned decimal (bit number)
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.12.2 Description

In this configuration list the bit numbers of the operation data are programmed, which are contained in the consumer RTB word container (S-0-0145). The sequence of the bit numbers in the configuration list sets the numerical order in the consumer RTB word container. The first bit number in the configuration list sets bit 0, the last bit number sets bit 15 in the consumer RTB word container. Maximum 16 bit number can be taken into this list, therefore this list shall have a fixed length of 32 octets (see also S-0-0027).

A.3.13 IDN S-0-0360 MDT data container A1

A.3.13.1 Attributes

Table A.22 shows the possible attributes for this IDN.

Table A.22 – Attributes for IDN S-0-0360

Attribute	Value
Name	MDT data container A1
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—

Attribute	Value
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.13.2 Description

For the standard data container function in the MDT,

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0360 and S-0-0457). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0361 and S-0-0459) as well as one configuration list (S-0-0370) for all MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):

The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The MDT data container A1 (S-0-0360) and A9 (S-0-0457) are used also in the extended data container function.

A.3.14 IDN S-0-0361 MDT data container B1

A.3.14.1 Attributes

Table A.23 shows the possible attributes for this IDN.

Table A.23 – Attributes for IDN S-0-0361

Attribute	Value
Name	MDT data container B1
Version	—
Length	4
Display Format	Hexadecimal

Attribute	Value
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.14.2 Description

For the standard data container function in the MDT,

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0360 and S-0-0457). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0361 and S-0-0459) as well as one configuration list (S-0-0370) for all MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The MDT data container A1 (S-0-0360) and A9 (S-0-0457) are used also in the extended data container function.

A.3.15 IDN S-0-0362 MDT data container A list index

A.3.15.1 Attributes

Table A.24 shows the possible attributes for this IDN.

Table A.24 – Attributes for IDN S-0-0362

Attribute	Value
Name	MDT data container A list index
Version	—
Length	2
Display Format	Hexadecimal

Attribute	Value
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.15.2 Description

If in the MDT data container an IDN with a variable length (list parameter) is configured,

- the corresponding list element of this list parameter will be addressed via the list index,
- the master writes the addressed list element into the MDT data container.

The list index of the MDT data container consists of a 16 bit address. Via list index 65535 the MDT data container can be defined not valid by the master.

The list index of MDT data container can be configured in the cyclic data of the MDT. Thereby, a switching of the list elements in the MDT data container during the next communication cycle is possible.

If the list index is situated outside of the list parameter, the data in the corresponding MDT data containers will be ignored by the slave. In this case the slave shall set the list index (acknowledgment) on value 65535 and optionally the pointer of MDT data container on value 255.

The list index of MDT data container (see Table A.25) can also be configured into the cyclic data of the ATs. In this way an acknowledgment of the MDT data container is possible. The slave reads the list index of MDT data container from the MDT and acknowledges it in the AT.

Table A.25 – List index of MDT data container A

Bit number	Value	Description
15-0	MDT data container A list index structure:	
	0-65534	index of the configured list parameter
	65535	MDT data container A not valid (error)

A.3.16 IDN S-0-0363 MDT data container B list index

A.3.16.1 Attributes

Table A.26 shows the possible attributes for this IDN.

Table A.26 – Attributes for IDN S-0-0363

Attribute	Value
Name	MDT data container B list index
Version	—
Length	2
Display Format	Hexadecimal

Attribute	Value
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.16.2 Description

If in the MDT data container an IDN with a variable length (list parameter) is configured,

- the corresponding list element of this list parameter will be addressed via the list index,
- the master writes the addressed list element into the MDT data container.

The list index of the MDT data container consists of a 16 bit address. Via list index 65535 the MDT data container can be defined not valid by the master.

The list index of MDT data container can be configured in the cyclic data of the MDT. Thereby, a switching of the list elements in the MDT data container during the next communication cycle is possible.

If the list index is situated outside of the list parameter, the data in the corresponding MDT data containers will be ignored by the slave. In this case the slave shall set the list index (acknowledgment) on value 65535 and optionally the pointer of MDT data container on value 255.

The list index of MDT data container (see Table A.27) can also be configured into the cyclic data of the ATs. In this way an acknowledgment of the MDT data container is possible. The slave reads the list index of MDT data container from the MDT and acknowledges it in the AT.

Table A.27 – List index of MDT data container B

Bit number	Value	Description
15-0	MDT data container B list index structure:	
	0-65534	index of the configured list parameter
	65535	MDT data container B not valid (error)

A.3.17 IDN S-0-0364 AT data container A1

A.3.17.1 Attributes

Table A.28 shows the possible attributes for this IDN.

Table A.28 – Attributes for IDN S-0-0364

Attribute	Value
Name	AT data container A1
Version	—
Length	4
Display Format	Hexadecimal

Attribute	Value
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.17.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.18 IDN S-0-0365 AT data container B1

A.3.18.1 Attributes

Table A.29 shows the possible attributes for this IDN.

Table A.29 – Attributes for IDN S-0-0365

Attribute	Value
Name	AT data container B1
Version	—
Length	4
Display Format	Hexadecimal

Attribute	Value
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.18.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format’ are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data’s data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.19 IDN S-0-0366 AT data container A list index

A.3.19.1 Attributes

Table A.30 shows the possible attributes for this IDN.

Table A.30 – Attributes for IDN S-0-0366

Attribute	Value
Name	AT data container A list index
Version	—
Length	2
Display Format	Hexadecimal

Attribute	Value
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.19.2 Description

If in the AT data container an IDN with a variable length (list parameter) is configured,

- the corresponding list element of this list parameter will be addressed via the list index,
- the slave writes the addressed list element into the AT data container.

The list index of the AT data container consists of a 16 bit address. Via list index 65535 the AT data container can be defined not valid by the slave.

The list index of AT data container can be configured in the cyclic data of the MDT. Thereby, a switching of the list elements in the AT data container during the next communication cycle is possible.

If the list index is situated outside of the list parameter, the data in the corresponding AT data containers will be ignored by the master. In this case the slave shall set the list index (acknowledgment) on value 65535 and optionally the pointer of AT data container on value 255.

The list index of AT data container (Table A.31) can also be configured into the cyclic data of the ATs. In this way an acknowledgment of the AT data container is possible. The slave reads the list index of AT data container from the MDT and acknowledges it in the AT.

Table A.31 – List index of AT data container A

Bit number	Value	Description
15-0	AT data container A list index structure:	
	0-65534	index of the configured list parameter
	65535	AT data container A not valid (error)

A.3.20 IDN S-0-0367 AT data container B list index

A.3.20.1 Attributes

Table A.32 shows the possible attributes for this IDN.

Table A.32 – Attributes for IDN S-0-0367

Attribute	Value
Name	AT data container B list index
Version	—
Length	2
Display Format	Hexadecimal

Attribute	Value
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.20.2 Description

If in the AT data container an IDN with a variable length (list parameter) is configured,

- the corresponding list element of this list parameter will be addressed via the list index.
- The slave writes the addressed list element into the AT data container.

The list index of the AT data container consists of a 16 bit address. Via list index 65535 the AT data container can be defined not valid by the slave.

The list index of AT data container can be configured in the cyclic data of the MDT. Thereby, a switching of the list elements in the AT data container during the next communication cycle is possible.

If the list index is situated outside of the list parameter, the data in the corresponding AT data containers will be ignored by the master. In this case the slave shall set the list index (acknowledgment) on value 65535 and optionally the pointer of AT data container on value 255.

The list index of AT data container (Table A.33) can also be configured into the cyclic data of the ATs. In this way an acknowledgment of the AT data container is possible. The slave reads the list index of AT data container from the MDT and acknowledges it in the AT.

Table A.33 – List index of AT data container B

Bit number	Value	Description
15-0	AT data container B list index structure:	
	0-65534	index of the configured list parameter
	65535	AT data container B not valid (error)

A.3.21 IDN S-0-0368 Data container A pointer

A.3.21.1 Attributes

Table A.34 shows the possible attributes for this IDN.

Table A.34 – Attributes for IDN S-0-0368

Attribute	Value
Name	Data container A pointer
Version	—
Length	2
Display Format	hexadecimal

Attribute	Value
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.21.2 Description

The data container pointers contain an 8-bit pointer, that defines which operation data should be placed in the MDT and AT data container. The data container pointer is the offset, within the corresponding data container configuration list

- for standard data container S-0-0370 and S-0-0371,
- additionally for extended data container S-0-0490 to S-0-0498 and S-0-0500 to S-0-0508.

Herewith inside the configuration list an IDN is addressed for the MDT data container or AT data container.

The master writes the addressed operation data into the MDT data container. The slave writes the addressed operation data into the AT data container.

The IDN "Data container A pointer and B pointer" (S-0-0368, S-0-0369) can be configured in the cyclic data of the MDT. Thereby, a switching of the operation data in the data containers during the next communication cycle is possible.

The IDN "Data container A pointer and B pointer" (S-0-0368, S-0-0369) can also be configured in the cyclic data of the AT. In this case the addressing (acknowledgment) according to the contents of the data container will be transmitted. The slave generates the acknowledgment by copying the pointer of MDT to the pointer of AT.

If the pointer of the data container (Table A.35) is situated outside of the configuration list for the MDT or AT data container or the data is longer than the data container, the contents of the data container are not valid. The slave sets the pointer (acknowledgment) in the AT on 255. The data in MDT or AT data container will be ignored by the slave or master.

Table A.35 – Data container A pointer structure

Bit number	Value	Description
15-8	—	Address for AT data container
	0-254	address for AT data container A
	255	AT data container A not valid (error)
7-0	—	Address for MDT data container
	0-254	address for MDT data container A
	255	MDT data container not valid (error)

A.3.22 IDN S-0-0369 Data container B pointer

A.3.22.1 Attributes

Table A.36 shows the possible attributes for this IDN.

Table A.36 – Attributes for IDN S-0-0369

Attribute	Value
Name	Data container B pointer
Version	—
Length	2
Display Format	hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.22.2 Description

The data container pointers contain an 8-bit pointer, that defines which operation data should be placed in the MDT and AT data container. The data container pointer is the offset, within the corresponding data container configuration list

- for standard data container S-0-0370 and S-0-0371,
- additionally for extended data container S-0-0490 to S-0-0498 and S-0-0500 to S-0-0508.

Herewith inside the configuration list an IDN is addressed for the MDT data container or AT data container.

The master writes the addressed operation data into the MDT data container. The slave writes the addressed operation data into the AT data container.

The IDN "Data container A pointer and B pointer" (S-0-0368, S-0-0369) can be configured in the cyclic data of the MDT. Thereby, a switching of the operation data in the data containers during the next communication cycle is possible.

The IDN "Data container A pointer and B pointer" (S-0-0368, S-0-0369) can also be configured in the cyclic data of the AT. In this case the addressing (acknowledgment) according to the contents of the data container will be transmitted. The slave generates the acknowledgment by copying the pointer of MDT to the pointer of AT.

If the pointer of the data container (Table A.37) is situated outside of the configuration list for the MDT or AT data container or the data is longer than the data container, the contents of the data container are not valid. The slave sets the pointer (acknowledgment) in the AT on 255. The data in MDT or AT data container will be ignored by the slave or master.

Table A.37 – Data container B pointer structure

Bit number	Value	Description
15-8	—	Address for AT data container
	0-254	address for AT data container B
	255	AT data container A not valid (error)
7-0	—	Address for MDT data container
	0-254	address for MDT data container B

Bit number	Value	Description
	255	MDT data container not valid (error)

A.3.23 IDN S-0-0370 MDT data container A/B configuration list

A.3.23.1 Attributes

Table A.38 shows the possible attributes for this IDN.

Table A.38 – Attributes for IDN S-0-0370

Attribute	Value
Name	MDT data container A/B configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.23.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.24 IDN S-0-0371 AT data container A/B configuration list

A.3.24.1 Attributes

Table A.39 shows the possible attributes for this IDN.

Table A.39 – Attributes for IDN S-0-0371

Attribute	Value
Name	AT data container A/B configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—

Attribute	Value
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.24.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.25 IDN S-0-0394 List IDN

A.3.25.1 Attributes

Table A.40 shows the possible attributes for this IDN.

Table A.40 – Attributes for IDN S-0-0394

Attribute	Value
Name	List IDN
Version	—
Length	4
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.25.2 Description

The ident number of a parameter with variable length is stored in this parameter. This parameter shall be interpreted only if the list segment (S-0-0397) is accessed.

A.3.26 IDN S-0-0395 List index

A.3.26.1 Attributes

Table A.41 shows the possible attributes for this IDN.

Table A.41 – Attributes for IDN S-0-0395

Attribute	Value
Name	List index
Version	—
Length	2
Display Format	unsigned decimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.26.2 Description

The list index specifies the starting address inside the list. With list index = 0, the first list element is accessed after the lengths indication. The list index is always programmed according to the list elements.

- List index = 0 -> 1. List element (1, 2, 4 or 8 octets long)
- List index = 1 -> 2. List element (1, 2, 4 or 8 octets long) etc.

The list index shall be in the range between the first list element and the last list element + 1. It is not possible to place a gap in a list but it is possible to extend the list at the end up to its maximum length.

A.3.27 IDN S-0-0396 Number of list elements

A.3.27.1 Attributes

Table A.42 shows the possible attributes for this IDN.

Table A.42 – Attributes for IDN S-0-0396

Attribute	Value
Name	Number of list elements
Version	—
Length	2
Display Format	unsigned decimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.27.2 Description

This parameter stores the number of list elements which shall be transmitted. The number counts list elements, not octets. The calculation of the length in octets shall be done by the device itself. At a read access the slave generates the current length in octets of the list segment. For write access this parameter shall be ignored. The length information for write access is stored in the current length of the list segment.

A.3.28 IDN S-0-0397 List segment

A.3.28.1 Attributes

Table A.43 shows the possible attributes for this IDN.

Table A.43 – Attributes for IDN S-0-0397

Attribute	Value
Name	List segment
Version	—
Length	like configured data, variable
Display Format	like configured data
Min input value	like configured data
Max input value	like configured data
Positions after decimal point	like configured data
Write protection	Not defined
Conversion factor	1
Scaling/resolution	like configured data
Unit	

A.3.28.2 Description

With the access of this IDN the slave shall evaluate and save the contents of the IDNs S-0-0394, S-0-0395 and if necessary S-0-0396. Modifications at these 3 parameters get effective only with an element change in the service channel.

- Read access
  - The slave provides a list depending on the contents of list index and number of list elements. The current length is calculated by the slave (number of list elements * element width). The maximum length of the list segment shall be set to the current length of the list IDN (S-0-0394).
  - Access to any element excepting element 7 shall provide a copy of the corresponding element of the parameter stored in S-0-0394.
- Write access
  - IDN S-0-0396 shall be ignored on write accesses.
  - The current length of the list segment shall be used in the slave.
  - Only Element 7 shall be writeable.
  - A write access overwrites existing list elements. It shall not be possible to paste a list segments in the destination list. It is possible to extend a list with a list segment at the end of the list up to its maximum length.
  - The list segment immediately gets effective. It behaves so as if the complete list had been written.

A.3.29 IDN S-0-0398 IDN list of configurable real-time bits as producer

A.3.29.1 Attributes

Table A.44 shows the possible attributes for this IDN.

Table A.44 – Attributes for IDN S-0-0398

Attribute	Value
Name	IDN list of configurable real-time bits as producer
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.29.2 Description

This IDN list contains ident numbers, whose bits are configurable as real-time bits in a producer connection.

A.3.30 IDN S-0-0399 IDN list of configurable real-time bits as consumer

A.3.30.1 Attributes

Table A.45 shows the possible attributes for this IDN.

Table A.45 – Attributes for IDN S-0-0399

Attribute	Value
Name	IDN list of configurable real-time bits as consumer
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.30.2 Description

This IDN list contains ident numbers, whose bits are configurable as real-time bits in a consumer connection.

A.3.31 IDN S-0-0444 IDN-list of configurable data in the AT data container

A.3.31.1 Attributes

Table A.46 shows the possible attributes for this IDN.

Table A.46 – Attributes for IDN S-0-0444

Attribute	Value
Name	IDN-list of configurable data in the AT data container
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.31.2 Description

This list contains the IDNs of operation data which can be processed by the sub-device in the AT data containers. This IDN list is used in the standard and extended data container.

A.3.32 IDN S-0-0445 IDN-list of configurable data in the MDT data container

A.3.32.1 Attributes

Table A.47 shows the possible attributes for this IDN.

Table A.47 – Attributes for IDN S-0-0445

Attribute	Value
Name	IDN-list of configurable data in the MDT data container
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.32.2 Description

This list contains the IDNs of operation data which can be processed by the sub-device in the MDT data containers. This IDN list is used in the standard and extended data container.

A.3.33 IDN S-0-0450 MDT data container A2

A.3.33.1 Attributes

Table A.48 shows the possible attributes for this IDN.

Table A.48 – Attributes for IDN S-0-0450

Attribute	Value
Name	MDT data container A2
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.33.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.34 IDN S-0-0451 MDT data container A3

A.3.34.1 Attributes

Table A.49 shows the possible attributes for this IDN.

Table A.49 – Attributes for IDN S-0-0451

Attribute	Value
Name	MDT data container A3
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.34.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format’ are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data’s data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.35 IDN S-0-0452 MDT data container A4

A.3.35.1 Attributes

Table A.50 shows the possible attributes for this IDN.

Table A.50 – Attributes for IDN S-0-0452

Attribute	Value
Name	MDT data container A4
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.35.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - – The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format’ are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - – The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data’s data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.36 IDN S-0-0453 MDT data container A5

A.3.36.1 Attributes

Table A.51 shows the possible attributes for this IDN.

Table A.51 – Attributes for IDN S-0-0453

Attribute	Value
Name	MDT data container A5
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.36.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.37 IDN S-0-0454 MDT data container A6

A.3.37.1 Attributes

Table A.52 shows the possible attributes for this IDN.

Table A.52 – Attributes for IDN S-0-0454

Attribute	Value
Name	MDT data container A6
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.37.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  – The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  – The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.38 IDN S-0-0455 MDT data container A7

A.3.38.1 Attributes

Table A.53 shows the possible attributes for this IDN.

Table A.53 – Attributes for IDN S-0-0455

Attribute	Value
Name	MDT data container A7
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.38.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.39 IDN S-0-0456 MDT data container A8

A.3.39.1 Attributes

Table A.54 shows the possible attributes for this IDN.

Table A.54 – Attributes for IDN S-0-0456

Attribute	Value
Name	MDT data container A8
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.39.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format’ are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data’s data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.40 IDN S-0-0457 MDT data container A9

A.3.40.1 Attributes

Table A.55 shows the possible attributes for this IDN.

Table A.55 – Attributes for IDN S-0-0457

Attribute	Value
Name	MDT data container A9
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.40.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  – The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  – The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.41 IDN S-0-0458 MDT data container A10

A.3.41.1 Attributes

Table A.56 shows the possible attributes for this IDN.

Table A.56 – Attributes for IDN S-0-0458

Attribute	Value
Name	MDT data container A10
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.41.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  – The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  – The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.42 IDN S-0-0459 MDT data container B2

A.3.42.1 Attributes

Table A.57 shows the possible attributes for this IDN.

Table A.57 – Attributes for IDN S-0-0459

Attribute	Value
Name	MDT data container B2
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.42.2 Description

For the extended data container function in the MDT,

- eight data containers (4 octets long, A1 to A8) and
- two data containers (8 octets long, A9 and A10)

are defined, serving as placeholders in the MDT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 to A10 (S-0-0360, S-0-0450 to S-0-0458), as well as a configuration list (S-0-0370, S-0-0490 to S-0-0498) for each of the MDT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the MDT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

A.3.43 IDN S-0-0480 AT data container A2

A.3.43.1 Attributes

Table A.58 shows the possible attributes for this IDN.

Table A.58 – Attributes for IDN S-0-0480

Attribute	Value
Name	AT data container A2
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.43.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.44 IDN S-0-0481 AT data container A3

A.3.44.1 Attributes

Table A.59 shows the possible attributes for this IDN.

Table A.59 – Attributes for IDN S-0-0481

Attribute	Value
Name	AT data container A3
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.44.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.45 IDN S-0-0482 AT data container A4

A.3.45.1 Attributes

Table A.60 shows the possible attributes for this IDN.

Table A.60 – Attributes for IDN S-0-0482

Attribute	Value
Name	AT data container A4
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.45.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  – The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  – The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.46 IDN S-0-0483 AT data container A5

A.3.46.1 Attributes

Table A.61 shows the possible attributes for this IDN.

Table A.61 – Attributes for IDN S-0-0483

Attribute	Value
Name	AT data container A5
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.46.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.47 IDN S-0-0484 AT data container A6

A.3.47.1 Attributes

Table A.62 shows the possible attributes for this IDN.

Table A.62 – Attributes for IDN S-0-0484

Attribute	Value
Name	AT data container A6
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.47.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.48 IDN S-0-0485 AT data container A7

A.3.48.1 Attributes

Table A.63 shows the possible attributes for this IDN.

Table A.63 – Attributes for IDN S-0-0485

Attribute	Value
Name	AT data container A7
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.48.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.49 IDN S-0-0486 AT data container A8

A.3.49.1 Attributes

Table A.64 shows the possible attributes for this IDN.

Table A.64 – Attributes for IDN S-0-0486

Attribute	Value
Name	AT data container A8
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.49.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.50 IDN S-0-0487 AT data container A9

A.3.50.1 Attributes

Table A.65 shows the possible attributes for this IDN.

Table A.65 – Attributes for IDN S-0-0487

Attribute	Value
Name	AT data container A9
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.50.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.51 IDN S-0-0488 AT data container A10

A.3.51.1 Attributes

Table A.66 shows the possible attributes for this IDN.

Table A.66 – Attributes for IDN S-0-0488

Attribute	Value
Name	AT data container A10
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.51.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.52 IDN S-0-0489 AT data container B

A.3.52.1 Attributes

Table A.67 shows the possible attributes for this IDN.

Table A.67 – Attributes for IDN S-0-0489

Attribute	Value
Name	AT data container B
Version	—
Length	4
Display Format	Hexadecimal
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.52.2 Description

For the standard data container function in the AT

- two data containers (4 octets long, A1 and B1) and
- two data containers (8 octets long, A9 and B2)

are defined, serving as placeholders in the AT. The contents of the data containers can be dynamically changed by the master as necessary. Additionally, the data container A pointer (S-0-0368) is required for the containers A1 and A9 (S-0-0364 and S-0-0487). The data container B pointer (S-0-0369) is required for the containers B1 and B2 (S-0-0365 and S-0-0489) as well as one configuration list (S-0-0371) for all AT container. If the configured operation data is only 2 or 4 octets long, it is placed in the lower part of the AT data container. The higher part is not used.

In configuring data container operation data, the slave can select between a minimum requirement and maximum requirement.

- Minimum required data block (access via service channel):
  - The configured operation data is represented in the data container in hexadecimal, without the units.
    - Attribute: Data type and display format' are set hexadecimal (bits 22-20 = 011)
    - Units: Not present
- Maximum required data block (access via service channel):
  - The configured operation data is represented in the data container not with the data block of the data container itself, rather the configured operation data's data block. In this case the operation data will be displayed with the IDN of the data container exactly the same as it would be with its own IDN.

NOTE The AT data container A1 (S-0-0364) and A9 (S-0-0487) are used also in the extended data container function.

A.3.53 IDN S-0-0490 MDT data container A2 configuration list

A.3.53.1 Attributes

Table A.68 shows the possible attributes for this IDN.

Table A.68 – Attributes for IDN S-0-0490

Attribute	Value
Name	MDT data container A2 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.53.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.54 IDN S-0-0491 MDT data container A3 configuration list

A.3.54.1 Attributes

Table A.69 shows the possible attributes for this IDN.

Table A.69 – Attributes for IDN S-0-0491

Attribute	Value
Name	MDT data container A3 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.54.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.55 IDN S-0-0492 MDT data container A4 configuration list

A.3.55.1 Attributes

Table A.70 shows the possible attributes for this IDN.

Table A.70 – Attributes for IDN S-0-0492

Attribute	Value
Name	MDT data container A4 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.55.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.56 IDN S-0-0493 MDT data container A5 configuration list

A.3.56.1 Attributes

Table A.71 shows the possible attributes for this IDN.

Table A.71 – Attributes for IDN S-0-0493

Attribute	Value
Name	MDT data container A5 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—

Attribute	Value
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.56.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.57 IDN S-0-0494 MDT data container A6 configuration list

A.3.57.1 Attributes

Table A.72 shows the possible attributes for this IDN.

Table A.72 – Attributes for IDN S-0-0494

Attribute	Value
Name	MDT data container A6 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.57.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.58 IDN S-0-0495 MDT data container A7 configuration list

A.3.58.1 Attributes

Table A.73 shows the possible attributes for this IDN.

Table A.73 – Attributes for IDN S-0-0495

Attribute	Value
Name	MDT data container A7 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.58.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.59 IDN S-0-0496 MDT data container A8 configuration list

A.3.59.1 Attributes

Table A.74 shows the possible attributes for this IDN.

Table A.74 – Attributes for IDN S-0-0496

Attribute	Value
Name	MDT data container A8 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1

Attribute	Value
Unit	—

A.3.59.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.60 IDN S-0-0497 MDT data container A9 configuration list

A.3.60.1 Attributes

Table A.75 shows the possible attributes for this IDN.

Table A.75 – Attributes for IDN S-0-0497

Attribute	Value
Name	MDT data container A9 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.60.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.61 IDN S-0-0498 MDT data container A10 configuration list

A.3.61.1 Attributes

Table A.76 shows the possible attributes for this IDN.

Table A.76 – Attributes for IDN S-0-0498

Attribute	Value
Name	MDT data container A10 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.61.2 Description

The Master enters into the MDT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding MDT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the master to the slave.

The IDNs in this IDN list may be taken from S-0-0445 IDN-list of configurable data in the MDT data container or from S-0-0188 IDN-list of configurable data as consumer, if S-0-0445 does not exist.

A.3.62 IDN S-0-0500 AT data container A2 configuration list

A.3.62.1 Attributes

Table A.77 shows the possible attributes for this IDN.

Table A.77 – Attributes for IDN S-0-0500

Attribute	Value
Name	AT data container A2 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.62.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.63 IDN S-0-0501 AT data container A3 configuration list

A.3.63.1 Attributes

Table A.78 shows the possible attributes for this IDN.

Table A.78 – Attributes for IDN S-0-0501

Attribute	Value
Name	AT data container A3 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.63.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.64 IDN S-0-0502 AT data container A4 configuration list

A.3.64.1 Attributes

Table A.79 shows the possible attributes for this IDN.

Table A.79 – Attributes for IDN S-0-0502

Attribute	Value
Name	AT data container A4 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—

Attribute	Value
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.64.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.65 IDN S-0-0503 AT data container A5 configuration list

A.3.65.1 Attributes

Table A.80 shows the possible attributes for this IDN.

Table A.80 – Attributes for IDN S-0-0503

Attribute	Value
Name	AT data container A5 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.65.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.66 IDN S-0-0504 AT data container A6 configuration list

A.3.66.1 Attributes

Table A.81 shows the possible attributes for this IDN.

Table A.81 – Attributes for IDN S-0-0504

Attribute	Value
Name	AT data container A6 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.66.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.67 IDN S-0-0505 AT data container A7 configuration list

A.3.67.1 Attributes

Table A.82 shows the possible attributes for this IDN.

Table A.82 – Attributes for IDN S-0-0505

Attribute	Value
Name	AT data container A7 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.67.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.68 IDN S-0-0506 AT data container A8 configuration list

A.3.68.1 Attributes

Table A.83 shows the possible attributes for this IDN.

Table A.83 – Attributes for IDN S-0-0506

Attribute	Value
Name	AT data container A8 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.68.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.69 IDN S-0-0507 AT data container A9 configuration list

A.3.69.1 Attributes

Table A.84 shows the possible attributes for this IDN.

Table A.84 – Attributes for IDN S-0-0507

Attribute	Value
Name	AT data container A9 configuration list
Version	—
Length	4, variable

Attribute	Value
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.69.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.70 IDN S-0-0508 AT data container A10 configuration list

A.3.70.1 Attributes

Table A.85 shows the possible attributes for this IDN.

Table A.85 – Attributes for IDN S-0-0508

Attribute	Value
Name	AT data container A10 configuration list
Version	—
Length	4, variable
Display Format	IDN
Min input value	—
Max input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.70.2 Description

The Master enters into the AT data container configuration list the IDN numbers for the operation data that are to be sent via the corresponding AT data containers (A and B of standard data container, and A1 to A10 of extended data container) as needed from the slave to the master.

The IDNs in this IDN list may be taken from S-0-0444 or from S-0-0187, if S-0-0444 does not exist.

A.3.71 IDN S-0-1000.0.1 Active SCP Classes

A.3.71.1 Attributes

Table A.86 shows the possible attributes for this IDN.

Table A.86 – Attributes of IDN S-0-1000.0.1

Attribute	Value
Name	Active SCP Classes
Version	—
Length	2, variable
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.71.2 Description

The Master programs the parameter IDN S-0-1000.0.1 with the classes of IDN S-0-1000.0.0 that the slave knows which classes are needed for the application. This parameter can only contain classes, which are also offered in IDN S-0-1000.0.0 by the slave. Each class shall be unique in IDN S-0-1000.0.1. This parameter is writable in CP2 only. Therefore, the slave can make a resource optimization of the communication functions in CP3.

A.3.72 IDN S-0-1000 SCP Type & Version

A.3.72.1 Attributes

Table A.87 shows the possible attributes for this IDN.

Table A.87 – Attributes of IDN S-0-1000

Attribute	Value
Name	SCP Type & Version
Version	—
Length	2, variable
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.72.2 Description

The SCP Type & Version (see Table A.88) contains a list of all Type 19 communication classes with the dedicated version supported by the slave. The support of a class version implies automatically the support of all lower versions of this class too. Nevertheless, a slave shall signal all versions of implemented classes as separate elements. The order of elements in this list is not defined.

The bits 15 ... 8 indicate the SCP class which is available in the slave device, bits 7 ... 4 are reserved, bits 3 ... 0 indicate the version of the SCP class.

Table A.88 – SCP type and version

Class code Bits 15-8	Reserved Bits 7-4	Version Bits 3-0	Class name	Defined in Type 19 version	Description
0x01	0x0	0x1	SCP_FixCFG	V1.1.1	Fix configuration of connections
0x01	0x0	0x2	SCP_FixCFG_0x02	V1.3	Fix configuration of connections
0x01	0x0	0x3	SCP_FixCFG_0x03	V1.3	Fix configuration of connections & connection stop
0x02	0x0	0x1	SCP_VarCFG	V1.1.1	Variable configuration of homogeneous connections
0x02	0x0	0x2	SCP_VarCFG_0x02	V1.3	Variable configuration of homogeneous connections
0x02	0x0	0x3	SCP_VarCFG_0x03	V1.3	Variable configuration of homogeneous connections & connection stop
0x03	0x0	0x1	SCP_Sync	V1.1.1	Synchronization
0x03	0x0	0x2	SCP_Sync_0x02	V1.3	Synchronization tSync > tScyc using MDT Extended field
0x03	0x0	0x3	SCP_Sync_0x03	V1.3	Synchronization tSync > tScyc using MDT Extended field
0x04	0x0	0x1	SCP_WD	V1.1.1	Watch dog of connection
0x04	0x0	0x2	SCP_WD_0x02	V1.3	Watchdog of connection with timeout & data losses
0x05	0x0	0x1	SCP_Diag	V1.1.1	Communication diagnoses
0x06	0x0	0x1	SCP_RTB	V1.1.1	Configuration of real-time bits
0x07	0x0	0x1	SCP_HP	V1.1.1	Hot-plug
0x08	0x0	0x1	SCP_SMP	V1.1.1	SMP
0x09	0x0	0x1	SCP_MuX	V1.1.1	Multiplex channel (standard data container)
0x0A	0x0	0x1	SCP_NRT	V1.1.1	-
0x0B	0x0	0x1	SCP_SIG	V1.1.1	Word of real-time bits as producer and consumer
0x0C	0x0	0x1	SCP_ListSeg	V1.3	segmented list transfer via the SVC
0x0D	0x0	0x1	SCP_IPS	V1.3	Support of IPS internet protocol services using the UC channel
0x0E	0xn	0xn	reserved		for future extensions
0x0F	0x0	0x1	SCP_Cap	V1.3	Connection Capabilities

Class code Bits 15-8	Reserved Bits 7-4	Version Bits 3-0	Class name	Defined in Type 19 version	Description
0x10	0x0	0x1	SCP_ExtMuX	V1.1.2	Extended Multiplex channel (extended data container)
0x11	0x0	0x1	SCP_RTBListProd	V1.3	List of real-time bits as producer (status)
0x12	0x0	0x1	SCP_RTBListCons	V1.3	List of real-time bits as consumer (control)
0x13	0x0	0x1	SCP_SysTime	V1.3	set Type 19 Time using MDT Extended field
0x14	0x0	0x1	SCP_RTBWordProd	V1.3	Word of real-time bits as producer
0x15	0x0	0x1	SCP_RTBWordCons	V1.3	Word of real-time bits as consumer
0x16	0x0	0x1	SCP_SafetyCon	V1.3	CSoS connection
0x17	0x0	0x1	SCP_OvSBasic	V1.3	Word of real-time bits as consumer
0x18	0x0	0x1	SCP_NRTPC	V1.3	UC channel (IP communication)
0x19	0x0	0x1	SCP_Cyc	V1.3	cyclic communication
0x03	0x0	0x1	SCP_Sync	V1.1.1	Synchronization
0x03	0x0	0x2	SCP_Sync_0x02	V1.3	Synchronization tSync > tScyc using MDT Extended field
0x03	0x0	0x3	SCP_Sync_0x03	V1.3	Synchronization tSync > tScyc using MDT Extended field
0x04	0x0	0x1	SCP_WD	V1.1.1	Watch dog of connection
0x04	0x0	0x2	SCP_WD_0x02	V1.3	Watchdog of connection with timeout & data losses
0x05	0x0	0x1	SCP_Diag	V1.1.1	Communication diagnoses
0x06	0x0	0x1	SCP_RTB	V1.1.1	Configuration of real-time bits
0x07	0x0	0x1	SCP_HP	V1.1.1	Hot-plug
0x08	0x0	0x1	SCP_SMP	V1.1.1	SMP
0x09	0x0	0x1	SCP_MuX	V1.1.1	Multiplex channel (standard data container)
0x0A	0x0	0x1	SCP_NRT	V1.1.1	-
0x0B	0x0	0x1	SCP_SIG	V1.1.1	Word of real-time bits as producer and consumer
0x0C	0x0	0x1	SCP_ListSeg	V1.3	segmented list transfer via the SVC
0x0D	0x0	0x1	SCP_IPS	V1.3	Support of IPS internet protocol services using the UC channel
0x0E	0xn	0xn	reserved		for future extensions
0x0F	0x0	0x1	SCP_Cap	V1.3	Connection Capabilities
0x10	0x0	0x1	SCP_ExtMuX	V1.1.2	Extended Multiplex channel (extended data container)
0x11	0x0	0x1	SCP_RTBListProd	V1.3	List of real-time bits as producer (status)
0x12	0x0	0x1	SCP_RTBListCons	V1.3	List of real-time bits as consumer (control)
0x13	0x0	0x1	SCP_SysTime	V1.3	set Type 19 Time using MDT

Class code Bits 15-8	Reserved Bits 7-4	Version Bits 3-0	Class name	Defined in Type 19 version	Description
					Extended field
0x14	0x0	0x1	SCP_RTBWordProd	V1.3	Word of real-time bits as producer
0x15	0x0	0x1	SCP_RTBWordCons	V1.3	Word of real-time bits as consumer
0x16	0x0	0x1	SCP_SafetyCon	V1.3	CSoS connection
0x17	0x0	0x1	SCP_OvSBasic	V1.3	Word of real-time bits as consumer
0x18	0x0	0x1	SCP_NRTPC	V1.3	UC channel (IP communication)
0x19	0x0	0x1	SCP_Cyc	V1.3	cyclic communication

A.3.73 IDN S-0-1002 Communication cycle time

A.3.73.1 Attributes

Table A.89 shows the possible attributes for this IDN.

Table A.89 – Attributes of IDN S-0-1002

Attribute	Value
Name	Communication cycle time (tScyc)
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	31 250 (31,250 µs)
Max. input value	65 000 000 (65000,000 µs)
Positions after decimal point	3
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.73.2 Description

The communication cycle time (tScyc) defines the intervals during which the configured MDTs and ATs shall be transferred by the master to all present slaves. The communication cycle times are defined as 31,25 µs, 62,5 µs, 125 µs, 250 µs or any integer multiple of 250 µs up to 65 ms. In CP2, tScyc needs to be transferred from the master to the slave and has to be activated by the master and slave in CP3 and CP4.

Min/max values are mandatory and shall be adopted by the manufacturer. The min/max values shall be the values that are theoretically possible.

A.3.74 IDN S-0-1003 Allowed MST losses in CP3/CP4

A.3.74.1 Attributes

Table A.90 shows the possible attributes for this IDN.

Table A.90 – Attributes of IDN S-0-1003

Attribute	Value
Name	Allowed MST losses in CP3/CP4
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.74.2 Description

This parameter defines the maximum number of successive communication cycles in which a slave may not receive the MST in CP3 and CP4. The slave shall switch from CP3 or CP4 back to NRT mode if it did not receive its MST within more than this user-defined quantity and

- sets the S-0-0390 Diagnostic number to 0xC30F4001 and
- sets the error (S-DEV.Bit7) in its S-0-1045 Device Status (S-DEV)

Less than 50 % of the operation data of this parameter, the slave shall

- resets the communication warning (S-DEV.Bit15) in its device status and
- resets the warning (S-DEV.Bit6) in its S-0-1045 Device Status (S-DEV)

Reaching more than 50 % of the operation data of this parameter, the slave shall

- sets the S-0-0390 Diagnostic number to 0xC30E4001
- sets the communication warning (S-DEV.Bit15) in its device status and
- flashes the Type 19 LED between red and green at least 2 seconds or as long as the communication warning (S-DEV.Bit15) is present and
- sets the warning (S-DEV.Bit6) in its S-0-1045 Device Status (S-DEV)

Whereas this parameter shall be set to a value that is large enough (compare to S-0-1050.x.11 Allowed Data Losses), so that the master can keep communicating with undisturbed sub-devices for safely stopping the other parts of the machine.

For example:

- S-0-1003 = 5 -> Warning is set with 3 telegram losses. Error is set with 6 telegram losses.
- S-0-1003 = 4 -> Warning is set with 3 telegram losses. Error is set with 5 telegram losses.

A.3.75 IDN S-0-1005 Minimum feedback processing time (t_5)

A.3.75.1 Attributes

Table A.91 shows the possible attributes for this IDN.

Table A.91 – Attributes of IDN S-0-1005

Attribute	Value
Name	Minimum feedback processing time (t_5)
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	3
Write protection	Always
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.75.2 Description

t_5 specifies the slave-specific maximum time duration that it needs between capturing its producer data (for example feedback values) and making them ready for transmission within ATs.

If IDN S-0-1060 (Connection capabilities) are available, t_5 covers the maximum value of IDN S-0-1060.x.07 (Maximum processing time) for all producer instances of IDN S-0-1060.

A.3.76 IDN S-0-1006 AT transmission starting time (t_1)

A.3.76.1 Attributes

Table A.92 shows the possible attributes for this IDN.

Table A.92 – Attributes of IDN S-0-1006

Attribute	Value
Name	AT transmission starting time (t_1)
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	3
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.76.2 Description

The AT0 transmission starting time (t1) determines the nominal time interval between the end of MST and beginning of AT0. The master sends its AT0 based on the MST in CP3 and CP4. This parameter shall be transferred by the master to the slave during CP2.

The maximum transmission starting time shall be less than tScyc.

A.3.77 IDN S-0-1007 Synchronization time (Tsync)

A.3.77.1 Attributes

Table A.93 shows the possible attributes for this IDN.

Table A.93 – Attributes for IDN S-0-1007

Attribute	Value
Name	Synchronization time (Tsync)
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	0
Max. input value	—
Positions after decimal point	3
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	0µs

A.3.77.2 Description

The synchronization time (TSync) defines the time at which all producer cycle times (producing and consuming connections) in a slave are synchronized. The master shall set the synchronization time smaller than the synchronization cycle time (tsync-cycle). The synchronization cycle time is the least common multiple of all producer cycle times (tPcyc) which shall be synchronized in a network.

In each communication cycle the synchronization reference point (TSref) is generated by the MSTs. One of these TSref times is selected with TSref-Counter = 0 (TSref0). The synchronization time (Tsync) defines the offset between the time TSref0 and TSync for the synchronization cycle (tsync-cycle).

All producer cycles (tPcyc) in the slave are synchronized at time TSync. The following synchronization times are defined by the corresponding producer cycle times. To distinguish the synchronization times of the connections, they are labeled as T4PCx. That means, synchronization time (T4) of the producer cycle (PC), with the structure index (x) of the connection (x = 0 … 255).

The sub-device shall enable the synchronization time during CP3.

Figure A.4 shows an example of the synchronization timing with different producer cycles.

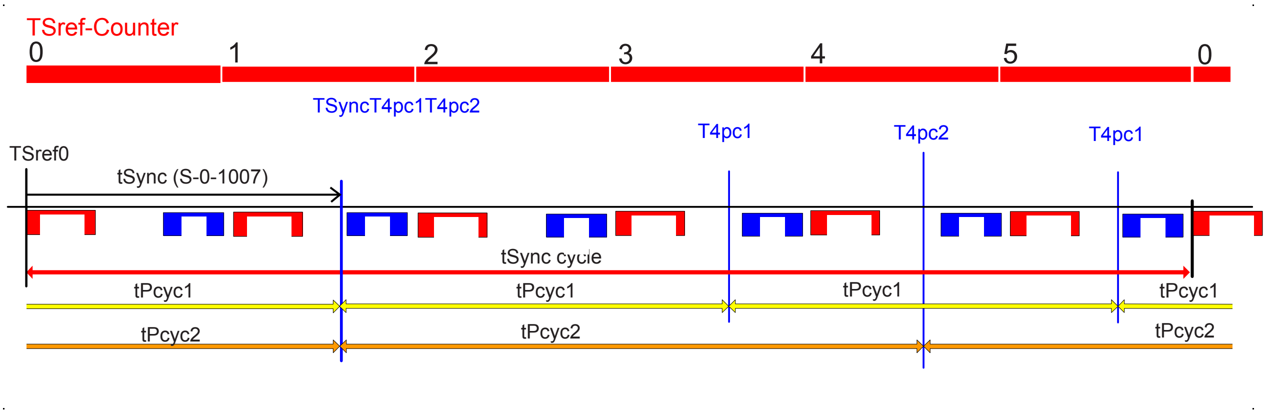

Figure A.4 – Example of synchronization timing with different producer cycles

A.3.78 IDN S-0-1008 Command value valid time (t_3)

A.3.78.1 Attributes

Table A.94 shows the possible attributes for this IDN.

Table A.94 – Attributes for IDN S-0-1008

Attribute	Value
Name	Command value valid time (t_3)
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	0
Max. input value	tScyc
Positions after decimal point	3
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.78.2 Description

The command value valid time indicates the time after which the slave can access the new values from the MDT related to the synchronization time. Thus the master can preset the same command value valid time for all coordinated applications.

A.3.79 IDN S-0-1009 Device Control (C-DEV) offset in MDT

A.3.79.1 Attributes

Table A.95 shows the possible attributes for this IDN.

Table A.95 – Attributes of IDN S-0-1009

Attribute	Value
Name	Device Control (C-DEV) offset in MDT
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.79.2 Description

The device control offset in the MDT defines MDT number and the position within this MDT for the device control. This parameter shall be transferred by the master to each slave during CP2. It shall become active during CP3 in the master and slave. This offset shall start after the Type 19 header within the MDT. The offset of device control shall be an even number.

Table A.96 shows the structure of this IDN.

Table A.96 – C-DEV Offset in MDT

Bit number	Bit value	Description
15-14	00	reserved
13-12	—	MDT-number
	00	MDT0
	01	MDT1
	10	MDT2
	11	MDT3
11-0	—	Offset in MDT (in octets)
	0..1492	C-DEV offset in MDT (shall be an even number)

A.3.80 IDN S-0-1010 Lengths of MDTs

A.3.80.1 Attributes

Table A.97 shows the possible attributes for this IDN.

Table A.97 – Attributes of IDN S-0-1010

Attribute	Value
Name	Lengths of MDTs
Version	—
Length	2, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.80.2 Description

The lengths of the MDTs shall be an even number and are expressed in octets (see Figure A.5). The parameter shall contain the lengths of the four possible master data telegrams. The lengths are necessary for the initialization of the Type 19 communication hardware. Always all four lengths have to be specified. Not configured MDTs shall be marked with length = 0 and the master shall not transmit these MDTs. Each slave shall be informed by the master during CP2 of the lengths of all configured MDTs. It shall become active in the master and slave during CP3.

The length includes all data between the end of Type 19 header to the beginning of FCS.

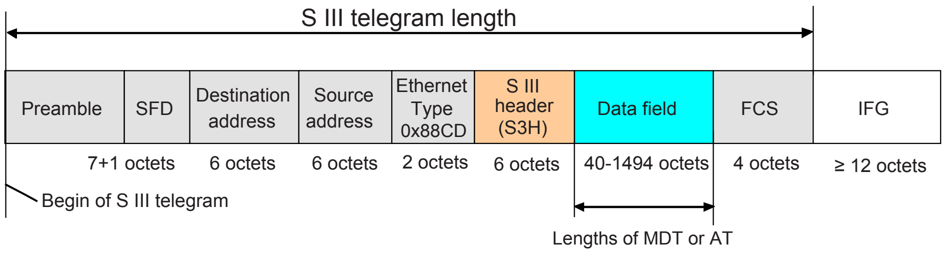

Figure A.5 – Definition of MDT length

The master shall set the lengths of any configured MDT in the range of 40 ≤ MDT length ≤ 1494, otherwise the slave generates the error message 0x7008 in the SVC.

The number of list elements of this parameter is fixed to four.

Figure A.6 shows the structure of this IDN as an example.

Figure A.6 – Lengths of MDTs (example)

A.3.81 IDN S-0-1011 Device Status (S-DEV) offset in AT

A.3.81.1 Attributes

Table A.98 shows the possible attributes for this IDN.

Table A.98 – Attributes of IDN S-0-1011

Attribute	Value
Name	Device Status (S-DEV) offset in AT
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.81.2 Description

The device status offset in AT defines the position of device status field of the slave in one of the ATs, expressed as a OCTET position. The offset shall start after the Type 19 header within the AT. This parameter shall be transferred by the master to each slave during CP2. It shall become active during CP3 in the master and slave. The offset of device status shall be an even number.

Table A.99 shows the structure of this IDN.

Table A.99 – S-DEV Offset in AT

Bit number	Bit value	Description
15-14	—	reserved
13-12	—	AT-number
	00	AT0
	01	AT1
	10	AT2
	11	AT3
11-0	—	S-DEV offset in AT (in octets)
	0..1492	S-DEV offset in AT (shall be an even number)

A.3.82 IDN S-0-1012 Lengths of ATs

A.3.82.1 Attributes

Table A.100 shows the possible attributes for this IDN.

Table A.100 – Attributes of IDN S-0-1012

Attribute	Value
Name	Lengths of ATs
Version	—
Length	2, variable
Display Format	Unsigned decimal
Min input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.82.2 Description

The lengths of the ATs shall be an even number and are expressed in octets (see Figure A.7). The parameter shall contain the lengths of the four possible ATs. The lengths are necessary for the initialization of the Type 19 hardware. Always all four lengths have to be specified. Not configured ATs shall be marked with length = 0 and the master shall not transmit these ATs. Each slave shall be informed by the master during CP2 of the lengths of all configured ATs. It shall become active in the master and slave during CP3. The length includes all data between the end of AT header to the beginning of the FCS. The number of list elements of this parameter is fixed to four.

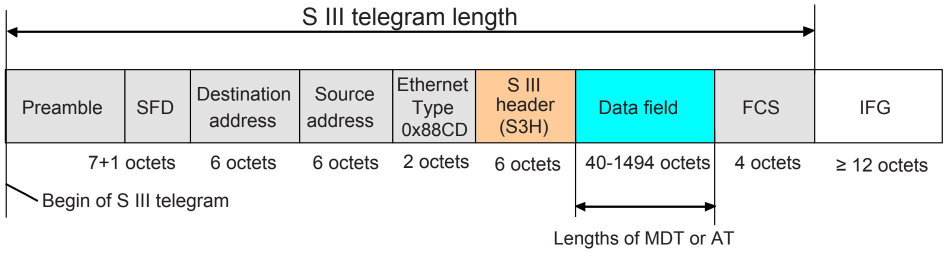

Figure A.7 – Definition of AT length

Figure A.8 shows the structure of this IDN, as an example.

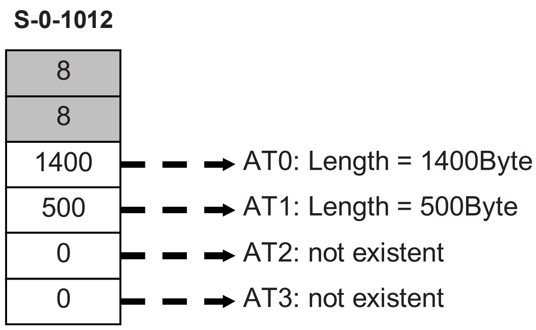

Figure A.8 – Lengths of ATs (example)

A.3.83 IDN S-0-1013 SVC offset in MDT

A.3.83.1 Attributes

Table A.101 shows the possible attributes for this IDN.

Table A.101 – Attributes of IDN S-0-1013

Attribute	Value
Name	SVC offset in MDT
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.83.2 Description

The SVC offset in the MDT defines the position of the service channel for the slave. This offset shall start after the Type 19 header within the related MDT. Every slave shall be informed by the master during CP2 of the offset of the service channel in the MDT. This parameter shall become active during CP3 in the master and slave. The SVC offset shall be an even number.

Table A.102 shows the structure of this IDN.

Table A.102 – SVC Offset in MDT

Bit number	Bit value	Description
15-14	—	(reserved)
	0	No others values allowed
13-12	—	MDT number
	00	MDT0
	01	MDT1
	10	MDT2
	11	(reserved)
11-0	—	MDT SVC-Offset (in octets)
	0…1484	MDT SVC-Offset (shall be an even number)

A.3.84 IDN S-0-1014 SVC offset in AT

A.3.84.1 Attributes

Table A.103 shows the possible attributes for this IDN.

Table A.103 – Attributes of IDN S-0-1014

Attribute	Value
Name	SVC offset in AT
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.84.2 Description

The SVC offset in AT defines the position of a service channel for the slave. The offset shall start after the Type 19 header within the related AT. Every slave shall be informed by the master during CP2 of the offset of the service channel in the AT. This parameter shall become active during CP3 in the master and slave. The SVC offset shall be an even number.

Table A.104 shows the structure of this IDN.

Table A.104 – SVC Offset in AT

Bit number	Bit value	Description
15-14	—	reserved
	0	No others values allowed
13-12	—	AT-number
	00	AT0
	01	AT1
	10	AT2
	11	AT3
11-0	—	AT SVC-Offset (in octets)
	0…1484	AT SVC-Offset (shall be an even number)

A.3.85 IDN S-0-1015 Ring delay

A.3.85.1 Attributes

Table A.105 shows the possible attributes for this IDN.

Table A.105 – Attributes of IDN S-0-1015

Attribute	Value
Name	Ring delay
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	1048,575
Positions after decimal point	3
Write protection	Never
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.85.2 Description

The master determines the entire ring delay and assigns it to the slaves. The slaves need the ring delay to eliminate the duration of the communication and to determine their synchronization reference time (TSref).

The Master calculates the ring delay for a given line or ring topology using the following formula:

S-0-1015 Ring delay = tRing + IFG jitter + extra delay

After each power turn on, this parameter contains different values. Therefore this IDN shall not be in the S-0-0327.x.0 (IDN list of checksum parameter).

A.3.86 IDN S-0-1016 Slave delay (P/S)

A.3.86.1 Attributes

Table A.106 shows the possible attributes for this IDN.

Table A.106 – Attributes of IDN S-0-1016

Attribute	Value
Name	Slave delay (P/S)
Version	—
Length	4, variable
Display Format	Unsigned decimal
Min. input value	0
Max. input value	—
Positions after decimal point	3
Write protection	Always
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.86.2 Description

After the master has assigned S-0-1015 Ring delay to the slave, it determines SYNCCNT-P and SYNCCNT-S, when S-0-1024 SYNC delay measuring procedure command is executed.

In the ring topology both list elements shall be filled with the measured values.

In the line topology:

- the greater measured value is written in the corresponding list element,
- the lower measured value is discarded and the unused list element is set to zero.

List elements:

- List element 0 is SYNCCNT-P
- List element 1 is SYNCCNT-S

A.3.87 IDN S-0-1017 UC channel transmission time

A.3.87.1 Attributes

Table A.107 shows the possible attributes for this IDN.

Table A.107 – Attributes of IDN S-0-1017

Attribute	Value
Name	UC channel transmission time
Version	—
Length	4, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	3
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.87.2 Description

The NRT transmission time includes two list elements (t6 & t7).

- First list element t6 (beginning of the UC channel)
  - T6 defines the time for the slave to switch the topology from RT channel to UC channel.
  - If UC channel is not required, the master shall set t6 to 0.
  - If UC channel is required master shall set t6 as specified in clause medium access.
- Second list element t7 (end of UC channel)
  - T7 defines the time for the slave to switch the topology from UC channel to RT channel.
  - If UC channel is not required, the time t7 is don't care.
  - If UC channel is required master shall set t7 as specified in clause medium access.

If the length of the UC channel is less than 125µs, S-0-1027.0.1 Requested MTU shall be adjusted accordingly.

The limits of t6 & t7 are described in 7.1.2.

A.3.88 IDN S-0-1019 MAC address

A.3.88.1 Attributes

Table A.108 shows the possible attributes for this IDN.

Table A.108 – Attributes of IDN S-0-1019

Attribute	Value
Name	MAC address
Version	—
Length	1, variable
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.88.2 Description

The slave inserts his MAC address in this parameter.

Figure A.9 shows the structure of this IDN.

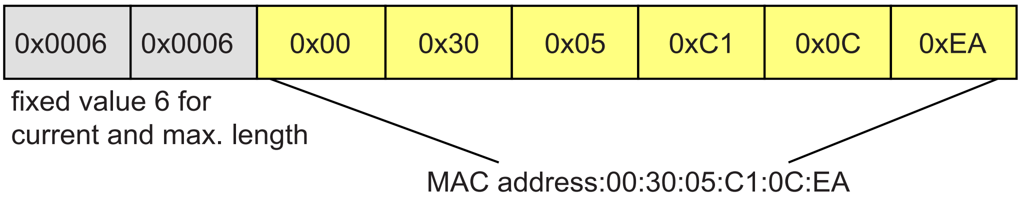

Figure A.9 – Structure of MAC address

A.3.89 IDN S-0-1020.0.1 Current IP address

A.3.89.1 Attributes

Table A.109 shows the possible attributes for this IDN.

Table A.109 – Attributes of IDN S-0-1020.0.1

Attribute	Value
Name	Current IP address
Version	—
Length	1, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.89.2 Description

This IDN contains the current IP address of the slave's Type 19 communication interface. The master may change the IP address by writing the IDN S-0-1020 (IP address) and executing the procedure command IDN S-0-1048 (Activate network settings).

Figure A.10 shows the structure of this IDN.

A.3.90 IDN S-0-1020 IP address

A.3.90.1 Attributes

Table A.110 shows the possible attributes for this IDN.

Table A.110 – Attributes of IDN S-0-1020

Attribute	Value
Name	IP address
Version	—
Length	1, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.90.2 Description

For SCP_NRT: This IDN contains the IP address of the slave's Type 19 communication interface. The master may change the IP address by writing this IDN.

For SCP_NRTPC: This IDN contains the requested IP address of the slave's Type 19 communication interface. The master may change the IP address by writing this IDN and executing the procedure command S-0-1048 (Activate network settings) to activate it. The current IP address is displayed in S-0-1020.0.1 (Current IP address).

Figure A.10 shows the structure of this IDN.

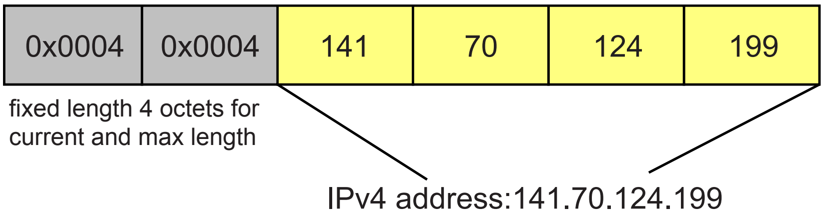

Figure A.10 – Structure of IP address

A.3.91 IDN S-0-1021.0.1 Current subnet mask

A.3.91.1 Attributes

Table A.111 shows the possible attributes for this IDN.

Table A.111 – Attributes of IDN S-0-1021.0.1

Attribute	Value
Name	Current subnet mask
Version	—
Length	1, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.91.2 Description

This IDN contains the currently activated subnet mask of the slave's Type 19 communication interface. The master may change the subnet mask by writing the S-0-1021.0.0 and executing the procedure command S-0-1048 according to subclause A.3.113.

Table A.112 shows the structure of this IDN.

A.3.92 IDN S-0-1021 Subnet mask

A.3.92.1 Attributes

Table A.112 shows the possible attributes for this IDN.

Table A.112 – Attributes of IDN S-0-1021

Attribute	Value
Name	Subnet mask
Version	—
Length	1, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.92.2 Description

For SCP_NRT: This IDN contains the subnet mask of the slave's Type 19 communication interface. The master may change the subnet mask by writing this IDN.

For SCP_NRTPC: This IDN contains the subnet mask of the slave's Type 19 communication interface. The master may change the subnetmask by writing this IDN and executing the procedure command S-0-1048 (Activate network) settings to activate it. The current subnet mask is displayed in S-0-1020.0.1 (Current IP address).

Figure A.11 shows the structure of this IDN.

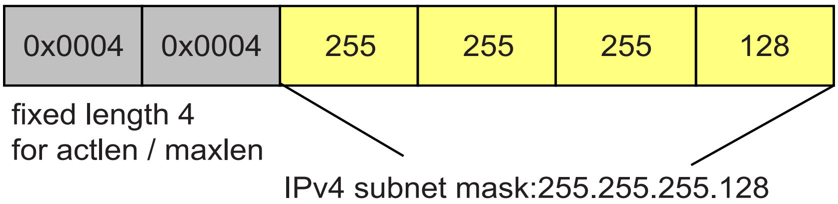

Figure A.11 – Structure of subnet mask

A.3.93 IDN S-0-1022.0.1 Current gateway address

A.3.93.1 Attributes

Table A.113 shows the possible attributes for this IDN.

Table A.113 – Attributes of IDN S-0-1022.0.1

Attribute	Value
Name	Gateway address
Version	—
Length	1, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.93.2 Description

This IDN contains the currently active gateway address of the slave's Type 19 communication interface. The master may change the gateway address by writing the S-0-1022.0.0 and executing the procedure command S-0-1048.

Figure A.12 shows the structure of this IDN.

A.3.94 IDN S-0-1022 Gateway address

A.3.94.1 Attributes

Table A.114 shows the possible attributes for this IDN.

Table A.114 – Attributes of IDN S-0-1022

Attribute	Value
Name	Gateway address
Version	—
Length	1, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.94.2 Description

For SCP_NRT: This IDN contains the gateway address of the slave's Type 19 communication interface. The master may change the gateway address by writing this IDN.

For SCP_NRTPC: This IDN contains the requested gateway address of the slave's Type 19 communication interface. The master may change the gateway address by writing this IDN and executing the procedure command S-0-1048 (Activate network settings) to activate it. The current gateway address is displayed in S-0-1020.0.1 (Current IP address).

Figure A.12 shows the structure of this IDN.

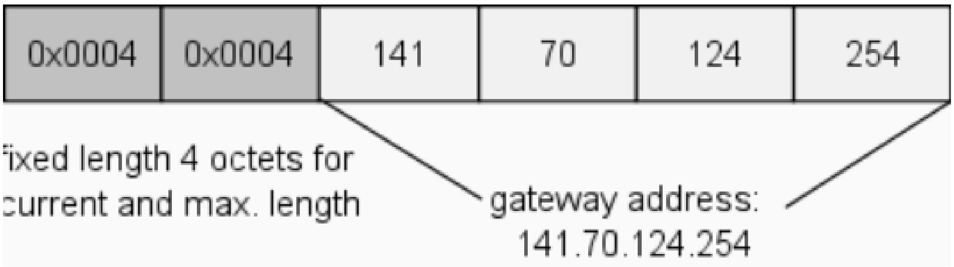

Figure A.12 – Structure of gateway address

A.3.95 IDN S-0-1023 SYNC jitter

A.3.95.1 Attributes

Table A.115 shows the possible attributes for this IDN.

Table A.115 – Attributes of IDN S-0-1023

Attribute	Value
Name	SYNC jitter
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	3
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.95.2 Description

The master shall calculate the maximum synchronization jitter. The jitter is used in the slave to determine the width of the MST window. The MST window is 2x synchronization jitter. The master shall transmit the parameter IDN S-0-1023 to all slaves supporting SCP_Sync.

The master shall calculate the maximum synchronization jitter as follows:

The maximum jitter of the transmitted MSTs (MST jitter) is dependent of the master hardware. The MST jitter is known by the master.

The maximum jitter of the Inter frame gap (IFG jitter) is dependent of the number of participants in the topology. Therefore the following formula shall be used:

Formula of IFG jitter calculation:

$$\text{IFG jitter} \geq \frac{27 * (S-0-1037)_{MAX} * \sqrt{2 * N}}{8000} \, [\mu s]$$

NOTE N is the number of participants in the topology.

The SYNC jitter shall include the MST jitter and the IFG jitter and shall be greater than or equal to the deviation of the accuracy of the crystal oscillator related to the S-0-1002 Communication Cycle time (tScyc).

Example:

- S-0-1002 = 10 ms;
- accuracy = 100 µHz/Hz;
- --> deviation = 1 µs;
- the Sync Jitter shall be greater than or equal to the deviation of 1µs.

Formula of SYNC jitter calculation:

$$\text{SYNC jitter} = \frac{\text{MST jitter} + \text{IFG jitter}}{2} \geq \frac{S-0-1002}{10000\,[100ppm]} \, [\mu s]$$

MST jitter of ±40 ns shall be used with 80 ns in the calculation of SYNC jitter.

A.3.96 IDN S-0-1024 SYNC delay measuring procedure command

A.3.96.1 Attributes

Table A.116 shows the possible attributes for this IDN.

Table A.116 – Attributes of IDN S-0-1024

Attribute	Value
Name	SYNC delay measuring procedure command
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.96.2 Description

After activation of this procedure command the slave shall determine SYNCCNT-P and SYNCCNT-S (S-0-1016 Slave delay (P&S)) depending on S-0-1015 Ring delay.

a) In CP2 the positive acknowledgment is generated if the slave was able to determine valid SYNCCNT-P and SYNCCNT-S. In this case the slave synchronizes in CP3 or CP4 automatically.
b) In CP3 or CP4 the positive acknowledgment is generated if the slave was able to determine valid SYNCCNT-P and SYNCCNT-S and the slave has synchronized again.

The slave will generate a negative acknowledgment and set the following diagnostics:

a) S-0-1015 Ring delay is invalid (for example value = 0) --> S-0-0390 Diagnostic number is set to diagnostic code 0xC30C5301.
b) SYNCCNT-P or SYNCCNT-S is 0, or the measuring of SYNCCNT-P or SYNCCNT-S was interrupted or disturbed --> S-0-0390 Diagnostic number is set to diagnostic code 0xC30C5302.

The master shall activate this procedure command

- and wait until it is finished in CP2 before activation of S-0-0127 CP3 transition check,
- in CP3 and CP4 after the ring recovery

in every slave which has to be synchronized.

A.3.97 IDN S-0-1026 Version of communication hardware

A.3.97.1 Attributes

Table A.117 shows the possible attributes for this IDN.

Table A.117 – Attributes of IDN S-0-1026

Attribute	Value
Name	Version of communication hardware
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.97.2 Description

This parameter includes the Type 19 communication hardware identification, for example hardware version and revision.

Example for FPGA: SERCON100M V02.01

A.3.98 IDN S-0-1027.0.1 Requested MTU

A.3.98.1 Attributes

Table A.118 shows the possible attributes for this IDN.

Table A.118 – Attributes of IDN S-0-1027.0.1

Attribute	Value
Name	Requested MTU
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	46
Max. input value	1500
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.98.2 Description

The requested MTU defines the maximum number of octets, that may be sent within the UC channel by higher layers. This IDN only defines the target value for the MTU and is used to calculate the S-0-1027.0.2 Effective MTU. The effective value may be different from the target value, if the target value exceeds the limits of the current communication phase.

- In case of SCP_NRT the effective MTU has to be set at the Type 19 interface immediately and updated when this IDN is written or a phase change has occurred.
- In case of SCP_NRTPC the effective MTU has to be recalculated and set to the Type 19 interface by executing procedure command S-0-1048 (Activate network settings) or a phase change has occurred.

E.g. if this IDN is set to 80, the effective MTU during NRT, CP0, CP1, CP2 and HP0 will be 576 (see Table A.119).

NOTE This parameter may be used to calculate the last transmission time of the UC channel.

If Type 19 V1.1.1 devices as well as Type 19 devices of newer versions are present in the same ring, this parameter shall be set to the value 1500 on every Type 19 device newer than Type 19 V1.1.1 by the master.

Calculation of the effective MTU:

tNRT = (t7 – t6) > 6,72 μs

$$\text{MTU}(\text{tNRT}) = \min\left\{1500; \frac{\text{tNRT}}{\text{s}} * 12.498.750 - 38\right\}$$

Table A.119 – Upper and lower Limit of MTU

Communication phase (CP)	Upper limit (CP)	Lower limit (CP)
NRT	1500	**576**
CP0	1500	**576**
CP1	1500	**576**
CP2	1500	**576**
CP3	MTU(tNRT)	46
CP4	MTU(tNRT)	46
HP0	1500	**576**
HP1	MTU(tNRT)	46
HP2	MTU(tNRT)	46

$MTU_{interim}$ = min {upperlimit(cp); $MTU_{requested}$}

$MTU_{effective}$ = max {lowerlimit(cp); $MTU_{interim}$}

Default value for this IDN shall be 1500.

A.3.99 IDN S-0-1027.0.2 Effective MTU

A.3.99.1 Attributes

Table A.120 shows the possible attributes for this IDN.

Table A.120 – Attributes of IDN S-0-1027.0.2

Attribute	Value
Name	Effective MTU
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	46
Max. input value	1500
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.99.2 Description

This IDN is the current MTU. The current MTU is calculated using IDN S-0-1017 and IDN S-0-1027.0.1. For further information see IDN S-0-1027.0.1.

A.3.100 IDN S-0-1028 Error counter MST-P/S

A.3.100.1 Attributes

Table A.121 shows the possible attributes for this IDN.

Table A.121 – Attributes of IDN S-0-1028

Attribute	Value
Name	Error counter MST-P/S
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.100.2 Description

The MST error counter is incremented by one if a valid MST is neither received on port 1 nor on port 2 inside the specified MST window, see S-0-1023 SYNC jitter in communication phases 3 and 4.

The MST error counter shall stop counting as soon as it reaches 65 535. It means that if the counter has a value of 65 535, there may have been more than 65 535 invalid MST's (for example noisy transmission over a long period of time).

This counter shall be reset by the slave, if the master switches from CP2 to CP3.

A.3.101 IDN S-0-1031 Test pin assignment Port 1 & Port 2

A.3.101.1 Attributes

Table A.122 shows the possible attributes for this IDN.

Table A.122 – Attributes of IDN S-0-1031

Attribute	Value
Name	Test pin assignment Port 1 & Port 2
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.101.2 Description

This parameter is used to assign communication related hardware signals to the test pins TS1 and TS2 (see Table A.123). The assigned signal is mapped to the test pins TS1 or TS2.

Table A.123 – Structure of test pin assignment Port 1 & Port 2

Bit No.	Value	Description
15-12		**(reserved)**
11-8		The selected signal is mapped to the test pin TS2 (selectable signals are shown in the following table).
7-4		(reserved)
3-0		The selected signal is mapped to the test pin TS1 (selectable signals are shown in the following table).

Table A.124 – Selectable output signals

Value	Signal slave	Description
0000	Port 1 MST	**MST pulse from the Rx MAC of port 1 (40 ns duration)**
0001	Port 2 MST	MST pulse from the Rx MAC of port 2 (40 ns duration)
0010	TMST	TMST signal after MST generator (TSref)
0011	CON_CLK	CON Clock from the TCNT timer
0100	DIV_CLK	DIV Clock from the DIV Clock unit (only if present)
0101	TCNT Reload	Overflow of the TCNT timer
0110	Port 1 TCNT Reload	Overflow of the Port 1 timer
0111	Port 2 TCNT Reload	Overflow of the Port 2 timer
1000	Port 1 IP Open	Port 1 IP window
1001	Port 1 IP Open Write	Port 1 IP transmit window
1010	Port 2 IP Open	Port 2 IP window
1011	Port 2 IP Open Write	Port 2 IP transmit window
1100	Port 1 MST Window Open	Port 1 MST window
1101	Port 2 MST Window Open	Port 2 MST window
1110	Port 1 Rx Frame	Reception of a frame on port 1
1111	Port 2 Rx Frame	Reception of a frame on port 2

A.3.102 IDN S-0-1034 PHY error counter Port 1 & Port 2

A.3.102.1 Attributes

Table A.125 shows the possible attributes for this IDN.

Table A.125 – Attributes of IDN S-0-1035

Attribute	Value
Name	PHY error counter Port 1 & Port 2
Version	—
Length	4
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.102.2 Description

These error counters monitor the Ethernet PHYs for error pulses.

The low word represents the error counter of port 1, the high word represents the error counter of port 2.

Due to the only supported 25MHz (100Mbit) of a Type 19 network, the value of these counters represent the amount of 40ns clock periods with "False Carrier" or "Data reception with errors" indication by the PHY over the MII interface.

This IDN is writable, so a human machine interface may reset these error counters. The maximum value for each counter is 0xFFFF. The error counters are not buffered and shall be set to 0 on power-on.

Table A.126 was extracted from the IEEE802.3 and ISO/IEC 8802-3 to show the coding of the PHY errors.

Table A.126 – Coding of PHY errors

RX_DV	RX_ER	RXD<3:0>	Indication
1	1	0000 through 1111	Data reception with errors
0	1	1110	False Carrier indication
0	1	0000	Normal inter-frame
0	1	0001 through 1101	(reserved)
0	1	1111	(reserved)

A.3.103 IDN S-0-1035 Error counter Port 1 & Port 2

A.3.103.1 Attributes

Table A.127 shows the possible attributes for this IDN.

Table A.127 – Attributes of IDN S-0-1035

Attribute	Value
Name	Error counter Port 1 & Port 2
Version	—
Length	4
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.103.2 Description

These error counters monitor MAC telegrams, these are Type 19 telegrams as well as Ethernet telegrams.

The low word represents the error counter of port 1, the high word represents the error counter of port 2.

The counter shall start in CP0 and will be incremented maximal one time per communication cycle, if an invalid MAC telegram is detected. The error counters are writable, so a human machine interface may reset these error counters. The maximum value for each counter is 0xFFFF. The error counters are not buffered and shall be set to 0 on power-on.

Invalid telegram: The contents of invalid MAC telegrams shall not be passed to sublayers. The occurrence of invalid MAC telegrams may be communicated to network management. Invalid MAC telegrams may be ignored or discarded.

An invalid telegram shall be defined as one that meets at least one of the following conditions:

- Alignment error: It is not an integral number of octets in length.
- FCS error: The bits of the incoming telegram (exclusive of the FCS field itself) do not generate a FCS value identical to the one received.
- CRC error: The bits of the incoming telegram (exclusive of the CRC field itself of Type 19 header) do not generate a CRC value identical to the one received.
- Type 19 telegram length error: The telegram was not received with the expected length.

The MAC telegrams are checked in different ways. This is specified as shown in Table A.128:

Table A.128 – Checking of MAC telegrams

MAC telegrams	FCS	CRC	Alignment	Length
Ethernet telegrams	Yes	No	Yes	No
Type 19 telegrams	Yes	Yes	Yes	Yes

A.3.104 IDN S-0-1036 Inter Frame Gap

A.3.104.1 Attributes

Table A.129 shows the possible attributes for this IDN.

Table A.129 – Attributes of IDN S-0-1036

Attribute	Value
Name	Inter Frame Gap
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	12
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	Octets

A.3.104.2 Description

A brief recovery time between frames allows devices to prepare for reception of the next frame. Ethernet specifies the minimum inter frame gap with 96 bits (12 octets), which is a time of 960 ns using a transmission rate of 100 Mbit/s.

The inter frame gap is dependent on S-0-1037 Slave Jitter and the number of participants (N) in the topology. The slave jitter is defined in the parameter S-0-1037 Slave Jitter and shall be read by the master of each slave. If this parameter is not supported by the slave, the master shall use a default value of 80ns. Only the greatest value of all S-0-1037 shall be used for the IFG calculation.

The formula of IFG shall be used by the master to calculate the inter frame gap for the given application.

Formula of IFG calculation:

$$S-0-1036 \geq \frac{27 * S-0-1037_{MAX} * \sqrt{2 * N}}{8000} * \frac{1 \text{octet}}{0{,}08\,\mu s} + 12\,\text{octets}$$

The master shall transmit this inter frame gap behind every transmitted Type 19 telegram.

If the master doesn't transfer this parameter during CP2 in this case the slave shall use the default value of 37 octets.

The slave shall use this inter frame gap for its Type 19 telegram timing calculation.

The formula of tIFG and IFG jitter shall be calculate the inter frame gap as time for the given application.

Formula of tIFG calculation:

$$tIFG \geq \frac{27 * S-0-1037_{MAX} * \sqrt{2 * N}}{8000} + 0{,}96\,\mu s$$

Formula of IFG jitter calculation:

$$IFG\ jitter \geq tIFG - 0{,}96\,\mu s$$

A.3.105 IDN S-0-1037 Slave jitter

A.3.105.1 Attributes

Table A.130 shows the possible attributes for this IDN.

Table A.130 – Attributes of IDN S-0-1037

Attribute	Value
Name	Slave Jitter
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	160
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	Octets

A.3.105.2 Description

The interface produces a telegram jitter which dependents on the hardware implementation. The slave shall show its jitter in this parameter. The master shall use this parameter to determine the inter frame gap for the transmission. The value of this parameter has to be specified by the manufacturer.

A jitter of ± 40 ns shall be shown with 80 ns in this parameter.

A.3.106 IDN S-0-1039.0.1 Current active hostname

A.3.106.1 Attributes

Table A.131 shows the possible attributes for this IDN.

Table A.131 – Attributes of IDN S-0-1039.0.1

Attribute	Value
Name	Current active hostname
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.106.2 Description

This parameter displays the current active network hostname of this device. The master may change the hostname by writing the S-0-1039.0.0 and executing the procedure command S-0-1048 to activate it. The device shall use this parameter in a DHCP request for identifying the node functionality in order to assign the IP address.

This parameter shall reserve at minimum a text of 16 octets.

A.3.107 IDN S-0-1039 Hostname

A.3.107.1 Attributes

Table A.132 shows the possible attributes for this IDN.

Table A.132 – Attributes of IDN S-0-1039

Attribute	Value
Name	Hostname
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.107.2 Description

This parameter shows a network hostname of this device. The device shall use this parameter in a DHCP request for identifying the node functionality in order to assign the IP address. The contents of S-0-1039 (Hostname) shall be remanent. This parameter shall reserve at minimum a text of 16 octets.

A.3.108 IDN S-0-1040 Sub-device address

A.3.108.1 Attributes

Table A.133 shows the possible attributes for this IDN.

Table A.133 – Attributes of IDN S-0-1040

Attribute	Value
Name	Sub-device address
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	0
Max. input value	511
Positions after decimal point	0
Write protection	Manufacturer-specific
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.108.2 Description

This parameter shall contain the sub-device address of a slave.

The address can either be set via the service channel or by means of an address switch (for example dip-switch).

Setting the sub-device address via the SVC

- The device shall apply the sub-device address immediately after the SVC write access
- The contents of S-0-1040 shall be stored remanently

Setting the sub-device address via an address switch

- The device shall always apply the sub-device address, which is shown on the address switch.

These values shall be displayed immediately.

If a device supports both ways of address configuration, it shall have the following behavior:

- If the address switch indicates an address, which is not 0
  - The device applies the address, which is shown on the address switch
  - S-0-1040 is write protected
- If the address switch indicates the address 0
  - The sub-device address can only be configured via the SVC
  - S-0-1040 is not write protected

Min/max values are mandatory and shall be adopted by the manufacturer.

A.3.109 IDN S-0-1041 AT Command value valid time (t9)

A.3.109.1 Attributes

Table A.134 shows the possible attributes for this IDN.

Table A.134 – Attributes of IDN S-0-1041

Attribute	Value
Name	AT Command value valid time (t9)
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	0
Max. input value	tScyc
Positions after decimal point	3
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.109.2 Description

After the AT Command value valid time (t9) the slave can access the new command values from the AT. Thus the master can preset the same AT Command value valid time (t9) for all coordinated applications.

A.3.110 IDN S-0-1044 Device Control (C-DEV)

A.3.110.1 Attributes

Table A.135 shows the possible attributes for this IDN.

Table A.135 – Attributes of IDN S-0-1044

Attribute	Value
Name	Device Control
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.110.2 Description

The device control contains the control information (for example, topology control) which are set by the master and evaluated by the slave (Table A.136). The device control is not part of a connection.

Table A.136 – Device control field (C-DEV)

Bit number	Bit value	Description
15	—	Identification
	0	No Identification request
	1	Identification request (slave shows the condition of this bit at the Type 19 LED or at the display). This function is used for the remote address allocation or for configuration errors between master and slave.
14	—	Topology HS (Initial value is 0 in every CP)
	toggle	The master toggles every time it requires a topology change.
13-12	—	Topology control (Master selects the new topology)
	00	Fast-Forward on both ports
	01	Loopback with Forward of P-Telegrams
	10	Loopback with Forward of S-Telegrams
	11	(reserved: slave shall ignore this bit combination)
11	—	Control physical topology (If the slave detects a toggle, then it shall drop the source address table. The control physical topology is used in the UC channel only)
	0	physical ring is broken
	1	physical ring is closed
10-9	—	(reserved)
8	—	Master valid (indicates if the master is processing data. In CP1 the slave detects the support of this function if this bit is set to 1 by the master)
	0	Master is not valid (The contents of device control C-DEV are invalid. Producer ready of all producer connections shall be set to 0)
	1	Master is valid (The contents of device control C-DEV are valid)
7-0	—	(reserved)

A.3.111 IDN S-0-1045 Device Status

A.3.111.1 Attributes

Table A.137 shows the possible attributes for this IDN.

Table A.137 – Attributes of IDN S-0-1045

Attribute	Value
Name	Device Status
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.111.2 Description

The device status (Table A.138) contains the status information (for example, topology status) which are set by the slave and evaluated by the master. The device status is not part of a connection.

The fastest reaction time to any event affecting device status except bits 11-10 (Status of inactive port) shall be within the slowest producer cycle time but at most 200 ms.

Table A.138 – Device status field

Bit number	Bit value	Description
15	—	Communication warning interface
	0	No warning
	1	Communication warning occurred (for example number of permitted MST losses has exceeded the half value of S-0-1003
14	—	Topology HS
	toggle	Initial value is 0 in every CP. Slave toggles, if the request of the master has been recognized, that means, the topology status may be updated after the toggle.
13-10	—	Topology status / Port status
	00-00	Fast-Forward on both ports (Diagnostic not available)
	01-00	Loopback with Forward of P-Telegrams (no link on inactive port --> no device connected)
	01-01	Loopback with Forward of P-Telegrams (LINK on inactive port --> device connected)
	01-10	Loopback with Forward of P-Telegrams (P LINK: P telegrams on inactive port --> Type 19 device connected)
	01-11	Loopback with Forward of P-Telegrams (S LINK: S telegrams on inactive port --> Type 19 device connected)
	10-00	Loopback with Forward of S-Telegrams (no link on inactive port --> no device connected)
	10-01	Loopback with Forward of S-Telegrams (LINK on inactive port --> device connected)
	10-10	Loopback with Forward of S-Telegrams

Bit number	Bit value	Description
		(P-LINK: P telegrams on inactive port --> Type 19 device connected)
	10-11	Loopback with Forward of S-Telegrams (S-LINK: S telegrams on inactive port --> Type 19 device connected)
	11-xx	store & forward or cut-through
	00-xx	Additional bit combinations:
	00-01	fast-forward on both ports (Diagnostic supported)
	00-10	fast-forward on both ports (P telegrams are missing)
	00-11	fast-forward on both ports (S telegrams are missing)
9	—	Error connection
	0	Error-free connection
	1	Error in the connection occurred (consumer recognized an error in a connection)
8	—	Slave valid (indicates if a slave is processing data)
	0	Slave not valid (Set to 0 when entering CP0. Modified during CPS. The contents of device status S-DEV are invalid. Producer ready of all producer connections shall be set to 0)
	1	Slave valid (CP > CP0. Modified during CPS the contents of device status (S-DEV) are valid)
7	—	Error (C1D), inclusive sub-device and resource errors
	0	No error
	1	Error (detailed information is shown in S-0-0390)
6	—	Warning (C2D), inclusive sub-device and resource warnings
	0	No Warning
	1	Warning (detailed information is shown in S-0-0390)
5	—	Procedure command change bit
	0	No change in procedure command acknowledgement
	1	Changing procedure command acknowledgment (procedure command is positive or negative acknowledged)
4	—	Sub-device level
	0	Operating level (OL) is active
	1	Parametrization level (PL) is active
3	—	(reserved)
2	—	(reserved)
1-0	—	(reserved)

A.3.112 IDN S-0-1047 Maximum Consumer Activation Time (t11)

A.3.112.1 Attributes

Table A.139 shows the possible attributes for this IDN.

Table A.139 – Attributes of IDN S-0-1047

Attribute	Value
Name	Maximum Consumer Activation Time (t11)
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	3
Write protection	always
Conversion factor	1
Scaling/resolution	0,001 μs
Unit	—

A.3.112.2 Description

The t11 specifies the slave-specific maximum time duration that it needs between collecting consumer data from the MDTs and ATs and activating it (for example command values) in the slave.

If S-0-1060 Connection capabilities are available, t11 covers the maximum value of S-0-1060.x.07 Maximum processing time for all consumer instances of S-0-1060.

A.3.113 IDN S-0-1048 Activate network settings

A.3.113.1 Attributes

Table A.140 shows the possible attributes for this IDN.

Table A.140 – Attributes of IDN S-0-1048

Attribute	Value
Name	Activate network settings
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.113.2 Description

This procedure command is used to activate requested IP-settings of the following IDN:

- S-0-1020.0.0 IP address
- S-0-1021.0.0 Subnet Mask
- S-0-1022.0.0 Gateway address

- S-0-1027.0.1

Active settings can be displayed within the following IDN:

- S-0-1020.0.1(optional)
- S-0-1021.0.1(optional)
- S-0-1022.0.1(optional)
- S-0-1027.0.2.

A.3.114 IDN S-0-1046 List of device addresses in device

A.3.114.1 Attributes

Table A.141 shows the possible attributes for this IDN.

Table A.141 – Attributes of IDN S-0-1046

Attribute	Value
Name	List of device addresses in device
Version	—
Length	2, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.114.2 Description

The multi-slave device stores the sub-device addresses of its slaves that participate in the communication (see Figure A.13). If only one slave exists in the device, this IDN may be absent.

0x0008	Current length = 8 octets
0x0008	Max. length = 8 octets
Sub-device address	List element 0 = slave index 0
Sub-device address	List element 1 = slave index 1
Sub-device address	List element 2 = slave index 2
Sub-device address	List element 3 = slave index 3

Figure A.13 – Structure of List of Sub-device addresses

A.3.115 IDN S-0-1050.x.1 Connection setup

A.3.115.1 Attributes

Table A.142 shows the possible attributes for this IDN.

Table A.142 – Attributes of IDN S-0-1050.x.1

Attribute	Value
Name	Connection setup
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.115.2 Description

This parameter configures connections (Table A.143).

Table A.143 – Connection setup

Bit number	Value	Description
15	—	Usage of connection (see note)
	0	Not used (slave shall not use this connection)
	1	Used (slave shall use this connection)
14	—	function within connection (see note)
	0	Consumer
	1	Producer
13-12	—	Source of connection configuration (see note)
	00	Master (configured by master)
	01	(reserved)
	10	External (not configured by master)
	11	(reserved)
11-6	—	(reserved)
5-4	—	type of configuration (see note)
	00	variable configuration of IDNs with S-0-1050.x.06
	01	configuration with connection length, see S-0-1050.x.05. S-0-1050.x.06 will not be considered in case of FSP I/O: The connection assignment is defined as follows: (C-CON - IO Control - S-0-1500.x.05) (C-CON - IO Status - S-0-1500.x.09) in case of FSP Drive: The connection assignment is defined as follows: (C-CON - Drive Control - S-0-1050.x.06) (C-CON - Drive Status - S-0-1050.x.06)
	10	standard telegram (see S-0-0015 of FSP Drive)
	11	reserved
3	—	mechanism of producing (for producers only, for consumers don't care)
	0	producer cycle synchronous
	1	asynchronous
2	—	reserved
1-0	—	mechanism of monitoring (for consumers only, for producers don't care)
	00	producer cycle synchronous operation (time defined by S-0-1050.x.10)
	01	asynchronous operation with watchdog (timeout for watchdog defined by product of S-0-1050.x.10 and S-0-1050.x.11)
	10	asynchronous operation without watchdog
	11	reserved

NOTE This bit is write protected if slave supports only SCP_FixCFG and additionally SCP-classes which contains S-0-1050.x.01.

A.3.116 IDN S-0-1050.x.2 Connection Number

A.3.116.1 Attributes

Table A.144 shows the possible attributes for this IDN.

Table A.144 – Attributes of IDN S-0-1050.x.2

Attribute	Value
Name	Connection Number
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	0
Max. input value	65 535
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.116.2 Description

The connection number is used to identify a connection. The producer and all consumers of the same connection shall have the same connection number.

A.3.117 IDN S-0-1050.x.3 Telegram assignment

A.3.117.1 Attributes

Table A.145 shows the possible attributes for this IDN.

Table A.145 – Attributes of IDN S-0-1050.x.3

Attribute	Value
Name	Telegram assignment
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.117.2 Description

The telegram assignment contains the

- telegram type (MDT or AT), the
- telegram number and the
- telegram offset of connection control

for this connection. The telegram type and telegram number define the Type 19 telegram of the connection. The telegram offset shall start at the end of the Type 19 header (S3H) and defines the position of the Connection control (C-CON) of this connection. The master determines the telegram assignment for each connection and shall transmit it to the related slaves in CP2 (Table A.146). The telegram assignment shall be an even number.

Table A.146 – Structure of telegram assignment

Bit number	Value	Description
15-14	—	reserved
13-12	—	Telegram number
	00	MDT0 or AT0
	01	MDT1 or AT1
	10	MDT2 or AT2
	11	MDT3 or AT3
11	—	Telegram type
	0	AT
	1	MDT
10-0	—	Telegram offset of connection control (in octets)
	0 … 1492	Telegram offset in MDT or AT (shall be an even number)

A.3.118 IDN S-0-1050.x.4 Max. Length of Connection

A.3.118.1 Attributes

Table A.147 shows the possible attributes for this IDN.

Table A.147 – Attributes of IDN S-0-1050.x.4

Attribute	Value
Name	Max. Length of Connection
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	2
Max. input value	product specific or S-0-1060.x.04
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.118.2 Description

This parameter defines the maximum length of this connection. The 2 octets for the connection control (C-CON) are part of this length. If the slave shows a length of n octets this length contains 2 octets C-CON and n-2 octets data (see Figure A.14).

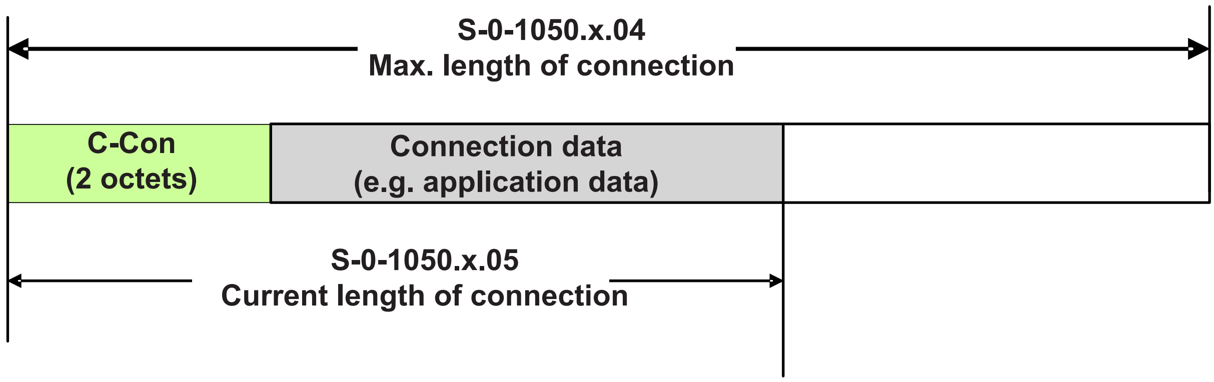

Figure A.14 – Definition of connection length

A.3.119 IDN S-0-1050.x.5 Current length of connection

A.3.119.1 Attributes

Table A.148 shows the possible attributes for this IDN.

Table A.148 – Attributes of IDN S-0-1050.x.5

Attribute	Value
Name	Current length of connection
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.119.2 Description

This parameter defines the current length of this connection and shall be an even number of octets. The 2 octets for the connection control (C-CON) are part of this length. If the slave shows a length of n octets this length contains 2 octets C-CON and n-2 octets data (see Figure A.14). This parameter shall be updated by the slave if configuration parameters are changed.

A.3.120 IDN S-0-1050.x.6 Configuration List

A.3.120.1 Attributes

Table A.149 shows the possible attributes for this IDN.

Table A.149 – Attributes of IDN S-0-1050.x.6

Attribute	Value
Name	Configuration List
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.120.2 Description

If the connection data is configured with IDNs (type of connection, bit 5-4 = 00, in S-0-1050.x.01) this parameter contains the list of IDNs within this connection. The sequence of the IDNs in this parameter and the sequence of the corresponding operation data in the connection is identical.

A.3.121 IDN S-0-1050.x.7 Assigned connection capability

A.3.121.1 Attributes

Table A.150 shows the possible attributes for this IDN.

Table A.150 – Attributes of IDN S-0-1050.x.7

Attribute	Value
Name	Assigned connection capability
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	0
Max. input value	255 and 0xFFFF
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.121.2 Description

This parameter shows the configured connection capability of this connection. Therefore the parameter contains the structure instance of the desired connection capability (S-0-1060). If this parameter is read only the slave has a fixed connection structure.

The default value shall 0xFFFF (connection capabilities S-0-1060) is not used.

A.3.122 IDN S-0-1050.x.8 Connection Control

A.3.122.1 Attributes

Table A.151 shows the possible attributes for this IDN.

Table A.151 – Attributes of IDN S-0-1050.x.8

Attribute	Value
Name	Connection Control
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.122.2 Description

This parameter contains the image of the control word C-CON of this connection (see Table 44).

A.3.123 IDN S-0-1050.x.10 Producer cycle time

A.3.123.1 Attributes

Table A.152 shows the possible attributes for this IDN.

Table A.152 – Attributes of IDN S-0-1050.x.10

Attribute	Value
Name	Producer cycle time
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	31 250 (31,250 µs)
Max. input value	product specific
Positions after decimal point	3
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

A.3.123.2 Description

The Producer cycle time should be an integer multiple of the communication cycle time:

$$tPcyc = tScyc \cdot n \quad \forall n \in N$$

Minimum and maximum input values are mandatory.

A.3.124 IDN S-0-1050.x.11 Allowed Data Losses

A.3.124.1 Attributes

Table A.153 shows the possible attributes for this IDN.

Table A.153 – Attributes of IDN S-0-1050.x.11

Attribute	Value
Name	Allowed Data Losses
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.124.2 Description

This parameter indicates the maximum amount of consecutive losses of producer data, before a connection is broken. If this connection is broken the consumer shall not process data anymore and sets the connection error in the device status. The default value = 1.

A.3.125 IDN S-0-1050.x.12 Error Counter Data Losses

A.3.125.1 Attributes

Table A.154 shows the possible attributes for this IDN.

Table A.154 – Attributes of IDN S-0-1050.x.12

Attribute	Value
Name	Error Counter Data Losses
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.125.2 Description

This parameter counts the amount of lost producer data. Therefore, the consumer compares the C.CON.counter with its expectations. In case of an erroneous difference this error counter is incremented by 1. This counter shall be reset with the positive edge of the Producer ready in the Connection control or additionally if the master switches from CP2 to CP3. This counter does not have any overrun and ends with 65 535.

A.3.126 IDN S-0-1050.x.20 IDN Allocation of real-time bit

A.3.126.1 Attributes

Table A.155 shows the possible attributes for this IDN.

Table A.155 – Attributes of IDN S-0-1050.x.20

Attribute	Value
Name	IDN Allocation of real-time bit
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.126.2 Description

In order to assign signals to the real-time bits (see S-0-0398 IDN list of configurable real-time bits as producer and S-0-0399 IDN list of configurable real-time bits as consumer), the IDN of the signal is written to this parameter. After the allocation of the IDN and the bit number (see S-0-1050.x.21 Bit allocation of real-time bit), the assigned signal is copied in the corresponding real-time bit. This parameter contains maximum 2 list elements.

Real-time bit 1 and 2 are present in the S-0-1050.x.08 Connection Control (C-CON).

- List element 0 corresponds to real-time bit 1: IDN of assigned signal
- List element 1 corresponds to real-time bit 2: IDN of assigned signal

see also S-0-1050.x.21 Bit allocation of real-time bit.

A.3.127 IDN S-0-1050.x.21 IDN Allocation of real-time bit

A.3.127.1 Attributes

Table A.156 shows the possible attributes for this IDN.

Table A.156 – Attributes of IDN S-0-1050.x.21

Attribute	Value
Name	Bit allocation of real-time bit
Version	—
Length	2, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.127.2 Description

This parameter contains the bit number of the operation data assigned in the S-0-1050.x.20. The signal assigned by an IDN (S-0-1050.x.20) and a bit number (S-0-1050.x.21) is copied into the corresponding real-time bit. This list contains a maximum of 2 entries.

- List element 0 corresponds to real-time bit 1
- List element 1 corresponds to real-time bit 2

A.3.128 IDN S-0-1051 Image of connection setups

A.3.128.1 Attributes

Table A.157 shows the possible attributes for this IDN.

Table A.157 – Attributes of IDN S-0-1051

Attribute	Value
Name	Image of connection setups
Version	—
Length	2, variable
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.128.2 Description

This IDN shows the actual state of all the connections of the slave, corresponding to S-0-1050.x.1.

The quantity of list elements shows the maximum number of connections of this slave.

A.3.129 IDN S-0-1060.x.1 Default configuration

A.3.129.1 Attributes

Table A.158 shows the possible attributes for this IDN.

Table A.158 – Attributes of IDN S-0-1060.x.01

Attribute	Value
Name	Default configuration
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.129.2 Description

The structure of this parameter is identical with S-0-1050.x.01 and contains the default settings of the connection capabilities (for example consumer or producer).

A.3.130 IDN S-0-1060.x.2 Configuration mask

A.3.130.1 Attributes

Table A.159 shows the possible attributes for this IDN.

Table A.159 – Attributes of IDN S-0-1060.x.02

Attribute	Value
Name	Configuration mask
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.130.2 Description

The structure of this parameter is identical with S-0-1050.x.01. The changeable configuration bits are set to 1. The not changeable configuration bits are set to 0.

A.3.131 IDN S-0-1060.x.3 Maximum quantity of this connection capability

A.3.131.1 Attributes

Table A.160 shows the possible attributes for this IDN.

Table A.160 – Attributes of IDN S-0-1060.x.03

Attribute	Value
Name	Maximum quantity of this connection capability
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	1
Max. input value	255
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.131.2 Description

This parameter limits the number of connections (S-0-1050.x.07) within this connection capability.

A.3.132 IDN S-0-1060.x.4 Max. connection length of connection capability

A.3.132.1 Attributes

Table A.161 shows the possible attributes for this IDN.

Table A.161 – Attributes of IDN S-0-1060.x.04

Attribute	Value
Name	Max. connection length of connection capability
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	2
Max. input value	Product specific
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.132.2 Description

This parameter defines the maximum connection length of this connection capability. The 2 octets of the connection control (C-CON) are part of this length. If the slave shows a length of n octets this length contains 2 octets C-CON and n-2 octets data.

A.3.133 IDN S-0-1060.x.6 Configurable IDNs of connection capability

A.3.133.1 Attributes

Table A.162 shows the possible attributes for this IDN.

Table A.162 – Attributes of IDN S-0-1060.x.06

Attribute	Value
Name	Configurable IDNs of connection capability
Version	—
Length	4, variable (list sorted by IDN)
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.133.2 Description

This parameter contains the configurable IDNs of the connection capability which can be configured in the configuration list of the connection (S-0-1050.x.06).

A.3.134 IDN S-0-1060.x.7 Maximum processing time

A.3.134.1 Attributes

Table A.163 shows the possible attributes for this IDN.

Table A.163 – Attributes of IDN S-0-1060.x.07

Attribute	Value
Name	Maximum processing time
Version	—
Length	4
Display Format	Decimal
Min. input value	—
Max. input value	—
Positions after decimal point	3
Write protection	Always
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	—

A.3.134.2 Description

With this time the sub-device defines how long the data processing of the connection data takes as producer or consumer (see Figure A.15).

- producer: the generation of the producer data needs the maximum processing time (tmp-P) after the synchronization time of the producer cycle (T4pc).
- consumer: the processing of the consumer data needs the maximum processing time (tmp-C) after the end of the corresponding telegram (MDT or AT).

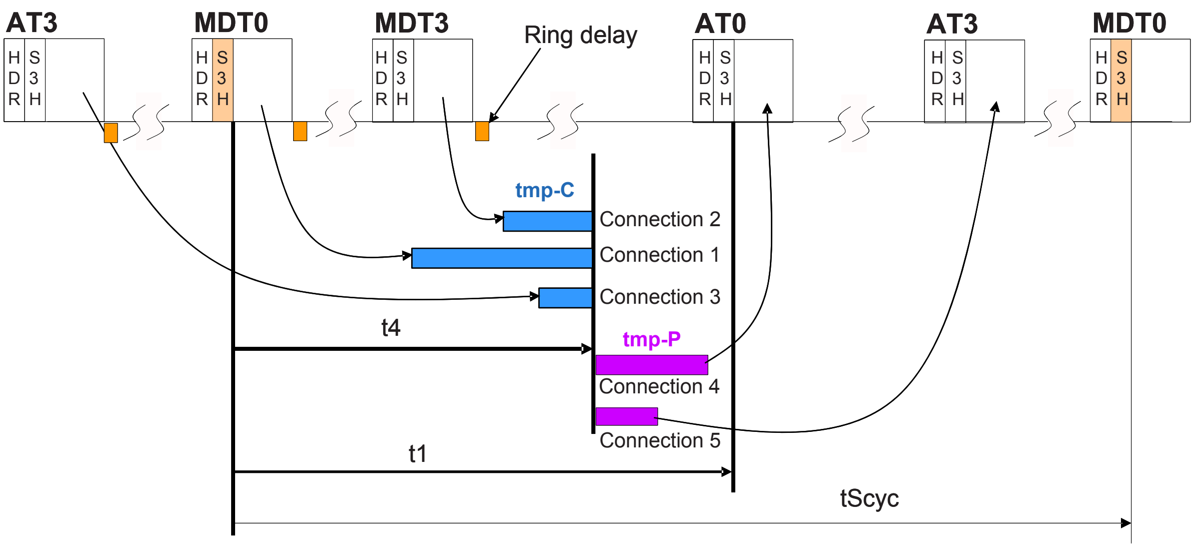

Figure A.15 – Synchronization with ring

A.3.135 IDN S-0-1060.x.10 Minimum producer cycle time

A.3.135.1 Attributes

Table A.164 shows the possible attributes for this IDN.

Table A.164 – Attributes of IDN S-0-1060.x.10

Attribute	Value
Name	Minimum producer cycle time
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	31 250 (31,250 µs)
Max. input value	—
Positions after decimal point	3
Write protection	Always
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	—

A.3.135.2 Description

This parameter contains the minimal producer cycle time which is supported for this connection capability.

A.3.136 IDN S-0-1061 Maximum TSref-Counter

A.3.136.1 Attributes

Table A.165 shows the possible attributes for this IDN.

Table A.165 – Attributes of IDN S-0-1061

Attribute	Value
Name	Maximum TSref-Counter
Version	—
Length	2
Display Format	Decimal
Min. input value	0
Max. input value	16 383
Positions after decimal point	0
Write protection	CP3, CP4
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.136.2 Description

This parameter contains the maximum value of the TSref-counter in the extended field of MDT0. This parameter is necessary if the application runs with different producer cycle times. The master shall support the following sequence:

- determination of the least common multiple (LCM) of all producer cycle times (tPcyc) which shall be synchronized for the application,
- the result is divided by the communication cycle time (tScyc) to get a LCM number based on tScyc,
- the LCM number shall be decrement by 1 and written in this parameter.

If this parameter is 0, then each of the producer cycle times are equal to the communication cycle time (each tPcyc = tScyc).

The TSref-counter in the extended field of MDT0 is 14 bit long, therefore the maximum value of this parameter is limited to the value of 16 383 (214 - 1). This parameter is used as a modulo value for TSref-counter.

A.3.137 IDN S-0-1080.x.02 Producer RTB list container

A.3.137.1 Attributes

Table A.166 shows the possible attributes for this IDN.

Table A.166 – Attributes of IDN S-0-1080.x.02

Attribute	Value
Name	Producer RTB list container
Version	—
Length	1, variable
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.137.2 Description

Signals can be transmitted in real-time from the producer to the consumer by means of the Producer RTB list container. For this purpose, the Producer RTB list container needs to be integrated in one producer connection.

Bits in the Producer RTB list container are definable by means of the configuration lists of the consumer RTB list container (see S-0-1080.x.03 and S-0-1080.x.04).

A.3.138 IDN S-0-1080.x.03 IDN allocation of producer RTB list container

A.3.138.1 Attributes

Table A.167 shows the possible attributes for this IDN.

Table A.167 – Attributes of IDN S-0-1080.x.03

Attribute	Value
Name	IDN allocation of producer RTB list container
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.138.2 Description

Bits in the Producer RTB list container (S-0-1080.x.02) are definable by means of the configuration list of the Producer RTB list container represented in this IDN. The sequence of the IDNs in the configuration list determines the bit numbering scheme in the Producer RTB list container. The first IDN of the configuration list with the first bit number of S-0-1080.x.04 defines bit 0 of the Producer RTB list container.

A.3.139 IDN S-0-1080.x.04 Bit allocation of producer RTB list container

A.3.139.1 Attributes

Table A.168 shows the possible attributes for this IDN.

Table A.168 – Attributes of IDN S-0-1080.x.04

Attribute	Value
Name	Bit allocation of producer RTB list container
Version	—
Length	2, variable
Display Format	Unsigned decimal (bit number)
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.139.2 Description

In this configuration list the bit numbers of the operation data are programmed, which are contained in the Producer RTB list container (S-0-1080.x.02). The sequence of the bit numbers in the configuration list sets the numerical order in the Producer RTB list container. The first bit number of the configuration list with the first IDN of S-0-1080.x.03 defines bit 0 of the Producer RTB list container.

A.3.140 IDN S-0-1081.x.02 Consumer RTB list container

A.3.140.1 Attributes

Table A.169 shows the possible attributes for this IDN.

Table A.169 – Attributes of IDN S-0-1081.x.02

Attribute	Value
Name	Consumer RTB list container
Version	—
Length	1, variable
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.140.2 Description

Signals can be transmitted in real-time from the producer to the consumer by means of the consumer RTB list container. For this purpose, the consumer RTB list container needs to be integrated in one consumer connection.

Bits in the consumer RTB list container are definable by means of the configuration lists of the consumer RTB list container (see S-0-1081.x.03 and S-0-1081.x.04).

A.3.141 IDN S-0-1081.x.03 IDN allocation of consumer RTB list container

A.3.141.1 Attributes

Table A.170 shows the possible attributes for this IDN.

Table A.170 – Attributes of IDN S-0-1081.x.03

Attribute	Value
Name	IDN allocation of consumer RTB list container
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.141.2 Description

Bits in the consumer RTB list container (S-0-1081.x.02) are definable by means of the configuration list of the consumer RTB list container represented in this IDN. The sequence of the IDNs in the configuration list determines the bit numbering scheme in the consumer RTB list container. The first IDN of the configuration list with the first bit number of S-0-1081.x.04 defines bit 0 of the Consumer RTB list container.

A.3.142 IDN S-0-1081.x.04 Bit allocation of consumer RTB list container

A.3.142.1 Attributes

Table A.171 shows the possible attributes for this IDN.

Table A.171 – Attributes of IDN S-0-1081.x.04

Attribute	Value
Name	Bit allocation of consumer RTB list container
Version	—
Length	2, variable
Display Format	Unsigned decimal (bit number)
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.142.2 Description

In this configuration list the bit numbers of the operation data are programmed, which are contained in the Producer RTB list container (S-0-1080.x.02). The sequence of the bit numbers in the configuration list sets the numerical order in the Producer RTB list container. The first bit number of the configuration list with the first IDN of S-0-1080.x.03 defines bit 0 of the Producer RTB list container.

A.3.143 IDN S-0-1099.0.1 Test-IDN Control for SCP Conformity Purpose

A.3.143.1 Attributes

Table A.172 shows the possible attributes for this IDN.

Table A.172 – Attributes of IDN S-0-1099.0.1

Attribute	Value
Name	Test-IDN Control for SCP Conformity Purpose
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.143.2 Description

This IDN controls (see Table A.173) the behavior of S-0-1099.0.2 Test-IDN Container for SCP Conformity purpose and other Type 19 functionality. It is used by the Type 19 certification to check if the slave processes the data correctly. Bits 14-8 define which test-group is activated. Test-group specific tests can be selected through bits 0-7.

Table A.173 – Structure of Test-IDN control

Bit number	Value	Description
15	—	Control type
	0	specification specific (Test-groups and test-cases follow the specification)
	1	manufacturer specific (Test-groups and test-cases are defined by the manufacturer)
14-8	—	Test-group - Describes which test-group (Bit 7-0) is used
	0	don't care (Test-IDN is not active)
	1	Real-time data
	2	SMP
7-4	—	Test-group - Real-time data
	0	don't care (Test-IDN is not active)
	1	Increment value of S-0-1099.0.2 by 1
	2	Stop toggling Con-Counter of the connection where S-0-1099.0.2 is defined cyclically for value defined in S-0-1099.0.2 cycles
3-0	—	Test-group – SMP
	0	Don't care (Test-IDN is not active)
	1	Receive SMP-message and send back SMP-message over configured SMP-container with Session-ID+1 and priority+1 as soon as SMP-message has been received completely.

A.3.144 IDN S-0-1099.0.2 Test-IDN Container for SCP Conformity purpose

A.3.144.1 Attributes

Table A.174 shows the possible attributes for this IDN.

Table A.174 – Attributes of IDN S-0-1099.0.2

Attribute	Value
Name	Test-IDN Container for SCP Conformity purpose
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.144.2 Description

This IDN is used by the Type 19 certification to check if the slave processes the data correctly. For that this parameter shall be handled like any other parameter, except this parameter can be configured in a produced and a consumed connection for one slave at the same time. Reading the operation data of this IDN shall return the value configured through S-0-1099.0.1 (Test-IDN Control for SCP Conformity purpose).

A.3.145 IDN S-0-1100.0.1 Diagnostic counter sent SMP fragments

A.3.145.1 Attributes

Table A.175 shows the possible attributes for this IDN.

Table A.175 – Attributes of IDN S-0-1100.0.1

Attribute	Value
Name	Diagnostic counter sent SMP fragments
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.145.2 Description

This parameter displays the number of SMP fragments transmitted via the SMP stack since switching to CP4. The counter is set to 0 before CP4 is activated. This counter has an auto-rollover at 2^{32}-1 to 0.

A.3.146 IDN S-0-1100.0.2 Diagnostic counter received SMP fragments

A.3.146.1 Attributes

Table A.176 shows the possible attributes for this IDN.

Table A.176 – Attributes of IDN S-0-1100.0.2

Attribute	Value
Name	Diagnostic counter received SMP fragments
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.146.2 Description

This parameter displays the number of SMP fragments received by the SMP stack since switching to CP4. This counter is set to 0 before CP4 is activated. This counter has an auto-rollover at 2^{32}-1 to 0.

A.3.147 IDN S-0-1100.0.3 Diagnostic counter discarded SMP fragments

A.3.147.1 Attributes

Table A.177 shows the possible attributes for this IDN.

Table A.177 – Attributes of IDN S-0-1100.0.3

Attribute	Value
Name	Diagnostic counter discarded SMP fragments
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.147.2 Description

This parameter displays the number of SMP fragments received by SMP stack that were discarded because its header did not match the receiver's expectations since switching to CP4. This counter is set to 0 before CP4 is activated. This counter has an auto-rollover at 2^{32}-1 to 0.

Reasons for this include:

- invalid session ID,
- wrong sequence counter,
- incorrect sequence of the FOS/LOS bits.

A.3.148 IDN S-0-1101.x.1 SMP Container Data

A.3.148.1 Attributes

Table A.178 shows the possible attributes for this IDN.

Table A.178 – Attributes of IDN S-0-1101.x.1

Attribute	Value
Name	SMP Container Data
Version	—
Length	2, variable
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.148.2 Description

This parameter contains the actual data transmitted via a SMP container.

A.3.149 IDN S-0-1101.x.2 List of session identifiers

A.3.149.1 Attributes

Table A.179 shows the possible attributes for this IDN.

Table A.179 – Attributes of IDN S-0-1101.x.2

Attribute	Value
Name	List of session identifiers
Version	—
Length	2, variable
Display Format	unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.149.2 Description

This parameter contains the list of all session identifiers currently set up for a SMP container. Each list entry corresponds to the list entry with the same index in S-0-1101.x.03 and defines the identifier for this session.

The lists in S-0-1101.x.02 and S-0-1101.x.03 shall have the same current length.

A.3.150 IDN S-0-1101.x.3 List of session priorities

A.3.150.1 Attributes

Table A.180 shows the possible attributes for this IDN.

Table A.180 – Attributes of IDN S-0-1101.x.3

Attribute	Value
Name	List of session priorities
Version	—
Length	2, variable
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.150.2 Description

This parameter contains a list of priority values for the sessions in this SMP container. Each list entry corresponds to the list entry with the same index in S-0-1101.x.02 and defines the priority for this session. The highest priority is 0, the lowest priority is 3.

The lists in S-0-1101.x.02 and S-0-1101.x.03 shall have the same current length.

A.3.151 IDN S-0-1150.x.01 OVS Control (C-OVS)

A.3.151.1 Attributes

Table A.181 shows the possible attributes for this IDN.

Table A.181 – Attributes of IDN S-0-1150.x.01

Attribute	Value
Name	OVS Control (C-OVS)
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.151.2 Description

The IDN contains a bit field with which the user can trigger various actions of the oversampling domain (Table A.182). The OVS control (C-OVS) is not part of the container. It may be mapped to the connection as its own IDN.

Table A.182 – OVS Control structure

Bit number	Value	Description
15	—	Quit error
	0	no action
	1	quit OVS error
14-9	—	(reserved)
8	—	Flow control (activates OVS)
	0	Run (OVS is active)
	1	Stop (OVS is passive)
7-0	—	(reserved)

A.3.152 IDN S-0-1150.x.02 OVS Status (S-OVS)

A.3.152.1 Attributes

Table A.183 shows the possible attributes for this IDN.

Table A.183 – Attributes of IDN S-0-1150.x.02

Attribute	Value
Name	OVS Status (S-OVS)
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.152.2 Description

The IDN contains a bit field with which the user is informed about various states of the oversampling domain (see Table A.184). The OVS status word (S-OVS) is not part of the container. It may be mapped to the connection as its own IDN.

Table A.184 – OVS Status structure

Bit number	Value	Description
15	—	Error
	0	the OVS state machine indicates an error
	1	no error occurs
14-9	—	(reserved)
8	—	OVS stop (indicates state of the OVS state machine)
	0	Run (OVS is active)
	1	Stop (OVS is passive)
7-0	—	(reserved)

A.3.153 IDN S-0-1150.x.03 OVS Container

A.3.153.1 Attributes

Table A.185 shows the possible attributes for this IDN.

Table A.185 – Attributes of IDN S-0-1150.x.03

Attribute	Value
Name	OVS Container
Version	—
Length	1, variable
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Conditional
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.153.2 Description

The operation data of this IDN contains the OVS data (see Table A.186 and Figure A.16).

The size of the container depends on S-0-1150.x.10 Number of Samples and the number of IDNs, which are configured in S-0-1150.x.06 Configuration List OVS - IDNs.

The order of the signals within this container follows the order in S-0-1150.x.06 Configuration List OVS - IDNs and respectively S-0-1150.x.07 Configuration List OVS - Offset and S-0-1150.x.08 Configuration List OVS - Length.

The data of each signal is stored in a bit-packed way. The data of the following signals starts at the next byte boundary (byte aligned).

If the slave is the producer of the container data, the container is always write protected.

If the slave is the consumer of the container data, the container is writable.

Table A.186 – Configuration example

S-0-1150.x.06 (IDN)	S-0-1150.x.07 (Offset)	S-0-1150.x.08 (Length)	Explanation
S-0-0135	0	0	16 bits, starting with bit 0 of IDN "S-0-0135" are configured.
S-0-1505.4.9	0	4	4 bits, starting with bit 0 of IDN "S-0-1505.4.9" are configured.
P-0-1213.7.5	2	3	3 bits, starting with bit 2 of IDN "P-0-1213.7.5" are configured.

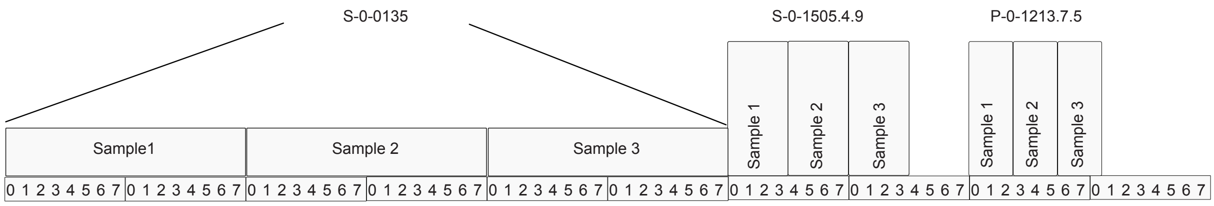

Figure A.16 – Configuration example

A.3.154 IDN S-0-1150.x.04 Sample time

A.3.154.1 Attributes

Table A.187 shows the possible attributes for this IDN.

Table A.187 – Attributes of IDN S-0-1150.x.04

Attribute	Value
Name	Sample time
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.154.2 Description

This IDN contains the sample time; that is the time distance between two signals. The selection is made according to S-0-1151.x.04 Minimum sample time.

The relationship between S-0-1150.x.04 Sample time and S-0-1150.x.10 Number of Samples is the following:

- If S-0-1150.x.10 Number of Samples is written, the operation data of S-0-1150.x.04 Sample time is calculated by the slave according to: S-0-1150.x.04 Sample time = S-0-1050.x.10 Producer Cycle Time / S-0-1150.x.10 Number of Samples

- If S-0-1150.x.04 Sample time is written, the operation data of S-0-1150.x.10 Number of Samples is calculated by the slave according to: S-0-1150.x.10 Number of Samples = S-0-1050.x.10 Producer Cycle Time / S-0-1150.x.04 Sample time. If the result is not an integer in nanoseconds it is rounded.
- If the corresponding OVS container has not been configured in a connection (for example: for OVS via SVC), no S-0-1050.x.10 is available, which can be used for the calculation of S-0-1150.x.04. In this case the configuration of the oversampling machine has to be done via S-0-1150.x.04 Sample time.

A.3.155 IDN S-0-1150.x.05 Phase shift

A.3.155.1 Attributes

Table A.188 shows the possible attributes for this IDN.

Table A.188 – Attributes of IDN S-0-1150.x.05

Attribute	Value
Name	Phase shift
Version	—
Length	4
Display Format	Signed decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.155.2 Description

The IDN contains the value of the phase shifting for recording/output.

Only integer multiples of the minimum sample time are permitted as entry.

With this entry the recording can be shifted back and forth. The producer cycle time determines the maximum phase shifting.

A.3.156 IDN S-0-1150.x.06 Configuration List OVS - IDNs

A.3.156.1 Attributes

Table A.189 shows the possible attributes for this IDN.

Table A.189 – Attributes of IDN S-0-1150.x.06

Attribute	Value
Name	Configuration List OVS - IDNs
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.156.2 Description

The list contains entries of the type IDN.

Every entry identifies a signal that is to be sampled or output by this oversampling machine.

The user shall create the list from the data of the S-0-1151.x.06 Configurable IDNs of OVS capability list.

A.3.157 IDN S-0-1150.x.07 Configuration List OVS - Offset

A.3.157.1 Attributes

Table A.190 shows the possible attributes for this IDN.

Table A.190 – Attributes of IDN S-0-1150.x.07

Attribute	Value
Name	Configuration List OVS - Offset
Version	—
Length	2, variable
Display Format	Unsigned integer
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.157.2 Description

This IDN contains a list of bit offsets, which can be used together with S-0-1150.x.08 Configuration List OVS - Length to address particular bit signals within the operation data of the corresponding IDN in S-0-1150.x.06 (Configuration list OVS).

A.3.158 IDN S-0-1150.x.08 Configuration List OVS - Length

A.3.158.1 Attributes

Table A.191 shows the possible attributes for this IDN.

Table A.191 – Attributes of IDN S-0-1150.x.08

Attribute	Value
Name	Configuration List OVS - Length
Version	—
Length	2, variable
Display Format	Unsigned integer
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.158.2 Description

This IDN contains a list of bit lengths, which can be used together with S-0-1150.x.07 Configuration List OVS - Offset to address particular bit signals within the operation data of the corresponding IDN in S-0-1150.x.06 (Configuration list OVS).

A.3.159 IDN S-0-1150.x.09 Assigned Oversampling Capability

A.3.159.1 Attributes

Table A.192 shows the possible attributes for this IDN.

Table A.192 – Attributes of IDN S-0-1150.x.09

Attribute	Value
Name	Assigned Oversampling Capability
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	0
Max. input value	255
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.159.2 Description

This parameter shows the configured OVS capability. Therefore the parameter contains the structure instance of the desired S-0-1151 OVS Capabilities.

If the slave has a static oversampling capability, this parameter shall be read only.

A.3.160 IDN S-0-1150.x.10 Number of Samples

A.3.160.1 Attributes

Table A.193 shows the possible attributes for this IDN.

Table A.193 – Attributes of IDN S-0-1150.x.10

Attribute	Value
Name	Number of Samples
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	1
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.160.2 Description

This IDN contains the number of values to be sampled.

The value of the operation data has to be less or equal than S-0-1151.x.01 Maximum number of samples.

NOTE The relationship between S-0-1150.x.04 Sample time and S-0-1150.x.10 Number of Samples is described in FG Oversampling.

A.3.161 IDN S-0-1151.x.01 Maximum number of samples

A.3.161.1 Attributes

Table A.194 shows the possible attributes for this IDN.

Table A.194 – Attributes of IDN S-0-1151.x.01

Attribute	Value
Name	Maximum number of samples
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.161.2 Description

The IDN contains the maximum number of sampled/output values.

The operation data of this IDN is a synonym for the available memory capacity and should be seen as a worst case value. This means, that there might be configurations of S-0-1150.x.06 Configuration List OVS - IDNs, which do not allow the maximum number of samples, as specified in this IDN.

A.3.162 IDN S-0-1151.x.02 Internal resolution

A.3.162.1 Attributes

Table A.195 shows the possible attributes for this IDN.

Table A.195 – Attributes of IDN S-0-1151.x.02

Attribute	Value
Name	Internal resolution
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	ns

A.3.162.2 Description

The IDN contains the internal resolution of signals that are to be sampled/output. This value will typically deviate from S-0-1151.x.04 Minimum sample time, because the acceptance/transfer, storage and management takes more time than permitted by the time basis generated in the hardware.

A.3.163 IDN S-0-1151.x.03 Maximum quantity of this oversampling capability

A.3.163.1 Attributes

Table A.196 shows the possible attributes for this IDN.

Table A.196 – Attributes of IDN S-0-1151.x.03

Attribute	Value
Name	Maximum quantity of this oversampling capability
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	1
Max. input value	255
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.163.2 Description

The IDN contains the maximum number of signals that can be sampled/output at the same time within a domain. This data item is a synonym for the available resources of the oversampling hardware. That means that several signals can be stored when the signals of the domain to be sampled/output (S-0-1151.x.06 Configurable IDNs of OVS capability) are selected well and the memory is managed efficiently.

A.3.164 IDN S-0-1151.x.04 Minimum sample time

A.3.164.1 Attributes

Table A.197 shows the possible attributes for this IDN.

Table A.197 – Attributes of IDN S-0-1151.x.04

Attribute	Value
Name	Minimum sample time
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	ns

A.3.164.2 Description

This IDN contains the minimum sample time, which is the smallest possible time distance between two sampled/output signal values.

The actual sampled/output time is defined in S-0-1150.x.04 Sample time.

A.3.165 IDN S-0-1151.x.06 Configurable IDNs of OVS capability

A.3.165.1 Attributes

Table A.198 shows the possible attributes for this IDN.

Table A.198 – Attributes of IDN S-0-1151.x.06

Attribute	Value
Name	Configurable IDNs of OVS capability
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.165.2 Description

The list contains entries of the type IDN. Every entry identifies a signal that can be sampled/output with this oversampling machine.

A.3.166 IDN S-0-1151.x.07 Configurable IDNs of OVS Capability - Offset

A.3.166.1 Attributes

Table A.199 shows the possible attributes for this IDN.

Table A.199 – Attributes of IDN S-0-1151.x.07

Attribute	Value
Name	Configurable IDNs of OVS Capability - Offset
Version	—
Length	2, variable
Display Format	Unsigned integer
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.166.2 Description

The list contains bit offsets of the corresponding IDNs listed in S-0-1151.x.06 Configurable IDNs of OVS capability. Every entry identifies a signal that can be sampled/output with this oversampling machine.

A.3.167 IDN S-0-1151.x.08 Configurable IDNs of OVS Capability - Length

A.3.167.1 Attributes

Table A.200 shows the possible attributes for this IDN.

Table A.200 – Attributes of IDN S-0-1151.x.08

Attribute	Value
Name	Configurable IDNs of OVS Capability - Length
Version	—
Length	2, variable
Display Format	Unsigned integer
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.167.2 Description

The list contains bit lengths of the corresponding IDNs listed in S-0-1151.x.06 Configurable IDNs of OVS capability. Every entry identifies a signal that can be sampled/output with this oversampling machine.

If "0" is entered as length, the corresponding entry in S-0-1151.x.07 Configurable IDNs of OVS Capability - Offset is ignored and the entire operation data of the IDN defined in S-0-1151.x.06 is selectable for sampling/output.

A.3.168 IDN S-0-1153 Amount of OVS Domains

A.3.168.1 Attributes

Table A.201 shows the possible attributes for this IDN.

Table A.201 – Attributes of IDN S-0-1151.x.08

Attribute	Value
Name	Amount of OVS Domains
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

A.3.168.2 Description

This IDN contains the amount of oversampling domains provided by the slave.

Annex B
(normative)

SCP– Classification

B.1 General concept of profiling

Type 19 offers two different views upon the grouping of IDNs. One view is the point of view of the specification, the other one is the point of view of an application.

Figure B.1 shows the technical view. Each IDN is existent in a function group. A function group is a functional grouping of IDNs. Each function group is existent in a so called profile area. This profile area may either be SCP, GDP or a FSP (for example: FSP_Drive, FSP_IO).

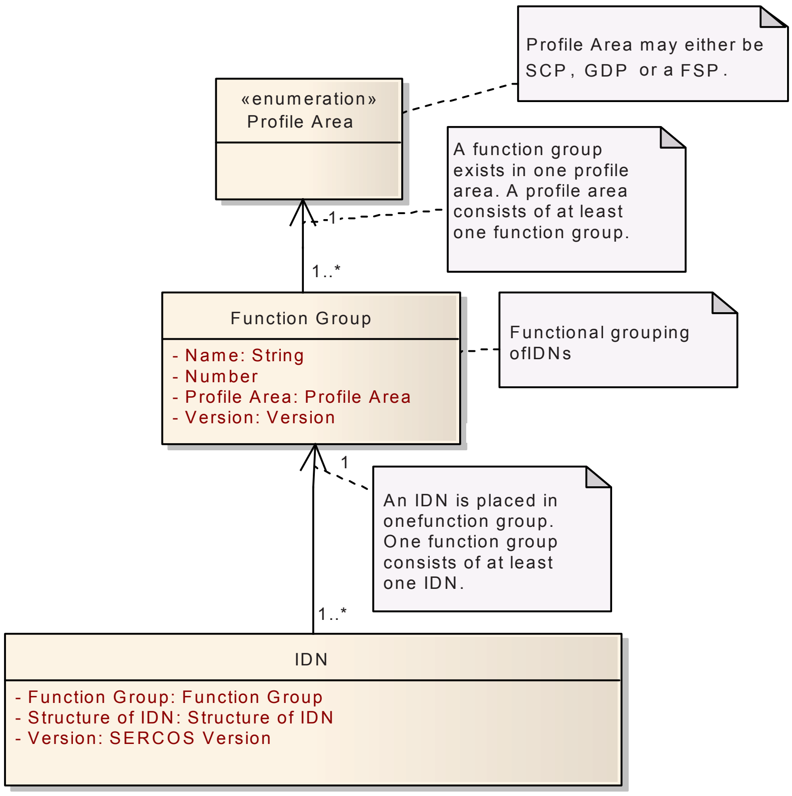

Figure B.1 – Technical Profiling in Type 19

An application expects certain functionalities in a device. So another view upon the profiling is introduced. This is called application profiling. Application profiling uses classes to group functionality described in the technical specification to groups that can be implemented in devices. This is to help device manufactures of slave devices to decide which functionality may be grouped in a reasonable way. On the other hand this view also helps master manufactures, since this classing provides an easy view upon the slave device.

For the profile areas SCP and GDP an IDN exists, which shows the classes implemented in a device:

a) SCP: IDN S-0-1000 once per slave

b) GDP: IDN S-0-1301 once per sub-device

For modular structured devices, the following IDN for the FSP profile area exists:

a) FSP: IDN S-0-1302.x.2 one for each resource in the sub-device

B.2 Function Groups related to the SCP

B.2.1 FG SCP Identification

The function group FG SCP Classification groups all IDNs that are related to the classification of a slave on SCP level.

This function group includes the following IDNs:

- IDN S-0-1000.0.1 Active SCP Classes
- IDN S-0-1000 SCP Type & Version

B.2.2 FG Timing

The function group FG Timing groups all IDNs that are related to the timing. This function group includes the following IDNs:

- IDN S-0-1002 Communication Cycle time (tScyc)
- IDN S-0-1005 Minimum feedback processing time (t5)
- IDN S-0-1006 AT0 transmission starting time (t1)
- IDN S-0-1008 Command value valid time (t3)
- IDN S-0-1023 SYNC jitter
- IDN S-0-1036 Inter Frame Gap
- IDN S-0-1037 Slave Jitter
- IDN S-0-1041 AT Command value valid time (t9)
- IDN S-0-1047 Maximum Consumer Activation Time (t11)
- IDN D-0-1060.x.01 Default Configuration
- IDN D-0-1060.x.02 Configuration mask
- IDN D-0-1060.x.03 Maximum quantity of connection capability
- IDN D-0-1060.x.04 Maximum connection length of connection capability
- IDN D-0-1060.x.06 Configurable IDNs of connection capability
- IDN D-0-1060.x.07 Maximum processing time
- IDN D-0-1060.x.10 Minimum producer cycle time

B.2.3 FG Telegram Setup

The function group FG Telegram Setup groups all IDNs that are related to the configuration of the telegrams. This function group includes the following IDNs:

- IDN S-0-1009 Device Control (C-DEV) offset in MDT
- IDN S-0-1010 Lengths of MDTs
- IDN S-0-1011 Device Status (S-DEV) offset in AT
- IDN S-0-1012 Length of ATs
- IDN S-0-1013 SVC offset in MDT
- IDN S-0-1014 SVC offset in AT

B.2.4 FG Control

The function group FG Control groups all IDNs that are related to the control of the communication state machine of the slave. This function group includes the following IDNs:

- IDN S-0-0021 IDN-list of invalid operation data for CP2
- IDN S-0-0022 IDN-list of invalid operation data for CP3
- IDN S-0-0127 CP3 transition check
- IDN S-0-0128 CP4 transition check

B.2.5 FG Bus-Diagnosis

The function group FG Bus-Diagnosis groups all IDNs, which are related to bus diagnosis. This function group includes the following IDNs:

- IDN S-0-0014 Interface Status
- IDN S-0-1003 Allowed MST losses in CP3/CP4
- IDN S-0-1026 Version of communication hardware
- IDN S-0-1028 Error counter MST-P/S
- IDN S-0-1031 Test pin assignment Port 1 & Port 2
- IDN S-0-1034 PHY error counter Port 1 & Port 2
- IDN S-0-1035 Error counter Port1 and Port2
- IDN S-0-1040 Sub-device address
- IDN S-0-1044 Device Control (C-DEV)
- IDN S-0-1045 Device Status (S-DEV)
- IDN S-0-1046 List of Sub-device addresses in device

B.2.6 FG Connection

The function group FG Connection groups all IDNs that are related the configuration of connections. This function group includes the following IDNs:

- IDN S-0-0187 IDN-list of configurable data as producer
- IDN S-0-0188 IDN-list of configurable data as consumer
- IDN S-0-1050.x.01 Connection setup
- IDN S-0-1050.x.02 Connection Number
- IDN S-0-1050.x.03 Telegram Assignment
- IDN S-0-1050.x.04 Max. Length Of Connection
- IDN S-0-1050.x.05 Current length of connection
- IDN S-0-1050.x.06 Configuration List
- IDN S-0-1050.x.07 Assigned connection capability
- IDN S-0-1050.x.08 Connection Control (C-CON)
- IDN S-0-1050.x.09 Connection State
- IDN S-0-1050.x.10 Producer Cycle Time
- IDN S-0-1050.x.11 Allowed Data Losses
- IDN S-0-1050.x.12 Error Counter Data Losses
- IDN S-0-1050 Connections
- IDN S-0-1051 Image of connection setups

B.2.7 FG NRT

The function group FG Connection groups all IDNs that are related to the communication in the UC channel. This function group includes the following IDNs:

- IDN S-0-1017 NRT transmission time
- IDN S-0-1019 MAC Address
- IDN S-0-1020.0.1 Current IP address
- IDN S-0-1020 IP address
- IDN S-0-1021.0.1 Current subnet Mask
- IDN S-0-1021 Subnet Mask
- IDN S-0-1022.0.1 Current active gateway address
- IDN S-0-1022 Gateway address
- IDN S-0-1027.0.1 Requested MTU
- IDN S-0-1027.0.2 Effective MTU
- IDN S-0-1039.0.1 Current active hostname
- IDN S-0-1039 Hostname
- IDN S-0-1048 Activate network settings

and the following Control and Status Bits

- C-DEV/Control physical topology

B.2.8 FG MUX

The function group FG Mux groups all IDNs that are related the usage of the multiplex channel within a connection. This function group includes the following IDNs:

- IDN S-0-0360 MDT data container A1
- IDN S-0-0361 MDT data container B1
- IDN S-0-0362 MDT data container A list index
- IDN S-0-0363 MDT data container B list index
- IDN S-0-0364 AT data container A1
- IDN S-0-0365 AT data container B1
- IDN S-0-0366 AT data container A list index
- IDN S-0-0367 AT data container B list index
- IDN S-0-0368 Data container A pointer
- IDN S-0-0369 Data container B pointer
- IDN S-0-0370 MDT data container A/B configuration list
- IDN S-0-0371 AT data container A/B configuration list
- IDN S-0-0444 IDN-list of configurable data in the AT data container
- IDN S-0-0445 IDN-list of configurable data in the MDT data container
- IDN S-0-0450 MDT data container A2
- IDN S-0-0451 MDT data container A3
- IDN S-0-0452 MDT data container A4
- IDN S-0-0453 MDT data container A5
- IDN S-0-0454 MDT data container A6

- IDN S-0-0455 MDT data container A7
- IDN S-0-0456 MDT data container A8
- IDN S-0-0457 MDT data container A9
- IDN S-0-0458 MDT data container A10
- IDN S-0-0459 MDT data container B2
- IDN S-0-0480 AT data container A2
- IDN S-0-0481 AT data container A3
- IDN S-0-0482 AT data container A4
- IDN S-0-0483 AT data container A5
- IDN S-0-0484 AT data container A6
- IDN S-0-0485 AT data container A7
- IDN S-0-0486 AT data container A8
- IDN S-0-0487 AT data container A9
- IDN S-0-0488 AT data container A10
- IDN S-0-0489 AT data container B2
- IDN S-0-0490 MDT data container A2 configuration list
- IDN S-0-0491 MDT data container A3 configuration list
- IDN S-0-0492 MDT data container A4 configuration list
- IDN S-0-0493 MDT data container A5 configuration list
- IDN S-0-0494 MDT data container A6 configuration list
- IDN S-0-0495 MDT data container A7 configuration list
- IDN S-0-0496 MDT data container A8 configuration list
- IDN S-0-0497 MDT data container A9 configuration list
- IDN S-0-0498 MDT data container A10 configuration list
- IDN S-0-0500 AT data container A2 configuration list
- IDN S-0-0501 AT data container A3 configuration list
- IDN S-0-0502 AT data container A4 configuration list
- IDN S-0-0503 AT data container A5 configuration list
- IDN S-0-0504 AT data container A6 configuration list
- IDN S-0-0505 AT data container A7 configuration list
- IDN S-0-0506 AT data container A8 configuration list
- IDN S-0-0507 AT data container A9 configuration list
- IDN S-0-0508 AT data container A10 configuration list

B.2.9 FG SMP

The function group FG SMP groups all IDNs that are related to the usage of the Type 19 Messaging Protocol (SMP). This function group includes the following IDNs:

- IDN S-0-1100.0.01 Diagnostic counter sent SMP fragments
- IDN S-0-1100.0.02 Diagnostic counter received SMP fragments
- IDN S-0-1100.0.03 Diagnostic counter dropped SMP fragments
- IDN S-0-1101.x.01 SMP Container Data
- IDN S-0-1101.x.02 List of session identifiers

- IDN S-0-1101.x.03 List of session priorities

B.2.10 FG RTB

The function group FG RTB groups all IDNs that are related to the usage of the Real Time Bits. This function group includes the following IDNs:

- IDN S-0-0026 IDN allocation of producer RTB word container
- IDN S-0-0027 IDN allocation of consumer RTB word container
- IDN S-0-0144 Producer RTB word container
- IDN S-0-0145 Consumer RTB word container
- IDN S-0-0328 Bit allocation of producer RTB word container
- IDN S-0-0329 Bit allocation of consumer RTB word container
- IDN S-0-0398 IDN list of configurable real-time bits as producer
- IDN S-0-0399 IDN list of configurable real-time bits as consumer
- IDN S-0-1050.x.20 IDN Allocation of real-time bit
- IDN S-0-1050.x.21 Bit allocation of real-time bit
- IDN S-0-1080.x.02 Producer RTB list container
- IDN S-0-1080.x.03 IDN allocation of producer RTB list container
- IDN S-0-1080.x.04 Bit allocation of producer RTB list container
- IDN S-0-1081.x.02 Consumer RTB list container
- IDN S-0-1081.x.03 IDN allocation of consumer RTB list container
- IDN S-0-1081.x.04 Bit allocation of consumer RTB list container

and the following Control and Status Bits

- C-CON/Real-time bit 1
- C-CON/Real-time bit 1

B.3 Type 19 communication classes

B.3.1 General

Type 19 defines several communication classes that may be implemented by slaves. Two of these define the basic communication and are mutually exclusive:

- SCP_FixCFG
- SCP_VarCFG

The other communication classes may be implemented on top of them.

B.3.2 SCP_FixCFG

SCP_FixCfg is a basic class in the SCP. A slave, that implements SCP_FixCfg, provides the following features at the communication level:

- A full-featured service channel (SVC).
- Cyclic device control and device status words.
- Exact two connections are supported, one as consumer and one as producer:
  - The connection which is produced may be placed in any AT, and uses structure instance 0 (IDN S-0-1050.0.y). The position of this connection is not dependent on the place where device status is placed.

– The connection which is consumed may either be placed in any MDT or AT, and uses structure instance 1 (IDN S-0-1050.1.y). The position of this connection is not dependent on the place where device control is placed.

- The content of the connections is defined by the slave and cannot be changed by the master
- All listed IDNs and Bits are mandatory in SCP_FixCFG.

This class includes the following IDNs:

- IDN S-0-0021 IDN-list of invalid operation data for CP2
- IDN S-0-0022 IDN-list of invalid operation data for CP3
- IDN S-0-0127 CP3 transition check
- IDN S-0-0128 CP4 transition check
- IDN S-0-1000 SCP Type & Version
- IDN S-0-1002 Communication Cycle time (tScyc)
- IDN S-0-1003 Allowed MST losses in CP3/CP4
- IDN S-0-1009 Device Control (C-DEV) offset in MDT
- IDN S-0-1010 Lengths of MDTs
- IDN S-0-1011 Device Status (S-DEV) offset in AT
- IDN S-0-1012 Length of ATs
- IDN S-0-1013 SVC offset in MDT
- IDN S-0-1014 SVC offset in AT
- IDN S-0-1017 NRT transmission time
- IDN S-0-1026 Version of communication hardware
- IDN S-0-1035 Error counter Port1 and Port2
- IDN S-0-1040 Sub-device address
- IDN S-0-1046 List of Sub-device addresses in device
- IDN S-0-1050.x.3 Telegram Assignment
- IDN S-0-1050.x.5 Current length of connection

This class includes the following Control and Status Bits:

- C-CON/New data (new producer data)
- C-CON/Producer ready
- C-DEV/Identification
- C-DEV/Topology HS
- C-DEV/Topology control
- S-DEV/Communication warning interface
- S-DEV/Error connection
- S-DEV/Port status
- S-DEV/Procedure command change bit
- S-DEV/Slave valid
- S-DEV/Sub-device level
- S-DEV/Topology HS
- S-Dev/Topology status

B.3.3 SCP_FixCFG_0x02

The class SCP_FixCFG_0x02 is version 0x02 of the class SCP_FixCFG. The support of a SCP class latest version implies automatically the support of all lower versions of this class too. This version is a backward compatible enhancement to the class SCP_FixCFG.

This class includes the following IDN:

- IDN S-0-1037 Slave Jitter

B.3.4 SCP_FixCFG_0x03

The class SCP_FixCFG_0x03 is version 0x03 of the class SCP_FixCFG. The support of a SCP class latest version implies automatically the support of all lower versions of this class too. This version is a backward compatible enhancement to the class SCP_FixCFG.

This class includes the following IDN:

- IDN S-0-1050.x.09 Connection State

This class includes the following Control and Status Bits:

- C-CON/Counter
- C-CON/Flow-control

B.3.5 SCP_VarCFG

SCP_VarCfg is a basic class in the SCP. A slave, that implements SCP_VarCfg, provides the following features on the communication level:

- A full-featured service channel.
- Cyclic device control and status words.
- A certain number of connections is supported. The slave defines this number and provides it to the master.
  - The connections which are consumed may either be placed in any MDT or AT. The position of this connection is not dependent on the place where device control is placed.
  - The connections which are produced may be placed in any AT. The position of this connection is not dependent on the place where device status is placed.
- The content of all connections have to be configured (for example by the master). The slave provides lists of IDNs that can be cyclically produced and consumed, so the master can find it out.
- All listed IDNs and Bits are mandatory in SCP_VarCFG.

This class includes the following IDNs:

- IDN S-0-0014 Interface Status
- IDN S-0-0021 IDN-list of invalid operation data for CP2
- IDN S-0-0022 IDN-list of invalid operation data for CP3
- IDN S-0-0127 CP3 transition check
- IDN S-0-0128 CP4 transition check
- IDN S-0-0187 IDN-list of configurable data as producer
- IDN S-0-0188 IDN-list of configurable data as consumer
- IDN S-0-1000 SCP Type & Version
- IDN S-0-1002 Communication Cycle time (tScyc)

- IDN S-0-1003 Allowed MST losses in CP3/CP4
- IDN S-0-1009 Device Control (C-DEV) offset in MDT
- IDN S-0-1010 Lengths of MDTs
- IDN S-0-1011 Device Status (S-DEV) offset in AT
- IDN S-0-1012 Length of ATs
- IDN S-0-1013 SVC offset in MDT
- IDN S-0-1014 SVC offset in AT
- IDN S-0-1017 NRT transmission time
- IDN S-0-1026 Version of communication hardware
- IDN S-0-1035 Error counter Port1 and Port2
- IDN S-0-1040 Sub-device address
- IDN S-0-1046 List of Sub-device addresses in device
- IDN S-0-1050.x.01 Connection setup
- IDN S-0-1050.x.02 Connection Number
- IDN S-0-1050.x.03 Telegram Assignment
- IDN S-0-1050.x.04 Max. Length Of Connection
- IDN S-0-1050.x.05 Current length of connection
- IDN S-0-1050.x.06 Configuration List
- IDN S-0-1051 Image of connection setups

This class includes the following Control and Status Bits:

- C-CON/New data (new producer data)
- C-CON/Producer ready
- C-DEV/Identification
- C-DEV/Topology HS
- C-DEV/Topology control
- S-DEV/Communication warning interface
- S-DEV/Error connection
- S-DEV/Port status
- S-DEV/Procedure command change bit
- S-DEV/Slave valid
- S-DEV/Sub-device level
- S-DEV/Topology HS
- S-DEV/Topology status

B.3.6 SCP_VarCFG_0x02

The class SCP_VarCFG_0x02 is an add-on to the class SCP_VarCFG. It contains information for heterogeneous connections. The class SCP_VarCFG_0x02 is version 0x02 of the class SCP_VarCFG. The support of a SCP class latest version implies automatically the support of all lower versions of this class too. This version is a backward compatible enhancement to the class SCP_VarCFG.

This class includes the following IDN:

- IDN S-0-1037 Slave Jitter

B.3.7 SCP_VarCFG_0x03

The class SCP_VarCFG_0x03 is version 0x03 of the class SCP_VarCFG. The support of a SCP class latest version implies automatically the support of all lower versions of this class too. This version is a backward compatible enhancement to the class SCP_VarCFG.

This class includes the following IDN:

- IDN S-0-1050.x.09 Connection State

This class includes the following Control and Status Bits:

- C-CON/Counter
- C-CON/Flow-control

B.3.8 SCP_Sync

SCP_Sync is a class in the SCP. A slave, that implements SCP_Sync, provides the ability to isochronously produce and consume cyclic data.

This class includes the following IDNs:

- IDN S-0-1005 Minimum feedback processing time (t5)
- IDN S-0-1006 AT0 transmission starting time (t1)
- IDN S-0-1007 Feedback acquisition capture point (t4)
- IDN S-0-1008 Command value valid time (t3)
- IDN S-0-1015 Ring delay
- IDN S-0-1016 Slave delay (P/S)
- IDN S-0-1023 SYNC jitter
- IDN S-0-1024 SYNC delay measuring procedure command
- IDN S-0-1041 AT Command value valid time (t9)
- IDN S-0-1050.x.01 Connection setup
- IDN S-0-1050.x.10 Producer Cycle Time
- IDN S-0-1050.x.11 Allowed Data Losses
- IDN S-0-1050.x.12 Error Counter Data Losses

This class includes the following Control and Status Bits:

- C-CON/Data field delay
- C-CON/Producer synchronization

B.3.9 SCP_Sync

SCP_Sync is a class in the SCP. A slave, that implements SCP_Sync, provides the ability to isochronously produce and consume cyclic data.

This class includes the following IDNs:

- S-0-1005 Minimum feedback processing time (t5)
- S-0-1006 AT0 transmission starting time (t1)
- S-0-1007 Synchronization time (Tsync)
- S-0-1008 Command value valid time (t3)
- S-0-1015 Ring delay

- S-0-1016 Slave delay (P&S)
- S-0-1023 SYNC jitter
- S-0-1024 SYNC delay measuring procedure command
- S-0-1041 AT Command value valid time (t9)
- S-0-1050.x.01 Connection setup
- S-0-1050.x.10 Producer Cycle Time
- S-0-1050.x.11 Allowed Data Losses
- S-0-1050.x.12 Error Counter Data Losses

And the following Control or Status Bits

- C-CON/Data field delay
- C-CON/Producer synchronization

B.3.10 SCP_Sync_0x02

SCP_Sync_0x02 is a class in the Type 19 Communication profile. A slave, that implements SCP_Sync_0x02, provides the ability to isochronously produce and consume cyclic data with a producing cycle time (tPcyc) > Type 19 cycle time (tScyc). The class SCP_Sync_0x02 is version 0x02 of the class SCP_Sync. The support of a SCP class latest version implies automatically the support of all lower versions of this class too. This version is a backward compatible enhancement to the class SCP_Sync.

This class includes the following IDNs:

- IDN S-0-1028 Error counter MST-P&S
- IDN S-0-1036 Inter Frame Gap
- IDN S-0-1047 Maximum Consumer Activation Time (t11)

B.3.11 SCP_Sync_0x03

The class SCP_Sync_0x03 is version 0x03 of the class SCP_Sync. The support of a SCP class latest version implies automatically the support of all lower versions of this class too. This version is a backward compatible enhancement to the class SCP_Sync.

This class includes the following IDN:

- IDN S-0-1061 Maximum TSref-Counter

B.3.12 SCP_WD

SCP_WD is a class in the SCP. A slave, that implements SCP_WD, provides the ability to monitor connections that are consumed by the slave.

This class includes the following IDNs:

- IDN S-0-1050.x.01 Connection setup
- IDN S-0-1050.x.10 Producer Cycle Time

B.3.13 SCP_WD_0x02

SCP_WD_0x02 is a class in the Type 19 communication profile. A slave, that implements this class x02, provides the ability to supervise the communication produce with watchdog. The class SCP_WD_0x02 is version 0x02 of the class SCP_WD. The support of a SCP class latest version implies automatically the support of all lower versions of this class too. This version is a backward compatible enhancement to the class SCP_WD.

This class includes the following IDN:

- IDN S-0-1050.x.11 Allowed Data Losses

B.3.14 SCP_Diag

SCP_Diag is a class in the SCP. A slave, that implements SCP_Diag, provides information that can help to do bus-diagnosis.

This class includes the following IDNs:

- IDN S-0-0021 IDN-list of invalid operation data for CP2
- IDN S-0-0022 IDN-list of invalid operation data for CP3
- IDN S-0-1031 Test pin assignment Port 1 & Port 2
- IDN S-0-1044 Device Control (C-DEV)
- IDN S-0-1045 Device Status (S-DEV)
- IDN S-0-1050.x.08 Connection Control (C-CON)
- IDN S-0-1050.x.12 Error Counter Data Losses

In addition the Type 19 LED is supported.

B.3.15 SCP_RTB

SCP_RTB is a class in the SCP. A slave that implements SCP_RTB provides the ability to produce and consume real-time bits. Real-time bits are signals which indicate some selected status or event which are represented in real time.

This class includes the following IDNs:

- IDN S-0-0398 IDN list of configurable real-time/status bits
- IDN S-0-0399 IDN list of configurable real-time/control bits
- IDN S-0-1050.x.20 IDN Allocation of real-time bit
- IDN S-0-1050.x.21 Bit allocation of real-time bit

This class includes the following Control or Status Bits

- C-CON/Real-time bit 1
- C-CON/Real-time bit 2

B.3.16 SCP_HP

SCP_HP is a class in the SCP. A slave that implements SCP_HP provides the ability to be hot-plugged. That means it can be integrated into a Type 19 network which is already in communication phase 4. There are no IDNs related to this ability but the slave must implement the usage of the Hotplug-fields in MDT and AT.

B.3.17 SCP_SMP

SCP_SMP is a class in the SCP. A slave, which implements SCP_SMP provides the ability to talk part in a connection that uses the Type 19 messaging protocol.

This class includes the following IDNs:

- IDN S-0-1100.0.01 Diagnostic counter sent SMP fragments
- IDN S-0-1100.0.02 Diagnostic counter received SMP fragments
- IDN S-0-1100.0.03 Diagnostic counter dropped SMP fragments

- IDN S-0-1101.x.01 SMP Container Data
- IDN S-0-1101.x.02 List of session identifiers
- IDN S-0-1101.x.03 List of session priorities

B.3.18 SCP_Mux

SCP_MuX is a class in the SCP. A slave, that implements SCP_MuX provides the ability to produce and consume cyclic data that are multiplexed a standard data container in one connection.

This class includes the following IDNs:

- IDN S-0-0360 MDT data container A1
- IDN S-0-0361 MDT data container B1
- IDN S-0-0364 AT data container A1
- IDN S-0-0365 AT data container B1
- IDN S-0-0368 Data container A pointer
- IDN S-0-0369 Data container B pointer
- IDN S-0-0370 MDT data container A/B configuration list
- IDN S-0-0371 AT data container A/B configuration list

B.3.19 SCP_Ext_Mux

SCP_ExtMuX is a class in the SCP. A slave, that implements SCP_ExtMuX provides the ability to produce and consume cyclic data that are multiplexed an extended data container in one connection.

This class includes the following IDNs:

- IDN S-0-0360 MDT data container A1
- IDN S-0-0364 AT data container A1
- IDN S-0-0368 Data container A pointer
- IDN S-0-0370 MDT data container A/B configuration list
- IDN S-0-0371 AT data container A/B configuration list
- IDN S-0-0450 MDT data container A2
- IDN S-0-0480 AT data container A2
- IDN S-0-0490 MDT data container A2 configuration list
- IDN S-0-0500 AT data container A2 configuration list

B.3.20 SCP_NRT

SCP_NRT is a class in the SCP. A slave, that implements SCP_NRT, provides the ability to an active use the NRT-channel for transmission and reception of non-SIII-Ethernet-frames.

This class shall not be implemented in new devices. Please implement SCP_NRTPC instead.

This class includes the following IDNs:

- IDN S-0-1019 MAC Address
- IDN S-0-1020 IP address
- IDN S-0-1021 Subnet Mask

- IDN S-0-1022 Gateway address
- IDN S-0-1027.0.1 Requested MTU
- IDN S-0-1027.0.2 Effective MTU

B.3.21 SCP_Sig

SCP_Sig is a class in the SCP. A slave, that implements SCP_Sig, provides the ability to produce a signal status word and to consume a signal control word. Signals can be transmitted in real-time of the signal control word and signal status word. For this purpose, the signal control and the signal status word have to be configured into a connection.

This class includes the following IDNs:

- IDN S-0-0026 Configuration list for signal status word
- IDN S-0-0027 Configuration list for signal control word
- IDN S-0-0144 Signal status word
- IDN S-0-0145 Signal control word
- IDN S-0-0328 Bit number allocation list for signal status word
- IDN S-0-0329 Bit number allocation list for signal control word
- IDN S-0-0398 IDN list of configurable real-time/status bits
- IDN S-0-0399 IDN list of configurable real-time/control bits

B.3.22 SCP_ListSeg

SCP_ListSeg is a sub profile of SCP it contains the segmentwise access of parameters with variable length

This class includes the following IDNs:

- IDN S-0-0394 List IDN
- IDN S-0-0395 List index
- IDN S-0-0396 Number of list elements
- IDN S-0-0397 List segment

B.3.23 SCP_IPS

SCP_IPS is a class in the SCP. A slave that implements SCP_IPS provides the ability to an active use of the NRT-channel for transmission and reception defined by Type 19 networks.

The IPS group the following two kinds of services:

- S/IP services, which are Type 19 specific services based on UDP or TCP
- Further TFTP based services

B.3.24 SCP_Cap

SCP_Cap is a class in the Type 19 CommunicationProfile. A slave, that implements SCP_Cap, provides the ability to show his communication capabilities

This class includes the following IDNs:

- IDN S-0-1050.x.07 Assigned connection capability
- IDN S-0-1060.x.01 Default configuration
- IDN S-0-1060.x.02 Configuration mask

- IDN S-0-1060.x.03 Maximum quantity of this connection capability
- IDN S-0-1060.x.04 Max. connection length of connection capability
- IDN S-0-1060.x.06 Configurable IDNs of connection capability
- IDN S-0-1060.x.07 Maximum processing time
- IDN S-0-1060.x.10 Minimum producer cycle time
- IDN S-0-1060 Connection capabilities

B.3.25 SCP_RTBListProd

If a slave shows the class SCP_RTBListProd, then it supports the Real-time bit list container as producer.

This class includes the following IDNs:

- IDN S-0-0398 IDN list of configurable real-time bits as producer
- IDN S-0-1080.x.02 Producer RTB list container
- IDN S-0-1080.x.03 IDN allocation of producer RTB list container
- IDN S-0-1080.x.04 Bit allocation of producer RTB list container

B.3.26 SCP_RTBListCons

If a slave shows the class SCP_RTBListCons, then it supports the Real-time bit list container as consumer.

This class includes the following IDNs:

- IDN S-0-0399 IDN list of configurable real-time bits as consumer
- IDN S-0-1081.x.02 Consumer RTB list container
- IDN S-0-1081.x.03 IDN allocation of consumer RTB list container
- IDN S-0-1081.x.04 Bit allocation of consumer RTB list container

B.3.27 SCP_SysTime

If a slave shows the class SCP_SysTime, then it supports the system time transmitted by the master in the extended field of MDT0.

B.3.28 SCP_RTBWordProd

If a slave shows the class SCP_RTBWordProd, then it supports the Real-time word as producer.

This class includes the following IDNs:

- IDN S-0-0026 IDN allocation of producer RTB word container
- IDN S-0-0144 Producer RTB word container
- IDN S-0-0328 Bit allocation of producer RTB word container
- IDN S-0-0398 IDN list of configurable real-time bits as producer

B.3.29 SCP_RTBWordCons

If a slave shows the class SCP_RTBWordCons, then it supports the Real-time word as consumer.

This class includes the following IDNs:

- IDN S-0-0027 IDN allocation of consumer RTB word container
- IDN S-0-0145 Consumer RTB word container
- IDN S-0-0329 Bit allocation of consumer RTB word container
- IDN S-0-0399 IDN list of configurable real-time bits as consumer

B.3.30 SCP_SafetyCon

SCP_SafetyCon is a class in the Type 19 Communication Profile. A slave, that implements SCP_SafetyCon, provides the CSoS functionality.

This class includes the following IDNs:

- IDN S-0-1810.x.01 SV Max data age
- IDN S-0-1810.x.02 Safety Validator state
- IDN S-0-1810.x.03 SV Error code
- IDN S-0-1810.x.04 Safety Validator type
- IDN S-0-1810.x.05 SV Time coord msg min multiplier
- IDN S-0-1810.x.06 SV Max consumer number
- IDN S-0-1810.x.07 SV Timeout multiplier
- IDN S-0-1810.x.08 SV Ping interval EPI multiplier
- IDN S-0-1810.x.09 SV Network time expectation multiplier
- IDN S-0-1830.x.01 Cyclic SMP container (out)
- IDN S-0-1830.x.02 Cyclic SMP Session ID (out)
- IDN S-0-1830.x.03 List of cyclic SMP containers (in)
- IDN S-0-1830.x.04 List of cyclic SMP Session IDs (in)
- IDN S-0-1830.x.05 List of UCM SMP containers (in)
- IDN S-0-1830.x.07 List of UCM SMP containers (out)
- IDN S-0-1830.x.08 List of UCM SMP Session IDs (out)
- IDN S-0-1830.x.09 List of consumer numbers

B.3.31 SCP_OvS_Basic

SCP_OvSBasic is a class in the Type 19 Communication Profile. A slave, that implements SCP_OvSBasic, provides the basic mechanism of the oversampling functionality.

This class includes the following IDNs:

- IDN S-0-1150.x.01 OVS Control (C-OVS)
- IDN S-0-1150.x.02 OVS Status (S-OVS)
- IDN S-0-1150.x.03 OVS Container
- IDN S-0-1150.x.06 Configuration List OVS - IDNs
- IDN S-0-1150.x.10 Number of Samples

And the following Control or Status Bits

- C-OVS Flow control
- C-OVS Quit error
- S-OVS error
- S-OVS stop

B.3.32 SCP_NRTPC

SCP_NRTPC is a class in the SCP. A slave, that implements SCP_NRTPC, provides the ability to an active use the NRT-channel for transmission and reception of “non Type 19” Ethernet-frames.

This class includes the following IDNs:

- IDN S-0-1019 MAC Address
- IDN S-0-1020 IP address
- IDN S-0-1021 Subnet Mask
- IDN S-0-1022 Gateway address
- IDN S-0-1027.0.1 Requested MTU
- IDN S-0-1027.0.2 Effective MTU
- IDN S-0-1048 Activate network settings

B.3.33 SCP_Cyc

SCP_Cyc is a class in the Type 19 Communication Profile. A slave, that implements SCP_Cyc, provides the ability to produce and consume cyclic data.

This class includes the following IDNs:

- IDN S-0-1005 Minimum feedback processing time (t5)
- IDN S-0-1006 AT0 transmission starting time (t1)
- IDN S-0-1047 Maximum Consumer Activation Time (t11)

Annex C
(normative)

GDP (Generic Device Profile)

C.1 General

The objective of the Generic Device Profile (GDP) is to provide for a sight to the sub-device that is not dependent of the function specific profile (FSP) that is implemented by the sub-device. The following parts are independent of the FSP:

- Identification
- Administration
- Archiving
- GDP state machines allow a decoupling of communication and applications state machine

C.2 Function Groups

C.2.1 Function Group Diagnosis

The grouping of IDNs in the function group diagnosis has the aim to provide a defined interface for diagnosis tasks to the master. This includes the tasks of accessing diagnostic numbers and messages and the corresponding timestamps as well as the resetting of those.

This function group includes the following IDNs:

- S-0-0095 Diagnostic message
- S-0-0099 Reset class 1 diagnostic
- S-0-0390 Diagnostic number
- S-0-1303.0.01 Diagnostic trace configuration
- S-0-1303.0.02 Diagnostic trace control
- S-0-1303.0.03 Diagnostic trace state
- S-0-1303.0.10 Diagnostic trace buffer no1
- S-0-1303.0.11 Diagnostic trace buffer no2
- S-0-1303.0.12 Diagnostic trace buffer no3
- S-0-1303 Diagnostic trace

In addition to these IDNs a light emitting diode (Type 19 LED) is part of this function group. Two LEDs are defined:

- one labeled with S for the indication of the communication status
- one for each sub-device labeled with SDx for the indication of the sub-device status of sub-device x.

If the SDx LED is supported, there shall be one for each sub-device in the device labeled SD1, SD2, SD3, etc.

If the device contains only one sub-device this label shall be SD.

S LED

It is mandatory that a Type 19 device shows the states as shown in Table C.1 via a LED which is labeled with S or via another display element.

The slave device indicates the communication state machine and three additional independent states. These are Loopback, C1D and identification. The colors and priority are shown in Table C.1. The state identification is shown with highest priority.

Table C.1 – Type 19 LED

Pattern	Color	Description	Prio.	Comment
#1	dark	NRT-Mode	0	no Type 19 communication
#2	orange	CP0	0	communication phase 0 is active
#3	orange	CP1	0	communication phase 1 is active
#4		CP2	0	communication phase 2 is active
#5		CP3	0	communication phase 3 is active
#6	green	CP4	0	communication phase 4 is active
#7		HP0	1	Device is in hot-plug phase 0
#8		HP1	1	Device is in hot-plug phase 1
#9		HP2	1	Device is in hot-plug phase 2
#10		Fast forward ⇒ Loopback	2	RT-state has changed from fast-forward to loopback
#11		application error	3	see GDP & FSP Status codes class error
#12		MST losses ≥ (S-0-1003/2)	4	as long as the communication warning (S-DEV.Bit15) in the Device Status is present, at least 2 sec.
#13	red	communication error	5	see SCP Status codes class error
#14		Identification	6	(C-DEV.Bit 15 in the Device Control) used for address allocation, configuration error or other identification purposes
#15		Watchdog error	7	

The time division for LED flashing shall be 250 ms (4 Hz).

***SDx* LED**

In addition to the Type 19 LED another optional LED is specified (see Table C.2). This additional LED shows the status of the sub-device.

Table C.2 – SDx LED

Pattern	Color	Description	Prio.	Comment
#1	dark	sub-device not active	0	
#2	orange	parametrization level (PL)	0	sub-device is in parametrization level (PL)
#3	green	operating level (OL)	0	sub-device is in operating level (OL)
#4	red	application error (C1D)	1	see GDP & FSP Status codes class error

C.2.2 Function Group Archiving

The grouping of IDNs in the function group archiving has the aim to provide a defined interface for archiving tasks to the master. This includes the tasks of accessing lists of data that has to be stored to do a backup, the checksums related to this data, procedure commands to perform the backup as well as restore it.

This function group includes the following IDNs:

- S-0-0192 IDN-list of all backup operation data
- S-0-0262 Load defaults procedure command
- S-0-0263 Load working memory procedure command
- S-0-0264 Backup working memory procedure command
- S-0-0269 Storage mode
- S-0-0270 IDN list of selected backup operation data
- S-0-0293 Selectively backup working memory procedure command
- S-0-0326.x.0 Parameter Checksum
- S-0-0327.x.0 IDN list of checksum parameter
- S-0-0531 Checksum for backup operation data
- S-0-1310 IDN List of operation data changed from default

C.2.3 Function Group Administration

The grouping of IDNs in the function group administration has the aim to provide a defined interface for administration tasks to the master. This includes the tasks of setting the language in the sub-device as well as setting a password for changing a list of data.

This function group includes the following IDNs:

- S-0-0017 IDN-list of all operation data
- S-0-0025 IDN-list of all procedure commands
- S-0-0265 Language selection
- S-0-0266 List of available languages
- S-0-0267 Password
- S-0-0279 IDN list of password protected data

C.2.4 Function Group Identification

The grouping of IDNs in the function group identification has the aim to provide a defined interface for identification tasks to the master. This includes the tasks of accessing all electronic labels of all available components of the device. Therefore each of these components is represented by a structure instance (SI) of the electronic label parameter (S-0-

1300). Another task covered by this GDP function group is to describe the functional view upon the sub-device. This means the description of the application layer within the sub-device (GDP + FSPs).

This function group includes the following IDNs:

- S-0-1300.x.01 Component Name
- S-0-1300.x.02 Vendor Name
- S-0-1300.x.03 Vendor Code
- S-0-1300.x.04 Device Name
- S-0-1300.x.05 Vendor Device ID
- S-0-1300.x.06 Connected to sub-device
- S-0-1300.x.07 Function Revision
- S-0-1300.x.08 Hardware Revision
- S-0-1300.x.09 Software Revision
- S-0-1300.x.10 Firmware Loader Revision
- S-0-1300.x.11 Order Number
- S-0-1300.x.12 Serial Number
- S-0-1300.x.13 Manufacturing Date
- S-0-1300.x.14 QA Date
- S-0-1300.x.20 Operational Hours
- S-0-1300.x.21 Service Date
- S-0-1300.x.22 Calibration Date
- S-0-1300.x.23 Calibration Due Date
- S-0-1300 Electronic Label
- S-0-1301 List of GDP classes & Version
- S-0-1302.x.01 FSP Type & Version
- S-0-1302.x.02 Function groups
- S-0-1302.x.03 Application Type
- S-0-1302 Resource Structures of sub-device

C.2.5 Function Group State machine

C.2.5.1 The sub-device state machine

In addition to the communication state machine each sub-device has to implement a sub-device state machine, which represents the operating state of the application (see Figure C.1).

In general the sub-device state machine has two states, the operating level and the parameterization level.

- Parameterization level (PL)
  - Within the parameterization level,
    - the sub-device deactivates the monitoring of the attached resources (for example encoder, motor, IO). The resource specific actions, which are required to change to parameterization level, are described within resource specific state machines (for example IO_state_machine or Drive state machine). With the activation of the parameterization level the corresponding monitoring bits of the resource status

words (for example S-0-1500.x.02 IO Status Bit 15 (outputs ready-to-operate) or S-0-0135 Drive status bits 15 and 14) are set to 0.

  - parameters which are write protected in parameterization level (PL) cannot be changed.
  - since certain monitoring functions are switched off, assemblies of a sub-device can be exchanged.
  - Bit 4 (Sub-device level) of the Device Status word is set to "1" (parameterization level (PL) is active).

- Operating level (OL)
  – Within the operating level:
    - The sub-device is ready for running the application. All monitoring systems are switched on.
    - Parameters which are write protected in operating level (OL) cannot be changed.
    - Bit 4 (Sub-device level) of the Device Status word is set to "0" (operating level (OL) is active).
    - As long as the sub-device does not support the class "GDP_StM" the sub-device state machine is coupled to the communication state machine. The following figure shows the dependency of the sub-device state machine from the communication state machine.

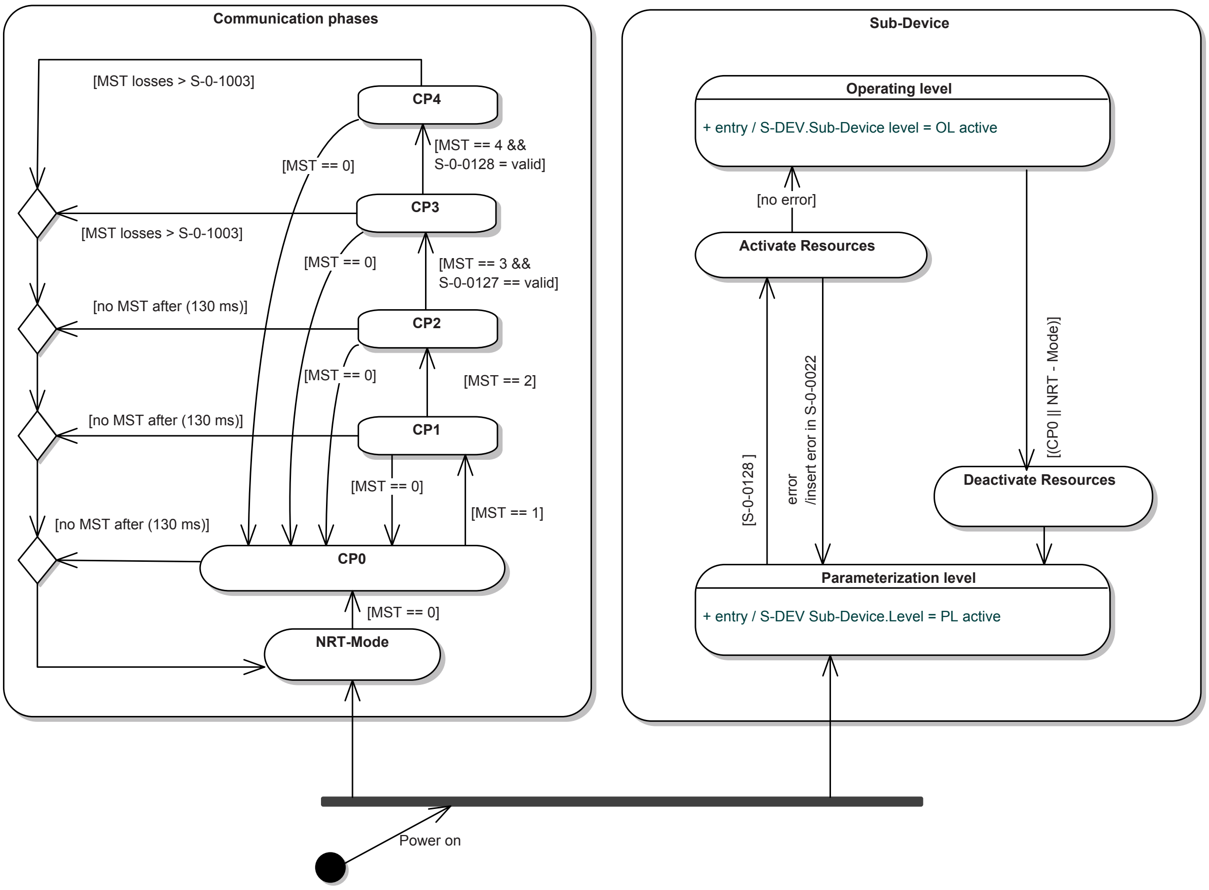

Figure C.1 – State machine without class GDP_StM

Without the support of the class "GDP_StM" the parameterization level gets active:

- After a restart of the slave.

- With an entry to NRT Mode or CP0.

Without the support of the class "GDP_StM" the operating level gets active:

- With the successful execution of procedure command S-0-0128 CP4 transition check

C.2.5.2 Decoupled state machines

C.2.5.2.1 Overview

The parameters of the function group "State Machine" have the aim to provide an interface, which controls the sub-device state machine independently from the communication state machine (Figure C.2). The following IDNs are defined within this function group:

- An IDN for the activation of the parameterization level (S-0-0420 Activate parametrization level procedure command (PL))
- An IDN for the activation of the operating level (S-0-0422 Exit parameterization level procedure command)
- An IDN for the error handling during the activation of the operating level (S-0-0423 IDN-list of invalid data for parameterization level)
- An IDN to control the coupling and decoupling of the communication and the sub-device state machine (S-0-0425 Sub-device state machine control)

The slave can realize this interface by supporting the class "GDP_StM".

In order to control the state machine of the sub-device independent from the communication state machine, the following commandos have to be executed:

- S-0-0420 Activate parametrization level procedure command (PL)
- S-0-0422 Exit parameterization level procedure command

In order to control, whether or not a change of the communication state machine leads to a transition of the sub-device state machine, S-0-0425 Sub-device state machine control is used.

The following figure shows how the IDNs, which are defined within the function group "State Machine", control the sub-device state machine.

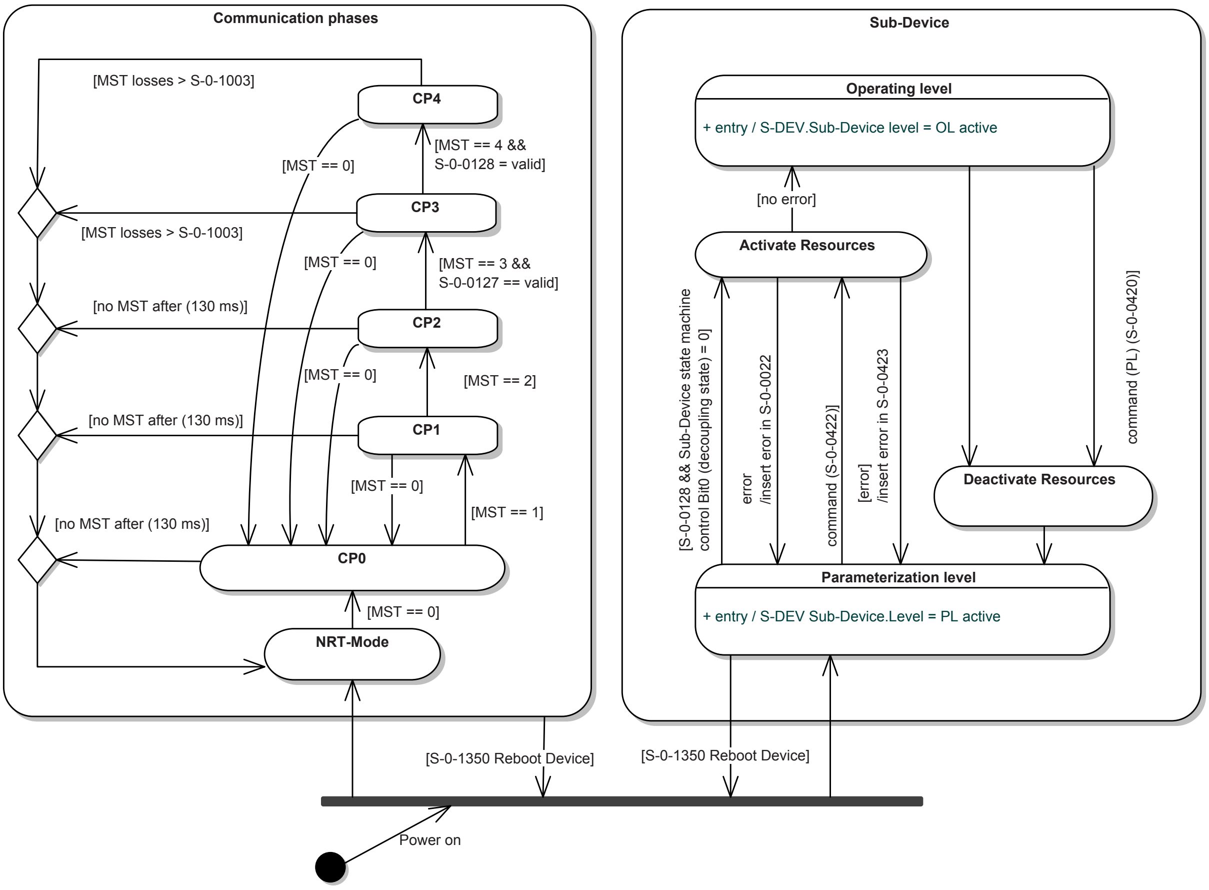

Figure C.2 – State machine without class GDP_StM

By supporting the class "GDP_StM" amongst others the following uses cases become possible.

C.2.5.2.2 Re-Parameterization during real time data transfer

The procedure command S-0-0420 Activate parametrization level procedure command (PL) makes it possible to change parameters, that are write protected in operating level and whose write protection is bound to the sub-device state machine, without leaving communication phase CP3/CP4. This can be useful if application specific parameters of a specific sub-device have to be adjusted while the applications of other connected slaves remain activated.

In order to exit the operating level the master has to activate the procedure command S-0-0420 Activate parametrization level procedure command (PL) in the sub-device. After the procedure command has been executed the sub-device shuts down its application and switches internally to the parameterization level, in which the communication of the active communication phase is maintained. While the sub-device is in parameterization level the master is able to change parameters which are writable in the current communication phase and in PL. After the parameter setup the master can re-activate the application of the sub-device by executing procedure command S-0-0422 Exit parameterization level procedure command.

C.2.5.2.3 Obviation of application activation

In some cases (for example during commissioning) it might not be possible to run the application of a specific sub-device. In this case a switch to CP4 without the support of the class "GDP_StM" is not possible, since the procedure command S-0-0128 CP4 transition check, which amongst others checks if the application is ready to run, would fail.

By supporting the class "GDP_StM" the sub-device offers the possibility to explicitly obviate the switch to the operating level and therefore allows a communication phase switch to CP4 with a deactivated application. This behavior is achieved by setting bit 0 (decoupling state) of S-0-0425 Sub-device state machine control to 1 (decoupled) before the execution of procedure command S-0-0128 CP4 transition check.

If the procedure command S-0-0128 CP4 transition check is executed and bit 0 of S-0-0425 Sub-device state machine control is set, the execution of S-0-0128 CP4 transition check is carried out in the sub-device without checks and generates a positive acknowledge.

C.2.5.2.4 Early run of the application

In some situations (for example during commissioning) where a cyclic communication (communication phases CP3 and CP4) is not possible, it is required to access a running application within the sub-device via the SVC or S/IP in order to test sub-functions of the application. The support of the class "GDP_StM" allows the user to activate the application within the device before the activation of the application is done implicitly with procedure command S-0-0128 CP4 transition check.

The early activation of the application is done by executing the procedure command S-0-0422 Exit parameterization level procedure command in earlier communication phases (NRT Mode-CP3, typically CP2). Once the application has been activated before S-0-0128 CP4 transition check, the sub-device state machine is already activated and thus the execution of S-0-0128 CP4 transition check is carried out in the sub-device without checks. In this case the procedure command S-0-0128 CP4 transition check generates a positive acknowledge.

C.2.5.2.5 Continuation of operating level

Some applications require a continuous operational state of specific slaves, independent from the current communication phase. By supporting the class "GDP_StM" the sub-device offers the possibility to keep the application running despite communication phase changes to a lower communication phase (for example NRT Mode, CP0).

In order to keep the application running, independent of communication phase changes, S-0-0425 Sub-device state machine control.Bit0 (decoupling state) has to be set to 1 (decoupled).

This function group includes the following IDNs:

- S-0-0420 Activate parametrization level procedure command (PL)
- S-0-0422 Exit parameterization level procedure command
- S-0-0423 IDN-list of invalid data for parameterization level
- S-0-0425 Sub-device state machine control
- S-0-1350.0.1 Reboot Device Delay
- S-0-1350 Reboot Device

And the following Control and Status Bits

- S-DEV/Sub-device level

C.2.6 Function Group Time

This function group describes the transmission and activating of the Type 19 time in the slave related to the current time in the master. The master shall calculate a predicted time (forecast) to compensate the transmission delay.

The parameter S-0-1305.0.01 contains the current Type 19 time in IEC 61588 format. The sub-device may mark events with this time (for example S-0-1305.0.01 marks the diagnoses trace with time stamps).

This function group includes the following IDNs:

- S-0-1305.0.01 Type 19 current time
- S-0-1305.0.02 Type 19 current fine time

C.2.7 Function Group Conformance Test GDP

This function group contains IDNs which are required by the conformance test.

The following IDNs shall not be listed within the operation data of S-0-0017 (IDN-list of all operation data)

- S-0-1399.0.1 Test IDN Diagnostic Event

C.3 Classification

C.3.1 General

Several GDP classes may be implemented by sub-devices.

The GDP defines one class (GDP_Basic) which provides the minimum of functionality that is needed in a sub-device on the application level. The class GDP_Basic shall be implemented by all sub-devices.

The other GDP classes may be implemented on top of them.

C.3.2 GDP_Basic

These are the essential IDNs for the Generic Device Profile. Every Type 19 device shall contain these parameter.

This class includes the following IDNs:

- S-0-0017 IDN-list of all operation data
- S-0-0099 Reset class 1 diagnostic
- S-0-0390 Diagnostic number
- S-0-1300.x.03 Vendor Code
- S-0-1300.x.05 Vendor Device ID
- S-0-1301 List of GDP classes & Version
- S-0-1302.x.01 FSP Type & Version
- S-0-1302.x.02 Function groups

C.3.3 GDP_DiagT

These are the essential IDNs for the Generic Device Profile, class Diagnosis Trace. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-1303.0.02 Diagnostic trace control
- S-0-1303.0.03 Diagnostic trace state
- S-0-1303.0.10 Diagnostic trace buffer no1
- S-0-1303.0.11 Diagnostic trace buffer no2

- S-0-1305.0.01 Type 19 current time

C.3.4 GDP_DiagTAdv

These are the essential IDNs for the Generic Device Profile, class Diagnosis Trace Advanced. This class requires the implementation of the class GDP_DiagT. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-1303.0.01 Diagnostic trace configuration
- S-0-1303.0.12 Diagnostic trace buffer no3

C.3.5 GDP_LNg

These are the essential IDNs for the Generic Device Profile, class LaNguage. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-0265 Language selection
- S-0-0266 List of available languages

C.3.6 GDP_PWD

These are the essential IDNs for the Generic Device Profile, class PassWorD. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-0267 Password
- S-0-0279 IDN list of password protected data

C.3.7 GDP_Id

These are the essential IDNs for the Generic Device Profile, class Identification. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-1300.x.01 Component Name
- S-0-1300.x.02 Vendor Name
- S-0-1300.x.04 Device Name
- S-0-1300.x.11 Order Number
- S-0-1300.x.12 Serial Number
- S-0-1302.x.03 Application Type

C.3.8 GDP_Rev

These are the essential IDNs for the Generic Device Profile, class Revisions. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-1300.x.07 Function Revision
- S-0-1300.x.08 Hardware Revision
- S-0-1300.x.09 Software Revision

- S-0-1300.x.10 Firmware Loader Revision

C.3.9 GDP_QA

These are the essential IDNs for the Generic Device Profile, class Quality Assurance. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-1300.x.13 Manufacturing Date
- S-0-1300.x.14 QA Date
- S-0-1300.x.21 Service Date
- S-0-1300.x.22 Calibration Date
- S-0-1300.x.23 Calibration Due Date

C.3.10 GDP_CKs

These are the essential IDNs for the Generic Device Profile, class Checksum. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-0192 IDN-list of all backup operation data
- S-0-0531 Checksum for backup operation data

C.3.11 GDP_CKsUser

These are the essential IDNs for the Generic Device Profile, class ChecKsumsUser. This class requires the implementation of the class GDP_CKs. If this class is announced the sub-device shall implement this parameter and the dedicated functions. These parameters may be instantiated by using the instances (SI).

This class includes the following IDNs:

- S-0-0326.x.0 Parameter Checksum
- S-0-0327.x.0 IDN list of checksum parameter

C.3.12 GDP_StM

These are the essential IDNs for the Generic Device Profile, class StateMachine. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-0420 Activate parametrization level procedure command (PL)
- S-0-0422 Exit parameterization level procedure command
- S-0-0423 IDN-list of invalid data for parameterization level
- S-0-0425 Sub-device state machine control

C.3.13 GDP_BKP

These are the essential IDNs for the Generic Device Profile, class BacKuP. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-0192 IDN-list of all backup operation data
- S-0-0262 Load defaults procedure command
- S-0-0263 Load working memory procedure command
- S-0-0264 Backup working memory procedure command
- S-0-0531 Checksum for backup operation data

C.3.14 GDP_BKPAdv

These are the essential IDNs for the Generic Device Profile, class BacKuPAdvance. This class requires the implementation of the class GDP_BKP. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-0270 IDN list of selected backup operation data
- S-0-0293 Selectively backup working memory procedure command
- S-0-1310 IDN List of operation data changed from default

C.3.15 GDP_RST

These are the essential IDNs for the Generic Device Profile, class Restart. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

This class includes the following IDNs:

- S-0-1350 Reboot Device

C.3.16 GDP_CIPSafetyDev

These are the essential IDNs for the Generic Device Profile, class CIP_SafetyDev. If this class is announced the sub-device shall implement this parameter and the dedicated functions.

C.4 List of all GPD related IDNs

C.4.1 IDN specification

See Clause A.1.

C.4.2 Identification numbers in numerical orders

Table C.3 lists the IDNs which are related to the GDP, and that devices of this type shall support.

Application-specific data content is specified in other relevant standards, for example IEC 61800-7-20x.

Table C.3 – List of relevant communication-related IDNs

IDN (N°)	Name
S-0-0000	Dummy IDN
S-0-0017	IDN-list of all operation data
S-0-0025	IDN-list of all procedure commands
S-0-0095	Diagnostic message
S-0-0099	Reset class 1 diagnostic

IDN (N°)	Name
S-0-0192	IDN-list of all backup operation data
S-0-0262	Load defaults procedure command
S-0-0263	Load working memory procedure command
S-0-0264	Backup working memory procedure command
S-0-0265	Language selection
S-0-0266	List of available languages
S-0-0267	Password
S-0-0269	Storage mode
S-0-0270	IDN list of selected backup operation data
S-0-0279	IDN list of password protected data
S-0-0293	Selectively backup working memory procedure command
S-0-0326.x.0	Parameter checksum
S-0-0327.x.0	IDN list of Parameter checksum
S-0-0390	Diagnostic number
S-0-0420	Activate parametrization level procedure command (PL)
S-0-0422	Exit parameterization level procedure command
S-0-0423	IDN-list of invalid data for parameterization level
S-0-0425	Sub-device state machine control
S-0-0531	Checksum for backup operation data
S-0-1300.x.1	Component Name
S-0-1300.x.2	Vendor Name
S-0-1300.x.3	Vendor Code
S-0-1300.x.4	Device Name
S-0-1300.x.5	Vendor Device ID
S-0-1300.x.6	Connected to sub-device
S-0-1300.x.7	Function revision
S-0-1300.x.8	Hardware Revision
S-0-1300.x.9	Software Revision
S-0-1300.x.10	Firmware Loader Revision
S-0-1300.x.11	Order number
S-0-1300.x.12	Serial Number
S-0-1300.x.13	Manufacturing date
S-0-1300.x.14	QA Date
S-0-1300.x.20	Operational hours
S-0-1300.x.21	Service Date
S-0-1300.x.22	Calibration Date
S-0-1300.x.23	Calibration Due Date
S-0-1301	List of GDP function groups & Version
S-0-1302.x.1	FSP Type & Version
S-0-1302.x.2	Function groups
S-0-1302.x.3	Application Type
S-0-1303.0.1	Diagnosis trace configuration
S-0-1303.0.2	Diagnosis trace control

IDN (N°)	Name
S-0-1303.0.3	Diagnosis trace state
S-0-1303.0.10	Diagnosis trace buffer no1
S-0-1303.0.11	Diagnosis trace buffer no2
S-0-1303.0.12	Diagnosis trace buffer no3
S-0-1305.0.1	Current time
S-0-1305.0.2	Current fine time
S-0-1310	IDN List of operation data changed from default
S-0-1399	Test IDN Diagnostic Event

C.4.3 Detailed specification of communication-related IDNs

C.4.3.1 IDN S-0-0000 Dummy IDN

C.4.3.1.1 Attributes

Table C.4 shows the possible attributes for this IDN.

Table C.4 – Attributes of IDN S-0-0000

Attribute	Value
Name	Dummy IDN
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.1.2 Description

The usage of this parameter maybe

- a placeholder in a IDN configuration list
- a placeholder of two octets in a connection
- to switch off allocated functions

C.4.3.2 IDN S-0-0017 IDN-list of all operation data

C.4.3.2.1 Attributes

Table C.5 shows the possible attributes for this IDN.

Table C.5 – Attributes of IDN S-0-0017

Attribute	Value
Name	IDN-list of all operation data
Version	—
Length	4, variable (list sorted by IDN)
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.2.2 Description

All IDNs of all procedure commands and parameters of the sub-device are stored in this IDN list.

C.4.3.3 IDN S-0-0025 IDN-list of all procedure commands

C.4.3.3.1 Attributes

Table C.6 shows the possible attributes for this IDN.

Table C.6 – Attributes of IDN S-0-0025

Attribute	Value
Name	IDN-list of all procedure commands
Version	—
Length	4, variable (list sorted by IDN)
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.3.2 Description

The IDNs of all procedure commands are stored in this IDN-list.

C.4.3.4 IDN S-0-0095 Diagnostic message

C.4.3.4.1 Attributes

Table C.7 shows the possible attributes for this IDN.

Table C.7 – Attributes of IDN S-0-0095

Attribute	Value
Name	Diagnostic message
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.4.2 Description

The currently relevant operating status is being monitored with diagnostic messages. The diagnostic messages are generated by the slave as a text and stored in the operation data of this IDN.

C.4.3.5 IDN S-0-0099 Reset class 1 diagnostics

C.4.3.5.1 Attributes

Table C.8 shows the possible attributes for this IDN.

Table C.8 – Attributes of IDN S-0-0099

Attribute	Value
Name	Reset class 1 diagnostics
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.5.2 Description

If this procedure command is received by the sub-device via the service channel, then all diagnostic events with the class error (C1D), the error bits and the shut-down mechanism are deleted. If the cause of an diagnostic event with the class "error" is still active, then the corresponding event will be generated again. Thus an additional entry for this event within the diagnostic trace has to be generated.

This procedure command is not interruptible and generates no negative acknowledgment even if an error cannot be deleted or no error exists in the sub-device.

C.4.3.6 IDN S-0-0192 IDN-list of all backup operation data

C.4.3.6.1 Attributes

Table C.9 shows the possible attributes for this IDN.

Table C.9 – Attributes of IDN S-0-0192

Attribute	Value
Name	IDN-list of all backup operation data
Version	—
Length	4, variable (list sorted by IDN)
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.6.2 Description

The IDN-list stores IDNs of all device parameter that have to be loaded in the device in order to guarantee correct operation. The master uses this list to generate a backup copy of the device parameters.

C.4.3.7 IDN S-0-0262 Load defaults procedure command

C.4.3.7.1 Attributes

Table C.10 shows the possible attributes for this IDN.

Table C.10 – Attributes of IDN S-0-0262

Attribute	Value
Name	Load defaults procedure command
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.7.2 Description

When the master sets and enables the load defaults procedure command, the default parameter values will be activated. The scope and contents of the default parameter values (for example limit values, velocity loop settings, etc.) are determined by the device supplier. The default parameter values are not optimized for the respective application. Therefore the default parameter values allow a problem free inter operation between the sub-device and its connected components.

C.4.3.8 IDN S-0-0263 Load working memory procedure command

C.4.3.8.1 Attributes

Table C.11 shows the possible attributes for this IDN.

Table C.11 – Attributes of IDN S-0-0263

Attribute	Value
Name	Load working memory procedure command
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.8.2 Description

When the master sets and enables the Load working memory procedure command, all data necessary for operation (see S-0-0192) will be loaded from the device’s nonvolatile memory into its “active memory”. After power on, the device automatically transfers the data from non-volatile memory into the active memory.

NOTE This procedure command will cause active parameters to be overwritten.

C.4.3.9 IDN S-0-0264 Backup working memory procedure command

C.4.3.9.1 Attributes

Table C.12 shows the possible attributes for this IDN.

Table C.12 – Attributes of IDN S-0-0264

Attribute	Value
Name	Backup working memory procedure command
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.9.2 Description

When the master sets and enables the Backup working memory procedure command, all data necessary for operation (see S-0-0192) will be loaded from the device's "active memory" into its non-volatile memory.

NOTE This procedure command will cause previously saved parameters to be overwritten.

C.4.3.10 IDN S-0-0265 Language selection

C.4.3.10.1 Attributes

Table C.13 shows the possible attributes for this IDN.

Table C.13 – Attributes of IDN S-0-0265

Attribute	Value
Name	Language selection
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.10.2 Description

This parameter can be used to select one of the languages available in the device (see S-0-0266). By changing the language selection, text from the device such as

- Name (element 2)
- Unit (element 4) and

- all parameter with data type and display format = text (for example S-0-0095)

will be displayed in the selected language (see Table C.14).

Table C.14 – Language codes

Value	Language
0	German
1	English
2	French
3	Spanish
4	Italian
5	Portuguese
6	Polish
7	Hungarian
8	Russian
9	Swedish
10	Danish
11	Norwegian
12-65535	(reserved)

C.4.3.11 IDN S-0-0266 List of available languages

C.4.3.11.1 Attributes

Table C.15 shows the possible attributes for this IDN.

Table C.15 – Attributes of IDN S-0-0266

Attribute	Value
Name	List of available languages
Version	—
Length	2, variable
Display Format	Unsigned decimal (Language codes see S-0-0165)
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.11.2 Description

This list contains codes for all languages currently available in the device for language selection (see S-0-0265). This list is required if the device cannot manage or save all languages in its memory simultaneously.

C.4.3.12 IDN S-0-0267 Password

C.4.3.12.1 Attributes

Table C.16 shows the possible attributes for this IDN.

Table C.16 – Attributes of IDN S-0-0267

Attribute	Value
Name	Password
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.12.2 Description

This parameter is used to write protect selected parameters of the device by means of a password.

The parameters, which are affected by this kind of write protection, are listed in S-0-0279 IDN list of password protected data.

The password, which is used for the write protection underlies the following rules:

- it shall include only the UTF8 characters
- spaces (UTF8 code 0x20) are not allowed
- the character recognition is case sensitive

The state machine of the write protection of all password protected parameters is illustrated in Figure C.3.

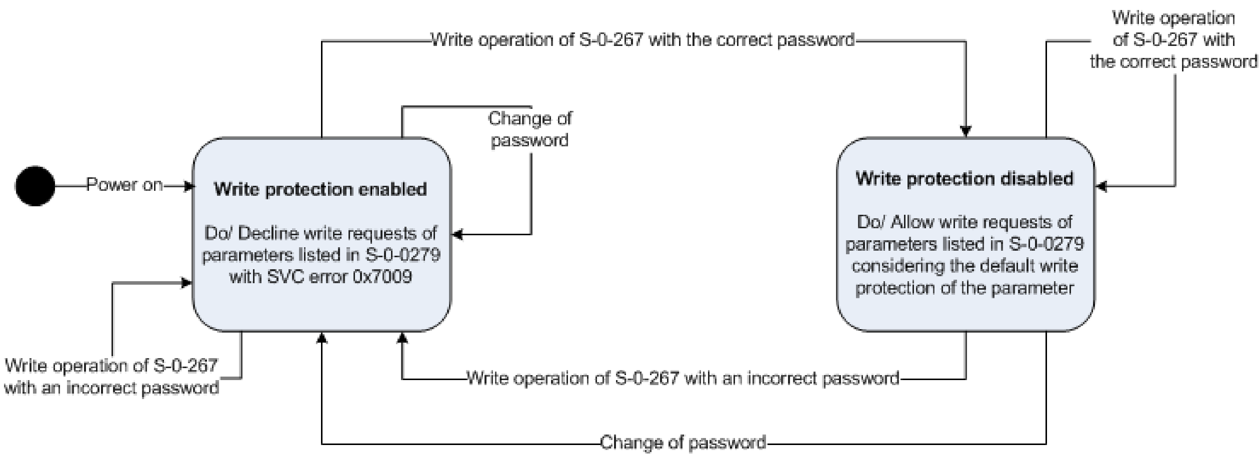

Figure C.3 – Password State Machine

The states can be described as follows (see Table C.17):

Table C.17 – States of the password state machine

State	Description
Write protection enabled	Write requests of parameters listed in S-0-0279 IDN list of password protected data and write requests of the parameter S-0-0279 IDN list of password protected data are declined with SVC error 0x7009 - Operation data is password write-protected. If the operation data of S-0-0267 Password is read in this state, the device will not return the password in plain text. The string "***" (UTF8 code 0x2A) will be sent instead.
Write protection disabled	Write requests of parameters listed in S-0-0279 IDN list of password protected data considering the default write protection of the parameter are possible. This means, that a write request of a parameter, which is write protected in CP4 and listed in S-0-0279 IDN list of password protected data is possible in a communication phase < CP4 but fails in CP4. If the operation data of S-0-0267 Password is read in this state, the device will not return the password in plain text. The string "$$$" (UTF8 code 0x24) will be sent instead.

The transitions can be described as follows (see Table C.18):

Table C.18 – Transitions of the password state machine

Transition			Description
Source	**Target**	**Condition**	
Write protection enabled	Write protection enabled	Write operation of S-0-0267 Password with an incorrect password	Sending a write request of the operation data of S-0-0267 Password with an incorrect password in the state of enabled password-specific write protection does not change the write protection of the parameters listed in S-0-0279 IDN list of password protected data. The write request shall not be declined by the slave with an SVC error (for example 0x7008 (Invalid operation data)), except boundary conditions of the SVC access (for example length of operation data > maximum length) are violated.
Write protection enabled	Write protection enabled	Change of password	A change of the password (see section below) in the state of enabled password-specific write protection, does not change the write protection of the parameters listed in S-0-0279 IDN list of password protected data.
Write protection enabled	Write protection disabled	Write operation of S-0-0267 Password with the correct password	Sending a write request of the operation data of S-0-0267 Password with the correct password (case sensitive), disables the password-specific write protection of all parameters listed in S-0-0279 IDN list of password protected data.
Write protection disabled	Write protection disabled	Write operation of S-0-0267 Password with the correct password	Sending a write request of the operation data of S-0-0267 Password with the correct password in the state of disabled password-specific write protection does not change the write protection of the parameters listed in S-0-0279 IDN list of password protected data.
Write protection disabled	Write protection enabled	Change of password	A change of the password (see section below) enables the password-specific write protection of all parameters listed in S-0-0279 IDN list of password protected data.
Write protection disabled	Write protection enabled	Write operation of S-0-0267 Password with an incorrect password	Sending a write request of the operation data of S-0-0267 Password with an incorrect password, enables the password-specific write protection of all parameters listed in S-0-0279 IDN list of password protected data. The write request shall not be declined by the slave with an SVC error (for example 0x7008 (Invalid operation data)), except boundary conditions of the SVC access (for example length of operation data > maximum length) are violated

Changing the password

In order to change the password, the current password, the new password, and a second verification of the new password have to be written via the SVC (see Table C.19).

A space character (UTF8 code 0x20) is used to delimit the passwords. The new password and the verification copy must match for the device to accept the change.

Table C.19 – Changing the password

1	2	3	4	5
current password	Space (UTF8 code 0x20)	New password	Space (UTF8 code 0x20)	New password

If the new password does not match the verification copy, the slave shall decline the change request with the SVC error 0x7008 - Invalid operation data.

The current value of the password has to be stored in non-volatile memory.

In the case of an unknown password, a supplier designed master password shall be available to deactivate password write protection.

This password shall always be available as a current password, in addition to the user specified password.

The device supplier shall provide this or another password in documentation shipped with the device, so that the user is able to set up their own password.

C.4.3.13 IDN S-0-0269 Storage mode

C.4.3.13.1 Attributes

Table C.20 shows the possible attributes for this IDN.

Table C.20 – Attributes of IDN S-0-0269

Attribute	Value
Name	Storage mode
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Manufacturer-specific
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.13.2 Description

The storage mode parameter (see Table C.21) setting determines whether data received over the service channel (UC channel) is stored temporarily (for example in RAM), or remanent (for example EEPROM). Which parameter are affected by the storage mode setting are to be defined by the device supplier in the device documentation.

Table C.21 – Structure of storage mode

Bit no.	Value	Description
15-1	—	(reserved)
0	—	Storage mode
	0	Data stored remanent
	1	Data stored not remanent

C.4.3.14 IDN S-0-0270 IDN list of selected backup operation data

C.4.3.14.1 Attributes

Table C.22 shows the possible attributes for this IDN.

Table C.22 – Attributes of IDN S-0-0270

Attribute	Value
Name	IDN list of selected backup operation data
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.14.2 Description

This IDN list is used to define a subset of the IDN list of backup operation data (S-0-0192) which should be stored into the non-volatile memory of the sub-device. The selectively backup working memory procedure command (S-0-0293), will only store the operation data of this IDN list into the non-volatile memory.

C.4.3.14.3 IDN S-0-0279 IDN list of password protected data

C.4.3.14.4 Attributes

Table C.23 shows the possible attributes for this IDN.

Table C.23 – Attributes of IDN S-0-0279

Attribute	Value
Name	IDN list of password protected data
Version	—
Length	4, variable (list sorted by IDN)
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.14.5 Description

The operation data of this IDN contains an IDN list, the operation data of which should be write-protected by means of S-0-0267 (Password).

Since this IDNs list must not be changed by an unauthorized user, this IDN always has to be write protected by means of S-0-0267 (Password), regardless of whether or not this parameter contains its own IDN (S-0-0279) as an entry of the IDN list.

C.4.3.15 IDN S-0-0293 Selectively backup working memory procedure command

C.4.3.15.1 Attributes

Table C.24 shows the possible attributes for this IDN.

Table C.24 – Attributes of IDN S-0-0293

Attribute	Value
Name	Selectively backup working memory procedure command
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.15.2 Description

When the master sets and enables the selectively backup working memory procedure command, all parameter programmed in the IDN list of selected backup operation data (S-0-0270) will be loaded from the device's "active memory" and stored into its non-volatile memory.

NOTE This procedure command will cause previously saved parameters to be overwritten.

C.4.3.16 IDN S-0-0326.x.00 Parameter checksum

C.4.3.16.1 Attributes

Table C.25 shows the possible attributes for this IDN.

Table C.25 – Attributes of IDN S-0-0326.x.00

Attribute	Value
Name	Parameter checksum
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.16.2 Description

After being switched on, the control unit is able to find out by comparing, whether the parameters or firmware have changed in the device. The device calculates the parameter checksum, when the IDN is read via the service channel.

For the calculation of the checksum, the parameters which are listed in S-0-0327.x.0 IDN list of checksum parameter are used. If S-0-0327.x.0 is not supported by the device, the IDN list S-0-0192 IDN-list of all backup operation data is taken for the checksum calculation.

C.4.3.17 IDN S-0-0327.x.00 IDN list of checksum parameter

C.4.3.17.1 Attributes

Table C.26 shows the possible attributes for this IDN.

Table C.26 – Attributes of IDN S-0-0327.x.00

Attribute	Value
Name	IDN list of checksum parameter
Version	—
Length	4, variable (list sorted by IDN)
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.17.2 Description

This IDN-list contains all ident numbers for the calculation of the parameter checksum. see S-0-0326.x.0 Parameter Checksum.

C.4.3.18 IDN S-0-0390 Diagnostic number

C.4.3.18.1 Attributes

Table C.27 shows the possible attributes for this IDN.

Table C.27 – Attributes of IDN S-0-0390

Attribute	Value
Name	Diagnostic number
Version	—
Length	4
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.18.2 Description

The operation data of this IDN contains detailed information about the diagnostic event with the highest priority which is currently active within the slave or sub-device.

In order to make this information language independent, a 16 bit status code is used to identify the specific diagnostic event. Using the status code, the operator interface has the ability to display diagnostic message text in languages which are not supported by the sub-device.

The prioritization of diagnostic events is described in Table C.28.

Table C.28 – Prioritization of diagnostic events

Diagnosis class (Bits 16-19)	Priority	Signaled (1)	Overwritten (2)
operational state	Priority 4 (lowest)	As long as relevant for presentation (for example until overwritten by the next information with the class operational state).	By any other diagnostic information with a higher priority. By diagnostic information with the same priority, which occurs subsequently.
procedure command specific state	Priority 3	As long as the procedure command is not reset by the master.	By any other diagnostic information with a higher priority. By diagnostic information with the same priority, which occurs subsequently.
warning (C2D)	Priority 2	As long as the corresponding threat is present.	By any other diagnostic information with a higher priority.
error (C1D)	Priority 1 (highest)	As long as S-0-0099 Reset class 1 diagnostic has not been executed after the cause for the corresponding failure has been removed	-

(1) Signaled – The diagnostic information is part of the operation data

(2) Overwritten – The corresponding diagnostic information is overwritten by other diagnostic information

If a diagnostic event gets inactive and there are no more active diagnostic events the operation data of this IDN has to contain the status code 0x0000.

A summary of all specified status codes are shown in the S-0-0390.

The structure of the diagnostic number is shown in Table C.29.

Table C.29 – Transitions of the password state machine

Bit no.	Value	Description
31-30	—	Interpretation of bits 29-0 (The bits 31-30 defines the interpretation of groups source type, class and status codes.)
	00	manufacturer specific status codes (bits 29-24 type and class defined by Type 19, bits 15-0 status codes defined by manufacturer)
	01	fully manufacturer specific (bits 29-0 are defined by manufacturer)
	10	(reserved)
	11	Standard (bits 29-0 are defined by Type 19)
29-24	—	Source type
	0x00	FSP Drive
	0x01	FSP IO
	0x02	GDP
	0x03	SCP
	0x04	CSoS
	0x05	FSP Encoder
	0x06	Safety Application
	0x07 … 0x3E	(reserved)
	0x3F	Unknown
23-20	—	(reserved)
19-16	—	Class
	0x00-0x09	(reserved)
	0x0A	Operational state (Priority 4 lowest) - Is used to inform about operational state related messages or other information (for example Drive HALT, Compatible replacement).
	0x0B	(reserved)
	0x0C	procedure command specific state (Priority 3) - Is used to inform about diagnostic events which occur during the execution of a procedure command.
	0x0D	(reserved)
	0x0E	Warning (C2D) – Priority 2
	0x0F	Error (C1D) – Priority 1 (highest)
15-00	—	Status code

C.4.3.19 IDN S-0-0420 Activate parametrization level procedure command (PL)

C.4.3.19.1 Attributes

Table C.30 shows the possible attributes for this IDN.

Table C.30 – Attributes of IDN S-0-0420

Attribute	Value
Name	Activate parametrization level procedure command (PL)
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.19.2 Description

By activating this procedure command the master instructs the sub-device to change from the operating level (OL) to the parameterization level (PL).

If the procedure command cannot be executed at this time, the sub-device generates the SVC error message 0x7012 - "Procedure command at this time not executable".

- The procedure command is acknowledged positively by the slave:
- After the sub-device state machine is decoupled from the communication state machine
- The sub-device has activated PL
- Bit 4 (Sub-device level) of the Device Status word has been set to 1 (parameterization level (PL) is active).

C.4.3.20 IDN S-0-0422 Exit parameterization level procedure command

C.4.3.20.1 Attributes

Table C.31 shows the possible attributes for this IDN.

Table C.31 – Attributes of IDN S-0-0422

Attribute	Value
Name	Exit parameterization level procedure command
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.20.2 Description

By activating this procedure command the master instructs the sub-device to change from the parameterization level (PL) to the operating level (OL).

After the activation of this procedure command the slave checks all corresponding parameters and turns all monitoring systems on. Necessary references shall be recovered by the control unit (for example by homing).

The procedure command is acknowledged positively by the sub-device:

- After the sub-device state machine is decoupled from the communication state machine
- After the sub-device has activated OL
- Bit 4 (Sub-device level) of the Device Status word has been set to 0 (operating level (OL) is active).
- When all corresponding parameters are checked without faults
- When the monitoring system has been switched on

The procedure command is acknowledged negatively if a fault has appeared during the checks. In this case all IDNs which have caused a fault are stored in the IDN list (S-0-0423) and the sub-device remains in parameterization level.

C.4.3.21 IDN S-0-0423 IDN-list of invalid data for parameterization level

C.4.3.21.1 Attributes

Table C.32 shows the possible attributes for this IDN.

Table C.32 – Attributes of IDN S-0-0423

Attribute	Value
Name	IDN-list of invalid data for parameterization level
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.21.2 Description

This IDN contains all IDNs which are considered as invalid by the sub-device during procedure command S-0-0422 (Exit parameterization level procedure command).

If the procedure command S-0-0422 is executed successfully the process data contains no IDNs.

C.4.3.22 IDN S-0-0425 Sub-device state machine control

C.4.3.22.1 Attributes

Table C.33 shows the possible attributes for this IDN.

Table C.33 – Attributes of IDN S-0-0425

Attribute	Value
Name	Sub-device state machine control
Version	—
Length	2
Display Format	Binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.22.2 Description

Table C.34 shows the structure of the sub-device state machine control.

Table C.34 – Structure of the sub-device state machine control

Bit no.	Value	Description
15-1	—	(reserved)
0	—	Decoupling state
	0	Coupled (The sub-device state machine is coupled with the communication state machine)
	1	Decoupled (The sub-device state machine shall be decoupled from the communication state machine. A transition of the communication state machine does not lead to a transition of the sub-device state machine.)

C.4.3.23 IDN S-0-0531 Dummy IDN

C.4.3.23.1 Attributes

Table C.35 shows the possible attributes for this IDN.

Table C.35 – Attributes of IDN S-0-0531

Attribute	Value
Name	Checksum for backup operation data
Version	—
Length	4
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.23.2 Description

The device calculates the checksum when the IDN is read via the service channel.

For the calculation of the checksum, the parameters which are saved in S-0-0192 are used.

C.4.3.24 IDN S-0-1300.x.1 Component Name

C.4.3.24.1 Attributes

Table C.36 shows the possible attributes for this IDN.

Table C.36 – Attributes of IDN S-0-1300.x.1

Attribute	Value
Name	Component Name
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.24.2 Description

The content of this IDN is manufacturer specific and contains the component name depending to the device, for example, motor, amplifier, power supply, bus coupler. The contents may be used for display purpose only.

C.4.3.25 IDN S-0-1300.x.2 Vendor Name

C.4.3.25.1 Attributes

Table C.37 shows the possible attributes for this IDN.

Table C.37 – Attributes of IDN S-0-1300.x.2

Attribute	Value
Name	Vendor Name
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.25.2 Description

This IDN contains the vendor name of the device.

C.4.3.26 IDN S-0-1300.x.3 Vendor Code

C.4.3.26.1 Attributes

Table C.38 shows the possible attributes for this IDN.

Table C.38 – Attributes of IDN S-0-1300.x.3

Attribute	Value
Name	Vendor Code
Version	—
Length	2
Display Format	unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.26.2 Description

The vendor code (see Table C.39) is a unique number assigned to each vendor and helps identifying a Type 19 device installed in the Type 19 network. The vendor shall apply for the vendor-code at the sercos user organisation.

Table C.39 – Vendor code

Value	Description
0x0000	Unregistered vendors
Other values	Registered vendors

C.4.3.27 IDN S-0-1300.x.4 Device Name

C.4.3.27.1 Attributes

Table C.40 shows the possible attributes for this IDN.

Table C.40 – Attributes of IDN S-0-1300.x.4

Attribute	Value
Name	Device Name
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.27.2 Description

The content of this IDN is manufacturer specific and identifies the device name published in vendor's price list.

C.4.3.28 IDN S-0-1300.x.5 Vendor Device ID

C.4.3.28.1 Attributes

Table C.41 shows the possible attributes for this IDN.

Table C.41 – Attributes of IDN S-0-1300.x.5

Attribute	Value
Name	Vendor Device ID
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.28.2 Description

The vendor device ID is a unique device ID managed by the vendor and identifies the component number.

C.4.3.29 IDN S-0-1300.x.6 Connected to sub-device

C.4.3.29.1 Attributes

Table C.42 shows the possible attributes for this IDN.

Table C.42 – Attributes of IDN S-0-1300.x.6

Attribute	Value
Name	Connected to sub-device
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.29.2 Description

If a device holds more than one sub-device and the component is assigned to a specific sub-device the operational data shows the assigned slave number.

C.4.3.30 IDN S-0-1300.x.7 Function revision

C.4.3.30.1 Attributes

Table C.43 shows the possible attributes for this IDN.

Table C.43 – Attributes of IDN S-0-1300.x.7

Attribute	Value
Name	Function revision
Version	—
Length	2
Display Format	Unsigned decimal
Min. input value	0
Max. input value	9999
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.30.2 Description

The function revision shall be adjusted on the case of functional corrections of this component. The manufacturer of the device guarantees the compatibleness. The function revision shall not to be checked on system startup.

C.4.3.31 IDN S-0-1300.x.8 Hardware Revision

C.4.3.31.1 Attributes

Table C.44 shows the possible attributes for this IDN.

Table C.44 – Attributes of IDN S-0-1300.x.8

Attribute	Value
Name	Hardware Revision
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.31.2 Description

This parameter contains the hardware revision of the device, therefore it can be used for the identification. The hardware revision includes the software revision S-0-1300.x.09 if it is not supported by the device. The hardware revision is specified by the manufacturer (for example 103).

C.4.3.32 IDN S-0-1300.x.9 Software Revision

C.4.3.32.1 Attributes

Table C.45 shows the possible attributes for this IDN.

Table C.45 – Attributes of IDN S-0-1300.x.9

Attribute	Value
Name	Software Revision
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.32.2 Description

This parameter contains the software or firmware version of the device, therefore it can be used for the identification. The software revision is specified by the manufacturer.

C.4.3.33 IDN S-0-1300.x.10 Firmware Loader Revision

C.4.3.33.1 Attributes

Table C.46 shows the possible attributes for this IDN.

Table C.46 – Attributes of IDN S-0-1300.x.10

Attribute	Value
Name	Firmware Loader Revision
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1

Attribute	Value
Scaling/resolution	—
Unit	—

C.4.3.33.2 Description

This parameter contains the firmware loader or boot loader revision which is implemented in the device. The firmware loader revision is specified by the manufacturer.

C.4.3.34 IDN S-0-1300.x.11 Order number

C.4.3.34.1 Attributes

Table C.47 shows the possible attributes for this IDN.

Table C.47 – Attributes of IDN S-0-1300.x.11

Attribute	Value
Name	Order number
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.34.2 Description

This parameter contains the order number of the device, therefore it can be used for the identification. The customer needs the order number to order this certain device. The order number is specified by the manufacturer.

C.4.3.35 IDN S-0-1300.x.12 Serial Number

C.4.3.35.1 Attributes

Table C.48 shows the possible attributes for this IDN.

Table C.48 – Attributes of IDN S-0-1300.x.12

Attribute	Value
Name	Serial Number
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.35.2 Description

This parameter contains the serial number of the device, therefore it can be used for the identification. The serial number is defined and assigned by the manufacturer that uniquely identifies each individual device (for example 1234567890).

C.4.3.36 IDN S-0-1300.x.13 Manufacturing date

C.4.3.36.1 Attributes

Table C.49 shows the possible attributes for this IDN.

Table C.49 – Attributes of IDN S-0-1300.x.13

Attribute	Value
Name	Manufacturing date
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.36.2 Description

This IDN contains the manufacturing date and time of the component (see Figure C.4). The information is provided by a text string formatted as described within ISO 8601 (extended format):

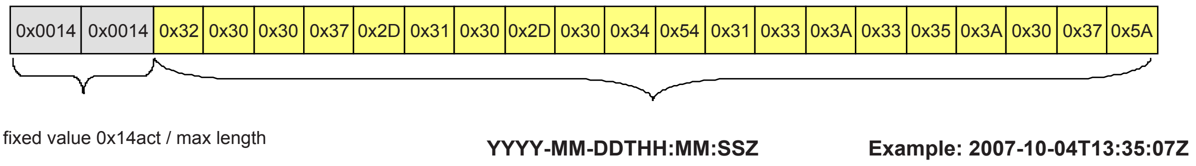

Figure C.4 – Structure of Date information

Date information is separated by hyphen "-" and time information by colon ":". Date and time are divided by a "T" character. The resulting text is terminated with Z (time zone UTC) with a fixed length of 20 characters.

C.4.3.37 IDN S-0-1300.x.14 QA Date

C.4.3.37.1 Attributes

Table C.50 shows the possible attributes for this IDN.

Table C.50 – Attributes of IDN S-0-1300.x.14

Attribute	Value
Name	QA Date
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.37.2 Description

This IDN contains the date and time when the quality assurance test has been performed for this component (see Figure C.5). Quality assurance is one of the final product test to ensure that the product keeps the desired quality level.

The information is provided by a text string formatted as described within ISO 8601 (extended format):

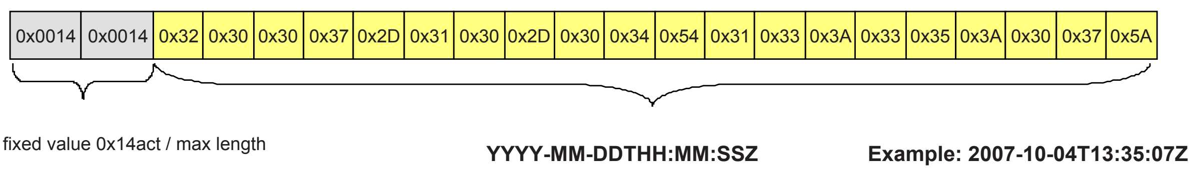

Figure C.5 – Structure of QA date information

Date information is separated by hyphen "-" and time information by colon ":". Date and time are divided by a "T" character. The resulting text is terminated with Z (time zone UTC) with a fixed length of 20 characters.

C.4.3.38 IDN S-0-1300.x.20 Operational hours

C.4.3.38.1 Attributes

Table C.51 shows the possible attributes for this IDN.

Table C.51 – Attributes of IDN S-0-1300.x.20

Attribute	Value
Name	Operational hours
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1 h
Unit	h

C.4.3.38.2 Description

This parameter contains the operational hours of the component. The device shall store this data in a retain memory.

C.4.3.39 IDN S-0-1300.x.21 Service Date

C.4.3.39.1 Attributes

Table C.52 shows the possible attributes for this IDN.

Table C.52 – Attributes of IDN S-0-1300.x.21

Attribute	Value
Name	Service Date
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.39.2 Description

This IDN contains the date and time of the last service maintenance for the device, for example firmware update (see Figure C.6).

The information is provided by a text string formatted as described within ISO 8601 (extended format):

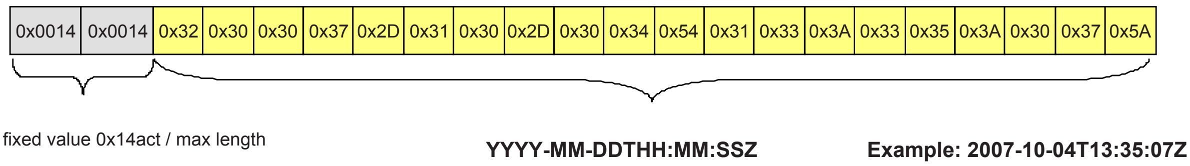

Figure C.6 – Structure of Service date information

Date information is separated by hyphen "-" and time information by colon ":". Date and time are divided by a "T" character. The resulting text is terminated with Z (time zone UTC) with a fixed length of 20 characters.

C.4.3.40 IDN S-0-1300.x.22 Calibration Date

C.4.3.40.1 Attributes

Table C.53 shows the possible attributes for this IDN.

Table C.53 – Attributes of IDN S-0-1300.x.22

Attribute	Value
Name	Calibration Date
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.40.2 Description

This IDN contains the date and time of the last calibration of the device (see Figure C.7). A service engineer shall write the current date and time into this IDN after a calibration service is done. At the same time, the date and time of the next required calibration date shall be written to S-0-1300.x.23.

The information is provided by a text string formatted as described within ISO 8601 (extended format):

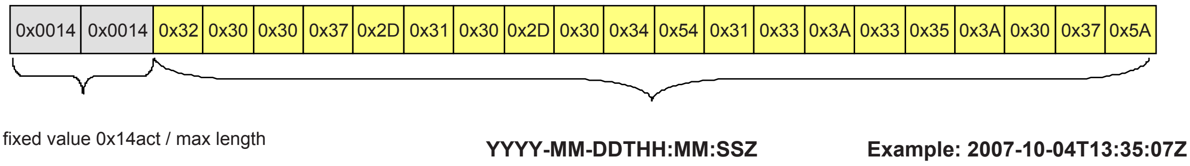

Figure C.7 – Structure of Calibration date information

Date information is separated by hyphen "-" and time information by colon ":". Date and time are divided by a "T" character. The resulting text is terminated with Z (time zone UTC) with a fixed length of 20 characters.

C.4.3.41 IDN S-0-1300.x.23 Calibration Due Date

C.4.3.41.1 Attributes

Table C.54 shows the possible attributes for this IDN.

Table C.54 – Attributes of IDN S-0-1300.x.23

Attribute	Value
Name	Calibration Due Date
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.41.2 Description

This IDN contains the date and time of the next calibration service of the device (see Figure C.8). After a service engineer has performed a calibration, the current date and time shall be written to S-0-1300.x.22 and the date and time of the next required calibration date shall be written to this IDN.

The information is provided by a text string formatted as described within ISO 8601 (extended format):

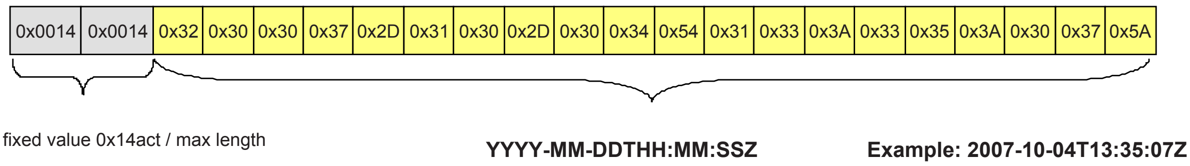

Figure C.8 – Structure of Calibration due date information

Date information is separated by hyphen "-" and time information by colon ":". Date and time are divided by a "T" character. The resulting text is terminated with Z (time zone UTC) with a fixed length of 20 characters.

C.4.3.42 IDN S-0-1301 List of GDP classes & Version

C.4.3.42.1 Attributes

Table C.55 shows the possible attributes for this IDN.

Table C.55 – Attributes of IDN S-0-1301

Attribute	Value
Name	List of GDP classes & Version
Version	—
Length	2, variable
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.42.2 Description

This parameter contains a list of the generic profile capabilities and the dedicated versions of the sub-device (see Table C.56).

- The bits 15 ... 8 indicate the GDP class,
- bits 7 ... 4 are reserved,
- bits 3 ... 0 indicate the version of this GDP class.

Table C.56 – Structure of GDP classes & version

Class Code Bits 15-8	Reserved Bits 7-4	Version Bits 3-0	Class name	Description
0x00	0x0	0x1	—	Not used
0x01	0x0	0x1	GDP_Basic	The generic profile shall be supported by each sub-device
0x02	0x0	0x1	GDP_Id	Identification
0x03	0x0	0x1	GDP_QS	Quality System
0x04	0x0	0x1	GDP_Rev	Revision
0x05	0x0	0x1	GDP_LNg	Language
0x06	0x0	0x1	GDP_StM	State Machine
0x07	0x0	0x1	GDP_CKs	Checksum
0x08	0x0	0x1	GDP_CKsUser	Checksum User
0x09	0x0	0x1	GDP_BKP	Backup
0x0A	0x0	0x1	GDP_BKPAdv	Backup Advanced
0x0B	0x0	0x1	GDP_DiagT	Diagnosis Trace
0x0C	0x0	0x1	GDP_DiagTAdv	Diagnosis Trace Advanced
0x0D	0x0	0x1	GDP_PWD	Password
0x0E-0xF	0x0	0x1	(reserved)	For future extensions
0x10	0x0	0x1	GDP_RST	Restart
0x11	0x0	0x1	GDP_CIPSafetyDev	CIP Safety Device
0x12-0xFF	0xn	0xn	(reserved)	For future extensions

C.4.3.43 IDN S-0-1302.x.1 FSP Type & Version

C.4.3.43.1 Attributes

Table C.57 shows the possible attributes for this IDN.

Table C.57 – Attributes of IDN S-0-1302.x.1

Attribute	Value
Name	FSP Type & Version
Version	—
Length	4
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.43.2 Description

The FSP Type & Version indicates the function specific type and the function dependent version of the resource as shown in Table C.58.

Table C.58 – Coding of S-1302.x.01

Bit No.	Value	Description
31	—	P/S
	0	Defined by Type 19
	1	Defined by manufacturer
30-16	—	Type 19 FSP types
	0x00	(reserved)
	0x01	FSP IO
	0x02	FSP Drive
	0x03	FSP Encoder
	0x04-0x7FFF	(reserved)
15-0	—	Version
	0x0000	(reserved)
	0x0001	First version

C.4.3.44 IDN S-0-1302.x.2 Function groups

C.4.3.44.1 Attributes

Table C.59 shows the possible attributes for this IDN.

Table C.59 – Attributes of IDN S-0-1302.x.2

Attribute	Value
Name	Function groups
Version	—
Length	4, variable (list sorted by SI)
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.44.2 Description

The operation data of this IDN contains a list of all instanced function groups.

This IDN is only present in case of a modular structured device according to for example FSP_IO. In case of a resource FSP IO this IDN is a list of IO function groups of FSP_IO. The elements of this list contain IDNs. The structure instance shows the slot number of the module (structure instance = module position), the structure element (SE) is always = 0. If the leftmost component is the bus coupler this list starts with the FG bus coupler (S-0-1500 .0.0). Otherwise the list starts with the leftmost IO-module at slot number 1, for example S-0-15xx.1.0. If a module contains more than one IO function group every IO function group has to be designated.

The mapping of data into the container InputData S-0-1500.x.09 and container OutputData S-0-1500.x.05 occurs in sequence of the list entries (see Figure C.9).

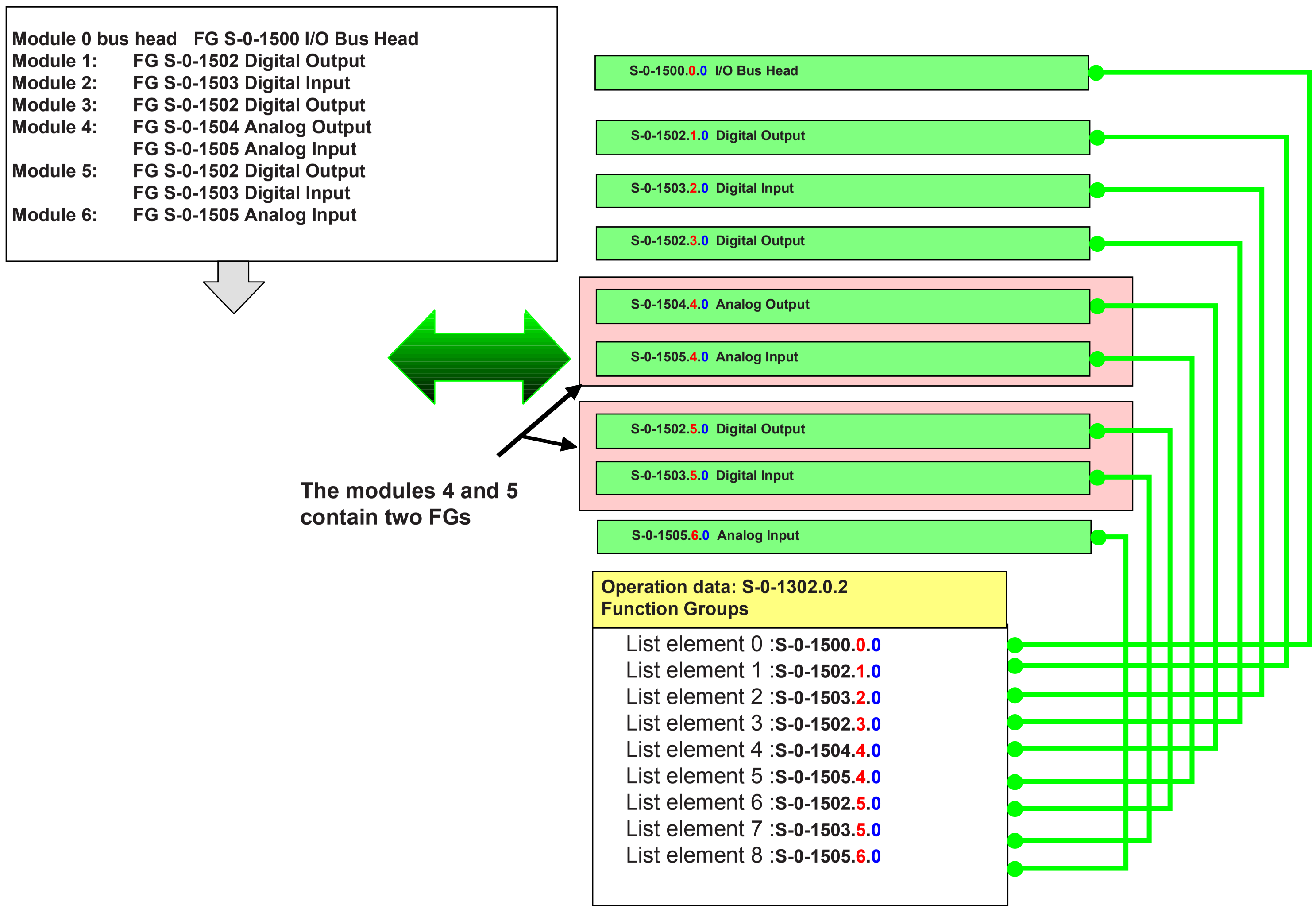

Figure C.9 – Mapping of data into the InputData and OutputData container

C.4.3.45 IDN S-0-1302.x.3 Application Type

C.4.3.45.1 Attributes

Table C.60 shows the possible attributes for this IDN.

Table C.60 – Attributes of IDN S-0-1302.x.3

Attribute	Value
Name	Application Type
Version	—
Length	1, variable
Display Format	Text
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.45.2 Description

The operation data of the application type contains the type of the sub-device application (for example: main spindle drive, round axis, X axis, etc.) The user can program this parameter if desired.

C.4.3.46 IDN S-0-1303.0.1 Diagnosis trace configuration

C.4.3.46.1 Attributes

Table C.61 shows the possible attributes for this IDN.

Table C.61 – Attributes of IDN S-0-1303.0.1

Attribute	Value
Name	Diagnosis trace configuration
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.46.2 Description

The IDN S-0-1303.0.01 describes the configured extension of the diagnosis trace listed in S-0-1303.0.12 to S-0-1303.0.127. This IDN shows a list of the additional IDNs which are added to the diagnosis trace in the same sequence (for example S-0-1500.x.33 Current IO Diagnosis Message in case of FSP IO).

The contents of IDN S-0-1303.0.01 are defined by the manufacturer.

C.4.3.47 IDN S-0-1303.0.2 Diagnosis trace control

C.4.3.47.1 Attributes

Table C.62 shows the possible attributes for this IDN.

Table C.62 – Attributes of IDN S-0-1303.0.2

Attribute	Value
Name	Diagnosis trace control
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Not defined
Conversion factor	1
Scaling/resolution	—
Unit	—

C.4.3.47.2 Description

This parameter controls the Diagnosis trace flow as shown in Table C.63.

Table C.63 – Coding of S-1303.0.02

Bit number	Value	Description
15	—	organization (master commands the organization of diagnosis buffer)
	0	Ring
	1	List
14-8	—	(reserved)
7-4	—	Threshold of Diagnosis class corresponds to Bits 19..16 S-0-0390 only diagnosis with classes equal or higher will be captured
3-2	—	(Reserved)
1		Reset buffer (The buffer is deleted and at the same time the status information. If the Level Indicator is supported, it is set to 1.)
0	—	Recording
	0	Capture (Diagnostic trace is capturing diagnostic events.)
	1	Freeze (Diagnostic trace does not capture any diagnostic events. Hence a consistent evaluation of the diagnostic buffers is possible.)

C.4.3.48 IDN S-0-1303.0.3 Diagnosis trace state

C.4.3.48.1 Attributes

Table C.64 shows the possible attributes for this IDN.

Table C.64 – Attributes of IDN S-0-1303.0.3

Attribute	Value
Name	Diagnosis trace state
Version	—
Length	2
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.48.2 Description

This parameter shows the state of the diagnosis trace flow as shown in Table C.65.

Table C.65 – Coding of S-1303.0.2

Bit number	Value	Description
15	—	organization (shows the organization of the buffer)
	0	ring (trace flow is organized as a ring)
	1	list (trace flow is organized as a list)
14-4	—	(reserved)
3	—	Overrun, indicates that the diagnostic trace buffer has been exceeded, i.e. – in case of a list new entries are lost, – in case of a ring old entries are overwritten.
	0	overrun has not occurred
	1	overrun has occurred
2-1	—	level indicator
	3	Full (Indicates that the diagnostic trace buffer is full but has not overrun yet.)
	2	Filled (Indicates that the diagnostic trace buffer contains at least one entry.)
	1	Empty (Indicates that the diagnostic trace buffer contains no entries.)
	0	— indicator level not present
0	—	Recording
	0	Capture
	1	Freeze

C.4.3.49 IDN S-0-1303.0.10 Diagnosis trace buffer no1

C.4.3.49.1 Attributes

Table C.66 shows the possible attributes for this IDN.

Table C.66 – Attributes of IDN S-0-1303.0.10

Attribute	Value
Name	Diagnosis trace buffer no1
Version	—
Length	4, variable
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	unit of the parameter

C.4.3.49.2 Description

This parameter contains a row of the diagnosis trace. The information is organized as a ring buffer or list.

Within this buffer each diagnostic event which occurs within the sub-device is recorded in form of the S-0-0390 (Diagnostic Number).

An event which is recorded within the diagnosis trace does not necessarily has to be indicated via S-0-0390 (Diagnostic Number) due to diagnostic priority reasons

C.4.3.50 IDN S-0-1303.0.11 Diagnosis trace buffer no2

C.4.3.50.1 Attributes

Table C.67 shows the possible attributes for this IDN.

Table C.67 – Attributes of IDN S-0-1303.0.11

Attribute	Value
Name	Diagnosis trace buffer no2
Version	—
Length	8, variable
Display Format	Type 19 time
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	Unit of the parameter

C.4.3.50.2 Description

This parameter contains a row of the diagnosis trace, and contains information organized as a ring buffer or list. The configuration is fixed to Type 19 current time S-0-1305.0.01 of the related diagnostic number.

C.4.3.51 IDN S-0-1303.0.12 Diagnosis trace buffer no3

C.4.3.51.1 Attributes

Table C.68 shows the possible attributes for this IDN.

Table C.68 – Attributes of IDN S-0-1303.0.12

Attribute	Value
Name	Diagnosis trace buffer no3
Version	—
Length	Length of parameter, variable
Display Format	Format of parameter
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	Unit of parameter

C.4.3.51.2 Description

This parameter contains a row of the diagnosis trace, and contains information organized as a ring buffer or list. The configuration is fixed to Type 19 current time S-0-1305.0.01 of the related diagnostic number. This IDN shows the next row of diagnosis trace.

C.4.3.52 IDN S-0-1305.0.1 Type 19 current time

C.4.3.52.1 Attributes

Table C.69 shows the possible attributes for this IDN.

Table C.69 – Attributes of IDN S-0-1305.0.1

Attribute	Value
Name	Type 19 current time
Version	—
Length	8
Display Format	Type 19 time
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.52.2 Description

This parameter contains the current Type 19 time in IEC 61588 format (Table C.70). The sub-device shall mark events with this time (for example S-0-1305.0.01 Type 19 current time marks the S-0-1303 Diagnostic trace with time stamps).

This parameter shall become active within start-up of the device with the initial value of 0 indicating that Type 19 current time has not been set.

The IEC has determined that the value = 0x0 corresponds to 1970-01-01, 00:00, 0 s, 0 ns.

If SCP_SysTime is activated in the slave, then this parameter shall be written protected in CP3 and CP4.

Table C.70 – Structure of Type 19 time

Bit no.	Value	Description
63-32		Seconds
31-0		Nanoseconds

C.4.3.53 IDN S-0-1305.0.2 Type 19 current fine time

C.4.3.53.1 Attributes

Table C.71 shows the possible attributes for this IDN.

Table C.71 – Attributes of IDN S-0-1305.0.2

Attribute	Value
Name	Type 19 current fine time
Version	—
Length	4
Display Format	Unsigned decimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	0,001 µs
Unit	µs

C.4.3.53.2 Description

This parameter contains the low 32 bit of the current Type 19 time in IEC 61588 format. The sub-device may use this to mark events with this time.

If SCP_SysTime is activated in the slave, then this parameter shall be write protected in CP3 and CP4.

C.4.3.54 IDN S-0-1310 IDN-list of operation data changed from default

C.4.3.54.1 Attributes

Table C.72 shows the possible attributes for this IDN.

Table C.72 – Attributes of IDN S-0-1310

Attribute	Value
Name	IDN-list of operation data changed from default
Version	—
Length	4, variable
Display Format	IDN
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Always
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.54.2 Description

This IDN list is managed by the device and contains a list of all IDN's which operation data is changed from default.

C.4.3.55 IDN S-0-1350 Reboot Device

C.4.3.55.1 Attributes

Table C.73 shows the possible attributes for this IDN.

Table C.73 – Attributes of IDN S-0-1350

Attribute	Value
Name	Reboot device
Version	—
Length	2
Display Format	binary
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	OL
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.55.2 Description

The master uses this procedure command to initialize a reboot of the whole device.

- If S-0-1350.0.1 Reboot Device Delay exists, the device waits for the time period, which is defined in S-0-1350.0.1, before the reboot is performed.
- If S-0-1350.0.1 Reboot Device Delay does not exist, the device waits for min. 2 seconds and max. 30 seconds before the reboot is performed. This gives the master the possibility to reboot several devices in the ring.

NOTE The reboot of a device leads to the fact that the Type 19 communication is interrupted and the communication phase up has to be performed again.

C.4.3.56 IDN S-0-1399.0.1 Test IDN Diagnostic Event

C.4.3.56.1 Attributes

Table C.74 shows the possible attributes for this IDN.

Table C.74 – Attributes of IDN S-0-1310

Attribute	Value
Name	Test IDN Diagnostic Event
Version	—
Length	4
Display Format	Hexadecimal
Min. input value	—
Max. input value	—
Positions after decimal point	0
Write protection	Never
Conversion factor	1
Scaling/resolution	1
Unit	—

C.4.3.56.2 Description

Writing this IDN causes an application specific diagnostic event within the sub-device (Table C.75). The virtually caused diagnostic shall be treated within the sub-device as a real diagnostic event. That means that the virtually caused diagnostic event has an impact on all supported Type 19 diagnostic mechanisms which are amongst others (not a complete list):

Communication Profile (SCP)

- S-0-0014 (Interface Status)
- S-0-1028 (Error counter MST-P/S)
- S-0-1035.0.0 (Error counter Port1 and Port2)
- S-0-1045 (Device Status (S-DEV))
- S-0-1050.x.08 (Connection Control)
- S-0-1050.x.12 (Error Counter Data Losses)

Generic Device Profile (GDP)

- S-0-0095 (Diagnostic Message)
- S-0-0390 (Diagnostic Number)
- S-0-1303 (Diagnostic Trace)
- Type 19 LED

Function Specific Profile Drive (FSP Drive)

- S-0-0011 (Class 1 diagnostic (C1D))
- S-0-0012 (Class 2 diagnostic (C2D))
- S-0-0135 (Drive Status)

Function Specific Profile IO (FSP IO)

- S-0-1500.x.02 (IO Status)
- S-0-1500.x.32 (IO Diagnostic Message)
- IO_FG.x.17 (DIAGIN) (manufacturer specific)

The structure of the operation data of this IDN is almost identical to the operation data of the S-0-0390 (Diagnostic Number) (except bits 30-31). Writing this IDN with bit 31 set shall cause a diagnostic event with the source type, class and source code specified within the operation data.

Furthermore diagnostic events caused by this IDN shall be resettable by writing this IDN containing the diagnostic event which should be reset and bit 31 set to 0.

Table C.75 – Structure of Test IDN Diagnostic Event

Bit no.	Value	Description
31	—	Activation level (The bit 31 defines whether the diagnostic event is activated or deactivated.)
	0	Deactivation of the diagnostic event (Writing the IDN with this bit set to 0 deactivates the corresponding diagnostic event within the slave)
	1	Activation of the diagnostic event (Writing the IDN with this bit set activates the corresponding diagnostic event within the slave)
30	—	Interpretation the status code (The bit 30 defines the interpretation of the status code)
	0	manufacturer specific status codes (Bits 15-0 status codes defined by manufacturer)
	1	Standard (Status codes are defined by Type 19)
29-24		Source type (The encoding is identical to S-0-0390 (Diagnostic Number) Bits 24-29)
23-20		(reserved)
19-16		Class (The encoding is identical to S-0-0390 Diagnostic Number Bits 16-19)
15-0		Status code

C.5 GDP status codes

The following article defines the status codes for the GDP, which are used for the language-neutral presentation of diagnostic information of Type 19 slave devices.

The status codes, which are listed in Table C.76 and Table C.77, are grouped into different categories according to their fix assigned diagnosis class.

Table C.76 – Status codes with the diagnosis class "operational state"

Code (hex)	Description
A010	The device has been restarted (Power on)
A100	Wrong password entered
A110	Password write protection deactivated
A120	Password changed
A200	Diagnostic trace started
A210	Diagnostic trace stopped
A220	Diagnostic trace buffer overrun
A300	Test IDN written

Table C.77 – Status codes with the diagnosis class "procedure command specific state"

Code (hex)	Description
200	S-0-0422 Exit parameterization level procedure command
201	Incorrect or incomplete set of parameters (see S-0-0423 IDN-list of invalid data for parameterization level)
202	Parameter limit violation (see S-0-0423 IDN-list of invalid data for parameterization level)
203	Parameter conversion error (see S-0-0423 IDN-list of invalid data for parameterization level)
400	S-0-0420 Activate parametrization level procedure command (PL)
401	Switching to parametrization level is not possible
500	S-0-0099 Reset class 1 diagnostic
700	S-0-0262 Load defaults procedure command
2200	S-0-0264 Backup working memory procedure command
2300	S-0-0263 Load working memory procedure command
2400	S-0-0293 Selectively backup working memory procedure command

Bibliography

IEC 61131 (all parts), *Programmable controllers*

IEC 61158-1:2014, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-3-19:2014, *Industrial communication networks – Fieldbus specifications – Part 3-19: Data-link layer service definition – Type 19 elements*

IEC 61158-5-16:2007, *Industrial communication networks – Fieldbus specifications – Part 5-16: Application layer service definition – Type 16 elements*

IEC 61158-5-19:2014, *Industrial communication networks – Fieldbus specifications – Part 5-19: Application layer service definition – Type 19 elements*

IEC 61158-6-16:2007, *Industrial communication networks – Fieldbus specifications – Part 6-16: Application layer protocol specification – Type 16 elements*

IEC 61158-6-19:2014, *Industrial communication networks – Fieldbus specifications – Part 6-19: Application layer protocol specification – Type 19 elements*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

IEC 61800-7 (all parts), *Adjustable speed electrical power drive systems – Generic interface and use of profiles for power drive systems*

IEC 61800-7-20x[1], *Adjustable speed electrical power drive systems – Part 7-20x: Generic interface and use of profiles for power drive systems – Profile type x specification*

---

1 At present, these subparts are IEC 61800-7-201, IEC 61800-7-202, IEC 61800-7-203 and IEC 61800-7-204.

This page deliberately left blank

This page deliberately left blank