

INTERNATIONAL STANDARD



**Electronic railway equipment – Train communication network (TCN) –
Part 2-8: TCN conformance test**

IECNORM.COM : Click to view the full PDF of IEC 61375-2-8:2021



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2021 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, ...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC online collection - oc.iec.ch

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 18 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IECNORM.COM : Click to view the full PDF IEC 6175-28:2021

INTERNATIONAL STANDARD



**Electronic railway equipment – Train communication network (TCN) –
Part 2-8: TCN conformance test**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 45.060.01

ISBN 978-2-8322-1035-5

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD.....	14
INTRODUCTION.....	16
1 Scope.....	17
2 Normative references	17
3 Terms, definitions, symbols and abbreviated terms.....	19
3.1 Terms and definitions.....	19
3.2 Symbols and abbreviated terms	19
4 Conformance test: approach, requirements and boundaries.....	20
4.1 Approach	20
4.1.1 General	20
4.1.2 Requirements	20
4.1.3 Requirements declaration statements for an Item Under Test (IUT)	22
4.2 Boundaries	23
4.2.1 General	23
4.2.2 Basic interconnection tests	24
4.2.3 Capability tests.....	24
4.2.4 Behaviour tests.....	25
4.2.5 Conformance resolution tests	25
4.2.6 Interpretation of clauses/subclauses and statements	26
4.2.7 Relation to interoperability	27
4.2.8 Relation to performance test.....	28
4.2.9 Definition of test cases	28
4.3 Conformance assessment process outline	29
4.3.1 General	29
4.3.2 Analysis of results, outcomes and verdicts.....	29
4.4 Mapping of IUT types to conformance test suites	30
5 Basic interface test.....	31
5.1 Scope	31
5.2 PICS pro-forma	31
5.2.1 General	31
5.2.2 PICS Tables	31
5.3 Testing framework	33
5.4 Physical layer test.....	33
5.4.1 Test purpose	33
5.4.2 Inspection of the Hardware Ethernet interface	33
5.4.3 Inspection of the mechanical Ethernet interface construction	33
5.4.4 Check of electrical Ethernet interface design	34
5.4.5 Check of Ethernet interface characteristics	34
5.4.6 Ethernet signal wave form test (IEEE standard)	34
5.4.7 Ethernet signal wave form test (amplified signals, optional)	34
5.4.8 Auto-Crossover- and Auto-Sensing-Test (only for switches).....	34
5.5 Basic communication protocol test.....	35
5.5.1 Test purpose	35
5.5.2 Simple communication test	35
5.5.3 UDP rising payload test (only for End Devices).....	35
5.5.4 TCP rising payload test (only for End Devices)	35

5.5.5	UDP long term test (only for End Devices)	35
5.6	Communication reliability tests	35
5.6.1	Test purpose	35
5.6.2	Burst immunity test	35
5.6.3	Ethernet interface isolation test	36
6	Conformance test of End Device	36
6.1	General	36
6.2	Related requirements	37
6.3	IUT and Test scope	37
6.4	PICS Pro-forma	38
6.4.1	General	38
6.4.2	PICS Tables	38
6.5	Testing framework	44
6.6	Test Suite IEC 61375-2-3	46
6.6.1	Test case TCCT23ED_001	46
6.6.2	Test case TCCT23ED_002	47
6.6.3	Test case TCCT23ED_003	49
6.6.4	Test case TCCT23ED_004	50
6.6.5	Test case TCCT23ED_005	51
6.6.6	Test case TCCT23ED_006	52
6.6.7	Test case TCCT23ED_007	53
6.6.8	Test case TCCT23ED_008	54
6.6.9	Test case TCCT23ED_009	55
6.7	Test Suite IEC 61375-2-5	55
6.7.1	Test case TCCT25ED_001	55
6.7.2	Test case TCCT25ED_002	56
6.7.3	Test case TCCT25ED_003	57
6.7.4	Test case TCCT25ED_004	58
6.7.5	Test case TCCT25ED_005	59
6.7.6	Test case TCCT25ED_006	60
6.7.7	Test case TCCT25ED_007	61
6.7.8	Test case TCCT25ED_008	62
6.7.9	Test case TCCT25ED_009	63
6.7.10	Test case TCCT25ED_010	64
6.8	Test Suite IEC61375-3-4	65
6.8.1	Test case TCCT34ED_001	65
6.8.2	Test case TCCT34ED_002	66
6.8.3	Test case TCCT34ED_003	67
6.8.4	Test case TCCT34ED_004	68
6.8.5	Test case TCCT34ED_005	69
6.8.6	Test case TCCT34ED_006	70
6.8.7	Test case TCCT34ED_007	71
6.8.8	Test case TCCT34ED_008	72
6.8.9	Test case TCCT34ED_009	73
6.8.10	Test case TCCT34ED_010	74
6.8.11	Test case TCCT34ED_011	75
6.8.12	Test case TCCT34ED_012	76
6.8.13	Test case TCCT34ED_013	77
6.8.14	Test case TCCT34ED_014	78

6.8.15	Test case TCCT34ED_015	79
6.8.16	Test case TCCT34ED_016	80
6.8.17	Test case TCCT34ED_017	81
6.8.18	Test case TCCT34ED_018	82
6.8.19	Test case TCCT34ED_019	83
6.8.20	Test case TCCT34ED_020	84
6.8.21	Test case TCCT34ED_021	85
6.8.22	Test case TCCT34ED_022	86
7	Conformance test of ECN-equipped consist	86
7.1	Scope	86
7.2	Related requirements	86
7.3	PICS Pro-forma	86
7.3.1	General	86
7.3.2	PICS tables	87
7.4	Test cases	93
7.4.1	General	93
7.4.2	Recovery in case of network failure	94
7.4.3	IP address assignment via DHCP	96
7.4.4	Name (TCN-URI address) resolution via DNS	97
7.4.5	Switch basic functions	98
7.4.6	Priority levels	99
7.4.7	Ingress rate limiting	99
7.4.8	Egress rate shaping	100
7.4.9	Untagged/tagged frames	100
7.4.10	Switching and routing performance (optional)	101
7.4.11	NTP support	102
7.4.12	Switch management	102
7.4.13	Network management via SNMP	103
7.4.14	ECSP Interface (optional)	104
7.4.15	TTDB manager interface (optional)	105
7.4.16	DNS TCN interface (optional)	105
7.4.17	ETBN control interface (optional)	106
8	Conformance test of ETBN	106
8.1	Test scope	106
8.1.1	General	106
8.1.2	References	106
8.2	PICS Pro-forma	107
8.2.1	General	107
8.2.2	PICS tables	107
8.3	Test cases	115
8.3.1	Testing framework	115
8.3.2	Common Consist Topologies	118
8.3.3	Test Suite: ETB port settings	121
8.3.4	Test Suite: ETB Line Status and Link Aggregation management	123
8.3.5	Test Suite: TTDP HELLO format and content	130
8.3.6	Test Suite: Unicast address assignment and unicast routing	134
8.3.7	Test Suite: IP Multicast forwarding	137
8.3.8	Test Suite: TOPOLOGY frame format and basic functionality	148
8.3.9	Test Suite: Physical topology management and calculation	152

8.3.10	Test Suite: Logical connectivity management and calculation	156
8.3.11	Test Suite: Failing/loss of node(s).....	162
8.3.12	Test Suite: Late/recovered node(s)	166
9	Conformance test of TRDP	172
9.1	General.....	172
9.2	Related requirements.....	173
9.3	SUT and Test scope	173
9.4	PICS Pro-forma	173
9.4.1	General	173
9.4.2	PICS Tables	173
9.5	Message Data test.....	187
9.5.1	Testing framework	187
9.5.2	Communication Model Test.....	188
9.5.3	Communication Pattern and Addressing Test.....	192
9.5.4	MD-PDU Test	204
9.5.5	TRDP Layer Service Primitives Test	206
9.5.6	TRDP Layer Filtering Rules Test.....	209
9.5.7	Caller TRDP Layer Test.....	214
9.5.8	Replier TRDP Layer Test.....	234
9.5.9	TCP Connection Handling Test.....	249
9.5.10	Message Data Echo Server Test.....	257
9.6	Process Data test	258
9.6.1	General	258
9.6.2	Test setup	258
9.6.3	PD push pattern test cases	259
9.6.4	PD pull pattern test cases	280
10	SDTv2 Conformance Test.....	298
10.1	Scope	298
10.1.1	General	298
10.1.2	Related requirements	298
10.1.3	Terms and abbreviations	298
10.1.4	Test scope.....	298
10.2	PICS Pro-forma	299
10.2.1	General	299
10.2.2	PICS Tables	299
10.3	SDSRC- SAFE DATA SOURCE	302
10.3.1	Purpose	302
10.3.2	Inspection of the SID Structure	302
10.3.3	Inspection of the Vital Data Packet	303
10.3.4	Exclusivity	304
10.3.5	SDSRC performance	305
10.4	SDSINK – SAFE DATA SINK	305
10.4.1	General	305
10.4.2	VDP sampling.....	305
10.4.3	VDP Integrity Check	305
10.4.4	Sink Time Supervision	306
10.4.5	Guard Time Supervision	306
10.4.6	Latency Monitoring	306
10.4.7	Channel Monitoring.....	307

10.4.8	SDTv2 Application Interface	307
10.4.9	Diagnostics and statistics	308
11	Conformance test of ETB-equipped consist	308
11.1	Scope	308
11.2	Abstract consist tester architecture	309
11.3	Consist test stages	311
11.4	PICS Pro-forma	311
11.4.1	General	311
11.4.2	PICS Tables	311
11.5	Related requirements	314
11.6	Testing configurations	314
11.6.1	General	314
11.6.2	Testing configuration TC1	315
11.6.3	Testing configuration TC2	316
11.6.4	Testing configuration TC3	316
11.7	Consist tester operation	317
11.7.1	General	317
11.7.2	ECS addressing	317
11.7.3	Test Stimuli	317
11.7.4	Expected result verification	320
11.7.5	Test execution timing considerations	321
11.8	Basic test	321
11.8.1	General	321
11.8.2	Test sequence for TTDB computation (middle consist, same orientation)	321
11.8.3	Test sequence for TTDB computation (middle consist, inverse orientation)	321
11.8.4	Test sequence for TTDB computation (end consist, no traction, same orientation)	322
11.8.5	Test sequence for TTDB computation (end consist, no traction, inverse orientation)	322
11.8.6	Test sequence for TTDB computation (end consist, traction, same orientation)	323
11.8.7	Test sequence for TTDB computation (end consist, traction, inverse orientation)	323
11.8.8	Test sequence for TTDB computation (train lengthening and shortening)	324
11.9	Leading test	325
11.9.1	General	325
11.9.2	Test sequence for Leading (TE1.1 leading, CuT middle)	326
11.9.3	Test sequence for Leading (TE2 leading, CuT middle)	326
11.9.4	Test sequence for Leading (TE1.1 leading, CuT end)	326
11.9.5	Test sequence for Leading (CuT leading direction 1)	327
11.9.6	Test sequence for Leading (CuT leading direction 2)	327
11.9.7	Test sequence for Leading (Leading double request)	328
11.9.8	Test sequence for Leading (Leading conflict)	329
11.10	Inhibit test	329
11.10.1	General	329
11.10.2	Test sequence for Inhibit (single)	329
11.10.3	Test sequence for Inhibit (multiple)	330
11.11	Confirmation/correction test	331
11.11.1	General	331

11.11.2	Test sequence for confirmation	332
11.11.3	Test sequence for correction single middle vehicle	333
11.11.4	Test sequence for correction two middle vehicles	333
11.11.5	Test sequence for correcting three middle vehicles	334
11.11.6	Test sequence for correcting end vehicle	335
11.12	Sleep mode function (optional)	336
11.12.1	General	336
11.12.2	Test sequence for sleep mode	336
11.13	Performance test	337
11.13.1	General	337
11.13.2	Inauguration performance	337
11.13.3	ECSP performance	338
11.14	End-to-End Communication	339
11.15	Multiple ETB test (option)	339
11.15.1	General	339
11.15.2	Conformance test	340
Annex A (normative)	Test laboratory role and client role	341
A.1	Test laboratory and client role	341
A.1.1	General	341
A.1.2	Overview	341
A.2	Preparation for testing	342
A.2.1	General	342
A.2.2	General administrative steps	342
A.2.3	Agreement on test methods and selection of test suites	342
A.2.4	Exchange of documentation for conformance assessment	343
A.3	Test operation	344
A.3.1	General	344
A.3.2	Static conformance review	344
A.3.3	Selection of test cases and test parameterisation	345
A.3.4	Test campaign	345
A.4	Production of test reports	346
A.4.1	General	346
A.4.2	IUT conformance test report	347
A.4.3	Protocol conformance test report	347
Annex B (informative)	Instructions for filling the PICS pro-forma	349
B.1	General	349
B.2	Abbreviations	349
B.3	Reference column	349
B.4	Supported subclause column	350
B.5	Supported capability column	350
B.6	Requirement column	350
B.7	Implementation column	350
B.8	Parameter values columns	351
B.8.1	Allowed min.	351
B.8.2	Default value	351
B.8.3	Allowed max.	351
B.8.4	Implemented value	351
Annex C (informative)	Test instrumentation and dedicated test bed	352
C.1	Test instrumentation	352

C.1.1	Standard instrumentation – Test suites standard instrumentation.....	352
C.1.2	Test bed architecture	352
C.1.3	Consist tester specification	353
	Bibliography.....	356
	Figure 1 – Setup for burst immunity test.....	36
	Figure 2 – End Device conformance testing process	37
	Figure 3 – Consist topology and logical view of ECN.....	38
	Figure 4 – Testing framework architecture	44
	Figure 5 – General configuration for ECN-equipped consist test	94
	Figure 6 – Example of configuration for the test of network redundancy	94
	Figure 7 – Example of configuration of network redundancy with dual homing End Device interfaces	95
	Figure 8 – Example of the configuration for the test of IP address assignment via DHCP	97
	Figure 9 – Example of the configuration for the test of Name resolution via DNS	98
	Figure 10 – Example of the configuration for the test of NTP support.....	102
	Figure 11 – Example of the configuration for the test of Network management via SNMP	104
	Figure 12 – TTS1 One ETBN connects with test system (monitoring).....	116
	Figure 13 – TTS2 single ETBN connects with test system (one side simulation).....	116
	Figure 14 – TTS3 Single ETBN connects with test system (both sides simulation)	117
	Figure 15 – TTS4 Multiple ETBNs connect with test system (monitoring)	117
	Figure 16 – TTS5 multiple ETBNs connect with test system (one side simulation).....	118
	Figure 17 – TTS6 Multiple ETBN connects with test system (both sides simulation).....	118
	Figure 18 – Consist topology with single ETBN and single CN	119
	Figure 19 – Consist topology with multiple ETBNs, each connected to a separate CN.....	119
	Figure 20 – Consist topology with a single ETBN connected to multiple CNs	119
	Figure 21 – Consist topology with single CN with redundant ETBNs	120
	Figure 22 – Consist topology with multiple CNs with redundant ETBNs.....	120
	Figure 23 – Consist topology with multiple ETNs and CNs connected asymmetrically	120
	Figure 24 – Setup to verify full duplex mode on ETB ports	122
	Figure 25 – Example where IUT sends TOPOLOGY on line B (here X is B, Y is A)	124
	Figure 26 – Simulating Logical line down for single line setup, by letting simulator stop sending HELLO on line X (line A).....	125
	Figure 27 – Simulating Logical line down for multiple line setup, by letting simulator stop sending HELLO on line X (here line B)	125
	Figure 28 – Simulating one-way transmission error, by letting simulator send HELLO with recvXStatus FALSE (here recvBStatus FALSE)	126
	Figure 29 – Testing correct handling of loss of HELLO transmitted by IUT, and loss of HELLO sent to IUT.....	127
	Figure 30 – ETB line failover setup, the line where IUT initially forwards the data stream is referred to as 'X' (here line B).....	128
	Figure 31 – Topology for general multicast routing tests	138
	Figure 32 – Topology for testing multicast with ETBN with multiple CNs	140
	Figure 33 – Multicast test setup for consist with ECNs connected to different ETBNs.....	141

Figure 34 – Testing multicast with redundant ETBNs and single CN	142
Figure 35 – Multicast routing in consist with redundant ETBN and multiple CNs.....	144
Figure 36 – Testing multicast handling after reinauguation.....	145
Figure 37 – Test of multicast between local CNs on same ETBN when inaugurating	146
Figure 38 – Test of multicast between local CNs on different ETBNs when inaugurating	147
Figure 39 – Test ability to translate multicast destination address	147
Figure 40 – Testing transmission of connectivity vector	154
Figure 41 – Testing transmission of ETBN Vectors, number of ETBNs and ConnTableCrc32.....	155
Figure 42 – Testing loss of intermediate ETBNs/consists when inauguration is inhibited.....	162
Figure 43 – Test of losing intermediate ETBN in existence of "unknown intermediate"	163
Figure 44 – Test ability to handle loss of end node	164
Figure 45 – Test ability to handle loss of end node during lengthening.....	165
Figure 46 – Test ability to handle loss of end node when “late intermediate” is present.....	166
Figure 47 – Test environment	172
Figure 48 – System under Test	173
Figure 49 – TTS, IUT connected to the test system.....	259
Figure 50 – The switch in test topology	259
Figure 51 – TRDP telegram format.....	260
Figure 52 – Dataset of CONFTEST_PUSH_FORMAT.....	261
Figure 53 – Dataset of CONFTEST_UDP_PORT.....	262
Figure 54 – Dataset of CONFTEST_FCS	264
Figure 55 – Dataset of CONFTEST_FCS_REPLY	264
Figure 56 – Dataset of CONFTEST_PROTO_VERSION.....	266
Figure 57 – Dataset of CONFTEST_PROTO_VERSION_REPLY.....	266
Figure 58 – Dataset of CONFTEST_SIMU_TOPO	269
Figure 59 – Dataset of CONFTEST_IDU_TOPO.....	269
Figure 60 – Dataset of CONFTEST_DATA_LEN_1432	271
Figure 61 – Dataset of CONFTEST_DATA_LEN_0.....	271
Figure 62 – Dataset of CONFTEST_DATA_LEN_512.....	271
Figure 63 – Byte alignment test of 441 bytes length telegram data.....	273
Figure 64 – Byte alignment test of 442 bytes length telegram data.....	273
Figure 65 – Byte alignment test of 443 bytes length telegram data.....	273
Figure 66 – Dataset of CONFTEST_PUSH_FORMAT	275
Figure 67 – Dataset of CONFTEST_TIMEOUT_BEHAVIOR_ZERO	275
Figure 68 – Dataset of CONFTEST_TIMEOUT_BEHAVIOR_KEEP	276
Figure 69 – Dataset of CONFTEST_IP_FILTER	277
Figure 70 – Dataset of CONFTEST_IP_FILTER_REPLY	277
Figure 71 – DSCP(QOS) and TTL test telegram data	278
Figure 72 – Dataset of CONFTEST_REDUNDANT.....	279
Figure 73 – Dataset of CONFTEST_UNICAST	280
Figure 74 – Dataset of CONFTEST_PULL_FORMAT_REQUEST	282
Figure 75 – Dataset of CONFTEST_PULL_FORMAT_REPLY	283

Figure 76 – Dataset of CONFTEST_FCS_REQUEST	284
Figure 77 – Dataset of CONFTEST_FCS_REPLY	284
Figure 78 – Dataset of CONFTEST_PROTO_VERSION_REQUEST	286
Figure 79 – Dataset of CONFTEST_PROTO_VERSION_REPLY	286
Figure 80 – Dataset of CONFTEST_SIMU_TOPO	288
Figure 81 – Dataset of CONFTEST_IUT_TOPO	289
Figure 82 – Dataset of CONFTEST_IP_FILTER	290
Figure 83 – Dataset of CONFTEST_IP_FILTER_REPLY	290
Figure 84 – Dataset of CONFTEST_TIMEOUT_REQUEST_ZERO	292
Figure 85 – Dataset of CONFTEST_TIMEOUT_REPLY_ZERO	293
Figure 86 – Dataset of CONFTEST_TIMEOUT_REQUEST_KEEP	293
Figure 87 – Dataset of CONFTEST_TIMEOUT_REPLY_KEEP	294
Figure 88 – Dataset of CONFTEST_PULL_UNICAST	295
Figure 89 – Dataset of CONFTEST_REPLY_COMID_REQUEST	296
Figure 90 – Dataset of CONFTEST_REPLY_COMID_REPLY	296
Figure 91 – Dataset of CONFTEST_REPLY_IPADDRESS_REQUEST	297
Figure 92 – Dataset of CONFTEST_REPLY_IPADDRESS_REPLY	297
Figure 93 – SDTv2 conformance testing process	298
Figure 94 – SID Generation	302
Figure 95 – Structure of a VDP	303
Figure 96 – VDP with detailed trailer structure	304
Figure 97 – Consist tester architecture	309
Figure 98 – ECSP proxy	310
Figure 99 – Testing configuration variants	315
Figure 100 – Testing configuration TC1	316
Figure 101 – Testing configuration TC2	316
Figure 102 – Testing configuration TC3	316
Figure 103 – TRDP ECSP control telegram	318
Figure C.1 – Coach tester architecture	353
Figure C.2 – Complete consist tester architecture	354
Figure C.3 – Multiple ETB tester architecture	355
Table 1 – Relation to interoperability	28
Table 2 – Relation to performance test	28
Table 3 – Test case categories	29
Table 4 – Mapping conformance testing suites to IUT types	31
Table 5 – Testing framework DEVICE description	45
Table 6 – Test case TCCT23ED_001	46
Table 7 – Test case TCCT23ED_002	47
Table 8 – Test case TCCT23ED_003	49
Table 9 – Test case TCCT23ED_004	50
Table 10 – Test case TCCT23ED_005	51
Table 11 – Test case TCCT23ED_006	52

Table 12 – Test case TCCT23ED_007	53
Table 13 – Test case TCCT23ED_008	54
Table 14 – Test case TCCT23ED_009	55
Table 15 – Test case TCCT25ED_001	55
Table 16 – Test case TCCT25ED_002	56
Table 17 – Test case TCCT25ED_003	57
Table 18 – Test case TCCT25ED_004	58
Table 19 – Test case TCCT25ED_005	59
Table 20 – Test case TCCT25ED_006	60
Table 21 – Test case TCCT25ED_007	61
Table 22 – Test case TCCT25ED_008	62
Table 23 – Test case TCCT25ED_009	63
Table 24 – Test case TCCT25ED_010	64
Table 25 – Test case TCCT34ED_001	65
Table 26 – Test case TCCT34ED_002	66
Table 27 – Test case TCCT34ED_003	67
Table 28 – Test case TCCT34ED_004	68
Table 29 – Test case TCCT34ED_005	69
Table 30 – Test case TCCT34ED_006	70
Table 31 – Test case TCCT34ED_007	71
Table 32 – Test case TCCT34ED_008	72
Table 33 – Test case TCCT34ED_009	73
Table 34 – Test case TCCT34ED_010	74
Table 35 – Test case TCCT34ED_011	75
Table 36 – Test case TCCT34ED_012	76
Table 37 – Test case TCCT34ED_013	77
Table 38 – Test case TCCT34ED_014	78
Table 39 – Test case TCCT34ED_015	79
Table 40 – Test case TCCT34ED_016	80
Table 41 – Test case TCCT34ED_017	81
Table 42 – Test case TCCT34ED_018	82
Table 43 – Test case TCCT34ED_019	83
Table 44 – Test case TCCT34ED_020	84
Table 45 – Test case TCCT34ED_021	85
Table 46 – Test case TCCT34ED_022	86
Table 47 – Push pattern telegram test	259
Table 48 – Push pattern telegram format content	260
Table 49 – Telegram parameter of push pattern test	261
Table 50 – Destination UDP port test	262
Table 51 – Destination UDP port test telegram parameter	262
Table 52 – FCS check test	263
Table 53 – FCS check test telegram parameter	263
Table 54 – FCS check test reply telegram parameter	263

Table 55 – Protocol version test.....	265
Table 56 – Protocol version test telegram parameter	265
Table 57 – Protocol version test reply telegram parameter.....	265
Table 58 – Topology counter test.....	267
Table 59 – Simulator sending telegram parameter	268
Table 60 – IUT sending telegram parameter	269
Table 61 – Data length test.....	270
Table 62 – Data length test of 1 432 bytes length telegram parameters	270
Table 63 – Data length test of 0 byte length telegram parameters	270
Table 64 – Data length test of 512 bytes length telegram parameters	270
Table 65 – Byte alignment test.....	272
Table 66 – Byte alignment test of 441 bytes length telegram parameters	272
Table 67 – Byte alignment test of 442 bytes length telegram parameters	272
Table 68 – Byte alignment test of 443 bytes length telegram parameters	272
Table 69 – Timeout and validity test.....	274
Table 70 – Timeout and validity test telegram parameters.....	274
Table 71 – Timeout and validity test reply telegram parameters(ZERO mode).....	274
Table 72 – Timeout and validity test reply telegram parameters(KEEP mode).....	275
Table 73 – Receiving port IP filtering test.....	276
Table 74 – Receiving port IP filtering test telegram parameters.....	276
Table 75 – Receiving port IP filtering test reply telegram parameters	277
Table 76 – DSCP(QOS) and TTL test.....	278
Table 77 – DSCP(QOS) and TTL test telegram parameters.....	278
Table 78 – Redundancy test	279
Table 79 – Redundancy test telegram parameter	279
Table 80 – point to point test of push pattern	280
Table 81 – point to point test telegram parameter	280
Table 82 – Pull pattern telegram test	281
Table 83 – Pull pattern request telegram format content	281
Table 84 – Pull pattern reply telegram format content	281
Table 85 – Request telegram parameter of pull pattern test	282
Table 86 – Reply telegram parameter of pull pattern test	282
Table 87 – FCS check test.....	283
Table 88 – FCS check test request parameter	283
Table 89 – FCS check test reply parameter	284
Table 90 – Protocol version test.....	285
Table 91 – Protocol version test request telegram parameter.....	285
Table 92 – Protocol version test reply telegram parameter.....	285
Table 93 – Topology counter test.....	287
Table 94 – Requester topology counter test telegram parameter.....	288
Table 95 – Receiving port IP filtering test.....	289
Table 96 – Receiving port IP filtering test telegram parameters.....	289
Table 97 – Receiving port IP filtering test reply telegram parameters	290
Table 98 – Timeout and validity test.....	291
Table 99 – Timeout and validity test request telegram parameters (ZERO mode).....	291
Table 100 – Timeout and validity test reply telegram parameters (ZERO mode).....	291

Table 101 – Timeout and validity test request telegram parameters (KEEP mode)	292
Table 102 – Timeout and validity test reply telegram parameters (KEEP mode)	292
Table 103 – Point to point test of pull pattern	294
Table 104 – Point to point test telegram parameter	294
Table 105 – Reply ComId test of pull pattern	295
Table 106 – Reply ComId test telegram parameter	295
Table 107 – Reply IP Address test of pull pattern	296
Table 108 – Reply IP Address test telegram parameter	297
Table 109 – Configuration TC1: single coach coupled to an open train	316
Table 110 – Configuration TC2: single coach coupled to an open train	316
Table 111 – Configuration TC3: single coach coupled to an open train	317
Table 112 – ECSP control telegram for Leading request	319
Table 113 – ECSP control telegram for inauguration inhibit request	319
Table 114 – ECSP control telegram for sleep request	320
Table 115 – Test sequence for TTDB computation (middle consist, same orientation)	321
Table 116 – Test sequence for TTDB computation (middle consist, inverse orientation)	322
Table 117 – Test sequence for TTDB computation (end consist, no traction, same orientation)	322
Table 118 – Test sequence for TTDB computation (end consist, no traction, inverse orientation)	323
Table 119 – Test sequence for TTDB computation (end consist, traction, same orientation)	323
Table 120 – Test sequence for TTDB computation (end consist, traction, inverse orientation)	324
Table 121 – Test sequence for TTDB computation (train lengthening and shortening)	325
Table 122 – Test sequence for Leading (TE1.1 leading, CuT middle)	326
Table 123 – Test sequence for Leading (TE2 leading, CuT middle)	326
Table 124 – Test sequence for Leading (TE1.1 leading, CuT end)	327
Table 125 – Test sequence for Leading (CuT leading direction 1)	327
Table 126 – Test sequence for Leading (CuT leading direction 2)	328
Table 127 – Test sequence for Leading (Leading double request)	328
Table 128 – Test sequence for Leading (Leading conflict)	329
Table 129 – Test sequence for Inhibit (single)	330
Table 130 – Test sequence for Inhibit (multiple)	331
Table 131 – Test sequence for confirmation	332
Table 132 – Test sequence for correction single middle vehicle	333
Table 133 – Test sequence for correction two middle vehicles	334
Table 134 – Test sequence for correcting three middle vehicles	335
Table 135 – Test sequence for correcting end vehicle	336
Table 136 – Test sequence for sleep mode	337
Table 137 – Performance of basic ETB events	338
Table 138 – Performance of user request	339

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ELECTRONIC RAILWAY EQUIPMENT –
TRAIN COMMUNICATION NETWORK (TCN) –****Part 2-8: TCN conformance test****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61375-2-8 has been prepared by IEC technical committee 9: Electrical equipment and systems for railways.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
9/2746/FDIS	9/2758/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

A list of all parts in the IEC 61375 series, published under the general title *Electronic railway equipment – Train communication network (TCN)*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

TCN is an International Standard with the aim of defining interfaces so as to achieve plug-in compatibility:

- a) between equipment located in different vehicles or consist, and
- b) between equipment and devices located within the same vehicle or consist.

One of the key success factors for the deployment of any technology is standardisation and ensuring interoperability among various implementations. To facilitate interoperability, a conformance test should be implemented.

In this part of IEC 61375, the conformance testing of the ECN and ETB defined in IEC 61375-2-3:2015, IEC 61375-2-5:2014 and IEC 61375-3-4:2014 is specified.

IECNORM.COM : Click to view the full PDF of IEC 61375-2-8:2021

ELECTRONIC RAILWAY EQUIPMENT – TRAIN COMMUNICATION NETWORK (TCN) –

Part 2-8: TCN conformance test

1 Scope

This part of IEC 61375 applies to all equipment and devices implemented according to IEC 61375-2-3:2015, IEC 61375-2-5:2014 and IEC 61375-3-4:2014, i.e. it covers the procedures to be applied to such equipment and devices when the conformance should be proven.

The applicability of this document to a TCN implementation allows for individual conformance checking of the implementation itself, and is a pre-requisite for further interoperability checking between different TCN implementations.

2 Normative references

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

IEC 60571:2012, *Railway applications – Electronic equipment used on rolling stock*

IEC 61000-4-4:2012, *Electromagnetic compatibility (EMC) – Part 4-4: Testing and measurement techniques – Electrical fast transient/burst immunity test*

IEC 61076-2-101:2012, *Connectors for electronic equipment – Product requirements – Part 2-101: Circular connectors – Detail specification for M12 connectors with screw-locking*

IEC 61076-3-104, *Connectors for electrical and electronic equipment – Product requirements – Part 3-104: Detail specification for 8-way, shielded free and fixed connectors for data transmissions with frequencies up to 2 000 MHz*

IEC 61156-6, *Multicore and symmetrical pair/quad cables for digital communications – Part 6: Symmetrical pair/quad cables with transmission characteristics up to 1 000 MHz – Work area wiring – Sectional specification*

IEC 61375-1:2012, *Electronic railway equipment – Train communication network (TCN) – Part 1: General architecture*

IEC 61375-2-1:2012, *Electronic railway equipment – Train communication network (TCN) – Part 2-1: Wire Train Bus (WTB)*

IEC 61375-2-3:2015, *Electronic railway equipment – Train communication network (TCN) – Part 2-3: TCN communication profile*

IEC 61375-2-5:2014, *Electronic railway equipment – Train communication network (TCN) – Part 2-5: Ethernet train backbone*

IEC 61375-3-4:2014, *Electronic railway equipment – Train Communication Network (TCN) – Part 3-4: Ethernet Consist Network (ECN)*

ISO/IEC 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts (Also available as ITU-T Recommendation X.290 (1995))*

ISO/IEC 9646-7:1995, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 7: Implementation Conformance Statements (Also available as ITU-T Recommendation X.296 (1995))*

ISO/IEC 11801 (all parts), *Information technology – Generic cabling for customer premises*

IEEE 802.1AB: 2009, *Station and Media Access Control Connectivity Discovery*

IEEE 802.1AX:2008, *IEEE Standard for Local and metropolitan area networks – Link Aggregation*

IEEE 802.1Qaz:2011, *IEEE Standard for Local and metropolitan area networks – Enhanced Transmission Selection*

IEEE 802.1D:1990, *IEEE Standard for Local and metropolitan area networks – Media Access Control (MAC) Bridges*

IEEE 802.1Q, *IEEE Standard for Local and metropolitan area networks – Virtual Bridged Local Area Networks*

IEEE 802.2, *Logical Link Control*

IEEE 802.3:2012, *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*

IEEE 802.3az, *Energy Efficient Ethernet*

IETF RFC 768, *User Datagram Protocol*

IETF RFC 791, *INTERNET PROTOCOL*

IETF RFC 792, *INTERNET CONTROL MESSAGE PROTOCOL*

IETF RFC 793, *TRANSMISSION CONTROL PROTOCOL*

IETF RFC 826, *An Ethernet Address Resolution Protocol*

IETF RFC 1034, *DOMAIN NAMES – CONCEPTS AND FACILITIES*

IETF RFC 1035, *DOMAIN NAMES – IMPLEMENTATION AND SPECIFICATION*

IETF RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*

IETF RFC 1305, *Network Time Protocol (Version 3) Specification, Implementation and Analysis*

IETF RFC 1361, *Simple Network Time Protocol (SNTP)*

IETF RFC 1901, *Introduction to Community-based SNMPv2*

IETF RFC 1905, *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*

IETF RFC 1906, *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)*

IETF RFC 1918, *Address Allocation for Private Internets*

IETF RFC 2131, *Dynamic Host Configuration Protocol*

IETF RFC 2236, *Internet Group Management Protocol, Version 2*

IETF RFC 2365, *Administratively Scoped IP Multicast*

IETF RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*

IETF RFC 2544, *Benchmarking Methodology for Network Interconnect Devices*

IETF RFC 3203, *DHCP reconfigure extension*

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*

IETF RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace*

TIA/EIA-568-B, *Commercial Building Telecommunications Cabling Standard*

ANSI X3.263:1995, *Information Technology – Fibre Distributed Data Interface (FDDI) – Token Ring Twisted Pair Physical Layer Medium Dependent (TP-PMD)*

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 61375-2-3, IEC 61375-2-5 and IEC 61375-3-4 apply.

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

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.2 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviated terms given in IEC 61375-2-3, IEC 61375-2-5 and IEC 61375-3-4 apply.

4 Conformance test: approach, requirements and boundaries

4.1 Approach

4.1.1 General

This document specifies a general methodology for testing the conformance to the TCN protocol standard of products in which the standard is claimed to be implemented.

This document is organised into clauses structured into different phases of the conformance testing process; these phases being characterised by the following roles:

- a) the specification of abstract test suites for particular TCN protocols according to ISO/IEC 9646-1;
- b) the derivation of executable test suites and associated testing tools according to ISO/IEC 9646-7.

Annex A specifies the rules on clients and laboratory specifying:

- c) the role of a client of a test laboratory, having an implementation of TCN protocols to be tested;
- d) the operation of conformance testing, culminating in the production of a conformance test report which gives the results in terms of the test suite(s) used and the relevant documentation produced.

In all clauses of this document, the scope is limited in order to meet the following objectives:

- e) to achieve an adequate level of confidence in the tests as a guide to conformance;
- f) to achieve comparability between the results of the corresponding tests applied in different places at different times;
- g) to facilitate communication between the parties responsible for the roles described above.

Each objective involves the framework for development of TCN test suites, as listed hereinafter:

- h) how they should relate to the various types of conformance requirement;
- i) the types of test to be standardised and the types not needing standardisation;
- j) the criteria for selecting tests for inclusion in a conformance test suite;
- k) the notation to be used for defining tests;
- l) the structure of a test suite.

Certification, an administrative procedure which may follow conformance testing, is outside the scope of this document.

Requirements for procurement and contracts are outside the scope of this document.

4.1.2 Requirements

4.1.2.1 General

In the context of TCN, a real system is said to exhibit conformance if it complies with the requirements of applicable TCN standard clauses in its communication with a reference system, i.e. the tester.

A TCN standard is a set of interrelated clauses which, together, define behaviour of TCN systems in their communication. Conformance of an IUT will, therefore, be expressed at two levels, conformance to each individual clause, and conformance to the set of clauses.

The following clauses define the conformance requirements and classify them according to attributes and into feasible groups. Attributes and grouping are defined from the general point of view with reference to a TCN specification itself and from the IUT point of view. In the second case, the requirement shall be declared in the appropriate PICS and PIXIT.

4.1.2.2 Conformance requirements

The conformance requirements can be:

- a) mandatory requirements: these are to be observed in all cases;
- b) conditional requirements: these are to be observed if the conditions, set out in the clause, apply;
- c) options: these can be selected to suit the implementation, provided that any requirements applicable to the option are observed.

TCN essential functionality are mandatory requirements; additional functionality can be either conditional or optional requirements.

Furthermore, conformance requirements can be stated:

- d) positively: they state what shall be done;
- e) negatively (prohibitions): they state what shall not be done;
- f) optionally: they state what can be done.

Finally, conformance requirements fall into two groups:

- g) static conformance requirements;
- h) dynamic conformance requirements;

these are discussed in 4.1.2.3 and 4.1.2.4, respectively.

4.1.2.3 Static conformance requirements

To facilitate interoperability, static conformance requirements define the allowed minimum capabilities of an implementation. These requirements may be at a broad level, such as the grouping of functional units and options into protocol classes, or at a detailed level, such as a range of values that have to be supported for specific parameters of timers.

Static conformance requirements and options in TCN parts can be of two varieties:

- a) those which determine the capabilities to be included in the implementation of the particular protocol;
- b) those which determine multi-layer dependencies, for example those which place constraints on the capabilities of the underlying layers of the system in which the protocol implementation resides. These are likely to be found in upper layer parts (e.g. network management versus real time protocols).

All capabilities not explicitly stated as static conformance requirements are to be regarded as optional.

4.1.2.4 Dynamic conformance requirements

Dynamic conformance requirements are all those requirements (and options) which determine what observable behaviour is permitted by the relevant TCN part in instances of communication. They form the bulk of each TCN protocol document. They define the set of allowable behaviours of an implementation or real system. This set defines the maximum capability that a conforming implementation or real system can have within the terms of the TCN protocol document.

A system exhibits dynamic conformance in an instance of communication if its behaviour is a member of the set of all behaviours permitted by the relevant TCN protocol part in a way which is consistent with the PICS.

4.1.2.5 A conforming system

A conforming system or implementation is one which is shown to satisfy both static and dynamic conformance requirements, consistent with the capabilities stated in the PICS, for each protocol declared in the system conformance statement.

4.1.2.6 Interoperability and conformance

The primary purpose of conformance testing is to increase the probability that different implementations are able to inter-operate.

Successful interoperability of two or more real open systems is more likely to be achieved if they all conform to the same subset of a TCN part, or to the same selection of TCN parts, than if they do not.

To prepare two or more systems to successfully inter-operate, it is recommended that a comparison is made of the system conformance statements and PICSs of these systems.

If there is more than one version of a relevant TCN part indicated in the PICSs, the differences between the versions need to be identified and their implications for consideration, including their use in combination with other parts.

While conformance is a necessary condition, it is not on its own a sufficient condition to guarantee interoperability capability. Even if two implementations conform to the same TCN protocol part, they may fail to interoperate because of factors outside the scope of this document.

Trial interoperability is recommended to detect these factors. Further information to assist interoperability between two systems can be obtained by extending the PICS comparison to other relevant information, including test reports and PIXIT. The comparison can focus on:

- a) additional mechanisms claimed to work around known ambiguities or deficiencies not yet corrected in the TCN standard or in peer real systems, for example solution of multi-layer problems;
- b) selection of free options which are not taken into account in the static conformance requirements of the TCN parts;
- c) the existence of timers not specified in the TCN parts and their associated values.

NOTE The comparison can be made between two individual systems, between two or more types of product, or, for the PICS comparison only, between two or more specifications for procurement, permissions to connect, etc.

4.1.3 Requirements declaration statements for an Item Under Test (IUT)

4.1.3.1 Protocol implementation conformance statement (PICS)

To evaluate the conformance of a particular implementation, it is necessary to have a statement of the capabilities and options which have been implemented, and any features which have been omitted, so that the implementation can be tested for conformance against relevant requirements, and against those requirements only. Such a statement is called a Protocol Implementation Conformance Statement (PICS).

In a PICS there should be a distinction between the following categories of information which it may contain:

- a) information related to the mandatory, optional and conditional static conformance requirements of the protocol itself;

- b) information related to the mandatory, optional and conditional static conformance requirements for multi-layer dependencies.

If a set of interrelated TCN protocol has been implemented in a system, a PICS is needed for each protocol. A system conformance statement will also be necessary, summarising all protocols in the system for each of which a distinct PICS is provided.

4.1.3.2 Protocol implementation extra information for testing (PIXIT)

In order to test a protocol implementation, the test laboratory will require information relating to the IUT and its testing environment in addition to that provided by the PICS. This Protocol Implementation eXtra Information for Testing (PIXIT) will be provided by the client submitting the implementation for testing, as a result of consultation with the test laboratory.

The PIXIT may contain the following information:

- a) information needed by the test laboratory in order to be able to run the appropriate test suite on the specific system (e.g. information related to the test method to be used to run the test cases, addressing information);
- b) information already mentioned in the PICS and which needs to be made precise (e.g. a timer value range which is declared as a parameter in the PICS should be specified in the PIXIT);
- c) information to help determine which capabilities stated in the PICS as being supported are testable and which are untestable;
- d) other administrative matters (e.g. the IUT identifier, reference to the related PICS).

The PIXIT should not conflict with the appropriate PICS.

The abstract test suite specifier, test implementer and test laboratory will all contribute to the development of the PIXIT pro-forma.

4.2 Boundaries

4.2.1 General

Conformance testing as discussed in this document is focused on testing for conformance to TCN clauses as they are specified in IEC 61375-2-3:2015, IEC 61375-2-5:2014 and IEC 61375-3-4:2014.

In principle, the objective of conformance testing is to establish whether the implementation being tested conforms to the specification in the relevant clause. Practical limitations make it impossible to be exhaustive, and economic considerations may restrict testing still further.

Therefore, this document distinguishes four types of testing, according to the extent to which they provide an indication of conformance:

- a) basic interconnection tests, which provide prima facie evidence that an IUT conforms;
- b) capability tests, which check that the observable capabilities of the IUT are in accordance with the static conformance requirements and the capabilities claimed in the PICS;
- c) behaviour tests, which endeavour to provide testing which is as comprehensive as possible over the full range of dynamic conformance requirements within the capabilities of the IUT;
- d) conformance resolution tests, which probe in depth the conformance of an IUT to particular requirements, to provide a definite yes/no answer and diagnostic information in relation to specific conformance issues; such tests are not covered by this document.

The tests a), b), c) and d) are described in detail by the following subclauses.

Relations to interoperability and performance are hereinafter considered and defined to clarify their boundaries.

4.2.2 Basic interconnection tests

4.2.2.1 General

Basic interconnection tests provide limited testing of an IUT to establish that there is sufficient conformance for interconnection to be possible, without trying to perform thorough testing.

4.2.2.2 Applicability of basic interconnection tests

Basic interconnection tests are appropriate:

- a) for detecting severe cases of non-conformance;
- b) as a preliminary filter before undertaking more costly tests;
- c) to give a prima facie indication that an implementation which has passed full conformance tests in one environment still conforms in a new environment (e.g. before testing a Layer (N)-implementation, to check that a tested Layer (N – 1)-implementation has not undergone any severe change due to being linked to the Layer (N)-implementation);
- d) for use by users of implementations, to determine whether the implementations appear to be usable for communication with other conforming implementations, for example as a preliminary to data interchange.

Basic interconnection tests are inappropriate:

- e) as a basis for claims of conformance by the supplier of an implementation;
- f) as a means of arbitration to determine causes for communications failure.

Basic interconnection tests are a standardised subset of a conformance test suite (including capability and behaviour tests). They can be used on their own or together with a conformance test suite. The existence and execution of basic interconnection tests are optional.

4.2.3 Capability tests

4.2.3.1 General

Capability tests provide limited testing of each of the static conformance requirements to ascertain what capabilities of the IUT can be observed and to check that those observable capabilities are valid with respect to the static conformance requirements and the PICS.

4.2.3.2 Applicability of capability tests

Capability tests are appropriate:

- a) to check as far as possible the consistency of the PICS with the IUT;
- b) as a preliminary filter before undertaking more in-depth and costly testing;
- c) to check that the capabilities of the IUT are consistent with the static conformance requirements;
- d) to enable efficient selection of behaviour tests to be made for a particular IUT;
- e) when taken together with behaviour tests, as a basis for claims of conformance.

Capability tests are inappropriate:

- f) on their own, as a basis for claims of conformance by the supplier of an implementation;
- g) for testing in detail, the behaviour associated with each capability which has been implemented or not implemented;
- h) for resolution of problems experienced during live usage or where other tests indicate possible non-conformance even though the capability tests have been satisfied.

Capability tests are standardised within a conformance test suite. They can either be separated into their own test group(s) or merged with the behaviour tests.

4.2.4 Behaviour tests

Behaviour tests test an implementation as thoroughly as is practical, over the full range of dynamic conformance requirements. Since the number of possible combinations of events and timing of events is infinite, such testing cannot be exhaustive. There is a further limitation, namely that these tests are designed to be run collectively in a single test environment, so that any faults which are difficult or impossible to detect in that environment are likely to be missed. Therefore, it is possible that a non-conforming implementation passes the conformance test suite; one aim of the test suite design is to minimise the number of times that this occurs.

Behaviour tests with capability tests are the basis for the conformance assessment process.

Behaviour tests are inappropriate:

- a) for resolution of problems experienced during live usage or where other tests indicate possible non-conformance even though the behaviour tests have been satisfied.

Behaviour tests are standardised as the bulk of a conformance test suite.

Behaviour tests include tests for valid behaviour by the IUT in response to valid, inopportune and syntactically invalid protocol behaviour by the real tester. This includes testing the rejection by the IUT of attempts to use features (capabilities) which are stated in the PICS as being not implemented. Thus, capability tests do not need to include tests for capabilities omitted from the PICS.

4.2.5 Conformance resolution tests

Conformance resolution tests provide diagnostic answers, as near to definitive as possible, to the resolution of whether an implementation satisfies particular requirements. Because of the problems of exhaustiveness, the definite answers are gained at the expense of confining tests to a narrow field.

The test architecture and test method will normally be chosen specifically for the requirements to be tested and need not be ones that are generally useful for other requirements. They may even be ones that are regarded as being unacceptable for (standardised) abstract conformance test suites, for example involving implementation-specific methods using, say, the diagnostic and debugging facilities of the specific operating system.

The distinction between behaviour tests and conformance resolution tests may be illustrated by the case of an event such as a reset. The behaviour tests may include only a representative selection of conditions under which a reset might occur and may fail to detect incorrect behaviour in other circumstances. The conformance resolution tests would be confined to conditions under which incorrect behaviour was already suspected to occur and would confirm whether or not the suspicions were correct.

Conformance resolution tests are appropriate:

- a) for providing a yes/no answer in a strictly confined and previously identified situation (e.g. during implementation development, to check whether a particular feature has been correctly implemented, or during operational use, to investigate the cause of problems);
- b) as a means for identifying and offering resolutions for deficiencies in a current conformance test suite.

Conformance resolution tests are inappropriate:

- c) as a basis for judging whether or not an implementation conforms overall.

Conformance resolution tests are not standardised. As a by-product of conformance testing, errors and deficiencies in protocol parts may be identified.

4.2.6 Interpretation of clauses/subclauses and statements

4.2.6.1 General

The TCN described in IEC 61375-2-3, IEC 61375-2-5 and IEC 61375-3-4 is subject to a sort of interpretation to translate some clauses/subclauses and requirements into realisable test suites. The complexity of most TCN protocols makes exhaustive testing impractical on both technical and economic grounds. To cope with a real implementation and extract from IEC 61375-2-3, IEC 61375-2-5 and IEC 61375-3-4, all the relevant tests and some criteria were used. The criteria were grouped according to their characteristics:

- a) imperatives;
- b) illustrative;
- c) options;
- d) weak statements.

The following subclauses describe the criteria.

4.2.6.2 Imperatives

4.2.6.2.1 General

Imperatives are those words and phrases commanding that something shall be provided and are classified as mandatory. They are:

- a) shall: dictates the provision of a functional capability;
- b) must: establishes performance requirements or constraints;
- c) is required: is a specification statement written in the passive voice;
- d) is applicable: includes, by reference, standards or other documentation as an addition to the requirements being specified;
- e) responsible for: is a requirement written for architectures already defined. As an example, "In extended reply delay applications, the master is responsible for spacing the master frames so that the minimum time to transmit to a slave frame and the following master frame is greater than T_{safe} .";
- f) will: is generally used to cite things that the operational or development environment are to provide to the capability being specified. For example, "If it was a strong master, it will signal its demoting to all nodes and it will remain in control of the bus as a weak master until a strong node is appointed";
- g) should: when it is used, the specification statement is considered to be very weak. For example, "Devices supporting the message data capability should have a device address smaller than 256".

4.2.6.2.2 Continuance

Phrases that follow an imperative and introduce the specification of requirements at a lower level, for a supplemental requirement count.

- a) as follows,
- b) below,
- c) following,
- d) in particular,
- e) listed,
- f) support.

Phrases that introduces temporal indication, that may lead to definite or indefinite actions, or enumerative that may lead to infinite test cases.

- g) for each,
- h) while.

The requirement containing temporal or enumerative is tested with a finite time or finite sample.

4.2.6.3 Illustrative

This is information within the requirements document. The data and information pointed to by illustrative strengthens the document's specification statements and whenever possible is used as sample category input for the test. Namely:

- a) figure;
- b) table;
- c) for example;
- d) note.

4.2.6.4 Options

Options is the category of words that give the developer latitude in satisfying the specification statements that contain them. This category clearly forms the basis for the option statements declaration into the PICS. However, those requirements containing such a category of words loosen the specification, increase the risk of non-interoperability, and widen the tests sets.

- a) can (Example: Gateways with Bus_Administrator capability can synchronise the busses.);
- b) may (Example: Class 5 devices may offer the Bus_Administrator capability.);
- c) optionally (Example: The User_Programmable capability is optional.);
- d) exclusion (Example: while the IUT is naming the nodes, one node responds to the naming frame but not to the status request or sends a wrong naming response frame).

Options shall drive the PICS production.

4.2.6.5 Weak statements

Weak statements are apt to cause uncertainty leaving room for multiple interpretations, such wording provides a basis for expanding a requirement or adding future requirements. For the extent of testing, this category generates test with test cases chosen among a representative set of samples. However, by no means such sets fully represent all significant cases foreseen by the clause under test.

- a) adequate,
- b) be able to,
- c) be capable,
- d) effective,
- e) normal,
- f) provide for.

4.2.7 Relation to interoperability

One of the aims of this conformance test is to lead to comparability and wide acceptance of test results produced by different testers, and thereby minimise the need for repeated conformance testing of the same system. Interoperability plays a principal role, since the conformance test is aimed to facilitate interoperability, it has been taken into account with the following domains in Table 1:

Table 1 – Relation to interoperability

Domain	Description
Application interoperability	the ability of TCN to provide a consistent implementation of the syntax and semantics of the data which are interchanged
Protocol interoperability	the ability of TCN to interchange PDUs via the communications platform
Service interoperability	the ability of TCN to support a subset of its intended services
User perceived interoperability	the ability of the service user (human, application, machine) to exchange information via the TCN

No provision is made in this document to implement or recommend an interoperability test.

4.2.8 Relation to performance test

Performance attributes relate deeply to services given by the TCN, even though this conformance test does not intend to implement a performance test, nevertheless performance attributes were taken into account in the following way in Table 2.

Table 2 – Relation to performance test

Performance attribute	Description
Speed	This performance attribute describes the time interval that is used to perform the function or the rate at which the function is performed. (The function may or may not be performed with the desired accuracy.). An example of speed attribute evaluation is: freshness time supervision test
Accuracy	This performance attribute describes the degree of correctness with which the function is performed, no matter if the function is or is not performed with the desired speed. An example of accuracy evaluation is: the receiver hysteresis test
Dependability	This performance attribute describes the degree of certainty (or surety) with which the function is performed regardless of speed or accuracy, but within a given observation interval. An example of dependability attribute evaluation is: Connection stability for the entire inauguration time

No provision is made in this document to implement or recommend a performance test as it is defined by IEC 60571.

4.2.9 Definition of test cases

Test cases and inspection items derived from the requirements and outlined in this document can be categorized into verification activities, measurement activities and test activities. The characteristics of these activities are defined in Table 3.

Table 3 – Test case categories

Category	Description	Keywords	Example
Verification activities	Verification of procedures, processes and documents	verification, verify	Verify that ...
Measurement activities	Measurement and recording of physical quantities (e.g. size, distance, voltage, current, temperature, time). Does not include a pass or fail verdict.	measurement, measure	Measure the
Test activities	Check of properties or measurement results. Includes a pass or fail verdict.	test, check	Check that ... Test the

4.3 Conformance assessment process outline

4.3.1 General

The main feature of the conformance assessment process is a configuration of equipment allowing exchanges of information between the IUT and a real tester. These are controlled and observed by the real tester.

In a conceptual outline, conformance testing should include several steps, involving both static conformance reviews and live testing phases, culminating in the production of a test report which is as thorough as is practical.

These steps are:

- compilation of the PICS;
- compilation of the PIXIT;
- test selection and parameterisation;
- basic interconnection testing (optional);
- capability testing;
- behaviour testing;
- review and analysis of test results;
- synthesis, conclusions and conformance test report production.

4.3.2 Analysis of results, outcomes and verdicts

The observed outcome (of the test execution) is the series of events which have occurred during execution of a test case; it includes all input to and output from the IUT at the points of control and observation.

The foreseen outcomes are identified and defined by the abstract test case specification taken in conjunction with the protocol part. For each test case, there may be one or more foreseen outcomes. Foreseen outcomes are defined primarily in abstract terms.

A verdict is a statement of pass, fail or inconclusive to be associated with every foreseen outcome in the abstract test suite specification.

The analysis of results is performed by comparing the observed outcomes with foreseen outcomes.

The verdict assigned to an observed outcome is that associated with the matching foreseen outcome. If the observed outcome is unforeseen then the abstract test suite specification will state what default verdict shall be assigned.

The means for the comparison of the observed outcomes with the foreseen outcomes are beyond the scope of this document.

NOTE Amongst the possibilities are:

- a) manual or automated comparison (or a mixture);
- b) comparison at or after execution time;
- c) translating the observed outcomes into abstract terms for comparison with the foreseen outcomes or translating the foreseen outcomes into the terms used to record the observed outcomes.

The verdict will be pass, fail or inconclusive:

- a) pass means that the observed outcome satisfies the test purpose and is valid with respect to the relevant TCN parts and with respect to the PICS;
- b) fail means that the observed outcome is syntactically invalid or inopportune with respect to the relevant TCN parts or the PICS;
- c) inconclusive means that the observed outcome is valid with respect to the relevant TCN parts but prevents the test purpose from being accomplished.

The verdict assigned to a particular outcome will depend on the test purpose and the validity of the observed protocol behaviour.

The verdicts made in respect of individual test cases will be synthesised into an overall summary for the IUT based on the test cases executed.

4.4 Mapping of IUT types to conformance test suites

Because this document defines conformance testing suites applicable for different IUT types, a mapping of IUT types to conformance test suites will indicate which test suites are mandatory or optional as it is listed in Table 4.

Table 4 – Mapping conformance testing suites to IUT types

Conformance test IUT Type	Basic interface test	End device test	ECN equipped consist test	ETBN test	TRDP test	SDTv2 test	ETB-equipped consist test
Consist local end device	M	O					
Train communication end device	M				O	O	
Train topology aware end device	M	O			M	O	
Repeater	M		O				
Unmanaged consist switch	M		O				
Managed consist switch	M		O				
Router	M		O				
ETBN	M		O ¹⁾	M	M ¹⁾	M ¹⁾	
Consist							M
O : optional M : mandatory 1) If ETBN includes ECSP.							

5 Basic interface test

5.1 Scope

This clause specifies conformance test cases for physical layer functions and basic communication for devices connected to ECN or ETB as given in IEC61375-2-5 and IEC 61375-3-4.

5.2 PICS pro-forma

5.2.1 General

The PICS provides a statement of the capabilities and options which have been implemented, and any features which have been omitted, see 4.1.3.1. Instructions about filling the PICS pro-forma are given in Annex B.

5.2.2 PICS Tables

5.2.2.1 Identification of PICS

Ref	Question	Response
1	Date of statement	
2	PICS serial number	

5.2.2.2 Identification of the implementation under test

Ref	Question	Requirement	Response
1	Implementation name	M	
2	Version number	M	
3	Special configuration	O	
4	Power supply voltage	M	
5	Power supply current	M	
6	Other information	O	

NOTE 1 Implementation name refers to the identifier of the IUT as indicated by the client. The specific conformance test is applied to the entity identified by the implementation name.

NOTE 2 This is the version number of the IUT. When a version number is defined for an IUT, no subsystem which composes it can progress without a change of this figure (the architecture is frozen and constitutes a configuration).

NOTE 3 Indicated if PIXIT is provided for this IUT.

NOTE 4 Indicates the applicable power supply voltage. Power supply voltage is chosen amongst the values specified in IEC 60571.

NOTE 5 Other information the client considers relevant for IUT identification.

5.2.2.3 Identification of the IUT supplier and/or test laboratory client

Ref	Question	Requirement	Response
1	Organisation name	M	
2	Contact name(s)	M	
3	Address:	M	
4	Telephone number	M	
5	Fax number	M	
6	e-mail address	M	
7	Other information	M	

If the IUT supplier and the test laboratory client are not the same entity, the PICS shall be agreed between the supplier and the test laboratory client.

5.2.2.4 Identification of the standards

Ref	Question	Response
1	Specification document title	
2	Specification document IEC reference number	
3	Specification document date of publication	
4	Specification document version number	
5	Conformance test document title	
6	Conformance test document number	
7	Conformance test document date of publication	
8	Conformance test document version number	

5.2.2.5 Global statement of conformance

Ref	Question	Requirement	Implementation
1	Are all mandatory capabilities implemented?	M	[]

NOTE Answering "No" to this section indicates non-conformance to the protocol specification. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conforming.

5.3 Testing framework

The test is executed in a set-up with an instrumented IUT providing one or more Ethernet interfaces, a tester equipment for Ethernet/IP/UDP/TCP frame sending and receiving, a 100 m railway suitable Ethernet cable connecting IUT and tester equipment, and necessary measurement instruments.

NOTE Test set-up including Ethernet cable characteristics are listed in the specific IUT's report, see Annex C.

5.4 Physical layer test

5.4.1 Test purpose

Test the correctness of mechanical and electrical parameters of the physical layer network interface.

5.4.2 Inspection of the Hardware Ethernet interface

Compliance with the physical layer requirements as specified in IEC 61375-3-4:2014, 4.10.2.1, 4.10.3 and IEC 61375-2-5:2014, 4.2.2 shall be demonstrated by the following test:

- Check that the physical Ethernet interface of the IUT is set to 100FDX.
- Check that the IUT has a MAC address set on delivery. (The MAC address shall be unique worldwide).
- Optional: check that the IUT supports Power over Ethernet (PoE).
- Check that ETB ports are bypassed via relays.
- Check that the IUT support 100 m cable length over CAT5e cables.
- Check that a single point of failure from the IUT does not disable more than the affected end devices (for example by executing burst test, see 5.6.2).

5.4.3 Inspection of the mechanical Ethernet interface construction

Compliance with the physical layer mechanical requirements as specified in IEC 61375-3-4:2014, 4.10.2.2, 4.10.2.3 and 4.10.2.4 shall be demonstrated by the following test:

- Check that the IUT uses a M12 D-coded female connector.
- Check that the M12 D-coded connector is located at the front panel.
- Check that the M12 socket has an adequate ground connection providing lasting connection of the shield to ground.
- Check that the IUT internal Ethernet signal path (length and characteristics of internal Ethernet cabling, number of intermediate connectors) is documented.
- Check that all connectors are labelled, also the ground connector (see IEC 60571:2012).
- Check that the M12 and the casing screws are tightened.
- Check that the cable shield and the connector shield are connected to chassis earth. A cable shield should be connected on a 360° circular basis in the connector.
- Check that a RJ45 connector is only used for connecting Temporary End Devices.

5.4.4 Check of electrical Ethernet interface design

- a) Check that the temperature range class TX according to IEC 60571:2012 (-40 to +85°C) for electronic parts is used.
- b) Check that the transceiver (PHY) and the transformer is compliant to IEEE 802.3. Check the correct Ethernet termination and common mode rejection.

If possible, the components shall be inspected on the PCB. If not feasible the schematics have to be checked. If Off-the-shelf units are used, the manufacturer has to provide a reasonable data sheet which shows the conformance with the temperature range and IEEE 802.3.

5.4.5 Check of Ethernet interface characteristics

This test is optional and can only be applied if the DUT supports auto negotiation.

- a) Measure that the single link pulse meets the waveform requirements from IEEE 802.3.
- b) Measure that the fast link pulse burst indicates that the interface supports at least 100BASE-TX full duplex.
- c) Check that all features which are specified by the fast link pulse burst code are supported by the IUT.

5.4.6 Ethernet signal wave form test (IEEE standard)

Measure that the Ethernet signal parameters

- Amplitude
- zero Volt axis symmetry
- Rise and fall times
- Duty cycle distortion
- Transmitter jitter
- Waveform overshoot

are in the ranges, which are specified in IEEE 802.3, IEEE 802.3az and ANSI X3.263.

5.4.7 Ethernet signal wave form test (amplified signals, optional)

Compliance with the Ethernet signal wave form test which is specified in IEC 61375-3-4:2014, Annex C.

Measure that the Ethernet signal parameters

- Amplitude
- zero Volt axis symmetry
- Rise and fall times
- Duty cycle distortion
- Transmitter jitter
- Waveform overshoot

are in the specified ranges.

5.4.8 Auto-Crossover- and Auto-Sensing-Test (only for switches)

Check that the IUT perform Auto-Crossover and Auto-Sensing.

5.5 Basic communication protocol test

5.5.1 Test purpose

Test the ability of sending and receiving UDP and TCP packets.

5.5.2 Simple communication test

This measurement shall be executed with a cable length of 100 m, which is specified in IEEE 802.3.

- a) Check that the IUT supports UDP communication.
- b) Check that the IUT supports TCP communication.

5.5.3 UDP rising payload test (only for End Devices)

This measurement shall be executed with a cable length of 100 m, which is specified in IEEE 802.3.

- a) Measure the maximum supported payload size for UDP datagrams.
- b) Check that the IUT handles a payload size of minimum 8kBytes.

5.5.4 TCP rising payload test (only for End Devices)

This measurement shall be executed with a cable length of 100 m, which is specified in IEEE 802.3.

- a) Measure the maximum supported payload size for TCP datagrams.

5.5.5 UDP long term test (only for End Devices)

This measurement shall be executed with a cable length of 100 m, which is specified in IEEE 802.3.

- a) Check that the IUT communication with 100 m cable length is stable over a time of 60 min.
- b) Check that the long-term test will be done with the maximum supported payload size with an upper limit of 32 kBytes.
- c) Measure the frame error rate and check that it is below 10^{-6} (1 error per 1 million frames).

5.6 Communication reliability tests

5.6.1 Test purpose

Test the reliability of the communication under railway specific stress conditions.

5.6.2 Burst immunity test

5.6.2.1 Test setup

These measurements shall be executed with a cable length of 100 m, which is specified in IEEE 802.3.

For signal and control ports test level 4 is specified using capacitive clamp (IEC 61000-4-4:2012, Figure 1) with 0,5 m distance to the DUT.

The defined test level 4 has the following parameters, see Figure 1

- Rise time: 5 ns $\pm$ 30 %
- Impulse duration (50 % of peak value): 50 ns $\pm$ 30 %
- Peak value 2 kV $\pm$ 10 % (positive and negative polarity)

- Repetition 5 kHz $\pm$ 20 %
- Duration 15 ms
- Period 300 ms

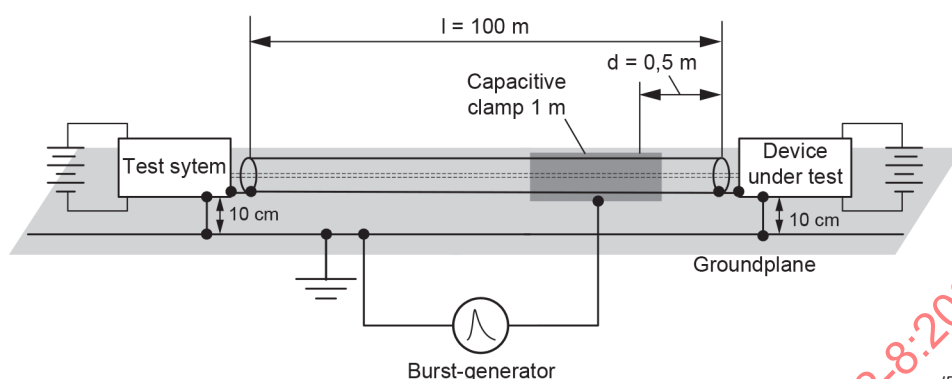


Figure 1 – Setup for burst immunity test

5.6.2.2 Test procedure

Connect the DUT to the burst test environment and power on the DUT.

- Switch on the burst generator configured with the parameters above, switch to positive polarity
- Start the Communication Test with “long term test configuration” at the Test System
- Stop the test after 10 min
- Repeat the test for negative polarity

5.6.2.3 Measurements

Check that the Ethernet port is stable against EMC stress with fast transients (bursts) according to IEC 61000-4-4:2012, by measuring that there are no transmission errors over a minimum period of 1 min or minimum amount of 40 000 frames for each burst polarity.

NOTE Since the chance to hit an Ethernet telegram is lower than 5 % per period (15 ms / 300 ms = 0,05) the minimum test duration for each burst polarity is $n \geq 1\,000$ transmit and receive cycles of a 32kBytes message (minimum more than 40 000 frames).

5.6.3 Ethernet interface isolation test

- Check that the signals are isolated against the shielding.
- Check that all ports are isolated against each other.

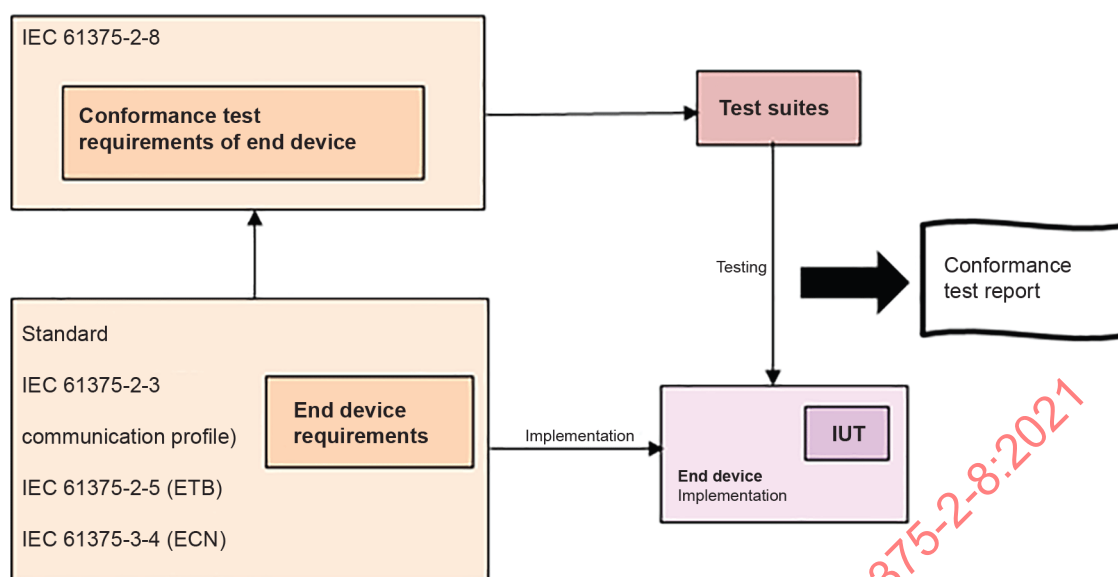
6 Conformance test of End Device

6.1 General

This clause defines conformance requirements for conformance testing for end devices as given in IEC 61375-2-3, IEC 61375-2-5 and IEC 61375-3-4. See also Table 6 through Table 46.

Those requirements are used to derive test suites and test cases in order to determine whether a system, in this case, a unit connected to one Consist Network or to one set of Consist Networks, complies with the requirements.

Figure 2 gives an overview of the End Device conformance testing process.



IEC

Figure 2 – End Device conformance testing process

6.2 Related requirements

As given in IEC 61375-2-3, IEC 61375-2-5 and IEC 61375-3-4.

6.3 IUT and Test scope

The Implementation Under Test (IUT) implements the base standard(s) being tested, so in this clause, the end device (ED).

The end device is a unit that could be connected to one Consist Network or to one set of Consist Networks prepared for redundancy reasons. However, the end device could be connected directly to ETBN (Ethernet Train Backbone Node).

Figure 3 shows a consist topology and a logical view of the ECN, in order to clarify where an end device is located.

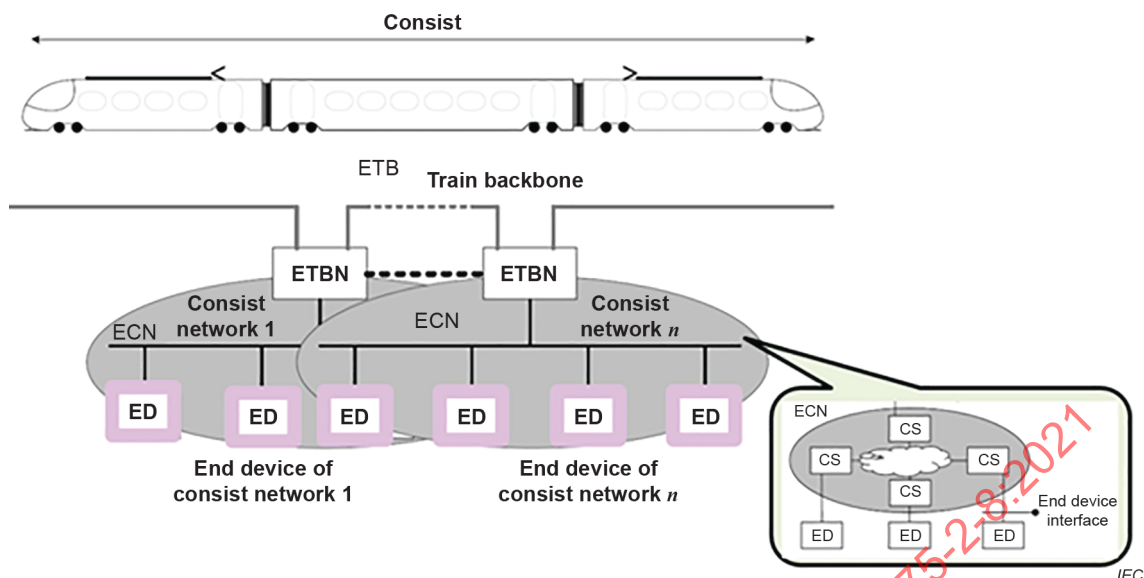


Figure 3 – Consist topology and logical view of ECN

6.4 PICS Pro-forma

6.4.1 General

The PICS provides a statement of the capabilities and options which have been implemented, and any features which have been omitted, see 4.1.3.1. Instructions about filling the PICS pro-forma are given in Annex B.

6.4.2 PICS Tables

6.4.2.1 PICS related to IEC 61375-2-3:2015

6.4.2.1.1 Common ETB Framework – Interoperability

Reference	Question	Subclause	Req.	Implementation
23ED_001	Follows a versioning scheme?	5.1.2	M	Yes ☐ No ☐

6.4.2.1.2 Common ETB Framework – Service Addressing

Reference	Question	Subclause	Req.	Implementation
23ED_002	Entities identified by TCN Domain Names	5.4.3	M	Yes ☐ No ☐
23ED_003	Sources/Destinations identified by TCN URI Identifier	5.4.4.3 5.4.4.6	M	Yes ☐ No ☐
23ED_004	– TCN URI User part	5.4.4.5	O	Yes ☐ No ☐
23ED_005	– Closed trains identification	5.4.4.6.5	O	Yes ☐ No ☐
23ED_006	IP routing in networks with multiple ETBs	5.4.5.5	M	Yes ☐ No ☐
23ED_007	Usage TRDP for data exchange	5.6.2	M	Yes ☐ No ☐
23ED_008	Service discovery	5.7	M	Yes ☐ No ☐

6.4.2.1.3 TRDP (Train Real-Time Data Protocol)

Reference	Question	Subclause	Req.	Implementation
23ED_009	TRDP for the Exchange of TCN process data and TCN message data over ETB	Annex A	M	Yes ☐ No ☐

6.4.2.1.4 SDTv2 (Safe Data Transmission)

Reference	Question	Subclause	Req.	Implementation
23ED_010	SDTv2 supported	Annex B	C	Yes ☐ No ☐

6.4.2.2 PICS related to IEC 61375-2-5:2014**6.4.2.2.1 ETB physical layer**

Reference	Question	Subclause	Req.	Implementation
25ED_001	PSE (Power Source Equipment) connected to a PD (Powered Device)	4.3	O	Yes ☐ No ☐

6.4.2.2.2 ETB network layer: IPv4 subnets definition – host on train subnet

Reference	Question	Subclause	Req.	Implementation
25ED_002	Particular hosts IP addresses on train subnet	6.5.2	M	Yes ☐ No ☐

6.4.2.2.3 ETB network layer: IPv4 subnets definition – host inside a closed train

Reference	Question	Subclause	Req.	Implementation
25ED_003	Particular hosts IP addresses inside a closed train	6.5.3	O	Yes ☐ No ☐

6.4.2.2.4 ETB transport layer

Reference	Question	Subclause	Req.	Implementation
25ED_004	ICMP Internet Control Message Protocol IETF RFC 792	7	M	Yes ☐ No ☐
25ED_005	IGMP v2 Internet Group Management Protocol IETF RFC 2236	7	M	Yes ☐ No ☐
25ED_006	UDP User Datagram Protocol IETF RFC 768	7	M	Yes ☐ No ☐
25ED_007	TCP Transmission Control Protocol IETF RFC 793	7	M	Yes ☐ No ☐

6.4.2.2.5 ETB physical train naming convention

Reference	Question	Subclause	Req.	Implementation
25ED_008	Hosts identified by a FQDN (Full Qualified Domain Name)	10	O	Yes ☐ No ☐

6.4.2.2.6 ETB Application interface

Reference	Question	Subclause	Req.	Implementation
25ED_009	ETB Process Data and Message Data protocols	13.3	M	Yes ☐ No ☐

6.4.2.3 PICS related to IEC 61375-3-4:2014

6.4.2.3.1 Architecture (Network structure)

Reference	Question	Subclause	Req.	Implementation
34ED_001	Ethernet ports according to IEEE 802.3	4.2.1	M	Yes ☐ No ☐

6.4.2.3.2 Redundancy

Reference	Question	Subclause	Req.	Implementation
34ED_002	Redundancy managed at End Device level	4.5.4	O	Yes ☐ No ☐

6.4.2.3.3 IP address and related definitions – Consist Network address

Reference	Question	Subclause	Req.	Implementation
34ED_003	IP address(es) as Consist Network address(es)	4.7.1	M	Yes ☐ No ☐
34ED_004	Unique Consist Network address	4.7.1	M	Yes ☐ No ☐
34ED_005	IPv4 private address space defined in IETF RFC1918	4.7.1	M	Yes ☐ No ☐
34ED_006	Addresses follow this binary presentation: 00001010.0ddddddd.dddddddd.dddd ddd / 9 Field [d] is defined in IEC61375-3-4:2014, 4.7.1.	4.7.1	M	Yes ☐ No ☐

6.4.2.3.4 IP address and related definitions – Train Network address

Reference	Question	Subclause	Req.	Implementation
34ED_007	IP address(es) as Train Network address(es)	4.7.2	M	Yes ☐ No ☐
34ED_008	IPv4 private address defined in IETF RFC 1918	4.7.2	M	Yes ☐ No ☐
34ED_009	Addresses follow the definitions in IEC 61375-2-5	4.7.2	M	Yes ☐ No ☐
34ED_010	Addresses follow this binary presentation: 00001010.1bbxssss.sshhhhhh.hhhhhhhh / 18 The fields are defined in IEC61375-3-4:2014, 4.7.2.	4.7.2	M	Yes ☐ No ☐

6.4.2.3.5 IP address and related definitions – Name resolution and naming definitions

Reference	Question	Subclause	Req.	Implementation
34ED_011	Destination devices of ETB addressable in train wide domain name space	4.7.4.1	M	Yes ☐ No ☐
34ED_012	127.0.0.1 "localhost" as self-addressing	4.7.4.2	M	Yes ☐ No ☐

6.4.2.3.6 IP address and network configuration management – Train network address management

Reference	Question	Subclause	Req.	Implementation
34ED_013	After a new inauguration, renewal of train network address	7.8.2	O	Yes ☐ No ☐
34ED_014	DHCP FORCERENEW message defined in IETF RFC 3203	7.8.2	O	Yes ☐ No ☐

6.4.2.3.7 IP address and network configuration management – DHCP configuration parameters

Reference	Question	Subclause	Req.	Implementation
34ED_015	Subnet mask	4.8.4	M	Yes ☐ No ☐
34ED_016	Router option	4.8.4	M	Yes ☐ No ☐
34ED_017	Domain Name server option	4.8.4	C	Yes ☐ No ☐
34ED_018	Host Name option	4.8.4	O	Yes ☐ No ☐
34ED_019	Broadcast Address	4.8.4	O	Yes ☐ No ☐
34ED_020	Network time protocol server option	4.8.4	C	Yes ☐ No ☐
34ED_021	Vendor Specific Information	4.8.4	O	Yes ☐ No ☐
34ED_022	IP address lease time	4.8.4	M	Yes ☐ No ☐
34ED_023	DHCP message type	4.8.4	M	Yes ☐ No ☐
34ED_024	Server Identifier	4.8.4	M	Yes ☐ No ☐
34ED_025	Parameter request list	4.8.4	M	Yes ☐ No ☐
34ED_026	Message	4.8.4	O	Yes ☐ No ☐
34ED_027	Client-Identifier	4.8.4	O	Yes ☐ No ☐
34ED_028	Relay Agent Information option	4.8.4	O	Yes ☐ No ☐

6.4.2.3.8 Network Device interface – Physical Layer – Protocols

Reference	Question	Subclauses	Req.	Implementation
34ED_029	100BASE-TX	4.9.4.1.1	M	Yes ☐ No ☐
34ED_030	Full Duplex mode defined in IEEE 802.3	4.9.4.1.1	M	Yes ☐ No ☐
34ED_031	Auto negotiation function defined in IEEE 802.3	4.9.4.1.1	M	Yes ☐ No ☐

6.4.2.3.9 Network Device interface – Physical Layer – Cables

Reference	Question	Subclause	Req.	Implementation
	When 100BASE-TX or 10BASE-T is used:			
34ED_032	-M12 D-coded socket/plug	4.9.4.3.1	O	Yes ☐ No ☐
34ED_033	-IEC 61076-3-104 socket/plug	4.9.4.3.1	O	Yes ☐ No ☐
34ED_034	-RJ45 socket/plug defined in TIA/EIA-568-B	4.9.4.3.1	O	Yes ☐ No ☐
34ED_035	Pinning for M12 connector as shown in Table 11	4.9.4.3.1	M	Yes ☐ No ☐

6.4.2.3.10 End Device interface (Table 12) – Physical Layer Protocols

Reference	Question	Subclause	Req.	Implementation
34ED_036	IEEE 802.3 supported	4.10.2.1	M	Yes ☐ No ☐
34ED_037	100BASE-TX	4.10.2.1	M	Yes ☐ No ☐
34ED_038	Full Duplex mode defined in IEEE 802.3	4.10.2.1	M	Yes ☐ No ☐
34ED_039	Auto negotiation function defined in IEEE 802.3	4.10.2.1	M	Yes ☐ No ☐

6.4.2.3.11 End Device interface – Physical Layer Cables

Reference	Question	Subclause	Req.	Implementation
34ED_040	ISO/IEC 11801 and IEC 61156-6	4.10.2.2	M	Yes ☐ No ☐
34ED_041	Class D (category 5e) with two twisted pairs	4.10.2.2	M	Yes ☐ No ☐

6.4.2.3.12 End Device interface – Physical Layer Connectors

Reference	Question	Subclause	Req.	Implementation
34ED_042	M12 D-coded socket/plug defined in IEC 61076-2-101	4.10.2.3	M	Yes ☐ No ☐
34ED_043	IEC 61076-3-104 socket/plug	4.10.2.3	O	Yes ☐ No ☐
34ED_044	RJ45 socket/plug defined in TIA/EIA-568-B	4.10.2.3	O	Yes ☐ No ☐
34ED_045	Pinning for M12 connector as shown in Table 11 and Figure 5	4.10.2.3	M	Yes ☐ No ☐

6.4.2.3.13 End Device interface – Link Layer

Reference	Question	Subclause	Req.	Implementation
34ED_046	MAC defined in IEEE 802.3	4.10.3	M	Yes ☐ No ☐
34ED_047	MAC with basic frame defined in IEEE 802.3	4.10.3	M	Yes ☐ No ☐

6.4.2.3.14 End Device interface – Network Layer

Reference	Question	Subclause	Req.	Implementation
34ED_048	IP version4 defined in IETF RFC 791	4.10.4	M	Yes ☐ No ☐
34ED_049	ICMP defined in IETF RFC 792	4.10.4	M	Yes ☐ No ☐
34ED_050	ARP defined in IETF RFC 826	4.10.4	M	Yes ☐ No ☐

6.4.2.3.15 End Device interface – Transport Layer

Reference	Question	Subclause	Req.	Implementation
34ED_051	UDP defined IETF RFC 768	4.10.5	M	Yes ☐ No ☐
34ED_052	TCP defined IETF RFC 793	4.10.5	M	Yes ☐ No ☐

6.4.2.3.16 End Device interface – Application Layer

Reference	Question	Subclause	Req.	Implementation
34ED_053	DHCP defined in IETF RFC 2131	4.10.6	M	Yes ☐ No ☐
34ED_054	DNS client function defined in IETF RFC 1034	4.10.6	M	Yes ☐ No ☐
34ED_055	SNTP client function defined in IETF RFC 1361 or NTP version 3 client function defined in IETF RFC 1305	4.10.6	O	Yes ☐ No ☐

6.4.2.3.17 Network management

Reference	Question	Subclause	Req.	Implementation
34ED_056	SNMP agent functions. (SNMPv2 defined in IETF RFC 1901, 1905 and 1906 is the minimum requirement.)	4.12.1	O	Yes ☐ No ☐

6.5 Testing framework

Figure 4 shows an example of the architecture of the testing framework and Table 5 describes each component:

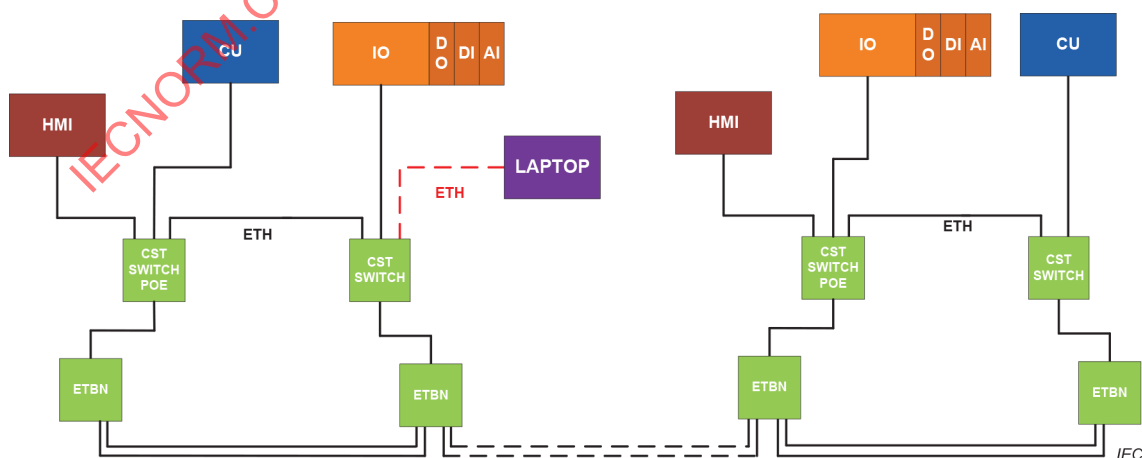


Figure 4 – Testing framework architecture

Table 5 – Testing framework DEVICE description

DEVICE	DESCRIPTION
HMI	Human Machine Interface
CU	Control Unit
IO	Input Output Device (with Digital Output Card, Digital Input Card, and Analog Input Card)
CST SWITCH	Consist switch for connecting different devices to the Ethernet Network
CST SWITCH POE	Switch for connecting different devices to the Ethernet Network, with Power over Ethernet
ETBN	Switch for coupling purposes
LAPTOP	Device equipped with common network tools such DHCP client, DNS client, ICMP (ping) or "Ethernet bus analyzer tool". It shall be also capable of exchanging TRDP frames over the network. The "Ethernet bus analyzer tool" could be defined as a software tool to capture Ethernet frames and dissect and analyze them. For example: "Wireshark".

IECNORM.COM : Click to view the full PDF of IEC 61375-2-8:2021

6.6 Test Suite IEC 61375-2-3

6.6.1 Test case TCCT23ED_001

Table 6 – Test case TCCT23ED_001

IDENTIFIER	TCCT23ED_001
TITLE	ETB Framework Interoperability
ITEM NUMBER	23ED_001
DESCRIPTION	The next test case will ensure that: All changes affecting normative clauses of this document shall be treated as incompatible changes, leading to an increment of parameter 'main Version', unless otherwise specified.
REQUIREMENT	Mandatory
DUT	ECSP
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Check that as defined in IEC 61375-2-5, following an ETB train inauguration, at least one entitled device in each consist shall send out a CSTINFO telegram over ETB to all other entitled devices within the train to inform about consist properties. Step 6: Check that the version has the next structure on the telegram: <pre> VERSION ::= RECORD { mainVersion UINT8 -- main version, incremented for incompatible changes value range: 1..255 (0 = reserved) subVersion UINT8 -- sub version, incremented for compatible changes value range: 0..255 }</pre>
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: the End Device sends the version in a correct format.
DEPENDENCIES	

6.6.2 Test case TCCT23ED_002**Table 7 – Test case TCCT23ED_002**

IDENTIFIER	TCCT23ED_002
TITLE	Sources/Destinations identified by TCN URI Identifier
ITEM NUMBER	23ED_002 / 23ED_003
DESCRIPTION	The next test case will ensure that: • Function devices shall be identified by labels following the syntax given in IEC61375-2-3:2015, Table 8. • Vehicles shall be identified by labels following the syntax given in IEC61375-2-3:2015, Table 10. • Consists shall be identified by labels following the syntax given in IEC61375-2-3:2015, Table 12.
REQUIREMENT	Mandatory
DUT	TCN-DNS Server
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off

IDENTIFIER	TCCT23ED_002
STEPS	Step 1: Supply the test bench. Step 2: Connect the laptop to an ETBN with the correct IP address and DNS server IP. Step 3: Perform a DNS translation from the laptop, using DNS names, checking that: - the function device label used is the one specified in IEC61375-2-3:2015, Table 8: <pre> fctdev.vehicle.consist.cltrain.train fctDev = "IDev" deviceName group devicename = label group = "grpAll" grpDev grpDev = "grp" 1 {uchar} 12 </pre> - the vehicle label used is the one specified in IEC61375-2-3:2015, Table 10: <pre> fctdev.vehicle.consist.cltrain.train veh = "IVeh" cstVehNo opVehNo "leadVeh" "aVeh" idVeh "anyVeh" cstVehNo = "veh" digit digit opVehNo = "opVeh" digit digit idVeh = label </pre> - the consist label used is the one specified in IEC61375-2-3:2015, Table 12: <pre> fctdev.vehicle.consist.cltrain.train consist = "ICst" "aCst" "anyCst" "leadCst" cstNo opCstNo idCst cstNo = "cst" digit digit opCstNo = "opCst" digit digit idCst = label </pre> Step 4: Check that the DNS Server replies with the correspondent IP address.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The DNS Server resolves the correspondent IP address
DEPENDENCIES	

6.6.3 Test case TCCT23ED_003**Table 8 – Test case TCCT23ED_003**

IDENTIFIER	TCCT23ED_003
TITLE	TCN URI User part
ITEM NUMBER	23ED_004
DESCRIPTION	The next test case will ensure that: Communicating entities (sources and destinations) shall be identified by a fully qualified TCN domain name which is derived from the TCN domain name space using a TCN URI scheme defined in RFC 3986.
REQUIREMENT	Mandatory
DUT	TCN-DNS Server
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the laptop to an ETBN with the correct IP address and DNS server IP. Step 3: Perform a DNS translation from the laptop, using DNS names, checking that the user part used is the one specified by the End Device: <i>usr@fctdev.vehicle.consist.cltrain.train</i> <i>train = ITm</i> Step 4: Check that the DNS Server resolves TCN URI user part.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The DNS Server resolves TCN URI user part
DEPENDENCIES	

6.6.4 Test case TCCT23ED_004

Table 9 – Test case TCCT23ED_004

IDENTIFIER	TCCT23ED_004
TITLE	Closed trains identification
ITEM NUMBER	23ED_005
DESCRIPTION	The next test case will ensure that: - Closed trains shall be identified by labels following the syntax given in Table 52. - Trains shall be identified by labels following the syntax given in Table 52.
REQUIREMENT	Optional
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Perform a ping from the laptop to the End Device, using DNS names, checking that closed trains and trains are identified by next labels: <i>trn://usr@fctdev.vehicle.consist.cltrain.train</i> <i>train = lTrn</i> Step 5: Check that the End Device responds to the ping performed by the laptop.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The End Device, connected in a different ETB network, responds to the ping performed from the laptop, with closed trains and trains correctly identified.
DEPENDENCIES	

6.6.5 Test case TCCT23ED_005**Table 10 – Test case TCCT23ED_005**

IDENTIFIER	TCCT23ED_005
TITLE	IP routing in networks with multiple ETBs
ITEM NUMBER	23ED_006
DESCRIPTION	The next test case will ensure that: For transmitting IP packets to a destination ED, which is related to another ETB than the ETB the source ED is related to, the IP packet shall be transferred over the ETB the destination ED is related to.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN of one ETB with the correct IP address. Step 3: Connect the laptop to an ETBN of another ETB with the correct IP address. Step 4: Perform a traceroute from the laptop to the End Device, using DNS names. Step 5: Check that the path followed by the IP frames to the End Device is the correspondent one.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The path path followed by the IP frames to the End Device is the correspondent one.
DEPENDENCIES	

6.6.6 Test case TCCT23ED_006

Table 11 – Test case TCCT23ED_006

IDENTIFIER	TCCT23ED_006
TITLE	Usage TRDP for data exchange
ITEM NUMBER	23ED_007
DESCRIPTION	The next test case will ensure that: All services of the operational network shall use the TRDP protocol as specified in Annex A (with supplementary information in Annex C and Annex D) of IEC 61375-2-3:2015 for the transfer of process data and message data between operational network functions.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off End Device shall be prepared for communicating using TRDP Process Data and Message Data Telegrams. The laptop shall be prepared for communicating using TRDP Process Data and Message Data Telegrams.
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Check that the End Device uses TRDP Process Data and Message Data for communicating with the rest of devices.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The End Device complies the requirements related to TRDP communication.
DEPENDENCIES	

6.6.7 Test case TCCT23ED_007**Table 12 – Test case TCCT23ED_007**

IDENTIFIER	TCCT23ED_007
TITLE	TRDP process Data and Message Data
ITEM NUMBER	23ED_008
DESCRIPTION	The next test cases will ensure the correct communication between different devices on the train, using TCN process data and TCN message data over ETB. These tests are defined in Clause 9 of IEC 61375-2-3:2015.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	
INITIAL CONDITIONS	
STEPS	
FINAL CONDITIONS	
ACCEPTANCE CRITERIA	
DEPENDENCIES	

IECNORM.COM : Click to view the full PDF of IEC 61375-2-8:2021

6.6.8 Test case TCCT23ED_008

Table 13 – Test case TCCT23ED_008

IDENTIFIER	TCCT23ED_008
TITLE	Service Discovery
ITEM NUMBER	23ED_009
DESCRIPTION	The next test case will ensure that: Each consist shall announce the functions it supports within its CSTINFO telegram.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off End Device shall be prepared for communicating using Message Data Telegrams. The laptop shall be prepared for communicating using Message Data Telegrams.
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the “Ethernet bus analyzer tool” to capture the frames that are going to be sent by the devices. Step 5: Check that when the laptop sends the MD request, on the reply sent by the End Device, the functions it supports are announced.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: the End Device announces the functions it supports within its CSTINFO telegram.
DEPENDENCIES	

6.6.9 Test case TCCT23ED_009**Table 14 – Test case TCCT23ED_009**

IDENTIFIER	TCCT23ED_009
TITLE	Safe Data Transmission (SDTv2)
ITEM NUMBER	23ED_010
DESCRIPTION	The next test cases will ensure the correct communication between different devices on the train, using an extension to the communication profile for the transmission of safety related (vital) data over ETB. These tests are defined in Clause 10.
REQUIREMENT	Conditional
DUT	End Device
TEST SETUP	
INITIAL CONDITIONS	
STEPS	
FINAL CONDITIONS	
ACCEPTANCE CRITERIA	
DEPENDENCIES	

6.7 Test Suite IEC 61375-2-5**6.7.1 Test case TCCT25ED_001****Table 15 – Test case TCCT25ED_001**

IDENTIFIER	TCCT25ED_001
TITLE	PSE connected to PD
ITEM NUMBER	25ED_001
DESCRIPTION	The next test case will ensure that: A PSE (Power Source Equipment) interface shall be connected to a PD (Powered Device) interface.
REQUIREMENT	Optional
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Connect the End Device to a consist switch port, that has POE. Step 2: Connect the laptop to a consist switch Step 3: Supply the test bench. Step 4: Check that the End Device gets powered when supplying the test bench. Step 5: Perform a "ping" from the laptop to the End Device IP address. Step 6: Check that there exists a response to the performed "ping".
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - the End Device is supplied. - There exists a response to the performed "ping".
DEPENDENCIES	

6.7.2 Test case TCCT25ED_002

Table 16 – Test case TCCT25ED_002

IDENTIFIER	TCCT25ED_002										
TITLE	Particular hosts IP addresses on train subnet										
ITEM NUMBER	25ED_002										
DESCRIPTION	The next test case will ensure that: Host IP addresses on ETB side are correctly defined.										
REQUIREMENT	Mandatory										
DUT	End Device										
TEST SETUP	Test Bench										
INITIAL CONDITIONS	Test Bench powered off										
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to the ETB network with the correct IP address. Step 3: Check that the IP address of the End Device has the next format: 00001010.1bb00000.00ttttt.hhhhhhhh/18 <table> <tr> <th colspan="2">Subnet number part:</th></tr> <tr> <td>[b]</td><td>« Backbone Id », between [0,3]. Identify some main backbone subnets. Up to 4 ETBs could be defined. 0, for TCMS 1, for multimedia 2, not specified 3, not specified</td></tr> <tr> <th colspan="2">Host number part:</th></tr> <tr> <td>[t]</td><td>« ETBN Id », configuration result, it is the ETBN number inside the train. Never Null, between [1,63].</td></tr> <tr> <td>[h]</td><td>« Host Id », unique host identification for this ETBN node. Range [1,254] due to 254 hosts maximum by ETBN node available.</td></tr> </table> IEC	Subnet number part:		[b]	« Backbone Id », between [0,3]. Identify some main backbone subnets. Up to 4 ETBs could be defined. 0, for TCMS 1, for multimedia 2, not specified 3, not specified	Host number part:		[t]	« ETBN Id », configuration result, it is the ETBN number inside the train. Never Null, between [1,63].	[h]	« Host Id », unique host identification for this ETBN node. Range [1,254] due to 254 hosts maximum by ETBN node available.
Subnet number part:											
[b]	« Backbone Id », between [0,3]. Identify some main backbone subnets. Up to 4 ETBs could be defined. 0, for TCMS 1, for multimedia 2, not specified 3, not specified										
Host number part:											
[t]	« ETBN Id », configuration result, it is the ETBN number inside the train. Never Null, between [1,63].										
[h]	« Host Id », unique host identification for this ETBN node. Range [1,254] due to 254 hosts maximum by ETBN node available.										
FINAL CONDITIONS	Test Bench powered off.										
ACCEPTANCE CRITERIA	OK if: The format of the IP address is correctly defined.										
DEPENDENCIES											

6.7.3 Test case TCCT25ED_003**Table 17 – Test case TCCT25ED_003**

IDENTIFIER	TCCT25ED_003
TITLE	Particular hosts IP addresses on a closed train
ITEM NUMBER	25ED_003
DESCRIPTION	The next test case will ensure that: Host IP addresses on a closed train ETB are correctly defined.
REQUIREMENT	Optional
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to the ECN network with the correct IP address. Step 3: Check that the IP address of the End Device has the next format: 00001010.0sstoooo.ooodddddd.dddddddd /12 with: s 0..3, scope (3 = train, 2 = closed train, 1 = Consist, 0 = Consist Network; currently relative IP addresses are only defined in the train scope) t 0..1, type (0 = absolute, 1 = relative; always 1 for relative IP addresses) o 0..63, offset of the destination Consist Network d 1 .. 16383, device address
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The format of the IP address is correctly defined.
DEPENDENCIES	

6.7.4 Test case TCCT25ED_004

Table 18 – Test case TCCT25ED_004

IDENTIFIER	TCCT25ED_004
TITLE	ETB Transport layer (ICMP)
ITEM NUMBER	25ED_004
DESCRIPTION	The next test case will ensure that: • ICMP is supported by the End Device.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Perform a ping from the laptop to the End Device. Step 6: Check on the bus analyzer tool that there exists an ICMP message sent by the End Device, and conforms to IETF RFC 792.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: • ICMP message sent by the End Device conforms to IETF RFC 792.
DEPENDENCIES	

6.7.5 Test case TCCT25ED_005**Table 19 – Test case TCCT25ED_005**

IDENTIFIER	TCCT25ED_005
TITLE	ETB Transport layer (IGMP v2)
ITEM NUMBER	25ED_005
DESCRIPTION	The next test case will ensure that: IGMP v2 Internet Group Management Protocol IETF RFC 2236 is supported by the End Device.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Start a Multicast communication between the End Device and the laptop. Multicast group addresses shall be defined. Step 6: Check on the "Ethernet bus analyzer tool" that: - Membership Query-s have been established. - The End Device reads correctly the multicast group sent by the laptop. - The laptop reads correctly the multicast group sent by the End Device.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The IGMP v2 is supported by the End Device.
DEPENDENCIES	

6.7.6 Test case TCCT25ED_006

Table 20 – Test case TCCT25ED_006

IDENTIFIER	TCCT25ED_006
TITLE	ETB Transport layer (UDP)
ITEM NUMBER	25ED_006
DESCRIPTION	The next test case will ensure that: UDP, which is defined in IETF RFC 768, is supported by the End Device.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off The End Device shall be prepared for sending UDP frames.
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: An UDP frame shall be sent by the End Device to the laptop. Step 6: Check on the bus analyzer tool that the UDP frame conforms to IETF RFC 768: - Source port (16 bits) - Destination port (16 bits) - Length (16 bits) - Checksum (16 bits) - Data
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: that the UDP frame sent by the End Device conforms to IETF RFC 768.
DEPENDENCIES	

6.7.7 Test case TCCT25ED_007**Table 21 – Test case TCCT25ED_007**

IDENTIFIER	TCCT25ED_007
TITLE	ETB Transport layer (TCP)
ITEM NUMBER	25ED_007
DESCRIPTION	The next test case will ensure that: TCP, which is defined in IETF RFC 793, shall be supported.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off The End Device shall be prepared for sending TCP frames.
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Open a command prompt on the laptop, and open a port for the transmission of TCP frames (for example: port 4567). Step 6: Establish a TCP connection on the End Device side on the same port opened on the laptop. Step 7: Send a message to the specified port and check on the laptop that the message is received.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: that the TCP frame sent by the End Device conforms to IETF RFC 793.
DEPENDENCIES	

6.7.8 Test case TCCT25ED_008

Table 22 – Test case TCCT25ED_008

IDENTIFIER	TCCT25ED_008
TITLE	Hosts identified by a FQDN
ITEM NUMBER	25ED_008
DESCRIPTION	The next test case will ensure that: - In a specific train, each host with a train IP address shall be identified by an unambiguous train FQDN. - Train FQDN shall always be translated to train IP address - ETB Train domain shall be defined using Train Network Directory contents - ETB Train domain shall be defined as on Figure 72 - End device train FQDN hostname part shall be defined by project in accordance with IETF RFC 1035.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Client and DNS client. There shall be a DHCP Server and DNS Server on the Test Bench.
STEPS	Step 1: Connect the End Device to an ETBN, with DHCP and DNS client configuration active. Step 2: Connect the laptop to an ETBN, with DHCP and DNS client configuration active. Step 3: Supply the test bench. Step 4: Check on the End Device that a correct IP address has been received. Step 5: Perform a “ping” using the DNS name of the End Device from the laptop, and check that there exists a response from the End Device. The format of the DNS name shall be: “hostname.domain.tid” where: For example: ping host1.cst1.ltrain
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The End Device responds to the “ping” sent by the laptop using the DNS correct format.
DEPENDENCIES	

6.7.9 Test case TCCT25ED_009**Table 23 – Test case TCCT25ED_009**

IDENTIFIER	TCCT25ED_009
TITLE	ETB Application interface (Process Data)
ITEM NUMBER	25ED_009
DESCRIPTION	The next test case will ensure that: To exchange data at train application level, a PD (Process Data) protocol shall be used.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Client. End Device shall be configured for transmitting Process Data. The laptop shall be prepared for receiving the process data sent by the End Device. There shall be a DHCP Server on the Test Bench.
STEPS	Step 1: Connect the End Device to an ETBN, with DHCP client configuration active. Step 2: Connect the laptop to an ETBN, with DHCP client configuration active. Step 3: Supply the test bench. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Check on the "Ethernet bus analyzer tool" that the format of the process data frames conform to the "Table A.3- PD-PDU parameters" of IEC 61375-2-3:2015.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The End Device sends the process data frames with the correct format specified in "Table A.3- PD-PDU parameters" of IEC 61375-2-3:2015.
DEPENDENCIES	

6.7.10 Test case TCCT25ED_010

Table 24 – Test case TCCT25ED_010

IDENTIFIER	TCCT25ED_010
TITLE	ETB Application interface (Message Data)
ITEM NUMBER	25ED_010
DESCRIPTION	The next test case will ensure that: To exchange data at train application level, a MD (Message Data) protocol shall be used.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Client. End Device shall be configured for transmitting Message Data. The laptop shall be prepared for receiving the Message data sent by the End Device. There shall be a DHCP Server on the Test Bench.
STEPS	Step 1: Connect the End Device to an ETBN, with DHCP client configuration active. Step 2: Connect the laptop to an ETBN, with DHCP client configuration active. Step 3: Supply the test bench. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Check on the "Ethernet bus analyzer tool" that the format of the message data frames conform to the "Table A.18 – MD-PDU parameters" of IEC 61375-2-3:2015.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The End Device sends the message data frames with the correct format specified in "Table A.18 – MD-PDU parameters".
DEPENDENCIES	

6.8 Test Suite IEC61375-3-4**6.8.1 Test case TCCT34ED_001****Table 25 – Test case TCCT34ED_001**

IDENTIFIER	TCCT34ED_001
TITLE	Architecture (Network structure)
ITEM NUMBER	34ED_001
DESCRIPTION	The next test case will ensure that: Ethernet ports between End Devices and Consist Switches shall follow IEEE 802.3.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench, visual inspection
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Check that ethernet ports between End Devices and Consist Switches follow IEEE 802.3.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: the ethernet ports between End Devices and Consist Switches follow IEEE 802.3.
DEPENDENCIES	

6.8.2 Test case TCCT34ED_002

Table 26 – Test case TCCT34ED_002

IDENTIFIER	TCCT34ED_002
TITLE	End Device Dual Homing
ITEM NUMBER	34ED_002
DESCRIPTION	The next test case will ensure that: - the End Device in dual homing uses separate physical network interfaces. - the End Device in dual homing does not create loop on the ECN.
REQUIREMENT	Optional
DUT	End Device
TEST SETUP	Test Bench, visual inspection
INITIAL CONDITIONS	Test Bench powered off End device must be prepared for sending UDP frames.
STEPS	Step 1: Visually check that the device uses separate physical network interfaces. Step 2: Supply the test bench. Step 3: Connect the End Device to 2 different switches. Step 4: Connect the laptop to a switch. Step 5: Using the "Ethernet bus analyzer tool" (for example Wireshark) on the laptop, check that the End Device is publishing the UDP information in one of the network interfaces, while the other is inactive. Step 6: Disconnect the ethernet cable from the port of the switch to which the End Device is sending the information. Step 7: Check that the End Device publishes the information through the port it has still connected.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - the device uses separate physical network interfaces. - the device only sends the information from 1 network interface at a time.
DEPENDENCIES	

6.8.3 Test case TCCT34ED_003**Table 27 – Test case TCCT34ED_003**

IDENTIFIER	TCCT34ED_003
TITLE	End Device Consist network address (No DHCP)
ITEM NUMBER	34ED_003 – 34ED_006
DESCRIPTION	The next test case will ensure that: - the End Device connected to ECN shall have one or several IP address(es) as Consist Network address(es). - The Consist Network address shall be unique within a Consist Network. - Consist Network address shall use IPv4 private address space defined in IETF RFC 1918. - The correct format specified in IEC 61375-3-4 table in 4.7.1 is used.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench, visual inspection
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to a Consist switch with the correct IP address. Step 3: Connect the laptop to a Consist switch with the correct IP address. Step 4: Perform a ping from the laptop to the End Device. Step 5: Check that there exists a response from the End Device to the laptop. Step 6: Check that the mac address corresponds to the End Device. Step 7: Check that the IP address of the End Device is on the next range: from 10.0.0.0 to 10.127.255.255. Step 8: Check that there is no other device with the same IP on the consist network.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - The device has one or several IP address(es) as Consist Network address(es). - The Consist Network address is unique within the Consist Network. - The device uses IPv4 private address space defined in IETF RFC 1918. - The correct format specified in IEC 61375-3-4, table in 4.7.1 is used.
DEPENDENCIES	

6.8.4 Test case TCCT34ED_004

Table 28 – Test case TCCT34ED_004

IDENTIFIER	TCCT34ED_004
TITLE	End Device Train network address (No DHCP)
ITEM NUMBER	34ED_007 – 34ED_010
DESCRIPTION	The next test case will ensure that: • Source and destination addresses in the communication on ETB shall be train network addresses. • Train Network address shall use IPv4 private address space defined in IETF RFC 1918. • The correct format specified in IEC 61375-2-5 and IEC 61375-3-4, table in 4.7.2 is used.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Perform a ping from the laptop to the End Device. Step 5: Check that there exists a response from the End Device to the laptop. Step 6: Check that the mac address corresponds to the End Device. Step 7: Check that the IP address of the End Device is on the next range: from 10.128.0.0 to 10.128.255.255. Step 8: Check that there is no other device with the same IP on the consist network.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: • Source and destination addresses in the communication on ETB are train network addresses. • The device uses IPv4 private address space defined in IETF RFC 1918. • The correct format specified in IEC 61375-2-5 and IEC 61375-3-4, table in 4.7.2 is used.
DEPENDENCIES	

6.8.5 Test case TCCT34ED_005**Table 29 – Test case TCCT34ED_005**

IDENTIFIER	TCCT34ED_005
TITLE	End Device Train network address name resolution
ITEM NUMBER	34ED_011
DESCRIPTION	The next test case will ensure that: Each communication device which can be a destination device for communication over ETB will be addressable in train wide domain name space.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. The test bench shall provide a Domain Name System Server. The End Device shall have configured the IP address of the DNS Server.
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Perform a ping from the laptop to the End Device using the Domain Name of the Device. Step 5: Perform a ping from the End Device to the laptop using the Domain Name of the Device. Step 6: Check that there exists a response from the laptop to the End Device.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - The End Device responds to the ping performed by the laptop using DNS addresses. - The laptop responds to the ping performed by the End Device using DNS addresses.
DEPENDENCIES	

6.8.6 Test case TCCT34ED_006

Table 30 – Test case TCCT34ED_006

IDENTIFIER	TCCT34ED_006
TITLE	End Device Internal Host Name
ITEM NUMBER	34ED_012
DESCRIPTION	The next test case will ensure that: for self-addressing End Device, internal hostname will be declared as "localhost" 127.0.0.1
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off.
STEPS	Step 1: Supply the test bench. Step 2: Open a terminal on the End Device using a defined protocol (for example ssh connection). Step 3: Perform a "ping" to the Ip address "127.0.0.1". Step 4: Check that a response to the "ping" exists.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: there exists a response to the performed "ping".
DEPENDENCIES	

6.8.7 Test case TCCT34ED_007**Table 31 – Test case TCCT34ED_007**

IDENTIFIER	TCCT34ED_007
TITLE	Renewal of Train Network Address End Device as DHCP Client
ITEM NUMBER	34ED_013
DESCRIPTION	The next test case will ensure that: - After a new inauguration, renewal of train network address is performed. - In case that train network addresses are managed by End Devices with DHCP client, End Devices and DHCP servers shall support DHCP FORCERENEW message defined in IETF RFC 3203.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Client.
STEPS	Step 1: Supply the test bench. Step 2: Connect the laptop with the "Ethernet bus analyzer tool" opened, where the ethernet traffic of the End Device can be checked. Step 3: Force an inauguration on the Test Bench. Step 4: Check that on the Ethernet bus analyzer tool that the End Device receives the "Force Renew" UDP frame. Step 5: Check that the End Device sends one of the next messages to the DHCP server: - DHCP Request - DHCP Discover Step 6: Check that in the end of the DHCP frame transfer, the End Device gets correctly the IP address.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - The End Device receives the "DHCP Force Renew" message after a new inauguration. - The End Device sends the "DHCP Renew" or "DHCP Discover" message to the DHCP server. - The End Device gets correctly the new IP address.
DEPENDENCIES	

6.8.8 Test case TCCT34ED_008

Table 32 – Test case TCCT34ED_008

IDENTIFIER	TCCT34ED_008
TITLE	Renewal of Train Network Address End Device as DHCP Server
ITEM NUMBER	34ED_014
DESCRIPTION	The next test case will ensure that: - After a new inauguration, renewal of train network address is performed. - In case that train network addresses are managed by End Devices with DHCP client, End Devices and DHCP servers shall support DHCP FORCERENEW message defined in IETF RFC 3203.
REQUIREMENT	Optional
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Server.
STEPS	Step 1: Supply the test bench. Step 2: Connect the laptop configured as DHCP client, with the "Ethernet bus analyzer tool" opened, to an ETBN. Step 3: Force an Inauguration on the Test Bench. Step 4: Check that on the Ethernet bus analyzer tool that the End Device sends the "Force Renew" UDP frame. Step 5: Check that in the end of the DHCP frame transfer, the laptop gets correctly an IP address.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - The End Device sends the "DHCP Force Renew" message after a new inauguration. - The End Device gets correctly the new IP address.
DEPENDENCIES	

6.8.9 Test case TCCT34ED_009**Table 33 – Test case TCCT34ED_009**

IDENTIFIER	TCCT34ED_009
TITLE	DHCP Configuration Parameters for DHCP Server
ITEM NUMBER	34ED_015 – 34ED_028
DESCRIPTION	The next test case will ensure that: The mandatory and conditional requirements for DHCP options are supported.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Server
STEPS	Step 1: Supply the test bench. Step 2: Connect the laptop to an ETBN, with DHCP client configuration active. Step 3: Open the “Ethernet bus analyzer tool” on the laptop. Step 4: Open a command prompt on the laptop, and execute the next command: <code>ipconfig /renew</code> Step 5: Check on the “Ethernet bus analyzer tool” that next requirements have been received on the laptop: - Option 1: Subnet Mask (mandatory) - Option 3: router (mandatory) - Option 6: Domain Name Server (conditional) - Host Name (optional) - Broadcast Address (optional). - NTP Server Options (conditional). - Vendor Specific Information (optional) - Option 51: IP Address lease Time (mandatory) - Option 53: DHCP message type (mandatory) - Option 54: Server Identifier (mandatory) - Option 55: Parameter request list (mandatory) - Message (optional) - Client-Identifier (optional) - Relay Agent Information option (optional)
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The mandatory and conditional DHCP requirements have been received.
DEPENDENCIES	

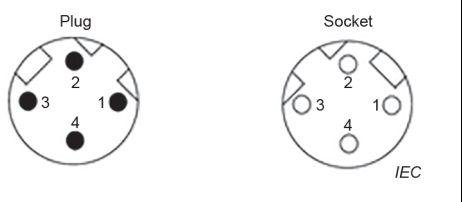
6.8.10 Test case TCCT34ED_010

Table 34 – Test case TCCT34ED_010

IDENTIFIER	TCCT34ED_010
TITLE	Physical Layer Protocol
ITEM NUMBER	34ED_029 – 34ED_031
DESCRIPTION	The next test case will ensure that: • End Devices shall support 100BASE-TX. • Full Duplex mode shall be supported to avoid collisions. • Auto negotiation function shall be able to be supported for connecting Temporary End Devices.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Server.
STEPS	Step 1: Supply the test bench. Step 2: Connect the laptop to a Switch, where the End Device will be connected. Step 3: Connect to the switch with the laptop, and force a 100BASE-TX Full Duplex configuration. Step 4: Connect the End Device to the Switch. Step 5: Check that the link of the End device is correct. Step 6: Open the "Ethernet bus analyzer tool" on the laptop. Step 7: Download a file from the End Device to the laptop, using FTP protocol. Step 8: Check on the "Ethernet bus analyzer tool" that the transmission speed is correct. Step 9: Change the Switch configuration to Auto negotiation function. Step 10: Repeat steps 5 to 9.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: • The communication between the End Device and the laptop is correct with 100BASE-TX Full Duplex configuration. • The communication between the End Device and the laptop is correct with Auto-Negotiation configuration.
DEPENDENCIES	

6.8.11 Test case TCCT34ED_011

Table 35 – Test case TCCT34ED_011

IDENTIFIER	TCCT34ED_011										
TITLE	Physical Layer Cables										
ITEM NUMBER	34ED_032 – 34ED_035										
DESCRIPTION	The next test case will ensure that: - M12 D-coded connector (socket), defined in IEC 61076-2-101, is supported on the - Network Device side. In this case, M12 D-coded plug connector will be used on the cable side. - IEC 61076-3-104 socket (outlet) can be used on the Network Device side. In this case, IEC 61076-3-104 plug connector will be used on the cable side. - RJ45 socket, defined in TIA/EIA-568-B, can be used for connecting Temporary End Devices on the Network Device side. In this case RJ45 plug connector shall be used on the cable side. - Pinning for M12 connector shall be as illustrated in 4.9.4.3.1, Table 11 of IEC 61375-3-4:2014.										
REQUIREMENT	Mandatory										
DUT	End Device										
TEST SETUP	Test Bench										
INITIAL CONDITIONS	Test Bench powered off. 100BASE-TX or 10BASE-T is used.										
STEPS	Step 1: Check that M12 D-coded connector (socket), defined in IEC 61076-2-101, is supported on the Network Device side. In this case, M12 D-coded plug connector will be used on the cable side. Step 2: Check that IEC 61076-3-104 socket (outlet) can be used on the Network Device side. In this case, IEC 61076-3-104 plug connector will be used on the cable side. Step 3: Check that RJ45 socket, defined in TIA/EIA-568-B, can be used for connecting Temporary End Devices on the Network Device side. In this case RJ45 plug connector shall be used on the cable side. Step 4: Check that the pinning for M12 connector is as shown in the figure below:  <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Pin</th><th>Signal</th></tr> </thead> <tbody> <tr> <td>1</td><td>TD+</td></tr> <tr> <td>2</td><td>RD+</td></tr> <tr> <td>3</td><td>TD–</td></tr> <tr> <td>4</td><td>RD–</td></tr> </tbody> </table> IEC	Pin	Signal	1	TD+	2	RD+	3	TD–	4	RD–
Pin	Signal										
1	TD+										
2	RD+										
3	TD–										
4	RD–										

IDENTIFIER	TCCT34ED_011
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The connectors used are those specified in the test.
DEPENDENCIES	

6.8.12 Test case TCCT34ED_012

Table 36 – Test case TCCT34ED_012

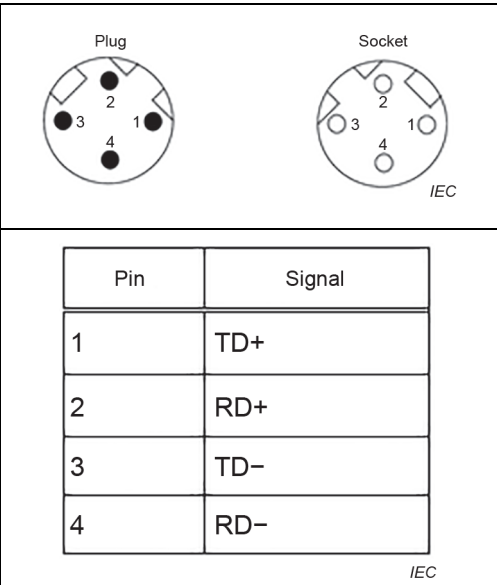
IDENTIFIER	TCCT34ED_012
TITLE	Physical Layer Protocol
ITEM NUMBER	34ED_036 – 34ED_039
DESCRIPTION	The next test case will ensure that: - End Devices shall support 100BASE-TX. - Full Duplex mode shall be supported to avoid collisions. - Auto negotiation function shall be able to be supported for connecting Temporary End Devices.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Server.
STEPS	Step 1: Supply the test bench. Step 2: Connect the laptop to a Switch, where the End Device will be connected. Step 3: Connect to the switch with the laptop, and force a 100BASE-TX, Full Duplex configuration. Step 4: Connect the End Device to the Switch. Step 5: Check that the link of the End device is correct. Step 6: Open the "Ethernet bus analyzer tool" on the laptop. Step 7: Download a file from the End Device to the laptop, using FTP protocol. Step 8: Check on the "Ethernet bus analyzer tool" that the transmission speed is correct. Step 9: Change the Switch configuration to Auto negotiation function. Step 10: Repeat steps 5 to 9.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - The communication between the End Device and the laptop is correct with 100BASE-TX Full Duplex configuration. - The communication between the End Device and the laptop is correct with Auto-Negotiation configuration.
DEPENDENCIES	

6.8.13 Test case TCCT34ED_013**Table 37 – Test case TCCT34ED_013**

IDENTIFIER	TCCT34ED_013
TITLE	Physical Layer Cables
ITEM NUMBER	34ED_040 – 34ED_041
DESCRIPTION	The next test case will ensure that: • Cables shall conform to ISO/IEC 11801 and IEC 61156-6. • Class D (Category 5e) with two twisted pairs shall be supported.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Client
STEPS	Step 1: Connect cables that conform to ISO/IEC 11801 and IEC 61156-6 between the End device and the Test Bench. Step 2: Check that the End Device gets a correct IP address via DHCP. Step 3: Perform a ping from the laptop to the End Device and check that there exists a response to the frames. Step 4: Connect Class D (Category 5e) with two twisted pairs between the End Device and the Test bench. Step 5: Check that the End Device gets a correct IP address via DHCP. Step 6: Perform a ping from the laptop to the End Device and check that there exists a response to the frames.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: • The communication between the End Device and the laptop is correct with cables that conform to ISO/IEC 11801 and IEC 61156-6 • The communication between the End Device and the laptop is correct with Class D (Category 5e) with two twisted pairs cable.
DEPENDENCIES	

6.8.14 Test case TCCT34ED_014

Table 38 – Test case TCCT34ED_014

IDENTIFIER	TCCT34ED_014										
TITLE	Physical Layer Connectors										
ITEM NUMBER	34ED_042 – 34ED_045										
DESCRIPTION	The next test case will ensure that: - M12 D-coded connector (socket), defined in IEC 61076-2-101, is supported on the End Device side. In this case, M12 D-coded plug connector shall be used on the cable side. - IEC 61076-3-104 socket (outlet) can be used on the End Device side. In this case, IEC 61076-3-104 plug connector shall be used on the cable side. - RJ45 socket, defined in TIA/EIA-568-B, can be used for the Temporary End Device on the End Device side. In this case RJ45 plug connector shall be used on the cable side. - Pinning for M12 connector is as specified.										
REQUIREMENT	Mandatory										
DUT	End Device										
TEST SETUP	Test Bench										
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Client										
STEPS	Step 1: Check that M12 D-coded connector (socket), defined in IEC 61076-2-101, is supported on the End Device side. In this case, M12 D-coded plug connector shall be used on the cable side Step 2: Check that IEC 61076-3-104 socket (outlet) can be used on the End Device side. In this case, IEC 61076-3-104 plug connector shall be used on the cable side. Step 3: Check that RJ45 socket, defined in TIA/EIA-568-B, can be used for the Temporary End Device on the End Device side. In this case RJ45 plug connector shall be used on the cable side. Step 4: Check that the pinning for M12 connector is as shown in the figure below:  <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Pin</th><th>Signal</th></tr> </thead> <tbody> <tr> <td>1</td><td>TD+</td></tr> <tr> <td>2</td><td>RD+</td></tr> <tr> <td>3</td><td>TD-</td></tr> <tr> <td>4</td><td>RD-</td></tr> </tbody> </table> IEC	Pin	Signal	1	TD+	2	RD+	3	TD-	4	RD-
Pin	Signal										
1	TD+										
2	RD+										
3	TD-										
4	RD-										

IDENTIFIER	TCCT34ED_014
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The connectors used are those specified in the test.
DEPENDENCIES	

6.8.15 Test case TCCT34ED_015

Table 39 – Test case TCCT34ED_015

IDENTIFIER	TCCT34ED_015
TITLE	End Device MAC Address
ITEM NUMBER	34ED_046 – 34ED_047
DESCRIPTION	The next test case will ensure that: - MAC in the Link Layer shall conform to IEEE 802.3. - MAC service with basic frame, which is defined in IEEE 802.3, shall be supported.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Perform a ping from the laptop to the End Device. Step 6: Check that there exists a response from the End Device to the laptop. Step 7: Check that the mac address conforms to IEEE 802.3. Step 8: Check that MAC service with basic frame is supported by the End Device.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - MAC in the Link Layer conforms to IEEE 802.3. - MAC service with basic frame is supported by the End Device.
DEPENDENCIES	

6.8.16 Test case TCCT34ED_016

Table 40 – Test case TCCT34ED_016

IDENTIFIER	TCCT34ED_016
TITLE	End Device Network Layer
ITEM NUMBER	34ED_048 – 34ED_050
DESCRIPTION	The next test case will ensure that: • IP version4, which is defined in IETF RFC 791, is supported by the End Device • ARP is supported by the End Device. • ICMP is supported by the End Device.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Perform a ping from the laptop to the End Device. Step 6: Check that there exists a response from the End Device to the laptop. Step 7: Check on the bus analyzer tool that there exists an ARP message sent by the End Device, and conforms to IETF RFC 792 standard. Step 8: Check on the bus analyzer tool that there exists an ICMP message sent by the End Device, and conforms to IETF RFC 792 standard. Step 9: Check on the same ICMP frame, that IPV4 format conforms to IETF RFC 792 standard.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: • MAC in the Link Layer conforms to IEEE 802.3. • MAC service with basic frame is supported by the End Device.
DEPENDENCIES	

6.8.17 Test case TCCT34ED_017**Table 41 – Test case TCCT34ED_017**

IDENTIFIER	TCCT34ED_017
TITLE	End Device Network Layer (UDP)
ITEM NUMBER	34ED_051
DESCRIPTION	The next test case will ensure that: • UDP, which is defined in IETF RFC 768, is supported by the End Device.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off The End Device shall be prepared for sending UDP frames.
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: An UDP frame shall be sent by the End Device to the laptop. Step 6: Check on the bus analyzer tool that the UDP frame conforms to IETF RFC 768: • Source port (16 bits) • Destination port (16 bits) • Length (16 bits) • Checksum (16 bits) • Data
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: • that the UDP frame sent by the End Device conforms to IETF RFC 768.
DEPENDENCIES	

6.8.18 Test case TCCT34ED_018

Table 42 – Test case TCCT34ED_018

IDENTIFIER	TCCT34ED_018
TITLE	End Device Network Layer (TCP)
ITEM NUMBER	34ED_052
DESCRIPTION	The next test case will ensure that: TCP, which is defined in IETF RFC 793, shall be supported.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off The End Device shall be prepared for sending TCP frames.
STEPS	Step 1: Supply the test bench. Step 2: Connect the End Device to an ETBN with the correct IP address. Step 3: Connect the laptop to an ETBN with the correct IP address. Step 4: Open the "Ethernet bus analyzer tool" to capture the frames that are going to be sent by the devices. Step 5: Open a command prompt on the laptop, and open a port for the transmission of TCP frames (for example: port 4567). Step 6: Establish a TCP connection on the End Device side on the same port opened on the laptop. Step 7: Send a message to the specified port and check on the laptop that the message is received.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: that the UDP frame sent by the End Device conforms to IETF RFC 768.
DEPENDENCIES	

6.8.19 Test case TCCT34ED_019**Table 43 – Test case TCCT34ED_019**

IDENTIFIER	TCCT34ED_019
TITLE	DHCP Client function
ITEM NUMBER	34ED_053
DESCRIPTION	The next test case will ensure that: DHCP client function, which is defined in IETF RFC 2131, may be supported.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Client There shall be a DHCP Server on the Test Bench.
STEPS	Step 1: Connect the End Device to a Consist switch, with DHCP client configuration active. Step 2: Connect the laptop to a Consist switch, with DHCP client configuration active. Step 3: Supply the test bench. Step 4: Check on the End Device that a correct IP address has been received. Step 5: Perform a "ping" from the laptop to the End Device, and check that there exists a response from the End Device.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: - The End Device gets a correct IP address. - The End Device responds to the "ping" sent by the laptop.
DEPENDENCIES	

6.8.20 Test case TCCT34ED_020

Table 44 – Test case TCCT34ED_020

IDENTIFIER	TCCT34ED_020
TITLE	DNS Client function
ITEM NUMBER	34ED_054
DESCRIPTION	The next test case will ensure that: DNS client function, which is defined in IETF RFC 1034, may be supported for mapping between IP addresses and names.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. End Device configured with DHCP Client and DNS client. There shall be a DHCP Server and DNS Server on the Test Bench.
STEPS	Step 1: Connect the End Device to a Consist switch, with DHCP and DNS client configuration active. Step 2: Connect the laptop to a Consist switch, with DHCP and DNS client configuration active. Step 3: Supply the test bench. Step 4: Check on the End Device that a correct IP address has been received. Step 5: Perform a "ping" using the DNS name of the End Device from the laptop, and check that there exists a response from the End Device.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The End Device responds to the "ping" sent by the laptop using the DNS name.
DEPENDENCIES	

6.8.21 Test case TCCT34ED_021**Table 45 – Test case TCCT34ED_021**

IDENTIFIER	TCCT34ED_021
TITLE	SNTP Client function
ITEM NUMBER	34ED_055
DESCRIPTION	The next test case will ensure that: NTP client function which is defined in IETF RFC 1361 or NTP version 3 client function which is defined in IETF RFC 1305 may be supported for synchronizing the time.
REQUIREMENT	Mandatory
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. Laptop prepared with SNTP Server.
STEPS	Step 1: Connect the End Device to a Consist switch, with SNTP client configured. Step 2: Connect the laptop to a Consist switch, with SNTP server configured. Step 3: Supply the test bench. Step 4: Check the time on the End Device. Step 5: Change the time on the SNTP Server. Step 6: Check that the End Device has synchronized the time.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The End Device synchronizes its time when changing the SNTP Server time.
DEPENDENCIES	

6.8.22 Test case TCCT34ED_022**Table 46 – Test case TCCT34ED_022**

IDENTIFIER	TCCT34ED_022
TITLE	SNMP Agent functions
ITEM NUMBER	34ED_056
DESCRIPTION	The next test case will ensure that: Communication devices in ECN should support SNMP agent functions for network management.
REQUIREMENT	Optional
DUT	End Device
TEST SETUP	Test Bench
INITIAL CONDITIONS	Test Bench powered off. Laptop prepared with SNMP monitoring tool. The End Device will provide the MIB file. The End Device will provide the SNMP community.
STEPS	Step 1: Connect the End Device to a Consist switch, with SNMP agent configured. Step 2: Connect the laptop to a Consist switch, SNMP monitoring tool installed. Step 3: Supply the test bench. Step 4: Execute the SNMP monitoring tool on the laptop, and using the IP address of the End Device. Step 5: Check that some parameters of the MIB can be requested.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	OK if: The End Device synchronizes its time when changing the SNTP Server time.
DEPENDENCIES	

7 Conformance test of ECN-equipped consist**7.1 Scope**

This clause specifies conformance test cases for testing the Network Devices within ECN.

7.2 Related requirements

As given in IEC 61375-2-3:2015 and IEC 61375-3-4:2014.

7.3 PICS Pro-forma**7.3.1 General**

PICS pro-forma shall be applied in case that ECN based on IEC 61375-3-4 and IEC 61375-2-3 is used.

The PICS provides a statement of the capabilities and options which have been implemented, and any features which have been omitted, see 4.1.3.1. Instructions about filling the PICS pro-forma are given in Annex B.

7.3.2 PICS tables

7.3.2.1 PICS for Recovery in case of network failure

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	Redundancy managed at network level – Measured recovery time – PD cycle time used for measuring	IEC 61375-3-4:2014, 4.5.2.2	C	[] [] ms max. [] ms max.	When redundancy scheme managed at network level is applied in the ECN, recovery time of network shall be measured in the conformance test.
2	Redundancy managed at network level – A single network component failure	IEC 61375-3-4:2014, 4.5.3	C	[]	When redundancy scheme is applied in ECN, a single network component failure shall not prevent the rest of the network from working without separation of the network so that the application can maintain its function.
3	Redundancy managed at network level – Connectivity loss duration time caused by the reconfiguration of the network	IEC 61375-3-4:2014, 4.5.3	C	[]	When a network component comes up late or a network component comes up again (reboot), connectivity loss duration time caused by the reconfiguration of the network shall be equal to or less than the value of the recovery time requirement.
4	Redundancy managed at network level – Forwarding loop not formed	IEC 61375-3-4:2014, 4.5.3	C	[]	When redundancy scheme managed at network level is applied in the ECN, forwarding loop shall not be formed in any time to avoid broadcast storm for instance.

7.3.2.2 PICS for IP address assignment via DHCP

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	Consist Network address – IPv4 private address	IEC 61375-3-4:2014, 4.7.1	M	[]	Consist Network address shall use IPv4 private address space defined in IETF RFC1918, class A private address should be used.
2	Consist Network address – class A private address	IEC 61375-3-4:2014, 4.7.1	C	[]	When – class A private address is used, and – ECN is connected to ETB, and – Consist Network address is not identical to train network address, addresses from 10.0.0.0 to 10.127.255.255 (10.0/9) shall be used.
3	Train network address	IEC 61375-3-4:2014, 4.7.2	C	[]	Train wide addressing of a communication device shall be possible with a train network address which is unique in the train if the ECN is connected to ETB. Source and destination addresses in the communication on ETB shall be train network addresses.
4	Service which maps train network addresses to Consist Network addresses	IEC 61375-3-4:2014, 4.7.2	C	[]	If train network address is not identical to Consist Network address, ECN shall support a service which maps train network addresses to Consist Network addresses.

7.3.2.3 PICS for Name (TCN-URI address) resolution via DNS

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	DNS server function	IEC 61375-3-4:2014 4.7.4.1	O	[]	ECN should provide DNS server function.
2	DNS server implemented in TBN	IEC 61375-3-4:2014, 4.7.4.1	O	[]	In case that the ECN is attached to ETB, it is recommended that the server is implemented in TBN.
3	Redundant DNS server	IEC 61375-3-4:2014, 4.7.4.1	O	[]	DNS server should be redundant.
4	Train wide domain name space addressable	IEC 61375-3-4:2014, 4.7.4.1	C	[]	Each communication device which could be a destination device for communication over ETB shall be addressable in train wide domain name space.
5	DNS client function (in ED)	IEC 61375-3-4:2014, 4.10.6	C	[]	Train Topology aware End Devices shall support DNS client function to resolve train network addresses from hostnames or function names; train network addresses may change by inaugurations.
6	Fully qualified TCN domain name	IEC 61375-2-3:2015, 5.4.3	C	[]	Communicating entities (sources and destinations) shall be identified by a fully qualified TCN domain name which is derived from the TCN domain name space.
7	TCN URI Identifier	IEC 61375-2-3:2015, 5.4.4.3	C	[]	The TCN URI Identifier shall be used to identify the source function/device and the destination function/device of information.
8	Physical addressing scheme	IEC 61375-2-3:2015, 5.4.4.3	O	[]	For addressing a physical device, the physical addressing scheme defined in IEC 61375-2-5 shall be used.
9	Function device identifier	IEC 61375-2-3:2015, 5.4.4.6.2	C	[]	Function devices shall be identified by labels following the syntax given in IEC 61375-2-3:2015, Table 8.
10	Vehicle identifier	IEC 61375-2-3:2015, 5.4.4.6.3	C	[]	Vehicles shall be identified by labels following the syntax given in 61375-2-3:2015, Table 10.
11	Consist identifier	IEC 61375-2-3:2015, 5.4.4.6.4	C	[]	Consists shall be identified by labels following the syntax given in IEC 61375-2-3:2015, Table 12.
12	Closed train identifier	IEC 61375-2-3:2015, 5.4.4.6.5	O	[]	Closed trains shall be identified by labels following the syntax given in 61375-2-3:2015, Table 14.
13	Train identifier	IEC 61375-2-3:2015, 5.4.4.6.6	C	[]	Trains shall be identified by labels following the syntax given in 61375-2-3:2015, Table 16.
14	TCN-URI mapping to IP addresses	IEC 61375-2-3:2015, 5.4.5.1	C	[]	TCN-URI shall be mapped to IP addresses in accordance to the IP addressing scheme defined in IEC 61375-2-5.
15	All-train group address	IEC 61375-2-3:2015, 5.4.5.2	C	[]	An all-train group address shall be assigned to a function if FUNCTION_INFO parameter 'etbld' is set to 255 (see 61375-2-3:2015, 5.3.3.2.4).
16	Other URI schemas	IEC 61375-2-3:2015, 5.4.6	O	[]	Other URI schemas like the commonly used uniform resource locators (URL) with the syntax scheme://domain:port/path?....." may be used.

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
17	Consist DNS server	IEC 61375-2-3:2015, 5.5.2	C	[]	Each consist shall provide at least one consist DNS name server which is authoritative for this consist domain.
18	Functional address resolution – Consist DNS server resolve TCN-URI host parts	IEC 61375-2-3:2015, 5.5.3	C	[]	The consist DNS server shall be able to resolve TCN-URI host parts as defined in 61375-2-3:2015, 5.4.4 to the corresponding IP address as defined in IEC 61375-2-5.
19	DNS protocol	IEC 61375-2-3:2015, 5.5.4	C	[]	The consist DNS name server shall implement the DNS protocols as specified in RFC 1034 and RFC 1035.
20	Consist DNS server – Recursive client queries	IEC 61375-2-3:2015, 5.5.4	C	[]	The consist DNS name server shall accept recursive client queries as shown in the example of 61375-2-3:2015, Figure 22.
21	Consist DNS server – Resolving time	IEC 61375-2-3:2015, 5.5.4	C	[]	TCN-URIs addressing destinations within the local or remote consists shall be resolved within a time of 1,0 s.

7.3.2.4 PICS for Switch basic functions

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	Ethernet port between ED and CS and between TBNs and CS	IEC 61375-3-4:2014, 4.2.1	M	[]	Ethernet ports between End Devices and Consist Switches and between TBNs and Consist Switches shall follow IEEE 802.3 standard.
2	Ethernet port between CS and CS	IEC 61375-3-4:2014, 4.2.1	O	[]	Ethernet ports between Consist Switches should follow IEEE 802.3 standard.
3	Frame forwarding (IEEE 802.3)	IEC 61375-3-4:2014, 4.4	M	[]	ECN shall receive MAC frames defined in IEEE 802.3 from End Devices and forward the MAC frames to the designated End Devices identified by the destination address fields of the MAC frames.
4	Frame forwarding (IEEE 802.1D)	IEC 61375-3-4:2014, 4.9.5.1	M	[]	Frame relaying, which is defined in IEEE 802.1D, shall be supported. Frame relaying provides frame reception, frame transmission, and frame forwarding.
5	Frame filtering (IEEE 802.1D)	IEC 61375-3-4:2014, 4.9.5.1	M	[]	Frame filtering, which is defined in IEEE 802.1D, shall be supported. Frame filtering provides learning of addresses and filtering database.

7.3.2.5 PICS for Priority levels

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment	Implementation
1	Default priority level	IEC 61375-3-4:2014 4.6.2	M	[]	According to IEEE 802.1D there are 8 priority levels, the highest priority level is 7 and the lowest priority level is 0. Default priority level shall be 0.	-
2	ETB priority level	IEC 61375-3-4:2014, 4.6.2	C	[]	In case of communication over ETB, priority level of a packet shall conform to IEC 61375-2-5.	-
3	Qos function mechanisms	IEC 61375-3-4:2014, 4.6.1	M	[]	Quality of service shall be provided by Consist Switches	
4	Number of queues	IEC 61375-3-4:2014, 4.6.2	M	[]	Consist Switches shall support 2 priority queues at minimum when QoS is provided.	
5	Priority mapping	IEC 61375-3-4:2014, 4.6.2	M	[]	Mapping of priority levels to data classes shall be determined according to the requirements from applications used in the ECN since data classes to be used and their performance parameters depend on applications.	
6	Classification	IEC 61375-3-4:2014, 4.9.5.1	M	[]	Frame queuing, which is defined in IEEE 802.1D, shall be supported by Consist Switches. Frame queuing can handle multiple data classes during frame relaying to achieve quality of service.	
7	Classification	IEC 61375-3-4:2014, 4.9.5.1	O	[]	Frame queuing can handle multiple data classes during frame relaying to achieve quality of service.	

7.3.2.6 PICS for Ingress rate limiting

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	Ingress rate limiting	IEC 61375-3-4:2014, 4.9.5.1	O	[]	Ingress rate limiting which is defined in 61375-3-4:2014, 4.6.5 may be supported.

7.3.2.7 PICS for Egress rate shaping

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	Egress rate shaping	IEC 61375-3-4:2014, 4.9.5.1	O	[]	Egress rate shaping which is defined in 61375-3-4:2014, 4.6.6 may be supported.

7.3.2.8 PICS for Untagged/tagged frames

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	Frame format	IEC 61375-3-4:2014, 4.4	M	[]	Consist Switches shall be able to relay both basic (untagged) and tagged MAC frames.
2	ECN VLAN function	IEC 61375-3-4:2014, 4.4	M	[]	ECN shall be able to provide VLAN functions defined in IEEE 802.1Q. VLAN functions required in ECN are defined in 4.9.
3	VLAN insert and remove function	IEC 61375-3-4:2014, 4.9.5.1	M	[]	Frame tagging and untagging, which is defined IEEE 802.1Q, shall be supported by Consist Switches. Frame tagging can insert the tag in the basic (untagged) frame for the ingress ports, and frame untagging can remove the tag from the tagged frame for the egress port.

7.3.2.9 PICS for Switching and routing performance

None.

7.3.2.10 PICS for NTP support

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	NTP server NTP client	IEC 61375-3-4:2014, 4.9.2, 4.10.6	O O	[] []	When NTP is used ECN shall provide NTP server function. NTP version 3

7.3.2.11 PICS for Management

Ref,	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	Management and remote management defined in IEEE802.1D	IEC 61375-3-4:2014, 4.9.5.1	C	[]	Management and remote management, which are defined in IEEE 802.1D, shall be supported by Managed Consist Switches.
2	ECN network management via SNMP	IEC 61375-3-4:2014, 4.12.1	O	[]	Communication devices in ECN should support SNMP agent functions for network management. SNMPv2 defined in IETF RFC 1901, 1905 and 1906 is the minimum requirement.
3	Standards MIB	IEC 61375-3-4:2014, 4.12.1	O	[]	Standards MIBs defined in IETF RFC 1213 should be supported.

7.3.2.12 PICS for ECSP interface (optional)

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	ECSP Interface	IEC 61375-2-3:2015, E.3.2 through E.3.4	O	[]	

7.3.2.13 PICS for TTDB manager interface (optional)

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	TTDB manager interface	IEC 61375-2-3:2015, E.4.2 through E.4.8	O	[]	

7.3.2.14 PICS for DNS TCN interface (optional)

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	DNS TCN interface	IEC 61375-2-3:2015, E.5.2.3	O	[]	

7.3.2.15 PICS for ETBN control interface (optional)

Ref	Question	Subclause	Req.	Impl. [Y/N]	Value/Comment
1	ETBN control interface	IEC 61375-2-3:2015, E.6.2 through E.6.3	O	[]	

7.4 Test cases**7.4.1 General**

Figure 5 shows a general configuration for ECN-equipped consist test.

The topology of ECN is not strictly defined in IEC 61375-3-4, but the requirements for Network Device interface and End Device interface are defined. The Network Device interface is located between Network Devices within ECN, and the End Device interface is between a Network Device and an End Device which is outside of ECN.

Furthermore, redundancy of the ECN is optional and may be formed in any topology.

The tests are performed in two communication cases. The one is the communication between the reference ED and the partner ED through the CSs under test. The other is the communication between the reference ED and the CS under test or the ETBN outside of the ECN.

Prior conditions for the test are in the following:

- Target of the test is Consist Switch which is a typical Network Device in ECN.
- For managing train network addresses, an ETBN may be connected with a CS in ECN, they may be integrated in a unit physically.
- Test application should be contained in Reference ED which is connected with the CS under test and also in another ED which works as a communication partner of the reference ED through the ECN.
- Decision of pass or fail should be made with the result in the test application within the reference ED.

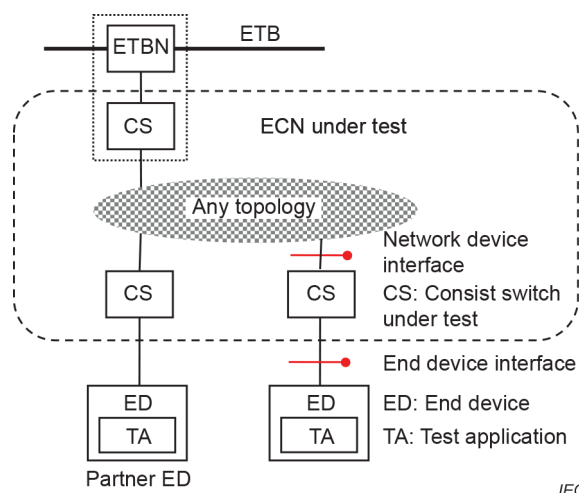


Figure 5 – General configuration for ECN-equipped consist test

7.4.2 Recovery in case of network failure

7.4.2.1 General

This test shall be executed in case that the ECN employs the network redundancy which can tolerate a single network component failure.

7.4.2.2 Purpose of test

To confirm the recovery function in case of network failure.

7.4.2.3 Prerequisite of test

Types of configurations for network redundancy are considered;

- One is, shown in Figure 6, network redundancy with redundant communication routes between the CSs, which is in the scope of this conformance test.
- Another is, shown in Figure 7, network redundancy with dual homing End Device interfaces, in which redundant routes between any ED to any ED are provided. This type of redundancy is mainly performed by the exclusive means in ED itself, therefore it is out of scope in the test.

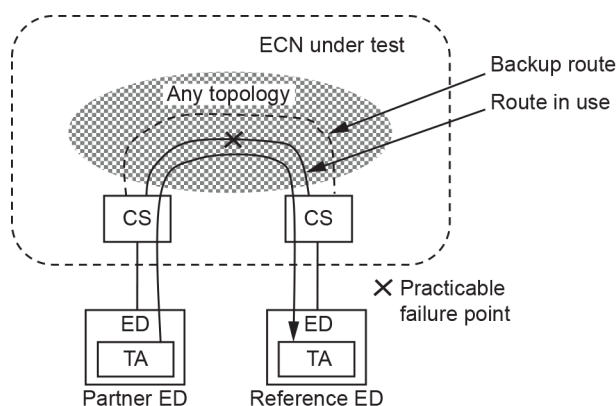


Figure 6 – Example of configuration for the test of network redundancy

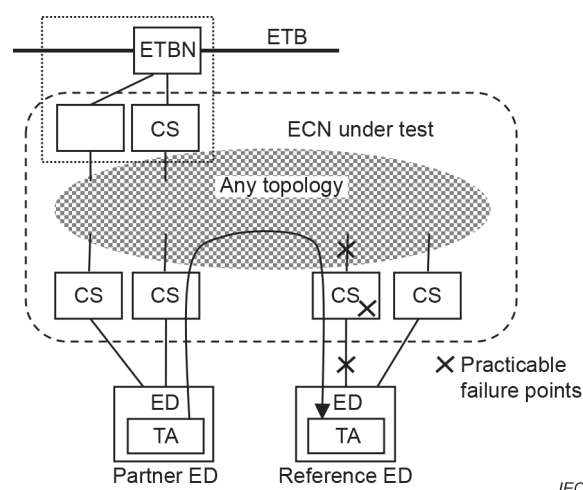


Figure 7 – Example of configuration of network redundancy with dual homing End Device interfaces

The ECN shall contain at least two CSs under test.

The reference ED for evaluating the test result and another ED for a communication partner are connected with the respective CSs under test.

One of practicable failures on the ECN occurs at a time.

Practicable failure points are:

- a link located on the communication route between the CSs which are connected with EDs for the test,
- an intermediate Network Device, if any, located on the same communication route in the above.

7.4.2.4 Method of test

In case of network redundancy with redundant communication routes between CSs, shown in Figure 6:

- The partner ED sends PD frames for test to the reference ED repeatedly with a certain period of time through the ECN.
- Force a single network component failure manually on the communication route in use between the CSs. The failure may be generated by removing the cable and so on.
- The recovery time shall be measured between the two time points in the following.
One is the time point when the last PD packet is received at the reference ED before the failure occurs, and the other is the time point when the first PD packet is received at the reference ED after recovery of communication route. These time points should be recorded in the test application or network analyzer software in the reference ED.
- Measuring of the recovery time should be performed at least ten times, and the maximum, the minimum and the average of the results should be calculated.

7.4.2.5 Criteria for judging

The recovery time measured in the above is within the maximum time which is determined by the applicant.

7.4.3 IP address assignment via DHCP

7.4.3.1 General

This test shall be executed in case that DHCP is supported in the ECN.

7.4.3.2 Purpose of test

To confirm IP address assignment by DHCP server to ED.

7.4.3.3 Prerequisite of test

DHCP server shall be implemented in the CS under test or available through the ECN.

ETBN may be connected with the ECN, in which DHCP server is implemented for managing train network address.

DHCP client uses UDP port 67 for the destination to DHCP server.

DHCP server uses UDP port 68 for the destination to DHCP client.

7.4.3.4 Method of test

After power up of the reference ED, verify the followings are assigned to the ED:

- IP address,
- Subnet mask,
- IP address of default gateway,
- DNS server address (Optional).

In case that train network addresses are managed by End Devices with DHCP client, DHCP server in the ETBN shall send FORCERENEW message to DHCP clients with train network addresses after inauguration, and DHCP clients shall renew train network addresses on receiving the FORCERENEW message.

Confirm the implementation of mandatory DHCP options with the PICS pro-forma.

Figure 8 shows an example of the configuration for the test of IP address assignment via DHCP in which DHCP client is located in the reference ED and DHCP server is in the CS under test or in the ETBN.

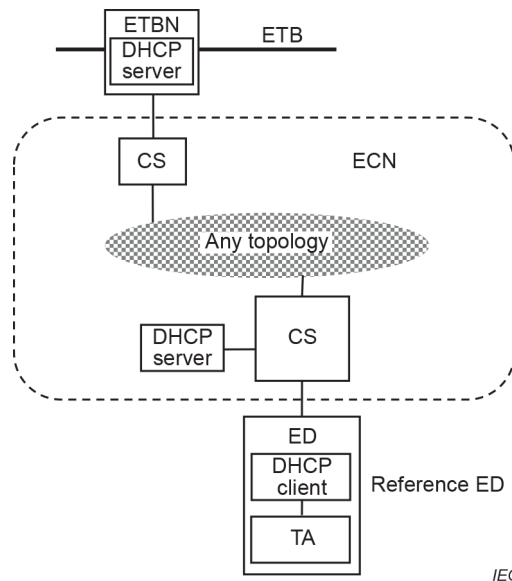


Figure 8 – Example of the configuration for the test of IP address assignment via DHCP

7.4.3.5 Criteria for judging

The assigned values are identical to the ones determined previously for test.

Implementations of the mandatory DHCP options are stated in the PICS pro-forma.

7.4.4 Name (TCN-URI address) resolution via DNS

7.4.4.1 General

This test shall be executed in case that DNS is supported in the ECN and IEC 61375-2-3 is applied to the ECN.

7.4.4.2 Purpose of test

To confirm the translation between the Name in TCN-URI schema defined in IEC 61375-2-3 and the IP address.

7.4.4.3 Prerequisite of test

DNS server shall be available through the ECN.

Reference ED as a DNS client knows IP addresses of the DNS server for test previously.

DNS client shall use UDP port 53 for the destination to DNS server.

DNS server shall use UDP port 53 for the sender to DNS client.

Test samples of Names in TCN-URI and corresponding IP addresses are provided previously.

Names in TCN-URI shall accord with the following schema:

trn://usr@fctdev.vehicle.consist.cltrain.train

7.4.4.4 Method of test

Reference ED as a DNS client sends a Name in TCN-URI.

After the IP address is returned from the DNS server, the reference ED verifies if it is correct.

The time shall be measured from the query reception until sending the answer to the client is completed at the server side.

7.4.4.5 Criteria for judging

Mapping between the Name and the IP address is correctly worked.

The time for resolving TCN-URI address is within 1,0 s.

Figure 9 shows an example of the configuration for the test in which DNS client is located in the reference ED and DNS server is in ETBN.

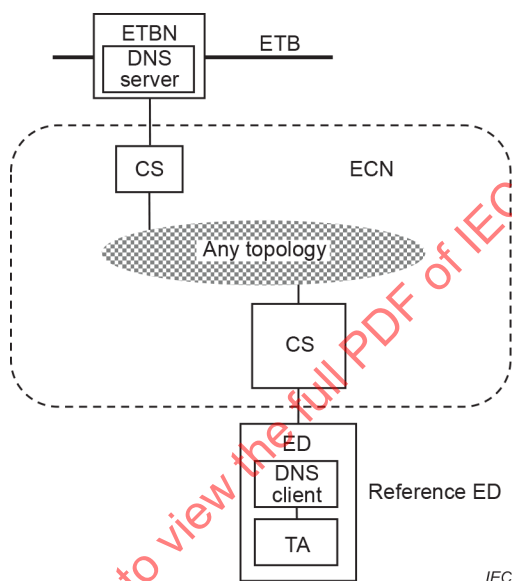


Figure 9 – Example of the configuration for the test of Name resolution via DNS

7.4.5 Switch basic functions

7.4.5.1 General

This test shall be executed in all cases.

7.4.5.2 Purpose of test

To confirm CS basic functions.

7.4.5.3 Prerequisite of test

None.

7.4.5.4 Method of test

Confirm the CS implementation with the PICS pro-forma for basic CS functions in 7.3.2.4.

7.4.5.5 Criteria for judging

Implementation of mandatory requirements for the CS basic functions is stated in the PICS pro-forma.

7.4.6 Priority levels

7.4.6.1 General

This test shall be executed in case that the ECN employs QoS service based on strict priority.

NOTE QoS support is conditional, but strict priority queuing is optional. See IEC 61375-3-4:2014, 4.6.4.

7.4.6.2 Purpose of test

To confirm QoS service based on strict priority.

7.4.6.3 Prerequisite of test

None.

7.4.6.4 Method of test

Connect plural EDs via ECN. One ED is a destination ED and others, two or more, are source EDs.

Source EDs as a group send plural different priority frames to a destination ED with port speed; each source ED sends frames with one assigned priority with port speed.

Capture the frames which are received at the destination ED.

Stop the source traffic in order from the highest priority to the lowest priority.

Capture the frames each time stopping a source traffic.

7.4.6.5 Criteria for judging

At the destination ED, all the highest priority frames in the traffic are captured even in the state that plural source EDs are sending different priority frames.

7.4.7 Ingress rate limiting

7.4.7.1 General

This test shall be executed in case that the ECN employs ingress rate limit function.

7.4.7.2 Purpose of test

To confirm Ingress rate limit function in CS.

7.4.7.3 Prerequisite of test

None.

7.4.7.4 Method of test

Connect a source ED and a destination ED via ECN.

Set ingress rate to the CS port which the source ED is connected.

Send plural different priority frames from a source ED to a destination ED with higher rate than ingress rate limit value.

Capture the frames which are received at the destination ED.

Measure traffic rate at the destination ED.

7.4.7.5 Criteria for judging

At the destination ED, low priority frames shall be discarded first to keep the rate limit on the ingress port.

7.4.8 Egress rate shaping

7.4.8.1 General

This test shall be executed in case that the ECN employs egress shaper function.

7.4.8.2 Purpose of test

To confirm egress shaper function in CS.

7.4.8.3 Prerequisite of test

None.

7.4.8.4 Method of test

Connect a source ED and a destination ED via ECN.

Set egress rate to the CS port which the destination ED is connected.

Send plural different priority frames from a source ED to a destination ED with higher rate than egress shaper rate.

Capture the frames which are received at the destination ED.

Measure traffic rate at the destination ED.

7.4.8.5 Criteria for judging

At the destination ED, low priority frames shall be discarded first to keep the egress shaper rate on the destination port.

7.4.9 Untagged/tagged frames

7.4.9.1 General

This test shall be executed to confirm VLAN untagged/tagged frames transfer.

7.4.9.2 Purpose of test

To confirm VLAN untagged/tagged frames transfer function.

7.4.9.3 Prerequisite of test

None.

7.4.9.4 Method of test

Untagged frames are transferred from source ED to destination ED via ECN.

Tagged frames are transferred from source ED to destination ED via ECN.

7.4.9.5 Criteria for judging

Receiving both untagged and tagged frame on the destination ED via ECN.

The number of untagged frames which are received on the destination ED shall the same number as the untagged frames which are sent from the source ED.

The number of tagged frames which are received on the destination ED shall the same number as the tagged frames which are sent from the source ED.

For test of VLAN tag insert function, the number of tagged frames which are received on the destination ED shall the same number as the untagged frames which are sent from the source ED.

For test of VLAN tag removal function, the number of untagged frames which are received on the destination ED shall the same number as the tagged frames which are sent from the source ED.

7.4.10 Switching and routing performance (optional)

7.4.10.1 General

This test should be executed to confirm Consist Switch performance.

7.4.10.2 Purpose of test

To confirm specific definition of the service parameters which are determined according to the requirements from the specific applications.

7.4.10.3 Prerequisite of test

None.

7.4.10.4 Method of test

Connect EDs for each application to a CS.

Send each data from EDs for Process, Message, Stream, Supervisory and Best effort data as a background traffic.

7.4.10.5 Criteria for judging

Measure minimum cycle time, maximum latency and maximum jitter on the receiving side of each application ED.

For each application, calculate values as follows based on measured values and number of CS deploy in ECN.

Confirm each values meet the values given in IEC 61375-3-4:2014, Table 6.

- [Minimum cycle time] × [number of CS]
- [Maximum latency] × [number of CS]
- [Maximum jitter] × [number of CS]

7.4.11 NTP support

7.4.11.1 General

This test shall be executed in case that NTP is supported in the ECN.

7.4.11.2 Purpose of test

To confirm the operation of the NTP server against NTP v3 client.

7.4.11.3 Prerequisite of test

NTP v3 client shall be implemented in the reference ED.

Server/client mode shall be used as synchronous mode.

NTP client shall use UDP port 123 for both the destination and the sender to NTP server.

7.4.11.4 Method of test

The reference ED sends a query for time to NTP server.

After receiving the response, NTP client adjusts the current time of the inner clock.

7.4.11.5 Criteria for judging

The operation of NTP query and response between NTP client and NTP server is correctly worked.

Figure 10 shows an example of the configuration for the test of NTP support in which NTP client is located in the reference ED and NTP server is in ETBN.

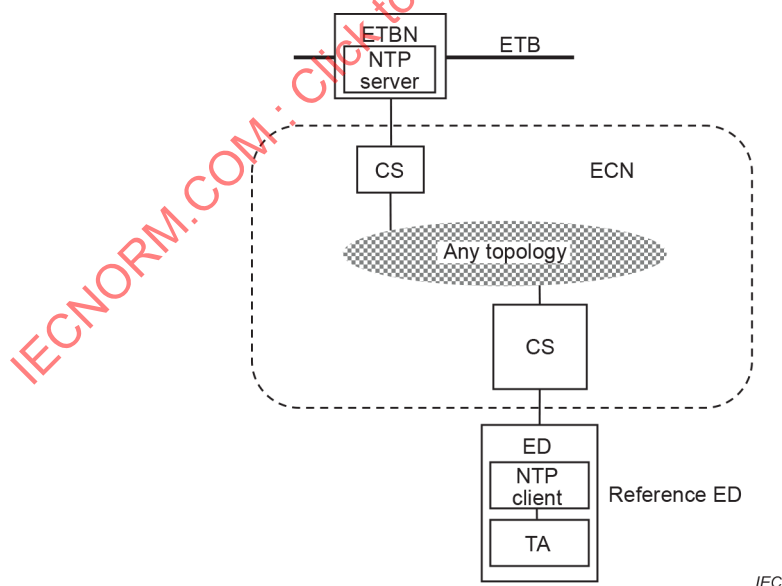


Figure 10 – Example of the configuration for the test of NTP support

7.4.12 Switch management

7.4.12.1 General

This test shall be executed in case that the CS is a managed switch.

7.4.12.2 Purpose of test

To confirm the bridge management and remote management function defined in IEEE 802.1D in the CS under test.

7.4.12.3 Prerequisite of test

None.

7.4.12.4 Method of test

Confirm the PICS pro-forma for IEEE 802.1D:1990, Annex A.14 Bridge management, and Annex A.15 Remote Management.

7.4.12.5 Criterion for judging

Implementation of the mandatory functions of IEEE 802.1D:1990, Clause 14 Bridge management, and Clause 15 Remote Management are stated.

7.4.13 Network management via SNMP

7.4.13.1 General

This test shall be executed in case that the CS is a managed switch.

7.4.13.2 Purpose of test

To confirm the functions of reading a MIB.

To confirm the functions of reading the MIB defined in IEC 61375-2-5:2014, Annex C: TTDP-MIB via SNMP in case that the ECN is connected with ETBN.

7.4.13.3 Prerequisite of test

SNMP v2 or more shall be supported in the reference ED.

The reference ED works as SNMP manager and the ETBN as SNMP agent.

SNMP manager shall use UDP port 161 for the destination to SNMP agent.

SNMP agent shall use UDP port 162 for the destination to SNMP manager.

A MIB shall be implemented in the CS under test. The standard MIBs defined in IETF RFC 1213 is recommended to be implemented.

TTDP-MIB shall be implemented in ETBN in case that the ECN is connected with ETBN.

7.4.13.4 Method of test

The reference ED reads out the MIB from the CS under test.

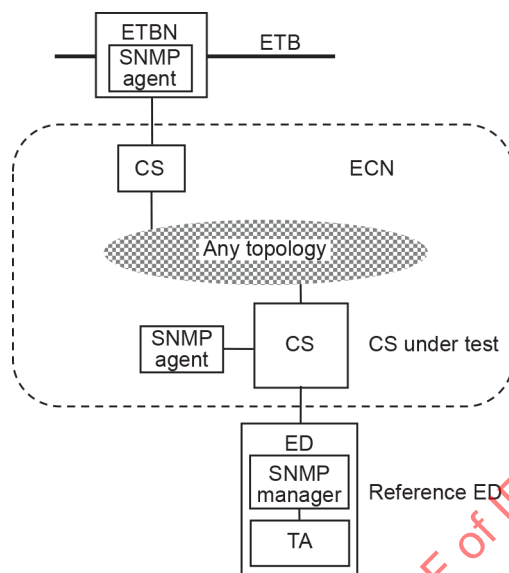
The reference ED read out the TTDP-MIB from the ETBN through the CS in the ECN in case that the ECN is connected with ETBN.

7.4.13.5 Criteria for judging

Contents of the MIB received at the reference ED are correct.

Contents of the TTDP-MIB received at the reference ED are identical to the TTDP-MIB in the ETBN in case that the ECN is connected with ETBN.

Figure 11 shows an example of the configuration for the test of Network management via SNMP in which SNMP manager is located in the reference ED and SNMP agents are in ETBN and CS.



IEC

Figure 11 – Example of the configuration for the test of Network management via SNMP

7.4.14 ECSP Interface (optional)

7.4.14.1 General

This test shall be executed in case that the ECN supports ECSP interface defined in IEC 61375-2-3:2015, Clause E.3.

7.4.14.2 Purpose of test

To confirm the communication interface between ECSP and ECSC.

7.4.14.3 Prerequisite of test

None.

7.4.14.4 Method of test

Connecting ECSP and ECSC, and capturing PD/MD telegram between ECSP and ECSC.

Confirming data contents of PD/MD telegram.

- ECSC (ED) sends control telegram to ECSP (ETBN).
 - ECSC receive status telegram from ECSP.
 - ECSC send confirmation message to ECSP.
- ECSC receive confirmation reply from ECSP.

7.4.14.5 Criterion for judging

Transmission is performed correctly.

The telegrams are same as the telegrams defined in IEC 61375-2-3:2015, E.3.2 through E.3.4.

7.4.15 TTDB manager interface (optional)

7.4.15.1 General

This test shall be executed in case that the ECN supports TTDB manager interface defined in IEC 61375-2-3:2015, Clause E.4.

7.4.15.2 Purpose of test

To confirm the communication interface between TTDB manager and ECSC.

7.4.15.3 Prerequisite of test

None.

7.4.15.4 Method of test

Connecting TTDB manager and ECSC and capturing PD/MD telegram between TTDB manager and ECSC.

Confirming data contents of PD/MD telegram.

- a) ECSC (ED) is pushed TTDB status from TTDB manager.
- b) ECSC is pushed TTDB notification from TTDB manager.
- c) ECSC send info request to TTDB manager.
ECSC receive info reply from TTDB manager.

7.4.15.5 Criterion for judging

Transmission is performed correctly.

The telegrams are same as the telegrams defined in IEC 61375-2-3:2015, E.4.2 through E.4.8.

7.4.16 DNS TCN interface (optional)

7.4.16.1 General

This test shall be executed in case that the ECN supports DNS TCN interface defined in IEC 61375-2-3:2015, Clause E.5.

7.4.16.2 Purpose of test

To confirm the communication interface between DNS server and ECSC.

7.4.16.3 Prerequisite of test

None.

7.4.16.4 Method of test

Connecting ED and DNS server, and capturing PD/MD telegram between ED and DNS server.

Confirming data contents of MD telegram.

- ED sends DNS request MD telegram to DNS server.
- ECSC receive DNS reply MD telegram from DNS server.

7.4.16.5 Criterion for judging

URI is resolved correctly.

The telegrams are the same as the telegrams defined in IEC 61375-2-3:2015, E.5.2.3.

7.4.17 ETBN control interface (optional)

7.4.17.1 General

This test shall be executed in case that the ECN supports ETBN control interface defined in IEC 61375-2-3:2015, Clause E.6.

7.4.17.2 Purpose of test

To confirm the communication interface between ETBN and ED.

7.4.17.3 Prerequisite of test

None

7.4.17.4 Method of test

Connecting ED and ETBN and capturing PD/MD telegram between ED and ETBN.

Confirming data contents of MD telegram.

- a) ED send control request telegram to ETBN.
ED receives status reply telegram from ETBN.
- b) ED send info request message to ETBN.
ED receives info reply message from ETBN.

7.4.17.5 Criterion for judging

Transmission is performed correctly.

The telegrams are same as the telegrams defined in IEC 61375-2-3:2015, E.6.2 through E.6.3.

8 Conformance test of ETBN

8.1 Test scope

8.1.1 General

This clause specifies PICS pro-forma and associated conformance test suites and test cases for testing ETBN functionality specified in IEC 61375-2-5:2014.

8.1.2 References

References used in this clause:

- [2-3]: IEC 61375-2-3:2015
- [2-5]: IEC 61375-2-5:2014
- [2-5-COR1]: Appendix C of "D6.3 ECN-ETB Conformance Testing Protocol", CONNECTA project <https://projects.shift2rail.org/>
Appendix C provides additional clarifications to IEC 61375-2-5:2014
- [2-8]: this document

[802.1AB]: IEEE 802.1AB:2009, IEEE Standard for Local and metropolitan area networks – Station and Media Access Control Connectivity Discovery

Subclause references in this clause refer to subclauses in IEC 61375-2-5:2014, unless explicitly stated otherwise.

8.2 PICS Pro-forma

8.2.1 General

The PICS provides a statement of the capabilities and options which have been implemented, and any features which have been omitted, see 4.1.3.1. Instructions about filling the PICS pro-forma are given in Annex B.

Additional symbols used in PICS tables:

- O.n: optional, but support of at least one of the group of options labelled by the same numeral n is required
- X: prohibited
- pred: conditional-item symbol, including predicate identification

Subclause references in PICS tables refer to subclauses in IEC 61375-2-5:2014, unless explicitly stated otherwise.

8.2.2 PICS tables

8.2.2.1 ETBN major capabilities

Item	Question	Req.	Subclause	Implementation
PORT	Does ETBN implementation support ETB port and bypass relay capabilities as specified in 4.2.2-4.2.4 and 8.3.2.2?	M	4.2.2-4.2.4, 8.3.2.2	Y[]
POE	Does ETBN implementation provide PoE support on ETB ports as specified in 4.2 and 4.3?	O	4.2.2-4.2.4, 4.3	Y[] N[]
LAG	Does ETBN provide link aggregation control using TTDP HELLO as specified in 4.4, 8.6.4 and 8.9.1?	M	4.4, 8.6.4, 8.9.1	Y[]
LAG_RED	Does ETBN provide ETB line redundancy?	O	4.4, 8.9.1, Annex A	Y[] N[] If "Yes", state nb of ETB lines: 2[] 4[]
ETBN_RED	Is the ETBN capable of acting as redundant ETBN?	O	9	Y[] N[]
MULT_CN	Is the ETBN capable of connecting multiple CNs?	O	6.5.1.1, 8.7.6, Annex A	Y[] N[]
HELLO	Does implementation transmit and accept TTDP HELLO packets with format and content as specified in 8.7.2-8.7.5	M	8.7.2-8.7.5	Y[]
LINK	Does implementation support Data Link Layer as specified in Clause 5?	M	5	Y[]
NET	Does implementation support general ETBN Network services	M	6.1	Y[]
UCAST	Does the implementation support ETBN IP unicast services?	M	6, 9	Y[]
MCAST	Does the implementation support ETBN IP multicast services?	M	6, 9	Y[]

Item	Question	Req.	Subclause	Implementation
TRANS	Does implementation support ETB Transport Layer as specified in Clause 7?	M	7	Y[]
TOPO-FRAME	Does implementation transmit and accept TTDP TOPOLOGY packets with format and content as specified in 8.7.2-8.7.4 and 8.7.6?	M	8.7.2-8.7.4, 8.7.6	Y[]
TOPO-PHY	Does implementation manage Physical Topology (Connectivity Table) based on exchange of TTDP TOPOLOGY messages?	M	8.2.1, 8.5.2, 8.6.1, 8.7.6, 8.8.1-8.8.4, 8.9.2	Y[]
TOPO-TND-GEN	Does the ETBN correctly calculate and transmit etbTopoCnt in TTDP TOPOLOGY frames?	M	8.8.5, 8.8.6	Y[]
TOPO-TND-CST	Can the ETBN be used to form at least the mandatory consist topology?	M	6.4.2.3.2-6.4.2.3.4., 6.6 6.4.2.3.3, 8.8.5 [2-8] 8.3.2.2-8.3.2.7	Y[]
TOPO-TND-TRAIN	Can the ETBN interact with ETBNs in remote consists being part of consist and train topologies compliant with the standard?	M	6.4.2.2-6.4.2.3.4, 6.6, 8.8.5, Annex A [2-8] 8.3.2.2-8.3.2.7	Y[]
TRANS-INHIBIT	Is the ETBN able to handle topology transitions correctly when inauguration is inhibited?	M	8.5.1, 8.5.2, 8.6.2, 8.11.1-8.11.3	Y[]

8.2.2.2 ETB port and bypass relay capabilities

Item	Question	Req.	Subclause	Implementation
PORT-1	Does ETB port implementation support 100 Mbit/s full duplex with auto-negotiation disabled?	M	4.2.2-4.2.4	Y[]
PORT-2	Does ETB port implementation support M12 port pinout specified in Table1 (4.2.2) with fixed-MDI mode (auto-sensing disabled)?	M	4.2.2 Item 2 of [2-5-COR1]	Y[]
PORT-3	Is ETB bypass relay set in bypass mode when unit has no power?	M	4.2.2, 8.3.2.2	Y[]
PORT-4	Does ETB bypass relay support single cross-over in bypass mode?	M	4.2.2-4.2.4 Item 5 of [2-5-COR1]	Y[]

8.2.2.3 Power Over Ethernet (PoE) capabilities

Regarding PoE Support (optional, as stated in Tables 1-3 of [2-5]), no PICS statements or test suites are defined as part of this document [2-8]. Instead a vendor with PoE support on ETBNs is referred to Clause 33 of IEEE 802.3:2012 for PoE Conformance.

8.2.2.4 ETB Line Status and Link Aggregation management

Item	Question	Req.	Subclause	Implementation
LAG-1	Does the implementation only include ETB lines which are logically up to distribute data in the link aggregate?	M	8.6.4 Item 15 of [2-5-COR1]	Y []
LAG-2	Does the implementation determine a line as logically up when two-way connectivity is verified (receiving TTDP HELLO with "Peer own line rcv status ok")?	M	8.6.4 Item 15 of [2-5-COR1]	Y []
LAG-3	Does implementation manage TTDP HELLO sending periods as specified in 8.9.1?	M	8.9.1	Y []
LAG-4	Does the implementation provide recovery within 200 ms in case of ETB line link down?	LAG_RED:M	8.9.1	Y [] N/A []
LAG-5	Does the implementation provide recovery within 200 ms in case of ETB line link up?	LAG_RED:M	8.9.1	Y [] N/A []
LAG-6	Is ETB interruption (recovery) max 200 ms when losing an intermediate ETBN?	M	8.6.4, 8.11.1 Item 13 of [2-5-COR1]	Y []
LAG-7	Is ETB interruption (recovery) max 200 ms when an intermediate ETBN comes up?	M	8.6.4, 8.11.1 Item 13 of [2-5-COR1]	Y []

Further explanations:

IEC 61375-2-5:2014, 4.4.3.4 and items 19-20 of IEC 61375-2-5 CORRECTION 1 define additional requirements with respect to link aggregation. Those requirements do not result in any PICS statements or test suites defined in this document. Instead the related 802.1AX PICS should be enough, with adjustments specified in item 20 of IEC 61375-2-5 CORRECTION 1.

8.2.2.5 Data link layer of ETB ports – Table 4 of IEC 61375-2-5:2014, Clause 5

Item	Question	Req.	Subclause	Implementation
LINK-1	MAC services and addressing IEEE 802.3	M	5 (Table 4)	Y []
LINK-2	LLC services, IEEE 802.2	X	5 (Table 4)	Y []
LINK-3	Frame Relaying, IEEE 802.1D PICS A.7	M	5 (Table 4)	Y []
LINK-4	Frame Filtering (layer 2 filtering), IEEE 802.1D, Clause 7, PICS A.8	M	5 (Table 4)	Y []
LINK-5	Frame Queuing, IEEE 802.1D, 7.7.3, 7.7.4 PICS A16- Annex G	M	5 (Table 4)	Y []
LINK-6	Frame tagging/untagging, IEEE 802.3, 3.5, IEEE 802.1Q (VLAN)	M	5 (Table 4)	Y []
LINK-7	VLAN Services, IEEE 802.1Q (VLAN), PICS A.21	M	5 (Table 4)	Y []
LINK-8	Port mirroring	O	5 (Table 4)	Y [] N []
LINK-9	Flow Control, IEEE 802.3, Part 2 Annex	O	5 (Table 4)	Y [] N []
LINK-10	Ingress rate limiting (policing)	O	5 (Table 4)	Y [] N []
LINK-11	Egress rate shaping	O	5 (Table 4)	Y [] N []
LINK-12	Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), IEEE 802.1D	X	5 (Table 4)	Y []
LINK-13	Management and Remote Mgt IEEE 802.1D, Clause 14 PICS A.14, A.15	M	5 (Table 4)	Y []

Further explanations:

For more details, see Clause 5 (Table 4) of 61375-2-5:2014.

From Table 4 the requirements for Link Aggregation and LLDP are left outside this PICS table, as the use of these protocols are covered in other parts of this document [2-8].

The requirements of data link layer services listed in the table above do not have any corresponding test suites in this conformance test. Some items are based on IEEE standards, and for them the associated standard PICS sections are assumed to be sufficient, e.g., for LINK-3, the test suites related to IEEE 802.1D PICS A.7 for “Frame relaying” applies.

8.2.2.6 ETB Network Layer

8.2.2.6.1 Table 5 of IEC 61375-2-5:2014, Clause 6

Item	Question	Req.	Subclause	Implementation
NET-1	Does ETBN support ARP Address Resolution Protocol IETF RFC 826?	M	6 (Table 5)	Y[]
NET-2	Does ETBN support IPv4 Internet Protocol IETF RFC 791?	M	6 (Table 5)	Y[]
NET-3	Is ETBN capable of setting train-wide addresses in range 10.128/9?	M	6 (Table 5), 6.4.2, 6.5.2	Y[]
NET-4	Is ETBN capable of setting ETB and CN subnets prefix length to /18?	M	6 (Table 5)	Y[]
NET-5	Is ETBN capable of setting IPv4 static routes?	O	6 (Table 5)	Y[] N []
NET-6	Is ETBN capable of acting as DNS client?	O	6 (Table 5)	Y[] N []
NET-7	Ability to let priority be based on IP Differentiated Services Field (DSCP: Differentiated Services CodePoint Field) IETF RFC 2474	O	6 (Table 5)	Y[] N []

Further explanations:

NET-1 (ARP), NET-2 (IPv4), NET-5 (IPv4 static routes) and NET-6 (DNS client): Items based on IETF standards, and the general support for these standards are not covered by test suites within this conformance document.

NET-3 and NET-4: Top level requirement for train-wide IP addresses and subnets. These requirements are covered in more detail by UCAST requirements below, thus are not referenced further in this document.

NET-7: No test suite is included for testing priority based on DSCP.

8.2.2.6.2 ETB Network Layer

Item	Question	Req.	Subclause	Implementation
UCAST-1	Which Backbone IDs (0-3) can the ETBN implementation assign IP settings for?	O.1	6.4.2.2	Backbone ID: 0[] 1[] 2[] 3[]
UCAST-2	Is the implementation able to assign a train-wide IP address for its ETB interface based on TTDP inauguration results?	M	6.4.2.2, 6.5.1.2, 6.5.1.3	Y[]
UCAST-3	Is the implementation able to allocate train-wide IP subnets for its local CNs based on TTDP inauguration results?	M	6.4.2	Y[]
UCAST-4	Does the implementation support setting unicast route entries to consist networks with redundant and non-redundant ETBNs respectively?	M	6.4.2.2, 6.7.1, 9	Y[]

Item	Question	Req.	Subclause	Implementation
UCAST-5	Does the ETBN enable/disable forwarding of IP unicast packets with train-wide addresses (dst or src IP 10.128.0.0/9) when entering/leaving INAUGURATED state?	M	6.4.2.1, 8.5.1-8.5.2 Item 30 of [2-5-COR1]	Y[]
UCAST-6	Does the ETBN support IP forwarding from a local CN back to the same CN when the CN's train-wide address range is used as destination IP? (hair-pin)	M	6.4.2, 6.7.1	Y[]
UCAST-7	Does the implementation support acting as redundant ETBN router with respect to IP unicast forwarding?	RED_ETBN:M	6.5.1.2, 6.6, 6.7.1, 9	Y[] N/A[]
UCAST-8	Is the ETBN capable of connecting EDs on the ETB and dynamically assigning addresses to these EDs?	O	6.5.2	Y[] N[]
UCAST-9	Does the ETBN support IP relative addressing plan for Closed Train?	O	6.5.3	Y[] N[]

Further explanations:

UCAST-8: Conformance testing of address assignment to hosts on ETB is not intended within this test document.

UCAST-9: Conformance testing of IP relative addressing is not intended within this test document.

8.2.2.6.3 ETBN Multicast Services

Item	Question	Req.	Subclause	Implementation
MCAST-1	Does the ETBN enable/disable forwarding of IP multicast packets with train-wide addresses (dst IP 239.192.0.0/14 or src IP 10.128.0.0/9) when entering/leaving INAUGURATED state?	M	6.4.2.1, 6.4.4, 8.5.1-8.5.2 Item 30 of [2-5-COR1]	Y[]
MCAST-2	Does the implementation support acting as redundant ETBN router for multicast traffic?	RED_ETBN:M	6.4.4, 6.7.2.2, 9 Item 39 of [2-5-COR1]	Y[] N/A[]
MCAST-3	CN to ETB multicast forwarding: Is ETBN capable of forwarding all multicast packets in range 239.192.0.0/14 range from CN to the ETB? a statically configured subset of multicast packets in range 239.192.0.0/14 range from CN to the ETB?	O.1	6.4.4, 6.7.2.2	Whole range[] Statically configured subset[]
MCAST-4	ETB to CN multicast forwarding: Does the ETBN support forwarding of "All Consist" (239.192.0.0), and for its own consist "All host in Consist X" (239.192.0.X)?	M	6.4.4	Y[]
MCAST-5	ETB to CN multicast forwarding: Does the ETBN support setting additional multicast routes: using static configuration? dynamically within 239.192.0.0/14 range via IGMP?	O.1	6.7.2.2	Static multicast route[] Dynamic multicast route (IGMP)[]

Item	Question	Req.	Subclause	Implementation
MCAST-6	Local CN to CN multicast forwarding: Is ETBN capable of forwarding all multicast packets in range 239.192.0.0/14 range between local CNs? a statically configured subset of multicast packets in range 239.192.0.0/14 range between local CNs?	MULT_CN:O.1	6.4.4	Whole range[] Statically configured subset[]
MCAST-7	Local CN to CN multicast forwarding: Is the ETBN able to forward multicast between local CNs for “All Consist” (239.192.0.0), and for its own consist “All host in Consist X” (239.128.0.X)?	MULT_CN:M	6.4.4	Y[] N/A[]
MCAST-8	Is multicast routing table updated upon train inauguration to take new IP unicast mapping and new “All hosts in Consist X (239.192.0.X)” into account?	M	6.4.4, 6.7.2.1	Y[]
MCAST-9	Is ETBN capable of NAT translation of destination multicast addresses when routing from ETB to CN, at least including consist specific multicast to local multicast translation?	O	5.4.5.3 [2-3] 6.7.2.2 [2-5]	Y[] N[]

8.2.2.7 ETB Transport Layer requirements on ETBN – Application end device common interface for devices connected to ETB subnet

Item	Question	Req.	Subclause	Implementation
TRANS-1	ICMP Internet Control Message, Protocol IETF RFC 792	M	7 (Table 11)	Y[]
TRANS-2	IGMP v2 Internet Group Management Protocol IETF RFC 2236 (as end device support multicast)	M	7 (Table 11)	Y[]
TRANS-3	UDP User Datagram Protocol, IETF RFC 768	M	7 (Table 11)	Y[]
TRANS-4	TCP Transmission Control, Protocol IETF RFC 793	M	7 (Table 11)	Y[]

Further explanations:

The requirements of transport layer services listed in the table above do not have any corresponding test suites in this conformance test. Instead the vendor is assumed to rely on general conformance methods for IETF standards.

8.2.2.8 TTDP HELLO format and content

Item	Question	Req.	Subclause	Implementation
HELLO-FRAME-1	Does <i>encapsulation of transmitted</i> TTDP HELLO TLV messages comply with the standard specification?	M	8.7.2-8.7.5	Y[]
HELLO-FRAME-2	Does <i>format and content of transmitted</i> TTDP HELLO TLVs comply with the standard specification?	M	8.7.5	Y[]
HELLO-FRAME-3	Does implementation accept the <i>encapsulation of incoming</i> TTDP HELLO TLVs compliant with the standard specification?	M	8.7.2-8.7.5	Y[]
HELLO-FRAME-4	Does implementation accept format and content of incoming TTDP TLVs compliant with the standard specification?	M	8.7.5 Item 11 of [2-5-COR1]	Y[]

8.2.2.9 TTDP TOPOLOGY format and content

Item	Question	Req.	Subclause	Implementation
TOPO-FRAME-1	Does the <i>encapsulation</i> of ETB-TLV and CN-TLV of <i>transmitted</i> TTDP TOPOLOGY messages comply with the standard specification?	M	8.7.2-8.7.4, 8.7.6 Item 12 of [2-5-COR1]	Y []
TOPO-FRAME-2	Does the <i>format and content</i> of <i>transmitted</i> ETB-TLV and CN-TLV comply with the standard specification?	M	8.7.6	Y []
TOPO-FRAME-3	Does implementation accept the <i>encapsulation</i> of ETB-TLV and CN-TLV of <i>incoming</i> TTDP TOPOLOGY frames in compliant with the standard specification?	M	8.7.2-8.7.4, 8.7.6	Y []
TOPO-FRAME-4	Does implementation accept <i>format and content</i> of <i>incoming</i> ETB-TLV and CN-TLV in compliant with the standard specification?	M	8.7.6 Item 11 of [2-5-COR1]	Y []

8.2.2.10 Physical topology management – Connectivity Table Management and related fields in TTDP TOPOLOGY frame

Item	Question	Req.	Subclause	Implementation
TOPO-PHY-1	Does the implementation correctly transmit the Connectivity Vector in the TTDP TOPOLOGY frame?	M	8.8.1, 8.7.6	Y []
TOPO-PHY-2	Does the implementation correctly transmit the ETBN Vector(s) and number of ETBNs in the TTDP TOPOLOGY frame?	M	8.6.1, 8.8.2, 8.7.6, 8.9.2	Y []
TOPO-PHY-3	Does the implementation correctly calculate and transmit the ConnTableCrc32 checksum in the TTDP TOPOLOGY frame?	M	8.2.1, 8.5.2, 8.8.3, 8.8.4, 8.7.6	Y []

8.2.2.11 Train Network Directory Management and supported consist topologies**8.2.2.11.1 General Train Network Directory Management and consist topology support**

Item	Question	Req.	Subclause	Implementation
TOPO-GEN-1	Does TOPOLOGY frame include an etbTopoCnt computed as defined in 8.8.5 and 8.8.6?	M	8.8.5, 8.8.6	Y []
TOPO-GEN-2	Does etbTopoCnt in TOPOLOGY frame reflect current Train Network Directory when etbInhibition is FALSE?	M	8.8.6	Y []
TOPO-GEN-3	Does etbTopoCnt reflect last stable value of etbTopoCnt when etbInhibition is TRUE?	M	8.8.6	Y []

8.2.2.11.2 declares the types of local consist topologies this implementation can be part of, which is also the topologies the ETBN is capable of advertising in its TOPOLOGY frames.

8.2.2.11.2 Local consist topologies which can be formed with (units of) this ETBN implementation

Item	Question	Req.	Subclause	Implementation
TOPO-TND-CST-1	Ability to act as ETBN in consist topology "Single ETBN/CN"	M	6.4.2.3.2, 6.6 [2-8] 8.3.2.2	Y []
TOPO-TND-CST-2	Ability to act as ETBN in consist topology "Multiple Single ETBN/CN"	O	6.4.2.3.3, 6.6 [2-8] 8.3.2.3	Y [] N []
TOPO-TND-CST-3	Ability to act as ETBN in consist topology "Single ETBN with multiple CNs"	O	8.8.5 [2-8] 8.3.2.4	Y [] N []
TOPO-TND-CST-4	Ability to act as ETBN in consist topology "Redundant ETBNs with single CN"	O	6.4.2.3.4, 6.6 [2-8] 8.3.2.5	Y [] N []
TOPO-TND-CST-5	Ability to act as ETBN in consist topology "Redundant ETBNs with multiple CNs"	O	8.8.5 [2-8] 8.3.2.6	Y [] N []
TOPO-TND-CST-6	Ability to act as ETBN in "Asymmetric consist topology"	O	8.8.5 [2-8] 8.3.2.7	Y [] N []

8.2.2.11.3 declares the types of topologies in remote consists which the implementation is capable of interacting with. This is the consist topologies received from other ETBNs it is capable of handling.

8.2.2.11.3 Remote consist topologies which the implementation can interact with

Item	Question	Req.	Subclause	Implementation
TOPO-TND-TRAIN-1	Ability to interact with ETBNs in other consists of basic topologies (as listed in PICS 8.3.2)	M	6.4.2.2-6.4.2.3.4, 6.6, 8.8.5 [2-8] 8.3.2	Y []
TOPO-TND-TRAIN-2	Ability to interact with ETBNs in topologies with max number of ETBN/train (63), ETBN/consist (32), CN/train (63) and consists/train (~63)?	M	8.8.5, Annex A	Y []

8.2.2.12 Topology transitions

This subclause concerns ETBN behaviour after inauguration has succeeded and inauguration inhibition has been set.

Item	Question	Req.	Subclause	Implementation
TRANS-INHIBIT-1	Is implementation as ETBN end-node able to handle train lengthening?	M	8.6.2, 8.6.3	Y []
TRANS-INHIBIT-2	Is implementation able to handle loss of ETBN without triggering a new inauguration?	M	8.11.2	Y []
TRANS-INHIBIT-3	Is implementation as ETBN intermediate node able to take over as end-node if current end-node is lost?	M	8.11.2, 8.6.3	Y []
TRANS-INHIBIT-4	Is implementation as late/recovered intermediate ETBN, in consist included in etbTopoCnt, capable of conducting node recovery when expected?	M	8.11.1, 8.5.1, 8.5.3	Y []
TRANS-INHIBIT-5	Is implementation as late/recovered intermediate ETBN waiting in state "NOT_INAUGURATED" when node recovery is not possible?	M	8.11.1, 8.5.1, 8.5.3	Y []

Item	Question	Req.	Subclause	Implementation
TRANS-INHIBIT-6	Is implementation acting as temporary ETBN end-node capable of managing recovery of ETBN(s) outside its extremity port?	O	8.11.3	Y [] N []
TRANS-INHIBIT-7	Is implementation capable of being recovered as end-node (or group of lost end-nodes) by delaying its transition from READY_FOR_INAUGURATION to INAUGURATED?	M	8.5.3, 8.11.3	Y []

Further explanations:

TRANS-INHIBIT-6 refers to the capability for a node acting as temporary end-node to detect and support recovery of previously lost end-nodes.

TRANS-INHIBIT-7 refers to the capability for a late or recovering end-node to include a delay before transitioning to INAUGURATED state. Thereby it enables itself to be included in the internal topology, given that the current temporary end-node supports end-node recovery (TRANS-INHIBIT-6).

8.3 Test cases

8.3.1 Testing framework

ETBN test should connect ETBN with the test system following the application scenarios, each node should follow the same access direction. Test system should be able to monitor the whole communication process of test ports and should be able to simulate peer-to-peer ETB network and assess the conformance of products. During the test process, the products should meet the following requirements:

Single ETBN connects to test system as IUT, the test system only monitors and records ETB ports of IUT. It then assesses that the communication behaviour meets the requirements of specification.

Single ETBN connects to test system as IUT, test system monitors and records ETB ports of IUT, and simulates the ETB network communication on any side or both two sides. It then assesses that the communication behaviour of IUT meets the requirements of specification.

Multiple ETBNs connect to test system as System under Test (SUT), the test system only monitors and records the ETB ports of SUT. It then assesses that the communication behavior of SUT meets the requirements of specification.

Multiple ETBNs connect to test system as SUT, the test system monitors and records the ETB port of SUT, and simulates the ETB communications on any side or both two sides. It then assesses that the communication behavior of SUT meets the requirements of specification.

Figure 12 shows the scenario that single ETBN(IUT) connects the test system, the test system only monitors and records the ETB ports of IUT.

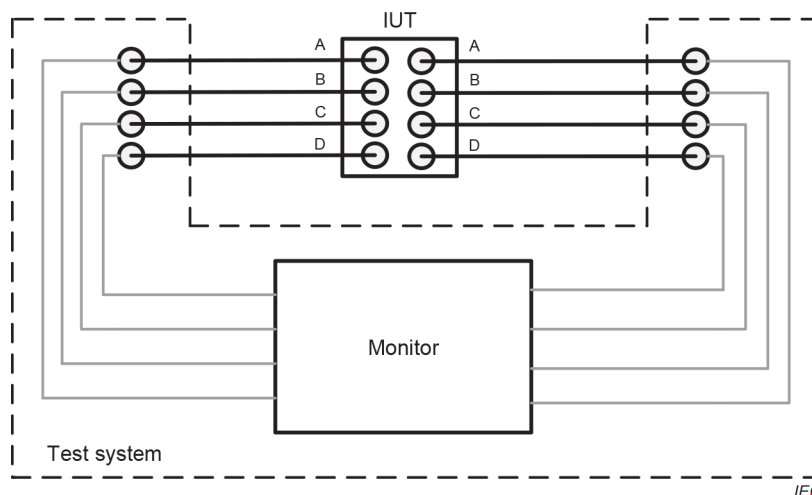


Figure 12 – TTS1 One ETBN connects with test system (monitoring)

Figure 13 shows the scenario that one ETBN (IUT) connects to the test system, the test system monitors and records the ETB ports of IUT and simulates one side of ETB network communication.

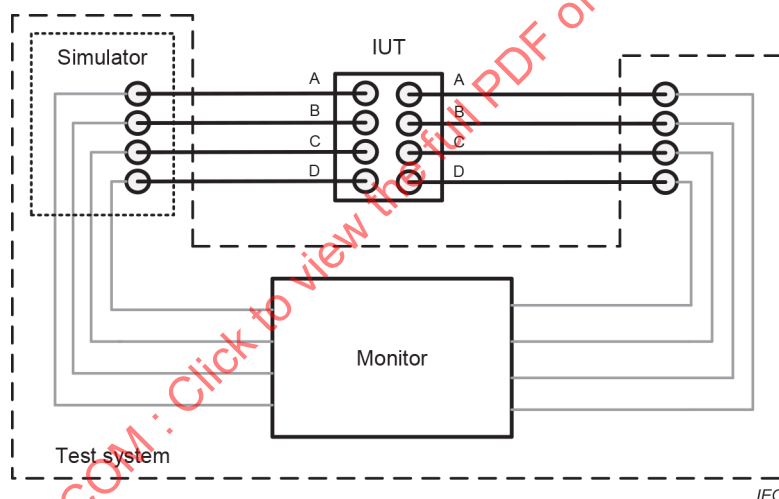


Figure 13 – TTS2 single ETBN connects with test system (one side simulation)

Figure 14 shows the scenario that when single ETBN connects to test system as IUT, test system monitors and records the ETB ports and simulates the communications in both sides of ETB networks.

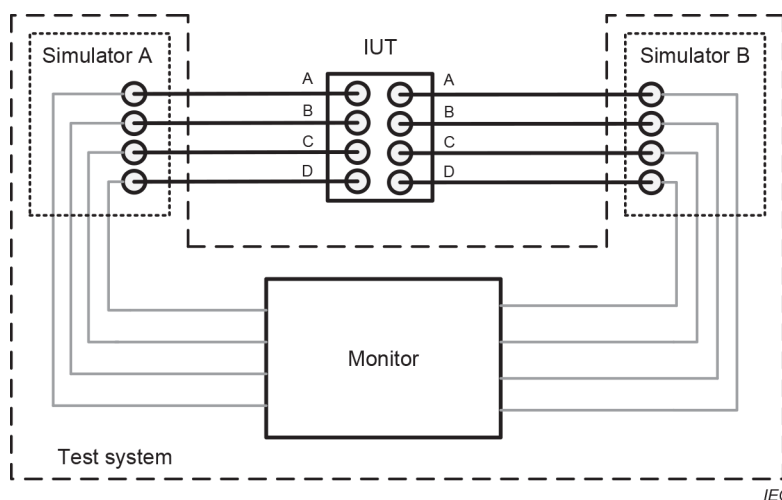


Figure 14 – TTS3 Single ETBN connects with test system (both sides simulation)

Figure 15 shows the scenario that N ($N \geq 2$, depends on test subject) ETBNs (SUT) connect with the test system, the test system only monitors and records the ETB ports of SUT.

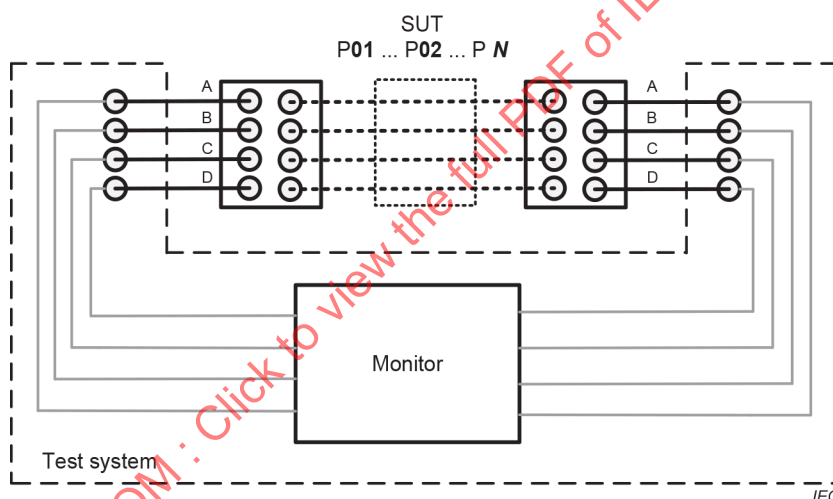


Figure 15 – TTS4 Multiple ETBNs connect with test system (monitoring)

Figure 16 shows the scenario that N ETBNs (SUT) connect with the test system, the test system monitors and records the ETB ports of IUT and simulates one side of ETB network communication.

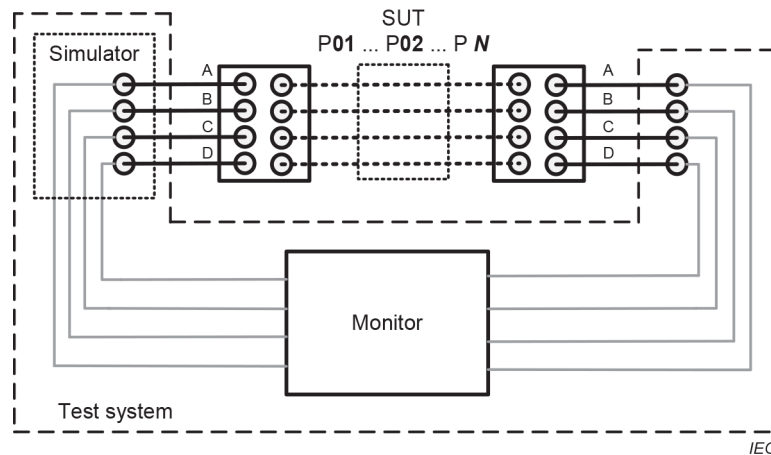


Figure 16 – TTS5 multiple ETBNs connect with test system (one side simulation)

Figure 17 shows the scenario that N ETBNs (SUT) connect with the test system, the test system monitors and records the ETB ports and simulates the communications in both sides of ETB networks.

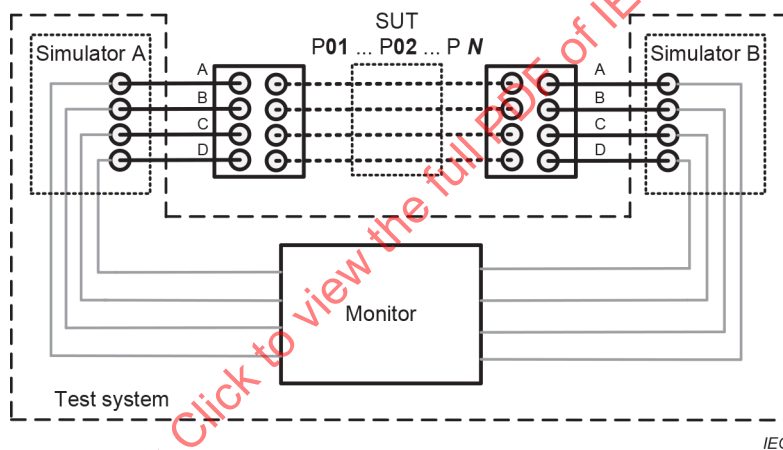


Figure 17 – TTS6 Multiple ETBN connects with test system (both sides simulation)

During the ETBN test, all of ETB ports of IUT/SUT (including both aggregation and redundant lines) should connect with test system.

8.3.2 Common Consist Topologies

8.3.2.1 General

This subclause lists a set of common consist topologies used as reference in later subclauses, see Figure 18 through Figure 23.

8.3.2.2 Single ETBN/CN

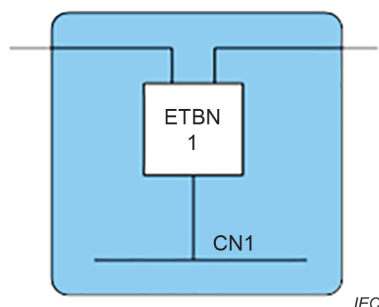


Figure 18 – Consist topology with single ETBN and single CN

This is the simplest consist topology (see also Figure 23 in [2-5]).

8.3.2.3 Multiple Single ETBN/CN

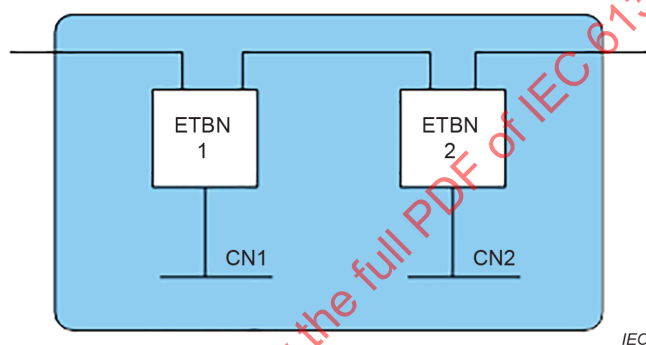


Figure 19 – Consist topology with multiple ETBNs, each connected to a separate CN

This is a generalisation of the “Single ETBN/CN” consist topology, with multiple ETBNs (here two) in the consist, each with a single CN. (See also Figure 27 in [2-5]).

8.3.2.4 Single ETBN with multiple CNs

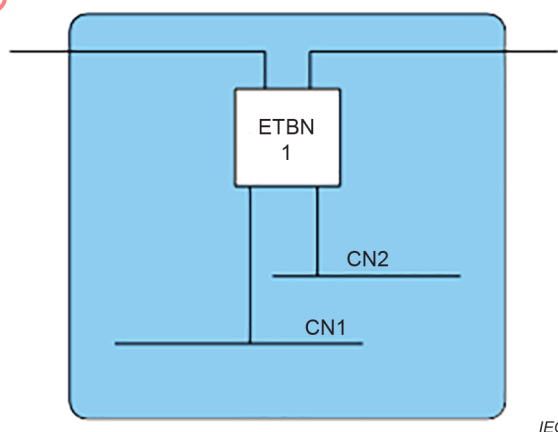


Figure 20 – Consist topology with a single ETBN connected to multiple CNs

This is another generalisation of the “Single ETB/CN” consist topology, where multiple CNs (here two) are attached to the ETBN.

8.3.2.5 Redundant ETBNs with single CN

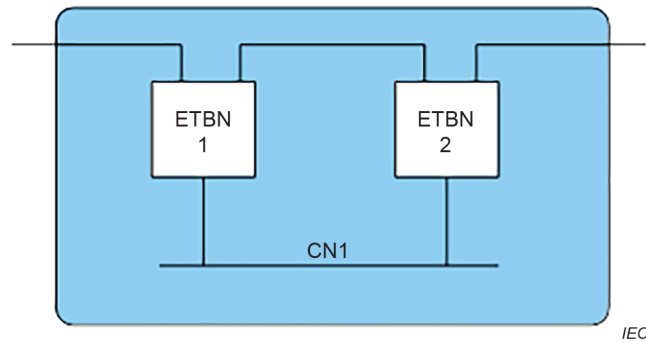


Figure 21 – Consist topology with single CN with redundant ETBNs

With redundant ETBN consist topologies, two (or more) ETNs are attached to a CN. (See also Figure 25 in [2-5].)

8.3.2.6 Redundant ETBNs with multiple CNs

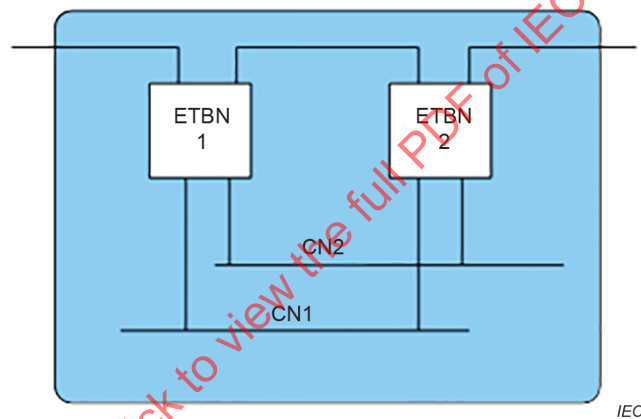


Figure 22 – Consist topology with multiple CNs with redundant ETBNs

This is a generalisation of the “redundant ETBN, single CN topology” where multiple CNs (here two) are attached to each ETBN.

8.3.2.7 Asymmetric Consist Topology

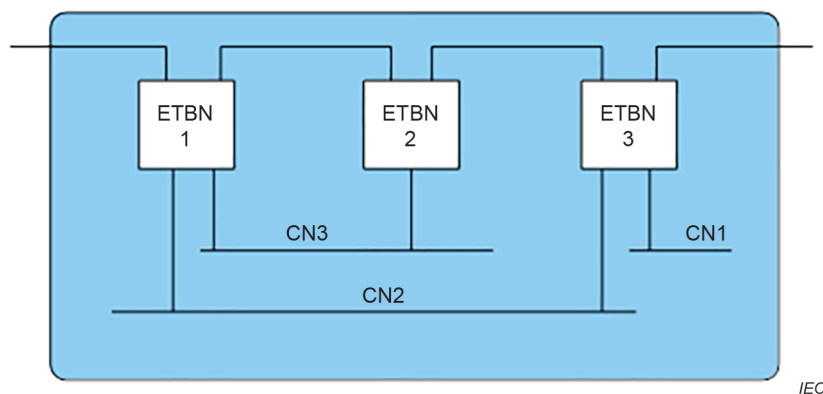


Figure 23 – Consist topology with multiple ETNs and CNs connected asymmetrically

This topology is probably not to consider as common. It is included here as an example of an odd and complex consist topology whose implementations should be able to interoperate with and possibly to support building. (See also Figure 43 in [2-5].)

8.3.3 Test Suite: ETB port settings

8.3.3.1 Purpose

Verify Correct ETB port settings with respect to speed/duplex, auto-sensing and auto-polarity.

PICS: PORT-1, PORT-2

References: 4.2.2 Table 1 of [2-5] and Item 2 of [2-5-COR1], and specifically the following items:

- “Full Duplex Mode” (Required), (PORT-1)
- “Physical Layer auto-negotiation” (Prohibited), (PORT-1)
- “Physical layer in fixed MDI mode (auto-sensing disabled)”, (PORT-2)
- “Connector for active Network Devices M12 D coded” (Required), (PORT-2)

The test suite should verify that IUT ETB ports comply to these items:

- The first two items imply that ETB ports shall be configured to 100 Mbit/s full duplex with auto-negotiation of speed/duplex disabled (PORT-1).
- Item 3 states that auto-MDI/MDIX is disabled (no auto-sensing). For interoperability on the ETB, it is important to verify that correct pin-pairs are used for TX (TX+/TX-) and which are used for RX (RX+/RX-). If the setting is incorrect at one end, the link will not come up. The pin-pairs are specified for M12 D coded connector in Table 1 (bullet 4). This is part of PORT-2.

Verify functionality of bypass relay in bypass mode

- PICS: PORT-3, PORT-4
- References: 4.2.2 and 8.3.2.2 of [2-5], Item 5 of [2-5-COR1]
- Verify that bypass relay is set in bypass-mode when unit has no power
- Verify that bypass relay conducts an internal cross-over in bypass mode.

8.3.3.2 Test Cases: ETB port settings

PICS: PORT-1, PORT-2

Test Setup: Single IUT (TTS3), one port/line in each direction of the IUT.

Test Steps:

- 1) Verify that IUT does not conduct auto-negotiation (part of PORT-1). Note that there are multiple ways to verify this, thus the proposals below should just be considered as examples, not as mandatory methods.
 - Alternative 1: Use low-level functionality at the test system to verify that the IUT does not transmit Ethernet FLP Bursts (28.2.3.1 of IEEE 802.3:2012) on its ETB ports. Test system may use auto-MDI/MDIX on its side.
Acceptance criteria is that test system verifies that IUT sends NLPs but not FLPs on the ETB port(s).
 - Alternative 2: The test system could itself configure auto-negotiation when connecting to the ETB ports on the IUT. Test system may use auto-MDI/MDIX on its side. If the IUT has auto-negotiation disabled (fixed 100 Mbit/s Full Duplex), the ports

on the test system shall come up with correct speed (100 Mbit/s), but shall come up with wrong duplex mode (Half Duplex), see NOTE 2 of 28.2.3.1 of IEEE 802.3:2012.

Acceptance criteria is that port on test system reports link up with speed/duplex 100 Mbit/s half duplex.

- 2) Verify that IUT is set to 100 Mbit/s Full Duplex (part of PORT-1): There is some overlap with previous step, but here the auto-negotiation is not monitored.
 - Let test system set ports to fixed 100 Mbit/s Full Duplex (no auto-negotiation). Test system may use auto-MDI/MDIX on its side.
 - Acceptance criteria for *speed* (100 Mbit/s) is that link to IUT indicates physical link up at the test system port. that come up (Physical UP), which indicates that speed is correct.

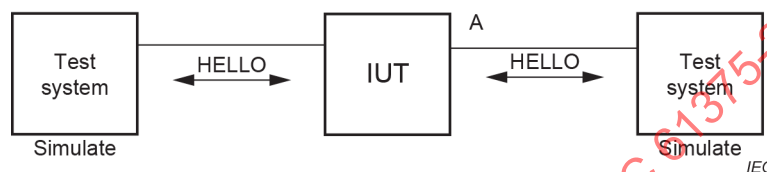


Figure 24 – Setup to verify full duplex mode on ETB ports

This approach uses a setup as in Figure 24, and assumes the use of one ETB line, where the test system sends data through the IUT via a single ETB line (A).

- Ensure the Lines on each side of IUT are logically up. For this to happen, the Test System needs to exchange HELLO messages with the IUT. (The Test System should verify that it actually exchanges HELLOs with the IUT, i.e., the IUT should not have its relay in bypass mode.) Note that there are separate test suites to verify logical link up on ETB lines (see section 8.3.4.3); here this step is just a means to enable switching of data through the IUT.
 - Test system send 60 Mbit/s data (more than 50 Mbit/s) in each direction through the IUT. Use some suitable packet size from RFC 2544 list of sizes, e.g., 512 bytes.
 - Verify that there is no packet loss
- If the IUT uses full duplex there should be no problem, but if half duplex is used. packet loss would occur. But in order to exchange data over the ETB lines, the lines need to be logically up. Thus, the test system shall be able to exchange HELLO.
- 3) Verify that IUT uses correct pinout with auto-sensing disabled (fixed MDI setting) (PORT-2):

The main objective here is to verify that the IUT uses correct pin pair for TX and correct pin pairs for RX and that auto-sensing is disabled. The Test System should set its Ethernet ports to 100 Mbit/s Full Duplex and fixed MDI (i.e., auto-sensing disabled), but keep auto-polarity enabled. The connection to the ETB ports on the IUT should be using cross-over cables.

- Acceptance criteria: Link should come up physically, indicating 100 Mbit/s Full Duplex on the test system side
- Then conduct the same test, now using a straight cable to connect the IUT and the test system. (Alternatively, the test system could change configuration to use fixed MDIX, keeping the cross-over cable)

Acceptance criteria: Link should not come up.

8.3.3.3 Test Cases: Bypass Relay Functionality

PICS: PORT-3, PICS-4

Test setup: Single IUT (TTS1)

The test system should connect to one port on each direction of the IUT, and they should be selected such that the bypass relay on the IUT connects the ports when in bypass mode. The test system should be configured with fixed 100 Mbit/s full duplex, fixed MDI, fixed auto-polarity to achieve the pinout shown in table 1 of [2-5]. The cross-over cables should be used to connect test system to the two ports on the IUT.

The IUT should have no power.

The objective is to verify that bypass relay is set in bypass-mode when unit has no power (PICS-3), and that the bypass relay conducts an internal cross-over in bypass mode (PICS-4).

Test steps:

- 1) Ensure test system is configured and connected to the IUT as described above, and that the IUT has no power.

Acceptance criteria: The test system should get link up, and be able to send data through the IUT (e.g., ping).

8.3.4 Test Suite: ETB Line Status and Link Aggregation management

8.3.4.1 Purpose

The purpose of this test is to verify that:

- link aggregation control manages lines in aggregate correctly
 - (LAG-1) a line in an aggregate is only included to distribute packets when the line is logically up.
 - (LAG-2) a line is considered logically up by an ETBN when it receives TTDP HELLO messages indicating that the peer also “hears” the ETBN.
 - (LAG-2) A special error case to consider is where there is a one-way transmission error, e.g., when the RX pair is intact, but when the TX pair is broken.
- (LAG-3) Details of TTDP HELLO message exchange, correct handling loss of HELLO messages. Cases to check are (a) when the node detects loss of TTDP HELLO, and (b) when the node receives a TTDP HELLO with FastInterval from a neighbour, indicating its neighbour has detected loss of TTDP HELLO.
- (LAG-4 and LAG-5), Nodes supporting ETB redundancy) Link Aggregation Failover performance: Should meet standard requirement (200 ms).
- (LAG-6 and LAG-7), ETBN down/up performance: Traffic being switched over the ETB should not be interrupted more than 200 ms.
- References: 4.4, 8.6.4, 8.9.1, items 13 and 15 of [2-5-COR1].

All tests can make use of a setup where one ETBN (IUT) connects to the test system, and the test system emulates ETBN on one or both sides. See Figure 13 and Figure 14.

8.3.4.2 Test Case: Physical connectivity

The purpose in this test is to check that a line is included or excluded from the aggregate if the line is physically up or down respectively. It assumes that physical up implies logical up, thus the test system needs to send HELLO messages.

Test Setup: TTS2 (Figure 13), either with single line (line A) or with multiple lines (A+B). The alternative with multiple lines shall be used if implementation claims to support line redundancy (LAG_RED).

Requirement: LAG-1 (partial fulfilment, complements 8.3.4.3).

Alternative 1: Single line (line A), for vendors *not* claiming line redundancy (LAG_RED).

Initial condition: Start with intact line (line A).

Test steps:

- 1) Verify TTDP HELLO content transmitted by IUT (“recvAstatus” TRUE, “egressLine” “A”).
- 2) Verify that IUT sends TOPOLOGY frames on the line (line A).
- 3) Break line A.
- 4) After break: Verify that IUT considers line (line A) down by the method provided by vendor PIXIT, e.g., SNMP ifOperStatus (IF-MIB).

Final conditions: Line is down.

Acceptance criteria: Successful verification as specified in test steps

Figure 25; if vendor claims support for 4 lines, add lines C and D and adapt steps accordingly).

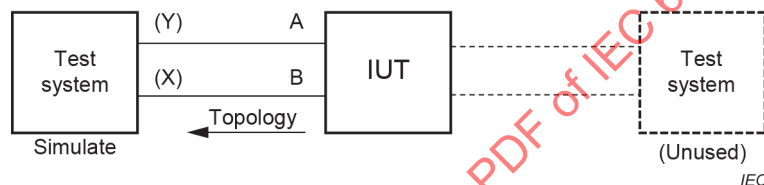


Figure 25 – Example where IUT sends TOPOLOGY on line B (here X is B, Y is A)

Initial condition: Start with intact lines (line A and B).

Test steps:

- 1) Verify that Test System receives TTDP TOPOLOGY on one of the lines (either line A or line B).
- 2) Verify TTDP HELLO content transmitted by IUT: (“recvAstatus” TRUE, “recvBstatus” TRUE, “egressLine” “A”) on line A and (“recvAstatus” TRUE, “recvBstatus” TRUE, “egressLine” “B”) on line B.
- 3) Verify that IUT sends TOPOLOGY frames on one (and only one) line, here referred to as line “X”. We refer to the other line as line “Y”. In the example in Figure 24, X would be line B, and Y would be line A.
- 4) Break the line where TOPOLOGY frames are received (line X).
- 5) Verify TTDP HELLO content transmitted by IUT: (“recvXstatus” FALSE, “recvYstatus” TRUE, “egressLine” “Y”) on line Y.
- 6) Verify that IUT now sends TOPOLOGY frames on second line (line Y).

Final conditions: IUT sends TOPOLOGY on the line which is up (line Y).

Acceptance criteria: OK if:

- Successful verification as specified in test steps.

8.3.4.3 Test Case: Logical connectivity

This test checks link aggregation failover and line status. The difference as opposed the previous test case is that here link failure is achieved logically, that is, there is no physical link down at the IUT.

The test emulates the situation is where a repeater is used between two ETBNs, and there is a link down between the repeater and one of the ETBNs (here the test system). In the test, this can be achieved by letting the test system stop sending HELLO messages (i.e., act as passive monitor).

Test Setup: TTS2 (Figure 13), either with single line (line A) or with multiple lines (A+B). The alternative with multiple lines shall be used if implementation claims to support line redundancy (LAG_RED).

Requirement: LAG-1 (partial fulfilment, complements 8.3.4.2), LAG-2 (partial fulfilment, complement 8.3.4.4).

Alternative 1: Single line (line A, as shown in Figure 26).

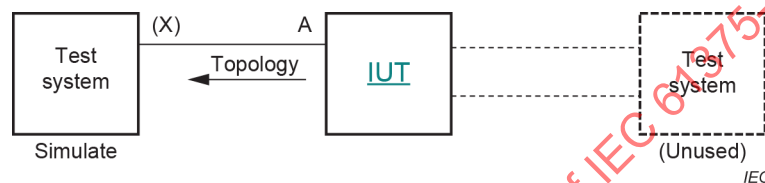


Figure 26 – Simulating Logical line down for single line setup, by letting simulator stop sending HELLO on line X (line A)

Test steps:

- 1) Verify TTDP HELLO content transmitted by IUT ("recvAstatus" TRUE, "egressLine" "A").
- 2) Verify that IUT sends TTDP TOPOLOGY over line A.
- 3) Let test system stop sending HELLO (this emulates a logical link down).
- 4) Verify TTDP HELLO content transmitted by IUT ("recvAstatus" FALSE, "egressLine" "A").
- 5) Verify that IUT sends TTDP TOPOLOGY over line A.

Final conditions: IUT sends no TTDP TOPOLOGY over line A.

Acceptance criteria: Successful verification as listed in test steps.

Alternative 2: Multiple lines (line A and B as shown in Figure 27; if vendor claims support for 4 lines, add lines C and D and adapt steps accordingly):

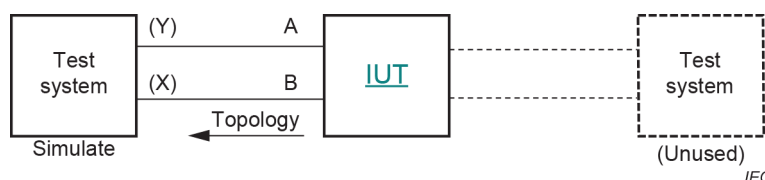


Figure 27 – Simulating Logical line down for multiple line setup, by letting simulator stop sending HELLO on line X (here line B)

Test steps:

- 1) Verify TTDP HELLO content transmitted by IUT: ("recvAstatus" TRUE, "recvBstatus" TRUE, "egressLine" "A") on line A and ("recvAstatus" TRUE, "recvBstatus" TRUE, "egressLine" "B") on line B.

- 2) Verify that IUT sends TOPOLOGY frames on one (and only one) line. Here we refer to that line as “X” and the other as “Y”. In the example in Figure 27, X would be line B, and Y would be line A.
- 3) Let the Test System stop sending TTDP HELLO on line X (emulating logical link down on line X).
- 4) Verify TTDP HELLO content transmitted by IUT: (“recvXstatus” FALSE, “recvYstatus” TRUE, “egressLine” “X”) on line X and (“recvXstatus” FALSE, “recvYstatus” TRUE, “egressLine” “Y”) on line Y.
- 5) Check that IUT now sends TOPOLOGY frames on second line (line Y).

Final conditions: IUT sends TOPOLOGY frames on the line logically up (line Y).

Acceptance criteria: Successful verification as listed in test steps.

8.3.4.4 Test Case: One-way transmission error

This test verifies that links with one-way transmission errors should not be included in the aggregate. An IUT receiving TTDP HELLO messages on a line (say “X”) with “recvXstatus” set to FALSE excludes the line from its aggregate but sends “recvXstatus” TRUE in its own TTDP HELLO messages.

Test Setup: TTS2 (Figure 13), either with single line (line A) or with multiple lines (A+B). The alternative with multiple lines shall be used if implementation claims to support line redundancy (LAG_RED).

Requirement: LAG-2 (partial fulfilment, complements 8.3.4.3).

Alternative 1: Single line (line A)

Initial condition: Start with intact line (line A). Test system should send TTDP HELLO on line A (with “recvAstatus” TRUE).

Test steps:

- 1) Verify TTDP HELLO content transmitted by IUT (“recvAstatus” TRUE, “egressLine” “A”).
- 2) Verify that TTDP TOPOLOGY are sent over line A.
- 3) Let Test System send HELLO with “recvAstatus” FALSE (to emulate broken TX line from IUT to Test System).
- 4) Verify TTDP HELLO content transmitted by IUT (“recvAstatus” TRUE, “egress line” “A”).
- 5) Verify that no TTDP TOPOLOGY is sent over line A.
- 6) Final conditions: No TTDP TOPOLOGY is sent over line A.

Acceptance criteria: Successful verification as listed in test steps.

Figure 28; if vendor claims support for 4 lines, add lines C and D and adapt steps accordingly):

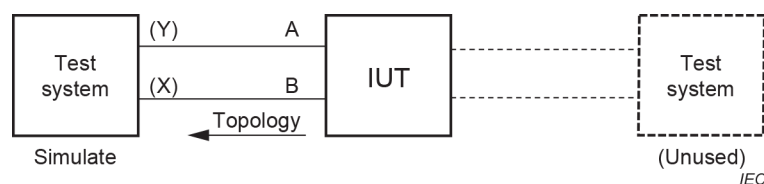


Figure 28 – Simulating one-way transmission error, by letting simulator send HELLO with recvXStatus FALSE (here recvBStatus FALSE)

Initial condition: Start with intact lines (line A and B). Test system should send TTDP HELLO on both line A and B (with “recvAstatus” TRUE and “recvBstatus” TRUE).

Test Steps:

- 1) Verify TTDP HELLO content transmitted by IUT: (“recvAstatus” TRUE, “recvBstatus” TRUE, “egressLine” “A”) on line A and (“recvAstatus” TRUE, “recvBstatus” TRUE, “egressLine” “B”) on line B.
- 2) Verify that IUT sends TOPOLOGY frames on one (and only one) line. Here we refer to that line as “X” and the other as “Y”.
- 3) Let Test System send HELLO with “recvXstatus” FALSE (to emulate broken TX line from IUT to Test System on line X) and “recvYstatus” TRUE on both line X and line Y.
- 4) Verify TTDP HELLO content transmitted by IUT: (“recvAstatus” TRUE, “recvBstatus” TRUE, “egressLine” “A”) on line A and (“recvAstatus” TRUE, “recvBstatus” TRUE, “egressLine” “B”) on line B. That is, IUT should still send TRUE, as it receives HELLOs from Test System.
- 5) Verify that IUT now sends TOPOLOGY frames on second line (line Y).

Final conditions: TOPOLOGY frames are now sent on second line (line Y).

Acceptance criteria: Successful verification as listed in test steps.

8.3.4.5 Test Case: Single Lost TTDP HELLO

This test checks correct IUT behaviour during the FastTimeout scenarios:

- If an IUT receives a TTDP HELLO with value FastTimeout, it should immediately respond with a TTDP HELLO
- If an IUT experience SlowTimeout, it should send a HELLO with a FastPeriod. Following that, the IUT
 - should switch back to SlowPeriod without failover if it receives a TTDP HELLO before FastTimeout occurs. This is part of this test case.
 - should conduct a failover if it does not receive a TTDP HELLO before FastTimeout occurs (resulting in recovery less than 200 ms). This is tested in another test case (0).

Figure 13). The test below assumes a single line (line A) is used, as that is sufficient for this test as shown in Figure 29.

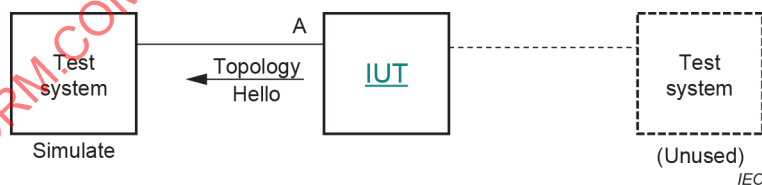


Figure 29 – Testing correct handling of loss of HELLO transmitted by IUT, and loss of HELLO sent to IUT

Requirement: LAG-3.

Initial condition: Start with intact line (line A). Test system should send TTDP HELLO on line A with value slowTimeout.

Test steps:

- 1) Verify IUT transmits TTDP HELLO with content: “recvAstatus” TRUE, “egressLine” “A”, slowTimeout.
- 2) Verify IUT transmits TTDP TOPOLOGY frames are sent over line A.

- 3) Verify correct handling of lost Hello transmitted by IUT (receiving Hello with fastTimeout):
 - i) Let Test System send TTDP HELLO with value FastTimeout at time T0, just after receiving a TTDP HELLO from IUT.
 - ii) Verify that IUT transmits TTDP HELLO content (“recvAstatus” TRUE, “egressLine” “A”, slowTimeout) within T0+15ms (upper limit set to ensure IUT takes advantage of the FastPeriod mechanism).
 - iii) Let Test System send regular TTDP HELLO (slowPeriod and value “slowTimeout”).
 - iv) Verify that IUT resumes sending TTDP HELLO at regular interval (slowPeriod) instead of triggered by reception of TTDP HELLO.
- 4) Verify correct handling when IUT experience slowTimeout due to single lost HELLO (Fast HELLO Initiation and Stopping):
 - i) Let Test system send a last TTDP HELLO at time T1.
 - ii) Verify that IUT transmits TTDP HELLO with value “fastTimeout”.
 - iii) Let Test System respond immediately with TTDP HELLO (content: “recvAstatus” TRUE, “egressLine” “A”, “slowTimeout”).
 - iv) Verify that IUT resumes with regular operation (line A stays in “Line OK”). IUT should not stop sending TTDP HELLO FastPeriod with “fastTimeout” (IUT should send at most one Hello with ‘fastTimeout’ after receiving the HELLO from the Test System).

Final conditions: IUT sends TTDP HELLO with SlowInterval and value “slowTimeout”.

Acceptance criteria: Successful verification as specified in test steps.

8.3.4.6 Test Case: ETB Line Failover performance test

Only applicable for IUTs supporting redundant ETB lines.

This test aims at verifying that recovery time of data on the ETB is 200 ms or less when an ETB line either goes down or comes up. To measure the recovery time, gaps in an offered data stream could be monitored (somewhat similar to reset test, see 26.4 of RFC 2544).

Only the case where there is logical link down/up is considered for performance measurements.

Figure 14). Two lines (A+B) are assumed in both directions (dir1 and dir2), as shown in Figure 30.

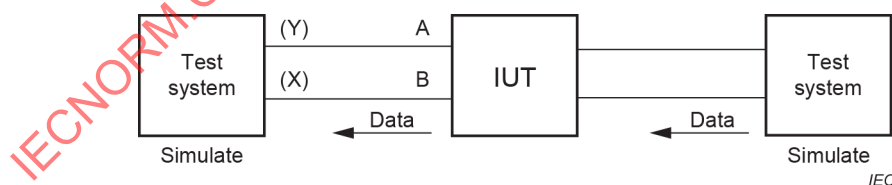


Figure 30 – ETB line failover setup, the line where IUT initially forwards the data stream is referred to as ‘X’ (here line B)

Requirement: LAG-4 and LAG-5.

Initial condition: Start with intact lines (line A and B) in both directions. Test System sends HELLO messages on all lines on both sides (dir 1 and dir 2) of the IUT. The test system should send a data stream on the dir2 side of the IUT, and monitor that it receives the data stream from the IUT over one of the lines on dir1 (here referred to as line “X”, while the other line is referred as line “Y”). (Link aggregates should map all traffic of a specific “data stream” to a specific port to ensure traffic of a stream is never received out of order. (Hint: sending UDP Data with same src/dst MAC, same src/dst IP and same src/dst UDP port is likely to be treated as one “data stream”).)

Test steps:

Test System sends traffic (dir2) and monitors reception (line X on dir1).

Logical link down:

- 1) Test system sends its last TTDP HELLO to IUT on dir1 line X at T0. The purpose is to emulate logical link down for line X.
- 2) Check that IUT starts to send TTDP HELLO with FastInterval no later than T0+SlowTimeout on line X.
- 3) After T0+SlowTimeout+FastTimeout the IUT should initiate the failover (removing line X from distribution on its aggregate in dir1). By using monitor at Test System, verify the gap in the data stream shifted from line X to line Y is 200 ms or less.

Logical link Up:

- 1) Test system resumes to send TTDP HELLO over dir 1 line A at T1.
- 2) When the IUT receives a TTDP HELLO on its line A (with "RecvAstatus" TRUE), it will include line A in its aggregate. The stream earlier moved from line X to line Y will either be mapped back to line X or stay on line Y.
- 3) Test System should verify that the gap in the data stream shifted from line Y to line X is 200 ms or less.

Final conditions: Monitor side of Test System receives all data streams it sends on the other side.

Acceptance criteria: Successful verification as specified in test steps.

8.3.4.7 Test Case: ETBN Down/Up performance test

This applies both to ETBNs with single line and redundant lines.

Requirement: LAG-6 and LAG-7.

Test setup: TTS6, In total there will be 5 ETBNs (A, B, C, D, E), where the SUT consisting of 3 IUTs (B, C, D), test system emulating ETBN on each side (A, E). Only a single ETB line is needed. (Even if the IUT supports multiple ETB lines, only one of them is connected.)

- a) For simplicity, each ETBN (A-E) should constitute a consist of its own, i.e., single ETBN/CN (see 8.3.2.2).
- b) To measure the recovery time, the test system should inject a traffic stream from A to E and one in the reverse direction (E to A). This should be short Ethernet data packets, sent at a suitable interval to achieve desired resolution.

Test steps:

- 1) Power up the IUTs (B, C, D) in the SUT and let the test system emulate nodes A and E. The nodes should be able to reach state INAUGURATED. Test Systems let both A and E set etbInhibition TRUE to avoid reininauguration.
- 2) Start the traffic streams from A to E and E to A and verify that it is received at a regular interval at the other side.
- 3) Now power off the middle IUT (C) and measure the gap in the data stream in each direction.
 - Acceptance criteria: Recovery time should be less or equal to 200 ms in both directions.
- 4) Continue to send data streams between A and E.
- 5) Now power up unit C) and measure the gap in the data stream in each direction. Note that the traffic needs to be monitored long enough for the IUT to boot up and shift the relay from bypass to connected.

Acceptance criteria: Recovery time should be less or equal to 200 ms in both directions.

8.3.5 Test Suite: TTDP HELLO format and content

8.3.5.1 Purpose

PICS: HELLO-FRAME-1, HELLO-FRAME-2, HELLO-FRAME-3, HELLO-FRAME-4.

(HELLO-FRAME-1): Verify that TTDP HELLO message encapsulation of transmitted TTDP HELLO messages comply with standard specification.

- Ethernet header, VLAN header and LLDP header have strict definitions.
- LLDP content shall include mandatory TLVs and may include optional TLVs before and after TTDP HELLO TLV.

(HELLO-FRAME-2): Verify that TTDP HELLO TLV structure and content of transmitted TTDP HELLO messages comply with standard specification.

- Static fields of TTDP HELLO TLV:
 - HELLO TLV header: Type, Length, OUI and Subtype all have static content for this specific TTDP version
 - HELLO message static fields: version, reserved1 and reserved2 fields all have constant values for this specific TTDP version
- Configurable fields of TTDP HELLO TLV: "cstUuid" and "vendor".
- Dynamic fields of TTDP HELLO TLV: tlvCS, lifeSign, etbTopoCnt, recvxStatus (X is A-D depending on nb of lines), timeoutSpeed, srcID, SrcPortId, egressLine, egressDir, inaugInhibition, remoteld.

(HELLO-FRAME-3): Verify that implementation accepts encapsulation of incoming TTDP HELLO compliant with [2-5] and [802.1AB].

- Mandatory TLVs: Chassis ID TLV and Port ID TLVs can have different sub-types, lengths and values. The TTDP implementation should accept correct Chassis and Port ID TLVs, but not use their values. EOF and TTL TLVs have no subtypes.
- Optional TLVs: Vendors may use additional (optional) TLVs before/after the TTDP HELLO TLV. The TTDP implementation should accept and ignore such optional TLVs.

(HELLO-FRAME-4): TTDP implementation should determine if incoming HELLO TLVs are correct or incorrect, and either accept or drop them. Fields to check:

- TLV header and TTDP version:
 - TLV type (127), OUI ('200E95'H), and TLV subtype (1) are used to identify the TTDP HELLO message. That is, without correct values here the message would not be treated by the TTDP implementation
 - TLV length: Should be 86 for the given (major) TTDP version ('01).
 - TTDP HELLO version: ('MMmmmmmm'H) where major (MM) shall be 01, while minor (mmmmmm) should be ignored.
- Reserved fields: content of should be ignored, that is, HELLO TLV should be accepted irrespective of the content in the Reserved fields.
- tlvCS: HELLO TLV should be dropped if checksum is incorrect

lifeSign should be incremented for each HELLO sent as long as the neighbour ETBN is the same (and does not restart), except when the counter wraps around. Still, no test is proposed as it obvious if a unit should drop the frame or accept it and log the event.

8.3.5.2 Test Case: TTDP HELLO frame encapsulation and static fields

PICS: HELLO-FRAME-1, and partially HELLO-FRAME-2.

Test Setup: TTS-1 (Figure 12).

Initial condition: IUT configured as ETBN. IUT ETB ports connected to monitoring test unit. Test Bench powered on and IUT booted up.

Test steps:

- 1) Monitor and record TTDP HELLO packets sent out by the IUT on each ETB line.
 - Verify that TTDP packets are sent out periodically on each ETB line.
 - Verify that packets have DMAC '0180C2-00000E'H and are VLAN tagged (type '8100'H) with VID 492 and priority 7.
 - Verify that the tagged packet includes an LLDP PDU (type '88CC'H). This LLDP PDU shall include the mandatory LLDP TLVs in correct order (Chassis TLV, Port TLV and TTL TLV). Refer to IEEE 802.1AX for acceptable content of these TLVs. (IEC 61375 does not mandate any specific subtype for the mandatory TLVs.) As LLDP was not designed for sub-second intervals, the acceptance of TTL TLV should be relaxed, although value 1 (second) is assumed to commonly used.
 - Verify that after the mandatory TLVs, the LLDP PDU may include 0 or more optional TLVs, followed by the mandatory HELLO TLV, and then 0 or more optional TLVs.
 - Verify that the LLDP PDU is ended by an EOF TLV.
 - Verify that TTDP HELLO TLV use correct byte and bit ordering, and the following fields have the stated values:
 - TLV type: 127
 - TLV length: 86
 - OUI: '200E95'H
 - TLV subtype: 1
 - TTDP HELLO version: '01000000'H
 - TTDP HELLO reserved fields set to "0"

Final conditions: Test Bench powered off.

Acceptance criteria: OK if:

- TTDP HELLO packets on all ETB-lines have structure and static content verified according to test steps.

8.3.5.3 Test Case: TTDP HELLO dynamic and configurable fields

PICS: HELLO-FRAME-2.

Test Setup: TTS-1 (Figure 12).

Initial condition:

- IUT configured as ETBN. IUT ETB ports connected to monitoring test unit.
- The ETBN is a single ETBN with a single CN. Suggested CstUUID '11111111-1111-1111-1111-111111111111'H.
- Test Bench powered on and IUT booted up.

Test Steps:

- 1) Monitor and record TTDP HELLO packets sent out by the IUT on each ETB line.
 - Verify that lifeSign is incremented for each sequential packet seen on an ETB line.
 - Verify that the vendor field is a zero-terminated string, padded with null characters.

- Verify that etbTopoCnt is constant and correctly calculated for the given topology. (If the suggested CstUUID is used, etbTopoCnt shall be '80EA82AC'H.)
- Receive status (recvAstatus, recvBstatus, etc.) shall be FALSE ('01'B) for the number of ETB lines supported, and UNDEFINED ('11'B) for remaining lines.
- Verify that srcID is constant and has same value in HELLO packets on all ETB lines.
- Verify that srcPortId is constant, and that its value is unique per ETB line.
- Verify that egressLine has correct value for each port (ASCII 65 for line 'A', etc.).
- Verify that egressDir has value dir1 (1) for port(s) in direction 1 and value dir2 (2) for port(s) in direction 2.
- Verify that inaugInhibition has value FALSE ('01'B).
- Verify that remote ID has value '000000000000'H.
- Verify that the tlvCS is correct (differs per packet as lifeSign is incremented) as per RFC 793.

Final conditions: Test Bench powered off.

Acceptance criteria: OK if:

- TTDP HELLO packets on all ETB-lines have content verified according to test steps.

8.3.5.4 Test Case: Accept various LLDP TLVs and subtypes

PICS: HELLO-FRAME-3.

Test Setup: TTS3 (Figure 14). There will be 3 ETBNs (A, B, C), where the IUT is the middle ETBN (B) and the test system emulates the ETBNs at the ends (A, C). The test system only needs to send TTDP HELLO. (A and C need not send TOPOLOGY messages.)

Verify IUT accepts HELLO with valid Mandatory TLVs.

Test steps:

- 1) First check that HELLO messages from IUT reports "recvAstatus" as "LineNotOK" in both directions.
- 2) Let A send correct HELLO messages with Chassis TLV of subtype "MAC address" and port TLV of subtype "Agent Circuit ID" as shown in the examples on page 63 of IEC 61375-2-5. Other mandatory TLVs (TTL and EOF) are also sent, but have no subtypes and are not tested further here (TTL value 1 is suggested).

Acceptance criteria: Inspect the HELLO message from IUT in order to check that it accepts the HELLO by verifying that IUT reports "recvAstatus" as "LineOK" towards A (and "LineNotOK" towards C).

- 3) Let C send correct HELLO messages with some other Chassis TLV and port TLV subtypes (allowed in the LLDP standard), e.g., Chassis TLV subtype 7 (locally assigned) and Port ID subtype 1 (ifAlias). The objective is to verify that the IUT looks at the Length value of the TLVs and is able to determine where the HELLO TLV starts. As before, other mandatory TLVs (TTL and EOF) are also sent but have no subtypes and are not tested further here.

Acceptance criteria: Inspect the HELLO message from IUT in order to check that it accepts the HELLO from C (and A) by verifying that IUT reports "recvAstatus" as "LineOK" in both directions.

Verify that IUT accepts Optional TLVs: Idea is that simulator inserts one optional TLVs before/after the TTDP-HELLO TLV. IUT should accept the TTDP-HELLO.

Test steps:

- 1) First check that HELLO messages from IUT reports "recvAstatus" as "LineNotOK" in both directions.
- 2) Let A send correct HELLO messages with
 - all the needed mandatory TLVs (Chassis TLV, port TLV, TTL TLV),
 - one optional TLV, e.g., an IEEE 802.1 Organizationally Specific TLVs (Annex E of [802.1AB])
 - the TTDP HELLO TLV (after the optional TLV)
 - EOF TLV

Acceptance criteria: Inspect the HELLO message from IUT in order to check that it accepts the HELLO by verifying that IUT reports "recvAstatus" as "LineOK" towards A (and "LineNotOK" towards C).

- 3) Let C send correct HELLO messages with
 - all the needed mandatory TLVs (Chassis TLV, port TLV, TTL TLV),
 - the TTDP HELLO TLV (before the optional TLV)
 - one optional TLV, e.g., an IEEE 802.1 Organizationally Specific TLVs (Annex E of IEEE 802.1AB:2009)
 - EOF TLV

Acceptance criteria: Inspect the HELLO message from IUT in order to check that it accepts the HELLO from C (and A) by verifying that IUT reports "recvAstatus" as "LineOK" in both directions.

8.3.5.5 Test Case: Basic test of processing of incoming TTDP HELLO TLVs

PICS: HELLO-4.

Test Setup: TTS2 (Figure 13).

The objective is to verify that the IUT discards invalid TTDP HELLO messages and that it ignores data of reserved fields and minor version number. This can be done by letting the test system inject specific TTDP HELLO in one direction (TTS2). Here we assume 2 ETBNs (A, B) where IUT is in the right (B) and test system emulates A. It is assumed a single line is used (line A).

Dropping Invalid frames:

Test steps:

- 1) Power up IUT. Let A send correct TTDP HELLOs as described in 8.7.5 of [2-5].
- 2) Verify that IUT reports "recvAstatus" as "LineOK" in its HELLO in towards A (and "LineNotOK" in the other).
- 3) Now let node A send invalid HELLO messages for some period of time (at least 1 second). Fields to be manipulated/tested.

- TLV header and major version

- TLV type (127), OUI ('200E95'H), and TLV subtype (1) are used to identify the TTDP HELLO message. That is, without correct values here the message should not be treated by the TTDP implementation. Change them to some other value for some time and then back again.

Acceptance criteria: IUT should report LineNotOK when these values are incorrect. Once correct values are sent, it should report LineOK again.

- TLV length: Should be 86 for the given (major) TTDP version ('01).

Acceptance criteria: IUT should report LineNotOK when setting Length to something else than 86. Once correct values are sent, it should report LineOK again.

- TTDP HELLO version: ('MMmmmmmm'H) where major (MM) shall be 01, while minor (mmmmmm) should be ignored. Test with MM equal to some other value(s), e.g, 02 and 03.

Acceptance criteria: IUT should report LineNotOK when setting major value to something else than 01. Once correct values are sent, it should report LineOK again.

- tlvCS: Let A send HELLOs with invalid tlvCS checksum for some period of time and then start so send correct again.
- Acceptance criteria: IUT should report LineNotOK when sending with incorrect tlvCS. Once correct values are sent, it should report LineOK again.

Ignoring reserved fields and minor version number:

Test steps:

- 1) Power up IUT. Let A send correct TTDP HELLOs as described in 8.7.5 of IEC 61375.
- 2) Verify that IUT reports "recvAstatus" as "LineOK" in its HELLO in towards A (and "LineNotOK" in the other).
- 3) Now let node A test send HELLO messages for some period of time (at least 1 second) with non-zero values in the reserved fields. Fields to manipulated/tested.

- Minor version. Let A send a non-zero value in its "minor version field" e.g., mmmmmm could be 000001.

Acceptance criteria: IUT should continue to report "LineOK" towards A. (For this to work, the IUT may need to be rebooted; it is not assumed that the minor version will be changed on a running train.)

- Reserved1 and reserved2 fields: Let A send a non-zero value in its "reserved1" and "reserved2" fields e.g., value 1 (decimal) in each field.

Acceptance criteria: IUT should continue to report "LineOK" towards A.

8.3.6 Test Suite: Unicast address assignment and unicast routing

8.3.6.1 Purpose

Verify that configuration of Backbone ID (0-3) affect train wide addresses assigned to backbone and CNs as described in 6.4.2. (UCAST-1).

- Verify subnet IP assignment for backbone (UCAST-1).
- Verify ETBN IP address on backbone (UCAST-2). This relates to logical topology and ETBN ID assignment., which is in turn tested more thoroughly in suites defined in 8.3.10.

Verify subnet IP assignment for CN networks:

- Verify that ETBN assigns correct train-wide subnets for its local CNs based on inauguration results, i.e., it will forward packets for this subnet between ETB and CN (conducting appropriate NAT operations). (UCAST-3).
- Verify ETBN installs correct next hop IP address to reach remote CNs using their trainwide address. This applies both when destination CN has redundant ETBNs (Virtual IP as next hop) or non-redundant ETBNs ("physical" IP as next hop). (UCAST-4).

Verify that IP unicast forwarding is only enabled between ETB and ECN when ETBN is in INAUGURATED state. (UCAST-5).

Verify that an ETBN supports route packets from local CN back to local CN when the CN's train-wide address range is used as destination IP (hairpin)? (UCAST-6).

For ETBNs capable of acting as redundant ETBNs (UCAST-7)

- Verify that an ETBN becoming “master” sends out gratuitous ARP for the virtual IP address, mapping it to its “physical” MAC (not “Virtual MAC”, as this is prohibited on ETB).
- Verify that only the “master” ETBN can forward unicast packets between ETB and ECN.

8.3.6.2 Test Cases: Basic IP assignment and route management

Initial Setup: TTS-3 (Figure 14), Three ETBNs in total (A, B, C). IUT represents ETBN “B”, in a consist with single ETBN/ECN (8.3.2.2). Test system represents ETBNs “A” and “C”, each representing a consist with single ETBN/ECN. Only a single line is required on the ETB. Let ETBN A have lower cstUUID than ETBN C.

Furthermore, the test system needs to connect to the ECN of the IUT and emulate a host on the ECN. It is assumed that IUT has IP 10.0.0.1/18 on its ECN, and that the host has IP address 10.0.0.4/18 with gateway set to 10.0.0.1. In one of the tests, a second host (10.0.0.5/18, gateway 10.0.0.1) is also emulated on the ECN.

- Verify backbone ID impact on addresses, and verify backbone IP assignment (UCAST-1 and UCAST-2).

This test is conducted for each backbone ID (0-3) claimed support for. The example below assumes backbone ID “0” (TCMS).

Test steps:

- 1) Ensure IUT is configured to operate on the desired backbone (here “0”).
- 2) Boot up the IUT and let the test system emulate nodes A and C. Check that all nodes (A, B, C) reach inaugurated state with same etbTopoCnt, by inspecting TTDP Topology from IUT. Then test system sets etbInhibition TRUE on A and C.
- 3) Verify ETB address: The IUT shall determine itself to be ETBN #2, thus it shall assign itself IP address 10.128.0.2 on its ETB interface. Let test system test this by sending an ARP request for 10.128.0.2 from 10.128.0.1 (ETBN “A”).

Acceptance criteria: IUT should respond to the ARP Request with its MAC address.

- Verify local IP subnet assignment. (UCAST-3).

This test is only required to be tested for one ETB ID, even if the vendor claims support for any ETB ID. The test below assumes ETB ID “0” (TCMS).

Test steps:

- 1) Ensure IUT is configured to operate on the desired backbone (here “0”).
- 2) Boot up the IUT and let the test system emulate nodes A and C. Check that all nodes (A, B, C) reach inaugurated state with same etbTopoCnt, by inspecting TTDP Topology from IUT. Then test system sets etbInhibition TRUE on A and C.
- 3) Verify local subnet assignment: The IUT shall determine its CN to have #2, thus, it shall assigned it trainwide subnet 10.128.128.0/18. Let test system send “ping” (ICMP Echo Request) from its emulated host (10.0.0.4) to ETBN “A” (10.128.0.1), and let “A” respond (ICMP Echo Response).

Acceptance criteria: Host should get “ping response” from “A”. In addition, IUT should map source IP of ICMP Request to the correct train wide subnet (10.0.0.4 mapped to 10.128.128.4).

- Verify routing table, i.e., correct next hop for remote CN in case of non-redundant ETBN or redundant ETBN (UCAST-4).

Test steps:

- 1) Do same steps 1 and 2 as in test above. Furthermore, ensure that test system does not transmit any gratuitous ARPs or ARP responses for their trainwide addresses (10.128.0.1 and 10.128.0.3) so to ensure the IUT do not have their MAC in their ARP cache.

- 2) Let the emulated host (10.0.0.4) send a ping to 10.128.64.1, which should be forwarded to ETBN A, and another ping to 10.128.192.1, which should be forwarded to ETBN C.

Acceptance criteria: The ping to 10.128.64.1 should be forwarded by the IUT to node A. The IUT should issue an ARP request for 10.128.0.1 to get the correct MAC for the ping packet. Similarly, the ping to 10.128.192.1 should be forwarded by the IUT to node C, and should be preceded by the IUT making an ARP resolution for 10.128.0.3.

- 3) Now let the test system emulate two ETBNs “to the right” of the IUT, i.e., the train now has 4 ETBNs (A, B, C, D), where C and D are redundant ETBNs of a single consist (see 8.3.2.5). This should lead to a new inauguration on the ETB. Avoid sending any gratuitous ARP from C or D for the virtual address of their CN (here virtual address 10.128.0.131).
- 4) Verify that IUT reaches inaugurated state with same etbTopoCnt in its TOPOLOGY messages as reported by the emulated ETBNs (A, B, C, D).
- 5) Let the emulated host (10.0.0.4) send a ping to 10.128.64.3. This should result in the IUT sending an ARP Request for the virtual address (10.128.0.131). Let “C” respond with its MAC address. IUT would then forward the ping to C.

Acceptance criteria: IUT sends ARP request for virtual address 10.128.0.131, and then forwards ping using the MAC address of C.

- 6) Then let “D” send gratuitous ARP for the virtual IP (10.128.0.131), and let the host (10.0.0.4) send another ping to 10.128.128.1. The IUT should forward it using the MAC address of D.

Acceptance criteria: IUT forwards ping using the MAC address of D.

- Verify that IP unicast forwarding is only enabled between ETB and ECN when in ETBN is in INAUGURATED state. (UCAST-5).

This test is only required to be tested for one ETB ID, even if the vendor claims support for any ETB ID. The test below assumes ETB ID “0” (TCMS).

Test steps:

- 1) Ensure IUT is configured to operate on the desired backbone (here “0”).
- 2) Boot up the IUT and let the test system emulate nodes A and C. Check that all nodes (A, B, C) reach inaugurated state with same etbTopoCnt, by inspecting TTDP Topology from IUT. This time, the test system lets etbInhibition be FALSE on A and C. Thus A, B and C will be in inaugurated state, but inauguration is not inhibited.
- 3) Let the emulated host (10.0.0.4) send a ping to 10.128.64.1 and a ping to 10.128.192.1. Verify that “B” does not forward the ping packet to ETBN A and ETBN C respectively.
- 4) Let A and B set etbInhibition TRUE.
- 5) Let the emulated host (10.0.0.4) send a ping to 10.128.64.1 and a ping to 10.128.192.1. Verify that “B” does forward the ping packet to ETBN A and ETBN C respectively.
- 6) Let A and B set etbInhibition FALSE.
- 7) Let the emulated host (10.0.0.4) send a ping to 10.128.64.1 and a ping to 10.128.192.1. Verify that “B” does not forward the ping packet to ETBN A and ETBN C respectively.

Acceptance criteria: Successful verification as described in the test steps.

- Verify that an ETBN supports route packets from local CN back to local CN when the CN's train-wide address range is used as destination IP (hairpin)? (UCAST-6).

This test is only required to be tested for one ETB ID, even if the vendor claims support for any ETB ID. The test below assumes ETB ID “0” (TCMS).

Test steps:

- 1) Ensure IUT is configured to operate on the desired backbone (here “0”).
- 2) Boot up the IUT and let the test system emulate nodes A and C. Check that all nodes (A, B, C) reach inaugurated state with same etbTopoCnt, by inspecting TTDP Topology from IUT. The test system lets etbInhibition be true on A and C. The test system is connected to the ECN of IUT (node B), emulating two hosts 10.0.0.4/18 and 10.0.0.5/18. The corresponding trainwide addresses are 10.128.0.4 and 10.128.0.5 respectively.
- 3) Let emulated host 10.0.0.4 send a ping to 10.128.128.5. Verify that the other host receives the ping and that proper NAT has been conducted (dst IP 10.0.0.5, src IP 10.128.0.4).
- 4) The other host responds to the ping (dst IP 10.128.0.4, src IP 10.0.0.5). Verify that the first host receives the ping response and that proper NAT has been conducted (dst IP 10.0.0.4, src IP 10.128.0.5).

Acceptance criteria: Successful verification as described in the test steps.

8.3.6.3 Test Suite: Ability to act as redundant ETBN

Test suite for vendors claiming support to to act as redundant ETBN (UCAST-7).

Test setup: TTS6 (Figure 17). The train will consist of 4 ETBNs (A, B1, B2, C), where there are:

- Two IUTs (B1 and B2) in the system under test, representing a consist with redundant ETBNs and single CN (see 8.3.2.5). The mechanism to elect virtual IP master is out of scope, but B1 is assumed to have higher “master election” priority.
- The test system simulates an ETBN on each side (A and C), each representing a consist with single ETBN/CN (see 8.3.2.2). Consist of A has lower cstUUID than C.
- The test system will simulate a host on the CN of ETBN B1 and B2. Here it is assumed that the host has address 10.0.0.4/18, B1 has 10.0.0.2/18, B2 has 10.0.0.3/18 and that they share virtual IP 10.0.0.1/18 on the local CN.

Test steps:

- 1) Boot up the IUTs and let the test system emulate nodes A and C. Check that all nodes (A, B1, B2, C) reach inaugurated state with same etbTopoCnt, by inspecting TTDP Topology from IUT. The test system lets etbInhibition be true on A and C.
- 2) Let A send out ARP request for virtual IP (10.64.0.130) corresponding to train-wide subnet of the CN shared by B1 and B2 (10.64.128.0/18). Verify that B1 responds with its physical MAC.
- 3) Now let A send a ping to 10.64.128.4. Verify that the ping reaches host 10.0.0.4 via B1, and that the response from the host comes back to ETBN A.
- 4) Now power off B1.
- 5) B2 should take over. It should send out gratuitous ARP with its physical MAC (it should not be the same as B1 sent out earlier). B2 should also take over virtual IP on its local CN (10.0.0.1).
- 6) Now let A send another ping to 10.64.128.64. Verify that the ping reaches host 10.0.0.4 (via B2), and that the response from the host comes back to ETBN A.

Acceptance criteria: Successful verification as described in the test steps.

8.3.7 Test Suite: IP Multicast forwarding**8.3.7.1 Purpose**

The purpose of this suite is to verify proper ETBN handling of multicast traffic with respect to multicast forwarding in relation to TTDP.

- General multicast forwarding support

- Verify that ETBN only routes multicast traffic when it is in inaugurated state. (MCAST-1)
- Verify that Multicast routing works from ECN to ETB (MCAST-3).
- Verify that Multicast routing from ETB to ECN (MCAST-4, MCAST-5)
- Multicast forwarding in case of multiple CNs/consist. (MCAST-6, MCAST-7)
- Multicast forwarding for redundant ETBN topologies (MCAST-2)
- Handling reinauguration (MCAST-8)
- Ability to conduct NAT of destination multicast address (MCAST-9)

8.3.7.2 Test Suite: General multicast forwarding test cases

Setup used for Test Cases in this section: TTS-3 (Figure 14), Three ETBNs in total (A, B, C), each representing a consist, see Figure 31.

- Consist A: Consist with single ETBN/CN (8.3.2.2), where test system simulates an ETBN (ETBN A) and an end device (H1).
- Consist B: Consist with single ETBN/CN (8.3.2.2), where IUT represents the ETBN (ETBN B) and test system emulates two end devices (H2 and H3).
- Consist C: Consist with single ETBN/CN (8.3.2.2), where test system simulates an ETBN (ETBN C) and an end device (H4).
- Only a single line is required on the ETB. Let ETBN A have lower cstUUID than ETBN C, such that A becomes consist 1, B becomes consist 2 and C becomes consist 3. Example local subnets and IP assignments are given. The ETBNs shall be able to translate (NAT) local source IP addresses to trainwide source IP addresses (NAT) when forwarding from ECN to ETB.

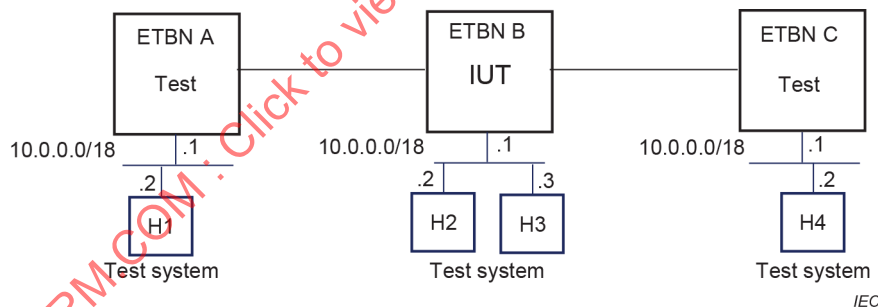


Figure 31 – Topology for general multicast routing tests

Case 1: Multicast forwarding only inaugurated state:

PICS: MCAST-1

Verify that ETBN only routes IP multicast traffic when it is in inaugurated state. The restriction to route IP multicast traffic concerns *multicast traffic with train-wide addresses*. The test only aims to verify this for “routable train IP (destination) multicast address”, using 239.192.0.0 (all consists) and 239.192.0.X (consist X) as representative examples.

Test steps:

- 1) Initial state: ETBN A, B and C in inaugurated state. No inhibition set for any ETBN.
- 2) Let Test System H1 send multicast streams to 239.192.0.0 and to 239.192.0.2. ETBN A routes these streams onto ETB with trainwide source address.
 - Verify packets are received by hosts in consist B (H2/H3), i.e., that IUT routes them to its ECN, keeping the trainwide source address.
- 3) Let Test System H2 send to multicast stream to 239.192.0.0 and to 239.192.0.1.

- Verify that IUT routes these multicast streams onto the ETB, with trainwide source address.
- 4) Force the IUT to leave inaugurated state by creating a mismatch in TOPOLOGY messages, e.g., by stop simulating consist C (IUT shall adapt to new topology), while ETBN A keeps sending its TOPOLOGY messages according to old topology (A, B and C).
 - Verify that multicast from H1 no longer reach H2/H3.
 - Verify that multicast from H2 no longer reach ETB.
- 5) Let IUT reach inaugurated state by removing the mismatch, e.g., by starting to simulate Consist C again.
 - Verify that multicast from H1 again reaches H2/H3.
 - Verify that multicast from H2 again reaches ETB.

Acceptance criteria: Successful verification in the listed steps.

Case 2: Multicast routing from ECN to ETB:

PICS: MCAST-3

Verify that packets with multicast address in global range (239.192.0.0/14) can be routed from CN to ETB. In case 1 above this was already verified for addresses in the reserved range (239.192.0.<0-255>). Therefore, this test is limited to verify capability to route other addresses in the non-reserved parts of the global range (239.192.0.0/14), with 239.192.1.2 as example. Consult vendor PIXIT for method to configure IUT to route that multicast address from ECN to ETB (either “whole range” or “configured subset” as stated by vendor for MCAST-3).

Test steps:

- 1) Initial state: ETBN A, B and C in inaugurated state. Inhibition optionally set (ETBN A).
- 2) Let Test System H2 send multicast stream to 239.192.1.2.

Verify that IUT routes this multicast stream onto the ETB, with trainwide source address.

Acceptance criteria: Successful verification in the steps above.

Verify that the IUT does *not* route multicast packets back to the originating CN (that would cause duplicates to local receivers). This should be tested for the case when there are routes for a multicast address going both from ECN to ETB and from ETB to ECN. One such example is 239.192.0.0 which the ETBN should route from ETB to ECN and from ECN to ETB.

Test steps:

- 1) Initial state: ETBN A, B and C in inaugurated state. Inhibition optionally set (ETBN A).
- 2) Let Test System H2 send multicast stream to 239.192.0.0.

Verify that IUT routes this multicast stream onto the ETB, with trainwide source address, but that IUT does not route the packet towards the originating ECN. That is, H3 should (only) receive a copy of each multicast packet directly from H2, not another copy routed by the IUT.

Acceptance criteria: Successful verification in the steps above.

Case 3: Multicast routing from ETB to ECN:

PICS: MCAST-4, MCAST-5

Verify that relevant addresses in reserved range are routed from ETB to ECN: 239.192.0.0 (all consists), 239.192.0.X (all nodes in consist X), where X is the consist of the ETBN. (MCAST-4). In case 1 above this was already verified; IUT representing consist 2. Therefore, this test is limited to verify that IUT does not route streams for other consists onto its own ECN.

Test steps:

- 1) Initial state: ETBN A, B and C in inaugurated state. Inhibition optionally set (ETBN A).
- 2) Let Test System H1 send multicast streams to 239.192.0.2 (IUT consist) and 239.192.0.3 (Consist C)
 - Verify that H2/H3 receives stream 239.192.0.2 but does not receive stream 239.192.0.3.

Acceptance criteria: Successful verification in the steps above.

Verify that routing works for multicast addresses to which EDs in ECN subscribes (static multicast route(s) or dynamically learnt via IGMP as claimed by vendor for MCAST-5). This has been verified in Case 1 for addresses in the reserved range (239.192.0.0 and 239.192.0.<X>), this test is limited to verify capability to route other addresses in the non-reserved parts of the global range (239.192.0.0/14), with 239.192.1.2 as example. Consult vendor PIXIT for method to configure IUT to route that multicast address from ETB to ECN.

Test steps:

- 1) Initial state: ETBN A, B and C in inaugurated state. Inhibition optionally set (ETBN A).
- 2) Let Test System H1 send multicast stream to 239.192.1.2 and let ETBN A route it onto ETB with train wide source IP address.

Verify that IUT routes this multicast stream onto its ECN (received by H2/H3), keeping the trainwide source address.

Acceptance criteria: Successful verification in the steps above.

8.3.7.3 Test Suite: Multicast forwarding in case of multiple CNs/consist

This is largely covered in the previous section. These special cases only apply for vendors claiming to support such consist topologies (ETBN with multiple CNs, see 8.3.2.4; or consists with CNs at different ETBNs, see 8.3.2.3).

Case 1: ETBN with multiple CNs:

PICS: MCAST-6, MCAST-7

Setup: Similar setup as in section 8.3.7.2. The differences are:

- that IUT consist (B) topology is “single ETBN, multiple CNs” (section 8.3.2.4), here with two CNs.

Figure 32.

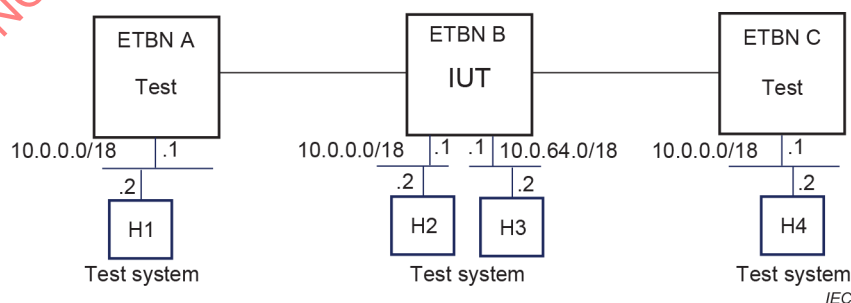


Figure 32 – Topology for testing multicast with ETBN with multiple CNs

Verify ability to forward train-wide multicast from one local ECN to another local ECN if there are receivers for those groups on the second ECN. Addresses 239.192.0.0 and 239.192.0.<X> are used in this test.

Test steps:

- 1) Initial state: ETBN A, B and C in inaugurated state. Inhibition optionally set (ETBN A).
- 2) Let Test System H2 send multicast streams to 239.192.0.0 and 239.192.0.2 and 239.192.0.1.

Verify that IUT routes streams 239.192.0.0 and 239.192.0.1 (and optionally 239.192.0.2) onto ETB.

Verify that IUT routes streams 239.192.0.0 and 239.192.0.2 (but not 239.192.0.3) onto the other local ECN, reaching H3.

Acceptance criteria: Successful verification in the steps above

Verify ability to forward train-wide multicast from ETB to both ECNs if there are receivers for those groups in both ECNs. Addresses 239.192.0.0 and 239.192.0.<X> are used in this test.

Test steps:

- 1) Initial state: ETBN A, B and C in inaugurated state. Inhibition optionally set (ETBN A).
- 2) Let Test System H1 send multicast streams to 239.192.0.0 and 239.192.0.2 and 239.192.0.2.

Verify that IUT routes streams 239.192.0.0 and 239.192.0.2 (but not 239.192.0.3) onto both local ECNs, reaching H2 and H3.

Acceptance criteria: Successful verification in the steps above.

Case 2: Consist with CNs at different ETBNs

PICS: MCAST-7

Figure 17), with three consists (A, B, C) and four ETBNs (A, B1, B2 and C) as shown in Figure 33.

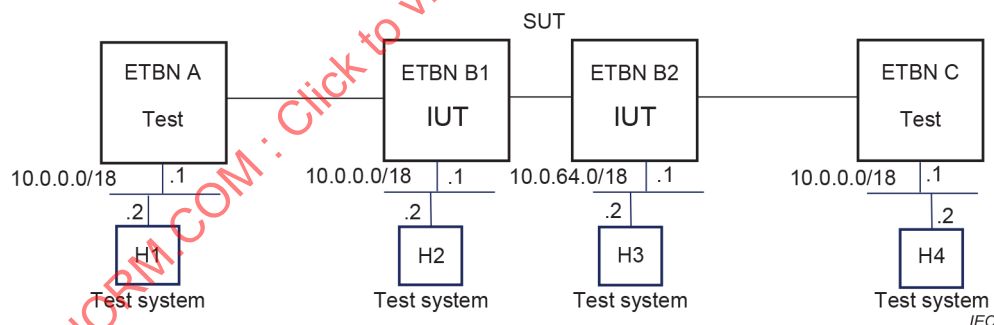


Figure 33 – Multicast test setup for consist with ECNs connected to different ETBNs

Verify ability to forward to multicast for its own consist (e.g., 239.192.0.X) to another ECN in its consist via the ETB (topology as in 8.3.2.3).

Test steps:

- 1) Initial state: ETBN A, B1, B2 and C in inaugurated state. Inhibition optionally set (ETBN A).
- 2) Let Test System H2 send multicast streams to 239.192.0.0 and 239.192.0.2 and 239.192.0.1.

Verify that IUT B1 routes streams 239.192.0.0, 239.192.0.1 and 239.192.0.2 onto ETB.

Verify that IUT B2 routes streams 239.192.0.0 and 239.192.0.2 (but not 239.192.0.3) onto its local ECN, reaching H3.

Acceptance criteria: Successful verification in the steps above.

8.3.7.4 Test Suite: Multicast forwarding for redundant ETBN topologies

These requirements only apply for vendors claiming support for redundant ETBN topologies. The election mechanism to decide which ETBN should forward multicast packets is not specified in [2-5]. The vendor needs to provide such a master/backup mechanism (specified in vendor PIXIT) to avoid duplicates and possible proliferation of multicast packets.

PICS: MCAST-2, for vendors claiming support for redundant ETBN topologies (TOPO-TND-CST-4 and/or TOPO-TND-CST-5).

Case 1: Basic function during stable network

PICS: MCAST-2, TOPO-TND-CST-4, TOPO-TND-CST-5.

Aim: Verify that the master ETBN (and only the master) forwards multicast packets from ETB to ECN and from ECN to ETBN. Also verify that a multicast forwarded by the master from ECN to ETB is not routed back from ETBN to the originated ECN by the backup ETBN.

Setup: TTS6 (Figure 17), with three consists (A, B, C) and four ETBNs (A, B1, B2 and C) as shown in Figure 34.

- Consist A and C are simulated by test system and have topology single ETBN/CN (8.3.2.2). Consist A has lower CstUUID than consist C. The test System emulates an end device in each of these CNs (H2 and H3).

8.3.2.5), here

with two ETBNs (B1, B2). Test system simulates an end device (H2).

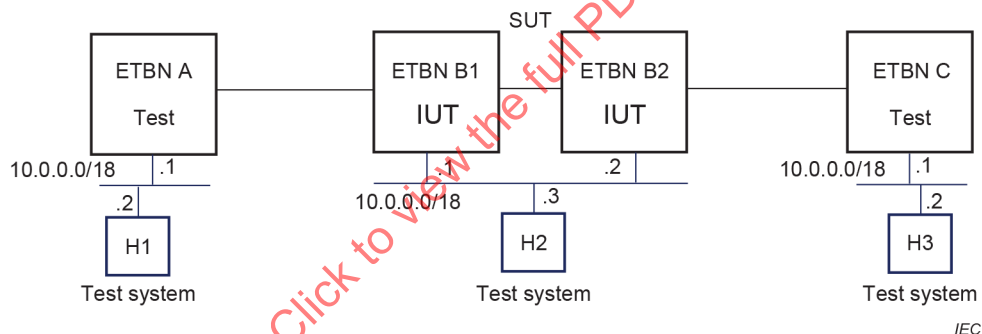


Figure 34 – Testing multicast with redundant ETBNs and single CN

Test steps:

- 1) Initial setup: ETBNs A, B1, B2, C have inaugurated, and inhibition is set (at least by ETBN A; aim is to avoid reinauguration during test). Only one of B1 and B2 shall forward multicast between ETB and its CN; consult vendor PIXIT for details. Here it is assumed that B1 takes that (master) role in both directions, while B2 is standby (backup), ready to take over if B1 goes down.
- 2) Let H1 send a multicast stream to 239.192.0.0 (all consists), which is forwarded onto ETB by ETBN A.
 - Verify that a single copy of the stream reaches H2, forwarded by B1.
- 3) Let H2 send a multicast stream to 239.192.0.0 (all consists).
 - Verify that a single copy of the stream reaches ETB, forwarded by B1.
 - Verify that B2 does not forward back to the originating CN (H2).

Acceptance criteria: Successful verification in steps above.

Case 2: Multicast during failover

PICS: MCAST-2, TOPO-TND-CST-4, TOPO-TND-CST-5.

Aim: Verify that failover works for multicast routing. This is a functional test based on vendor PIXIT instructions (rather than a performance test to measure recovery time):

- If the Master ETBN goes down (or loses connectivity with the CN), the backup ETBN shall take over and forward multicast packets.
- If the previous master comes up again (or regains connectivity to the CN), a failover may (or may not) occur depending on the vendor's master election mechanism. Verify multicast forwarding is recovered, and that only of the ETBNs forwards multicast packets.

Setup: Same as in case 1, see Figure 34.

Test steps:

- Continue where case 1 stopped:
 - H1 sends stream to 239.192.0.0 reaching H2 via ETBNs A and B1
 - H2 sends stream to 239.192.0.0 reaching ETB via B1
- Power down B1. B2 shall take over
 - Verify stream from H1 to 239.192.0.0 recovers, such that it reaches H2 via B2
 - Verify stream from H2 to 239.192.0.0 recovers, such that it reaches ETB via B2
- Power up B1. B1 may or may not take over (consult vendor PIXIT)
 - Verify stream from H1 to 239.192.0.0 recovers, such that it reaches H2 via B2 or via B1
 - Verify stream from H2 to 239.192.0.0 recovers, such that it reaches ETB via B2 or via B1

Acceptance criteria: Successful verification in steps above.

Case 3: Multicast routing for redundant ETBNs with multiple CNs

PICS: MCAST-2, TOPO-TND-CST-5.

Aim: If vendor claims support for multiple CN with ETBN redundancy (section 8.2.4.6), verify that the master ETBN (and only the master) forwards multicast.

A rudimentary failover test is also specified.

Setup: TTS6 (Figure 17), with three consists (A, B, C) and four ETBNs (A, B1, B2 and C) as shown in Figure 35.

- Consist A and C are simulated by test system and have topology single ETBN/CN (8.3.2.2). Consist A has lower CstUUID than consist C. The test system emulates an end device in each of these CNs (H2 and H4).
- that IUT consist (B) topology is "redundant ETBNs with multiple CNs" (8.3.2.6), here with two ETBNs (B1, B2) and two CNs. Test system simulates an end device in each CN (H2 and H3).

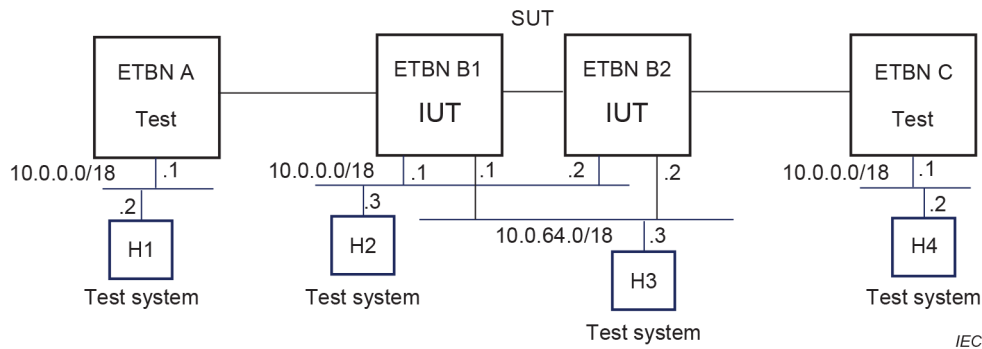


Figure 35 – Multicast routing in consist with redundant ETBN and multiple CNs

Test steps:

- Initial setup: ETBNs A, B1, B2, C have inaugurated and inhibition is set (at least by ETBN A; aim is to avoid reinauguration during test). Only one of B1 and B2 shall forward multicast between ETB and its CNs; consult vendor PIXIT for details. Here it is assumed that B1 takes that (master) role in both directions for both CNs, while B2 is standby (backup), ready to take over if B1 goes down.
- Let H1, H2 and H3 send a multicast stream each to 239.192.0.0 (all consists). The stream from H1 is forwarded onto ETB via ETNA.
 - Verify that H2 and H3 each receive the multicast stream from H1 without duplicates (here assumed streams are routed onto CNs by B1).
 - Verify that stream from H2 reaches ETB and H3 without duplicates (here assumed streams are routed by B1).
 - Verify that stream from H3 reaches ETB and H2 without duplicates (here assumed streams are routed by B1).
- Power down ETBN B1. ETBN B2 shall take over the role to route the multicast streams.
 - Verify that H2 and H3 each resume to receive the multicast stream from H1 via B2.
 - Verify that stream from H2 reaches ETB and H3 via B2.
 - Verify that stream from H3 reaches ETB and H2 via B2.

Acceptance criteria: Successful verification in steps above.

8.3.7.5 Test Suite: Handling reinauguration

After a reinauguration, the ETBN needs to ensure that any stale multicast forwarding state is not lingering.

PICS: MCAST-8.

Case 1: Verify that forwarding of to all hosts in consist X (239.192.0.X) from ETB to ECN(s) reflects the newly inaugurated consist number (X). Here MCAST-8 complements MCAST-4.

Setup: TTS4 (Figure 17), with four consists (A, B, C, D) as shown in Figure 36.

Consist A, B and D: Consists with single ETBN/CN (8.3.2.2), where test system simulates an ETBN and an end device.

Consist C: Consist with single ETBN/CN (8.3.2.2), where IUT represents the ETBN (ETBN C) and test system simulates an end device (H3).

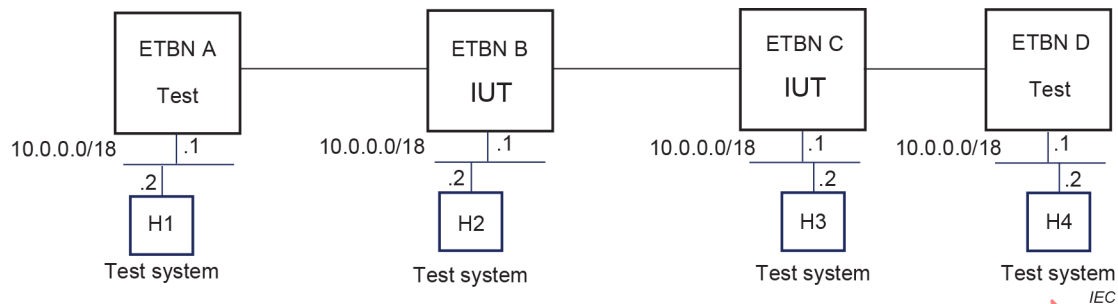


Figure 36 – Testing multicast handling after reinauguration

Test steps:

- 1) Initial state: ETBN A, B, C and D in inaugurated state. No inhibition set for any ETBN. IUT consist becomes "consist 3".
- 2) Let Test System H1 send multicast streams to 239.192.0.2 and 239.192.0.3. ETBN A routes these streams onto ETB with trainwide source address.
- 3) Verify that packets of stream 239.192.0.3 (and not 239.192.0.2) are routed by the IUT onto its ECN (to H3). Packets shall keep the trainwide source address.
- 4) Stop emulating consist B, thereby trigger a reinauguration where IUT consist becomes "consist 2".
- 5) Verify that packets of stream 239.192.0.2 (and not 239.192.0.3) are now routed by the IUT onto its ECN (to H3). Packets shall keep the trainwide source address.

Acceptance criteria: Successful verification in the steps above.

Case 2: ETBN performs 1-1 NAT of source IP (CN local to train-wide address) when routing multicast packets from CN to ETB. Verify that after a reinauguration, any multicast data stream routed from CN to ETB should have train-wide source IP reflecting the newly inaugurated train-wide subnet ID.

Setup: Same setup as in Case 1, see above.

Test steps:

- 1) Initial state: ETBN A, B, C and D in inaugurated state. No inhibition set for any ETBN. IUT consist becomes "consist 3" and its trainwide CN subnet 10.128.192.0/18 (assuming ETB0).
- 2) Let Test System H3 send multicast streams to 239.192.0.0.
- 3) Verify that packets of H3 stream 239.192.0.0 is routed by the IUT onto the ETB with the proper train-wide source address (10.128.192.2).
- 4) Stop emulating consist B, thereby trigger a reinauguration where IUT consist becomes "consist 2". Trainwide subnet for IUT CN now becomes 10.128.128.0/18.
- 5) Verify that packets of H3 stream 239.192.0.0 is routed by the IUT onto the ETB, now with train-wide source address (10.128.128.2).

Acceptance criteria: Successful verification in the steps above.

Case 3: For vendors claiming support for multiple CNs/consist, forwarding to 239.192.0.X from local CN to local CN should also reflect the newly inaugurated consist number (X). Here MCAST-8 complements MCAST-6 and MCAST-7.

Setup: TTS6 (Figure 17), four consists (A, B, C and D) as shown in Figure 37.

- Consists A, B and D: Consist with single ETBN/CN (8.3.2.2), where test system simulates an ETBN and an end device.
- Consist C (IUT Consist): Consist with topology “single ETBN, multiple CNs” (8.3.2.4), here with two CNs. IUT is ETBN and test system simulates an end device in each of these CNs (H3 and H4).

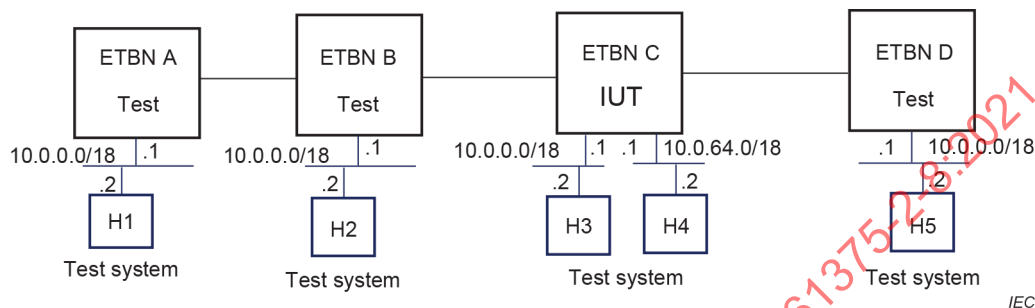


Figure 37 – Test of multicast between local CNs on same ETBN when inaugurating

Test steps:

- 1) Initial state: ETBN A, B, C and D in inaugurated state. No inhibition set for any ETBN. IUT consist becomes “consist 3”.
- 2) Let Test System H3 send multicast streams to 239.192.0.2 and 239.192.0.3.
 - Verify that packets of stream 239.192.0.3 are routed by the IUT onto its other ECN (to H4) and stream 239.192.0.2 (and optionally 239.192.0.3) is routed onto ETB.
- 3) Stop emulating consist B, thereby trigger a reinauguration where IUT consist becomes “consist 2”.
 - Verify that packets of stream 239.192.0.2 are routed by the IUT onto its other ECN (to H4) and stream 239.192.0.3 (and optionally 239.192.0.2) is routed onto ETB.

Acceptance criteria: Successful verification in the steps above.

Setup: TTS6 (Figure 17), four consists (A, B, C1, C2 and D) as shown in Figure 38.

- Consists A, B and D: Consist with single ETBN/CN (8.3.2.2), where test system simulates an ETBN and an end device.
- Consist C (IUT Consist): Consist with topology CNs on different ETBNs (8.3.2.3). A SUT of two IUTs act as ETBN (C1 and C2) and test system simulates an end device in each of these CNs (H3 and H4).
- CstUUID of consist A is lower than CstUUID of consist D, such that SUT consist becomes consist 3 when all consists are up.

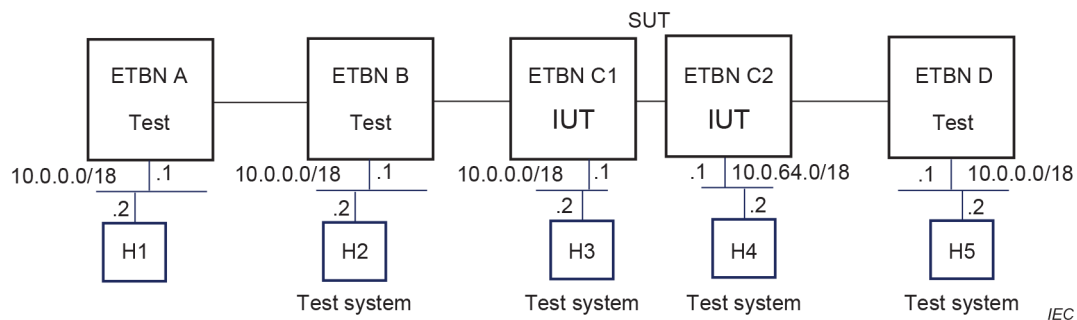


Figure 38 – Test of multicast between local CNs on different ETBNs when inaugurating

Test steps:

- 1) Initial state: ETBN A, B, C and D in inaugurated state. No inhibition set for any ETBN. IUT consist becomes “consist 3”.
- 2) Let Test System H3 send multicast streams to 239.192.0.2 and 239.192.0.3.
 - Verify that packets of stream 239.192.0.2 and 239.192.0.3 are routed by the IUT C1 onto the ETB and that IUT C2 routes stream 239.192.0.3 (but not 239.192.0.2) onto its ECN (to H4).
- 3) Stop emulating consist B, thereby trigger a reinauguration where IUT consist becomes “consist 2”.
 - Verify that packets of stream 239.192.0.3 and 239.192.0.2 are routed by the IUT C1 onto the ETB and that IUT C2 routes stream 239.192.0.2 (but not 239.192.0.3) onto its ECN (to H4).

Acceptance criteria: Successful verification in the steps above.

8.3.7.6 Test Suite: Ability to conduct NAT of destination multicast address

This only applies to vendors claiming support for translating multicast addresses.

PICS: MCAST-9.

Setup: TTS4 (Figure 17), with four consists (A, B, C, D) as shown in Figure 39.

Consist A, B and D: Consists with single ETBN/ECN (8.3.2.2), where test system simulates an ETBN and an end device.

Consist C: Consist with single ETBN/ECN (8.3.2.2), where IUT represents the ETBN (ETBN C) and test system simulates an end device (H3).

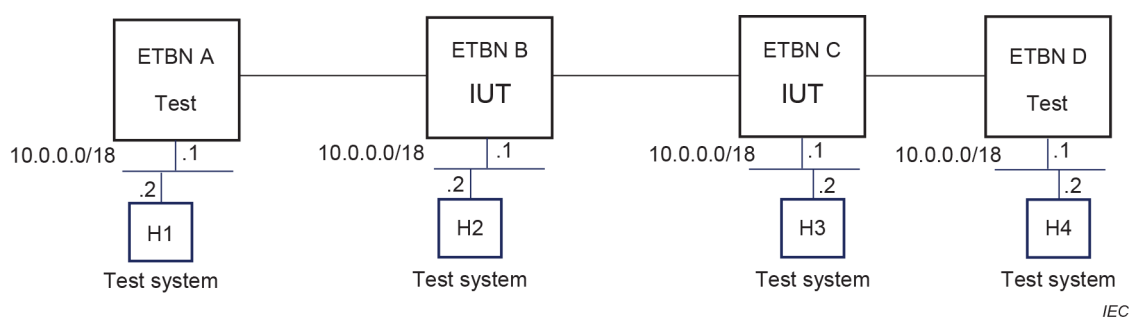


Figure 39 – Test ability to translate multicast destination address

Verify that packets to 239.192.0.X for inaugurated consist number (X) are forwarded by the ETBN onto the ECN with destination address translated to a local multicast address according to vendor PIXIT. Also verify that NAT translation adapts after reinaugurating to a new consist number (X).

Test steps:

- 1) Initial state: ETBN A, B, C and D in inaugurated state. No inhibition set for any ETBN. IUT consist becomes “consist 3”.
- 2) Let Test System H1 send multicast streams to 239.192.0.2 and 239.192.0.3. ETBN A routes these streams onto ETB with trainwide source address.
- 3) Verify that packets of stream 239.192.0.3 are routed by the IUT onto its ECN (to H3), with destination address translated to 239.192.0.0. Packets shall keep the trainwide source address.
- 4) Stop emulating consist B, thereby trigger a reinauguration where IUT consist becomes “consist 2”.
- 5) Verify that packets of stream 239.192.0.2 are now routed by the IUT onto its ECN (to H3), with destination address translated to 239.192.0.0. Packets shall keep the trainwide source address.

Acceptance criteria: Successful verification in the steps above.

8.3.8 Test Suite: TOPOLOGY frame format and basic functionality

8.3.8.1 Purpose

Correct transmission TOPOLOGY frames

- (TOPO-FRAME-1) Encapsulation of transmitted TOPOLOGY frames: Verify that the ETB-TLV and CN-TLV of transmitted TTDP TOPOLOGY messages are encapsulated in compliance with the standard.
 - References: 8.7.2-8.7.4, 8.7.6, Item 12 of [2-5-COR1]
- (TOPO-FRAME-2) Verify that the structure and content of transmitted ETB-TLV and CN-TLV are in compliance with the standard.
 - References: 8.7.6.
- (TOPO-FRAME-3) Verify that the implementation accepts the encapsulation of ETB-TLV and CN-TLV of incoming TTDP TOPOLOGY frames in compliance with the standard.
 - References: 8.7.2-8.7.4, 8.7.6
- (TOPO-FRAME-4) Verify that the implementation accepts format and content of incoming ETB-TLV and CN-TLV in compliance with the standard.
 - References: 8.7.6, Item 11 of [2-5-COR1].

8.3.8.2 Test Case: Encapsulation and content of transmitted TTDP TOPOLOGY frames

Test cases for TOPO-FRAME-1 and TOPO-FRAME-2.

(TOPO-FRAME-1) Encapsulation of transmitted TOPOLOGY frames: Verify that the ETB-TLV and CN-TLV of transmitted TTDP TOPOLOGY messages are encapsulated in compliance with the standard.

Test Setup: TTS3 (Figure 14). There will be 3 ETBNs (A, B, C), where the IUT is the middle ETBN (B) and the test system emulates the ETBNs at the ends (A, C).

The IUT ETBN (B) and the test system ETBNs (A, C) each represent a consist advertising a “Single ETBN/CN” topology described in 8.3.2.2.

Verify correct Ethernet and VLAN headers:

- DMAC: Use “All Bridges” MAC address as defined in Table 13 (8.7.4)
- SMAC: Shall match MAC used as “ownMacAddr” inside TOPOLOGY frame (and “srcID” inside HELLO frame) to enable “Internal peers detection” (8.6.1).
- VLAN header: Verify correct VLAN ID and priority (8.7.3).
- Ethertype: Verify correct Ethertype (8.7.6)

Verify correct TOPOLOGY frame padding and order of TLVs

- Padding byte: Verify presence of initial padding byte (8.7.6)
- TLV presence and order: Verify that TOPOLOGY frame contains one ETB-TLV, one CN-TLV and one EOL-TLV in that order. (8.7.6). This includes verification of “tlvType” and “tlvLength” of ETB-TLV and CN-TLV. Use of “optional TLVs” shall not be allowed (item 12 of [2-5-COR1]).

Acceptance criteria: Successful verification as listed in test steps.

(TOPO-FRAME-2) Verify that the structure and content of transmitted ETB-TLV and CN-TLV are in compliance with the standard.

Test Setup: TTS3 (Figure 14). There will be 3 ETBNs (A, B, C), where the IUT is the middle ETBN (B) and the test system emulates the ETBNs at the ends (A, C).

The IUT ETBN (B) and the test system ETBNs (A, C) each represent a consist advertising a “Single ETBN/CN” topology described in 8.3.2.2.

Verify ETB TLV of TOPOLOGY frame sent by IUT:

- TLV checksum: Verify that “tlvCS” is correctly computed
- Protocol ID: Verify that “protoID” field contains value as defined in 8.7.6 (ASCII “TTDP”)
- Protocol version: Verify that the “protoVersion” field contains the value defined in 8.7.6 (0100000000'H).
- Life-sign: Verify that the lifeSign counter field is incremented for each TOPOLOGY frame sent by the implementation. No requirement on selection of initial value. Wrap-around functionality only verified if implementation selects initial value so high that lifeSign reaches MAX within duration of test (this is unlikely).
- Consist UUID: Verify that the “cstUuid” contains value matching the configured value.
- Reserved fields: Verify that the fields “reserved1”, “reserved2” and “reserved3” contain “all zeros”.
- Inauguration state: Verify etbnInaugState converges to contain value “Inaugurated”. This assumes that the test environment is stable (emulating one consist on each side of the IUT, etbnInhibition FALSE). Then the etbnInaugState field should contain value “Inaugurated”, perhaps except for a few initial TOPOLOGY frames.
- Node role: Verify that etbnNodeRole contains value “EtbnRoleNotRedundant”. This assumes the IUT is configured to advertise a “Single ETBN/CN” topology described in 8.3.2.2.
- ETBN inhibition request: Verify that the “etbnInhibition” field contains value FALSE. This assumes that the IUT does not receive control stimuli to request inhibition.
- Remote inhibition flag: Verify that the “remoteInhibition” contains value “FALSE”.
- Connectivity Table CRC: Verify that the “connTableCrc32” field converges to contain the correct value for the given train topology. The test system is assumed to emulate one ETBN in each direction, each representing a “Single ETBN/CN” consist topology described in 8.3.2.2. (Covered further in 8.2.2.10.)
- Link Information: Verify that etbnDir1LinkInfo/etbnDir2LinkInfo converges to contain value

- “TRUE” for etbnLineAstatus in each direction, and “UNDEFINED” for lines B-D in each direction. (Value “FALSE” is acceptable for Lines B-D if IUT configuration requires those lines to be included).
- “A” for etbnLineAdistIdent in each direction, and “-” for lines B-D in each direction.
- Dir 1 and dir 2 MAC addresses: Verify that “dir1MacAddr” and “dir2MacAddr” fields converges to contain MAC addresses of the neighbour ETBNs emulated by the test system in each respective direction. (Covered further in 8.2.2.10.)
- Own MAC address: Verify that “ownMacAddr” field represents the MAC address used as TTDP identity for the IUT and shall match SMAC in Ethernet header of TTDP TOPOLOGY frame. (Covered further in 8.2.2.10.)
- Number of ETBNs in each direction: Verify that “nDir1Etbn” and “nDir2Etbn” both contain value “1”. (Covered further in 8.2.2.10.)
- ETBN Vectors: Verify that “dir1EtbnVector” and “dir2EtbnVector” fields each contain a single MAC with value corresponding to the MAC address of the neighbour ETBNs emulated by the test system in each respective direction. (Covered further in 8.2.2.10.)

Verify CN TLV of TOPOLOGY frame sent by IUT:

- TLV checksum: Verify that “tlvCS” is correctly computed.
- etbTopoCnt: Verify that the “etbTopoCnt” converges to represent the TND checksum for this rudimentary train topology. The test system is assumed to emulate one ETBN in each direction, each representing a “Single ETBN/CN” consist topology described in 8.3.2.2. (8.8.4).
- Static relative position: Verify that “ownEtbnNb” contains value “1”. (This assumes the IUT is configured to advertise a “Single ETBN/CN” topology described in 8.3.2.2.)
- Train lengthening/shortening: Verify that “lengthen” and “shorten” fields both contain value “FALSE”.
- Reserved: Verify that field “reserved1” contains “all zeros”.
- Nb of ETBNs and CNs in consist: Verify that the field “nEtbnCst” contains value “1” and that the field “nCnCst” contains value “1”. (This assumes the IUT is configured to advertise a “Single ETBN/CN” topology described in 8.3.2.2.)
- CN/ETBN mapping: Verify that field “cnToEtbnList” contains proper value for a “Single ETBN/CN” consist topology. 01 00 00 00 (hex), i.e., the first byte holds BITSET for CNs #1-#8, etc. (BITSET32 according to IEC 61375-2-1:2012).
- CN Types and padding: Verify that a single cnTypes field is sent (value in allowed range), and that three padding octets with zeros follow for 32-bit alignment. (This assumes the IUT is configured to advertise a “Single ETBN/CN” topology described in 8.3.2.2.)

Acceptance criteria: Successful verification as listed in test steps.

8.3.8.3 Test Case: Handling of incoming TTDP TOPOLOGY frames

Test cases for TOPO-FRAME-3 and TOPO-FRAME-4.

(TOPO-FRAME-3) Verify that the implementation accepts the encapsulation of ETB-TLV and CN-TLV of incoming TTDP TOPOLOGY frames in compliance with the standard.

Test Setup: TTS2 (Figure 13). There will be 2 ETBNs (A, B) where IUT is in the right (B) and test system emulates A. Both A and B would represent a consist with “Single ETBN/CN” consist topology described in 8.3.2.2.

The ability to accept incoming TTDP TOPOLOGY frames with encapsulation compliant with the standard is to a large extent covered by the tests outlined by TOPO-FRAME-1. Therefore, this test case is limited to the following steps:

- Reserved fields: Verify that IUT ignores the value of “reserved1” field (between EtherType and ETB-TLV) of incoming TOPOLOGY frames.

- Let the test system ETBN (A) send regular TOPOLOGY messages with “reserved1” set to “zeros” and verify that A and B reach inaugurated state with expected etbTopoCnt.
- Now, modify “reserved1” in the TOPOLOGY frames from A to hold at least one bit set to 1. A and B should still be in “inaugurated” state with same etbTopoCnt.
- Other TLVs: Verify that the IUT can handle (that is, ignore) “otherTLVs”, if such are present between CN-TLV and EOL-TLV. The test system can insert a dummy TLV, and the IUT should ignore the bytes of that TLV based length field.
 - Let the test system ETBN (A) send regular TOPOLOGY messages without any “other TLV”, and verify that A and B reach inaugurated state with expected etbTopoCnt.
 - Now, let TOPOLOGY frames from A to hold an additional TLV (e.g., type 3, length 2, value “0102” (hex)) between CN-TLV and EOL-TLV. A and B should still be in “inaugurated” state with same etbTopoCnt.

Acceptance criteria: Successful verification as listed in test steps.

(TOPO-FRAME-4) Verify that the implementation accepts format and content of incoming ETB-TLV and CN-TLV in compliance with the standard.

Test Setup: TTS3 (Figure 14). There will be 4 ETBNs (A, B, C, D and E), where the IUT is the second ETBN (B) and the test system emulates the ETBNs on each side of B (A, C, D and E).

The ability to accept incoming TTDP TOPOLOGY frames with format and content compliant with the standard is to a large extent covered by the tests outlined by TOPO-FRAME-2. Therefore, this test case is limited to the following steps:

ETB-TLV:

- Protocol Minor Version: Verify ability to accept protocol version with same “Major number” (1), but with other minor number (non-zero).
 - Let the test system ETBN (A, C, D and E) send regular TOPOLOGY messages with “minor version” set to “zeros” and verify that A and B reach inaugurated state with expected etbTopoCnt.
 - Now, modify “minor version” in the TOPOLOGY frames from A to hold a non-zero value in its “minor version field” e.g., mmmmmm could be 000001. A and B should still be in “inaugurated” state with same etbTopoCnt.
 - (For this to work, the IUT may need to be rebooted; it is not assumed that the minor version will be changed on a running train.)
- Protocol Major Version: Verify ability to reject protocol version with other “Major number” (2).
 - Let the test system ETBN (A, C, D and E) send Major version 2 in TOPOLOGY messages from A and version 1 from C, D and E. (Minor version “zeros” in TOPOLOGY. All units A, C, D, and E should send HELLO with Major version 1, and minor version “zeros”.)
 - Verify that B rejects TOPOLOGY messages from A, but accepts them from C, D and E. It should reach inaugurated state with expected etbTopoCnt.
- Reserved fields: Verify that IUT ignores value of “reserved2” and “reserved3” fields inside ETB-TLV.
 - Let the test system ETBN (A) send regular TOPOLOGY messages with “reserved2” and “reserved3” set to “zeros” and verify that IUT (B) reach inaugurated state with expected etbTopoCnt.
 - Now, modify “reserved2” and “reserved3” in the TOPOLOGY frames from A to each hold at least one bit set to 1. IUT (B) should still be in “inaugurated” state with same etbTopoCnt.

- Unordered ETBN vectors: Verify that IUT does not assume that “dir1EtbnVector” and “dir2EtbnVector” are ordered. To do this, the test system can shift the order in which ETBNs are sent in the vector.
 - Here we assume test system lets ETBN (C) have ETBN D and E in its dir2 vector in order D, E. Verify that IUT (B) reach inaugurated state with expected etbTopoCnt.
 - Now, let ETBN (C) send its dir2 vector in opposite order (E, D). IUT (B) should still be in “inaugurated” state with same etbTopoCnt.
- tlvCS: Let A send TOPOLOGY messages with invalid ETB-TLV tlvCS checksum for some period of time and then start so send correct again.
 - Acceptance criteria: IUT should reject TOPOLOGYs from A (etbTopoCnt based on B-E) when A is sending with incorrect tlvCS. Once correct values are sent, B should accept TOPOLOGYs from A again (etbTopoCnt based on A-E).

CN-TLV:

- Reserved fields: Verify that IUT ignores value of “reserved1” fields inside CN-TLV.
 - Let the test system ETBN (A) send regular TOPOLOGY messages with “reserved1” set to “zeros” and verify that IUT (B) reach inaugurated state with expected etbTopoCnt.
 - Now, modify “reserved1” in the TOPOLOGY frames from A to hold at least one bit set to 1. IUT (B) should still be in “inaugurated” state with same etbTopoCnt.
- tlvCS: Let A send TOPOLOGY messages with invalid CN-TLV tlvCS checksum for some period of time and then start so send correct again.
 - Acceptance criteria: IUT should reject TOPOLOGYs from A (etbTopoCnt based on B-E) when A is sending with incorrect tlvCS. Once correct values are sent, B should accept TOPOLOGYs from A again (etbTopoCnt based on A-E).

Acceptance criteria: Successful verification as listed in test steps.

8.3.9 Test Suite: Physical topology management and calculation

8.3.9.1 Purpose

Correct transmission of Connectivity Vector (TOPOLOGY frame)

References: 8.8.1, 8.7.6 (dir1MacAddr, OwnMacAddr, dir2MacAddr).

PICS: TOPO-PHY-1.

Check that IUT transmits correct Connectivity Vector when

- having no neighbour, but link is logically up (HELLO exchange)
- having one neighbour
- having two neighbours
- change of neighbour (current neighbour goes down, or new comes up).

Check that no TOPOLOGY frame is observed from IUT when links are logically down (even if links are physically up).

- Connectivity vector cannot be observed when links are logically down in both directions of the IUT.
- Verify that IUT considers itself to be alone by SNMP or the method provided by vendor PIXIT, e.g. SNMP OID ttdplsAlone.

Management and correct transmission of ETBN Vector(s) and number of ETBNs (TOPOLOGY frame)

References: 8.6.1 (detect peers) 8.8.2, 8.7.6 (dir1EtbnVector, dir2EtbnVector, nDir1Etbn, nDir2Etbn), 8.9.2.

PICS: TOPO-PHY-2.

Check that IUT can fill out correct ETBN Vectors (dir1EtbnVector, dir2EtbnVector) and number of ETBNs in each direction (nDir1Etbn, nDir2Etbn) for the following cases:

- when IUT is alone (timeout 1 s, 8.9.2) This assumes that test system emulates logical link up by transmitting HELLO frames, but no TOPOLOGY frames; otherwise no TOPOLOGY frames would be observed from IUT.
- with multiple other ETBNs on ETB; in particular, test extreme values (Annex A).
- when ETBNs are added (reception of new TOPOLOGY) or lost (timeout 400 ms, 8.9.2, 8.6.1).
- for different values of “inhibition status”. These values should always match the current physical topology when inhibition is false. When inhibition is true, the values should reflect ETBNs present up to the extremity ports (ports in DISCARDING).
- The following test cases are more relevant for connTableCrc32 calculation (below). The transmission of ETBN Vectors and number of ETBNs should be independent of the ETB reference direction and consist orientations.
 - when other ETBNs have different orientation in relation to ETB reference.
 - when IUT itself has different orientation relative ETB reference; this is more relevant for ConnTableCrc32 calculation (below).
 - when ETB reference direction changes; this is more relevant for ConnTableCrc32 calculation (below).
 - when the internal order of ETBN vectors come in different order (as the ETBN vectors are unordered, the IUT should make no assumptions about the order).

These cases are suitable to test together with “Connectivity Table Management” and “ConnTableCrc32 Calculation”, see below.

Management of Connectivity Table and correct calculation of ConnTableCrc32 (TOPOLOGY frame)

References: 8.2.1 (Physical Topology), 8.5.2 (BuildPhysicalTopology, StartTtdp, “ConnTableCrc32 Default Value”), 8.8.3, 8.8.4, 8.7.6 (connTableCrc32).

PICS: TOPO-PHY-3.

Check that IUT can compute correct ConnTableCrc32 for the same cases as described for transmission of ETBN Vectors and number of ETBNs above.

Test together with management of ETBN vectors (above).

8.3.9.2 Transmission of Connectivity Vector

Test Setup: TTS-3 (Figure 14), Test system emulates up to two neighbours on each side of the IUT, as shown in Figure 40.

PICS: TOPO-PHY-1.

Initial condition: ETBN connects to test system as IUT, test system monitor and record the ETB ports and simulate the communications in both sides of ETB networks. Unless stated otherwise, all ETBNs should report “etbnInhibition” as “False”, i.e, inauguration is allowed.

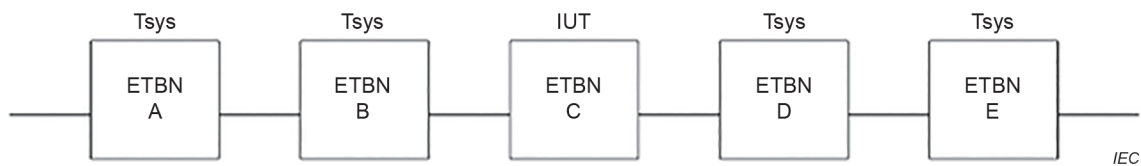


Figure 40 – Testing transmission of connectivity vector

Test steps:

- 1) Monitor and record TTDP topology frame of ETBN.
- 2) Start with no simulated neighbours (A, B, D and E logically down, but links physically up).
 - Verify that no TTDP TOPOLOGY is sent by IUT (neither in dir1 nor dir2).
 - Verify that IUT considers itself to be alone by SNMP request via IUTs CN interface (OID ttdplsAlone is TRUE).
- 3) Bring up A, i.e., simulate a new neighbour ETBN (A) in direction 1 of IUT.
 - Verify that IUT sends correct Connectivity Vector [A,Self,0] in dir 1 TOPOLOGY frames. No TOPOLOGY frame in dir2.
 - Verify that IUT considers itself to be “not alone” by SNMP request via IUTs CN interface (OID ttdplsAlone is FALSE).
- 4) Bring up D, i.e., simulate a new neighbour ETBN (D) in direction 2 of IUT.
 - Verify that IUT sends correct Connectivity Vector [A,Self,D] in dir 1 and dir 2 TOPOLOGY frames.
- 5) Bring up B, i.e., simulate a new neighbour ETBN (B) in direction 1 of IUT. Note that HELLO from A should not reach IUT.
 - Verify that IUT adapts, and sends correct Connectivity Vector [B,Self,D] in dir 1 and dir 2 TOPOLOGY frames.
- 6) Bring up E, i.e., simulate a new ETBN (E) in direction 2 of IUT. Note that HELLO from E should not reach IUT.
 - Verify that IUT continues sending Connectivity Vector [B,Self,D] in dir 1 and dir 2 TOPOLOGY frames.
- 7) Bring down D to simulate loss of neighbour ETBN (C) . Note that both HELLO and TOPOLOGY from E should reach IUT.
 - Verify that IUT adapts and sends correct Connectivity Vector [B,Self,E] in dir 1 and dir 2 TOPOLOGY frames.
- 8) Bring down E, i.e., simulate loss of last ETBN in on direction.
 - Verify that IUT adapts and sends correct Connectivity Vector [B,Self,0] in dir 1 TOPOLOGY frames. No TOPOLOGY frame in dir2.
- 9) Bring down all ETBNs in direction 1 (A and B) to simulate entering “alone” state.
 - Verify that no TTDP TOPOLOGY is sent by IUT (neither in dir1 nor dir2).

Final conditions: Test Bench powered off.

Acceptance criteria: OK if:

- The Connectivity Vector (dir1MacAddr, OwnMacAddr, dir2MacAddr) are consistent with the simulation value for each verification step, and that no TOPOLOGY frame is sent when IUT has no neighbour.

8.3.9.3 Transmission of ETBN Vector(s), number of ETBNs and ConnTableCrc32

Test Setup: TTS-3 (Figure 14), Test system emulates two ETBNs on each side of the IUT, as shown in Figure 41.

PICS: TOPO-PHY-2 and PICS: TOPO-PHY-3.

Simple setup

8.3.2.2).

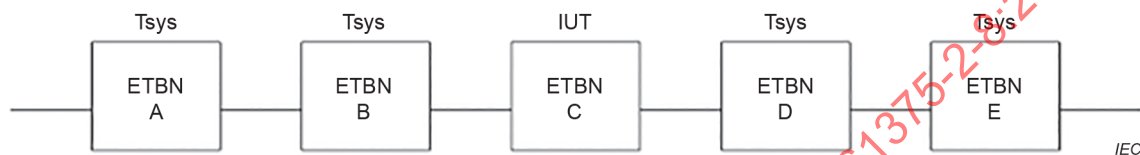


Figure 41 – Testing transmission of ETBN Vectors, number of ETBNs and ConnTableCrc32

CstUUID assignment: Let CstUUID of nodes be assigned in ascending order for the nodes: A, E, D, B, C (A lowest, C highest).

Cst Orientation: Let all ETBNs have their configured Cst Orientation such that dir1 is to the left, and dir2 to the right.

Unless stated otherwise, all ETBNs should report “etbnInhibition” as “False”, i.e, inauguration is allowed.

Test steps:

In the verification steps, “Vector Values” refer to dir1EtbnVector, dir2EtbnVector, nDir1Etbn and nDir2Etbn.

- 1) Monitor and record the TTDP TOPOLOGY frame of IUT.

IUT and one simulated ETBN (B) in direction 1 are up. (A, D, and E down). ETB Reference direction points left.

- Verify that IUT transmits correct Vector Values and ConnTable32. IUT should report one node (B) in its dir1 and no node in its dir2 direction.

- 2) Bring up simulated node D. ETB Reference direction points right.

- Verify that IUT transmits correct Vector Values and ConnTable32. IUT should report one node (B) in its dir1 and one node (D) in its dir2 direction.

- 3) Bring up simulated node E. ETB Reference direction still points right.

- Verify that IUT transmits correct Vector Values and ConnTable32. IUT should report 1 node (B) in its dir1 and two nodes (D,E; arbitrary order) in its dir2 direction.

- 4) Bring up simulated node A. ETB Reference direction now points left.

- Verify that IUT transmits correct Vector Values and ConnTable32. IUT should report two nodes in its dir1 (A,B; arbitrary order) and two nodes (D,E; arbitrary order) in its dir2 direction.

- 5) Let simulated node D have inverse configured direction, i.e., change D to have its configured direction1 to the right and direction2 to the left.
 - Verify that IUT reports correct Vector Values and ConnTable32. Vector Values should be equivalent to previous step, but ConnTable32 should differ.
- 6) Bring down simulated node A. ETB Reference direction now points right.
 - Verify that IUT transmits correct Vector Values and ConnTable32. IUT should report 1 node (B) in its dir1 and two nodes (D,E; arbitrary order) in its dir2 direction.

Acceptance criteria: OK if:

- IUT transmits correct Vector Values and ConnTable32 checksum according to given verification steps.

Setup with max ETBNs

ETBN connects to test system as IUT, test system monitors and records the ETB ports and simulate the communications in both sides of ETB networks, here with many ETBNs on either side. The purpose is to check that IUT handles maximum number of ETBNs for ETBN Vector Values and the ConnTable32 checksum. Each ETBN and the IUT represents a consist with own CstUUID, advertising a single ETBN/CN (see 8.3.2.2). In the following steps it assumed that the IUT has its direction1 pointing left, and its direction2 pointing right. The simulated ETBNs should have a mix of configured directions (some have dir1 to the left, while others have dir1 to the right).

Test steps:

- 1) Monitor and record the TTDP TOPOLOGY frame of IUT.
- 2) Simulate IUT with 62 ETBNs in its dir2 direction. Let CstUUID of simulated end (rightmost ETBN) be higher than of IUT, i.e. ETB reference direction points left.
 - Verify that IUT transmits correct Vector Values and ConnTable32. IUT should report no nodes in its dir1 and 62 nodes (arbitrary order) in its dir2 direction.
- 3) Let CstUUID of simulated end (rightmost ETBN) be lower than of IUT, i.e. ETB reference direction points right.
 - Verify that IUT transmits correct Vector Values and ConnTable32. As before, IUT should report no nodes in its dir1 and 62 nodes (arbitrary order) in its dir2 direction. The ConnTable32 checksum should differ.
- 4) Now simulate 32 ETBNs on each side of the IUT. The leftmost simulated ETBN should be given lower CstUUID than the ETBN in the right end. ETB reference direction points to the left.
 - Verify that IUT transmits correct Vector Values and ConnTable32. IUT should report 32 nodes in its dir1 and 32 nodes (arbitrary order) in its dir2 direction.
- 5) Now change CstUUID of the right end to be lower than the left end. ETB reference direction points to the right.
 - Verify that IUT transmits correct Vector Values and ConnTable32. As before, IUT should report 32 nodes in its dir1 and 32 nodes (arbitrary order) in its dir2 direction. The ConnTable32 checksum should differ.

Acceptance criteria: OK if successful verification as defined in test steps.

8.3.10 Test Suite: Logical connectivity management and calculation

8.3.10.1 Purpose

- a) Maintain Train Network Directory (TND).

PICS: TOPO-GEN-1 – TOPO-GEN-3, TOPO-TND-TRAIN-1, TOPO-TND-TRAIN-2.

References: 8.8.5 and 8.8.6 of [2-5].

Handle TND serialization (8.8.5):

- 1st level serialization (Consist), including ETB reference dir.
- 2nd level serialization (CN)
- 3rd level serialization (ETBN).

Verify by test correct etbTopoCnt calculation for typical and extreme train configurations, and events (adding/removing consist, changing reference direction, changing consist orientation).

Test correct etbTopoCnt computation:

- IUT: Single ETBN/CN ([2-8] 8.3.2.2)
- Remote (simulated by test system): That is the IUT capability to interact with the following topologies should be tested:
 - All the “common” consist topologies declared in [2-8] 8.3.2. (TOPO-TND-TRAIN-1).
 - Consists with extreme numbers of ETBNs and consists (TOPO-TND-TRAIN-2):
 - i) 62 more Single ETBN/CN consists ([2-8] 8.3.2.2).
 - ii) 1 consist with 32 ETBNs/CNs ([2-8] 8.3.2.3), 1 consist with 30 ETBNs/CNs ([2-8] 8.3.2.3).
 - iii) 31 consists with Redundant ETBN, dual CN ([2-8] 8.3.2.6).
- Verify: Proper etbTopoCnt.

Test correct transmission of etbTopoCnt

- Current etbTopoCnt when etbInhibition is false (TOPO-GEN-2).
- “Last Valid” etbTopoCnt when etbInhibition is false (TOPO-GEN-3).

b) Ability to transmit consist topology:

PICS: TOPO-TND-CST-1 – TOPO-TND-CST-6.

Only topology specified in TOPO-TND-CST-1 is mandatory to support. Other topologies are optional and only relevant if vendor claims support.

References: 8.7.6 and Annex A of [2-5].

Fields: ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList.

8.3.10.2 Train Network Directory (interact with common consist topologies)

PICS: TOPO-TND-TRAIN-1.

Test Setup: TTS-3 (Figure 14).

Initial conditions: IUT represents a consist with single ETBN/CN (8.3.2.2) with IUT direction point “left”. Test system will represent another consist of different types, starting with single ETBN/CN (8.3.2.2). etbInhibition is FALSE throughout the tests.

Tests: For each topology in 8.3.2.2-8.3.2.7 do steps given further below. Let test system simulate one of the ETBNs in the given consist topology. One is enough, since it announces the full consist topology. Special directives per simulated consist topology topology:

- “Single ETBN/CN” consist topology (8.3.2.2): Simulate a consist with a single ETBN and single CN.
- “Multiple single ETB/CN” consist topology (8.3.2.3): Simulate a topology with two ETBNs, each with a separate CN. Any of the two ETBNs can be simulated.
- “Single ETBN, multiple CN” consist topology (8.3.2.4): Simulate a topology with a single ETBN with two CNs, each with a separate CN.

- “Redundant ETBN, single CN” consist topology (8.3.2.5): Simulate a topology with two ETBNs connected to the same, single CN. Any of the two ETBNs can be simulated.
- “Redundant ETBN, multiple CN” consist topology (8.3.2.6): Simulate a topology with two ETBNs connected to the same two CNs. Any of the two ETBNs can be simulated.
- “Asymmetrical Consist” consist topology (8.3.2.7): Simulate the topology shown in the figure in 8.3.2.7. Any of the three ETBNs can be simulated.

Steps (done for each simulated consist topology):

- 1) Let the simulated ETBN be located to the left of the IUT, with orientation pointing left and with lower CstUuid lower than of IUT (ETB reference direction pointing left).
 - Verify that IUT transmits correct etbTopoCnt.
- 2) Change orientation of simulated ETBN, i.e., let its direction1 point to the right.
 - Verify that IUT transmits correct etbTopoCnt.
- 3) Change CstUuid of simulated ETBN to be higher than of IUT, i.e., ETB reference direction points to the right.
 - Verify that IUT transmits correct etbTopoCnt.

Final conditions: Test Bench powered off.

Acceptance criteria: OK if:

- All verification steps are successful.

8.3.10.3 Train Network Directory (interact with consists using extreme values)

PICS: TOPO-TND-TRAIN-2.

Test 1: Max ETBN per Consist

Test Setup: TTS-3 (Figure 14)

PICS: TOPO-TND-TRAIN-2 (partial fulfilment)

- Max ETBN/Consist (32)
- Max ETBN per train (63)

Initial condition: ETBN connects to test system as IUT, test system monitor and record the ETB ports and simulate the communications in both sides of ETB networks.

[Cst A: Simulated] – [Cst B: SUT] – [Cst C: Simulated]

- Cst A: Simulates consist with 32 ETBN nodes (Max ETBN/Consist), each connected to a CN (see 8.3.2.2). Simulator only needs to send TOPOLOGY frames from one ETBN in Cst A, as it contains the full topology of the consist.
- Cst C: Simulates consist with 30 ETBN nodes, each connected to a CN (see 8.3.2.2). Simulator only needs to send TOPOLOGY frames from one ETBN in Cst B, as it contains the full topology of the consist.
- In the SUT (TTS-3), N equals to 1 (i.e., single IUT). The etbInhibition status is “FALSE” in SUT and simulated nodes to allow inauguration.
- Let all consists (A, B and C) have their consist direction “towards A” (simulator 1); this is only a recommendation aiming to achieve a rudimentary test of limited compliance.

Test steps:

- 1) Start simulator. The simulator simulates an ETBN in each of consist A and C.
- 2) Start SUT/IUT and await that it boots up.
- 3) Monitor and record the TTDP TOPOLOGY frame of IUT ETBN. Inspect etbInaugState and etbTopoCnt.

Acceptance criteria: OK if IUT is capable of computing and exchanging correct etbTopoCnt and reach state “Inaugurated”.

Test 2: Max number of consists with dual ETBNs/CNs

Test Setup: TTS-2 (Figure 13) or TTS-4 (Figure 15).

PICS: TOPO-TND-TRAIN-2 (partial fulfilment).

- Max ETBNs/Train (63; here only 62 are tested)
- Max CNs/Train (63; here only 62 are tested)
- Max Consists/Train (~63; here only 32 are tested, due to redundant ETBN consist topology, which is anticipated to be a common)

Initial condition: ETBN connects to test system as IUT, test system monitor and record the ETB ports and simulate the communications in on one side of SUT.

[Cst 1-31: Simulated] – [Cst 32: SUT]

- There are 31 simulated consists in the ETB network. Each simulated consist has two ETBN nodes and two consist networks as described in 8.3.2.6.
- In the SUT, vendor can select between the following topologies:
 - Redundant ETBN with multiple CN (8.3.2.6), TTS-4 (Figure 15: Here with N equal to 2, i.e., two IUTs with two CNs, which is what the simulated consists also have. This alternative should be used if supported. Note that in the test it is possible to use only one physical ETBN as SUT, as it will advertise the topology for the whole consist.
 - Single ETBN/CN (8.3.2.2), TTS-2 (Figure 13): This is the minimum consist topology required to support and should be used if the previous alternative is not supported.
- Let all consists (1-32) have their consist direction “towards Consist 1” (simulator 1); this is only a recommendation aiming to achieve a test of limited compliance. Complementary tests with other orientations can be added by test institute.

Test steps:

- 1) Start simulator. The simulator simulates 31 consists, as described above.
- 2) Start SUT and await that it boots up.
- 3) Monitor and record the TTDP TOPOLOGY frame of IUT ETBN. Inspect etbnInaugState and etbTopoCnt.

Acceptance criteria: OK if ETBN(s) in SUT are capable of computing and exchanging correct etbTopoCnt and reach state “Inaugurated”.

Test 3: Max number of ETBNs, Consists and CNs

Test Setup: TTS-2 (Figure 13).

PICS: TOPO-TND-TRAIN-2 (partial fulfilment).

- Max ETBNs/Train (63)
- Max CNs/Train (63)
- Max Consists/Train (~63; here 63 are tested)

Initial condition: ETBN connects to test system as IUT, test system monitor and record the ETB ports and simulate the communications in on one side of SUT. There are 63 consists in the ETB network.

[Cst 1-62: Simulated] – [Cst 63: SUT]

- There are 62 simulated consists. Each consist has only one ETBN node, i.e., single ETBN/CN (8.3.2.2).
- In the SUT consist, N equals to 1, i.e., single ETBN/CN ().

- Let all consists (1-63) have their consist direction “towards Consist 1” (simulator 1); this is only a recommendation aiming to achieve a test of limited compliance. Complementary tests with other orientations can be added by test institute.

Test steps:

- 1) Start simulator. The simulator simulates 62 consists, as described above.
- 2) Start SUT and await that it boots up.
- 3) Monitor and record the TTDP TOPOLOGY frame of IUT ETBN. Inspect etbnInaugState and etbTopoCnt.

Acceptance criteria: OK if IUT is capable of computing and exchanging correct etbTopoCnt and reach state "Inaugurated".

8.3.10.4 Transmit consist topology

Test Setup: TTS-4 (Figure 15).

Test 1: ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList for “max CN per Consist” topology

Initial condition: ETBN connects to test system as IUT, test system monitor and record the ETB ports and simulate the communications in both sides of ETB networks. There are 32 ETBN nodes in a consist. ETBN nodes are not redundant, each of which is connected to a CN.

In the SUT, N equals to 1 (i.e., single IUT) and the inhibition status is “false” in the inauguration.

Test steps:

- 1) The simulator simulates 31 ETBN nodes and each of them is connected to a CN. IUT connects to a CN too. The direction 1 of the consist points to simulator 1.
- 2) Examine the logical topology.
- 3) Monitor and record the TTDP TOPOLOGY frame of ETBN.

Acceptance criteria: OK if:

- The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame are consistent with the values calculated by logical topology of the IUT.

After the successful inauguration, simulate a node is missing.

Monitor and record the TTDP TOPOLOGY frame of ETBN.

Acceptance criteria: OK if:

- After the successful re-inauguration, there is no missing node in the logical topology. The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame have changed and they are consistent with the values calculated by logical topology of the IUT.

After the successful re-inauguration, simulate a node joining.

Monitor and record the TTDP TOPOLOGY frame of ETBN.

Acceptance criteria: OK if:

- After the successful re-inauguration, there exists joined nodes in the logical topology. The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame have changed and they are consistent with the values calculated by logical topology of the IUT.
- Set the inhibition status as “inhibition” in the inauguration.
- Monitor and record the TTDP TOPOLOGY frame of ETBN.

Acceptance criteria: OK if:

- No new inauguration. The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame remain unchanged.

Test 2: ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList for “max dual ETBN/CN” topology

Initial condition: There are 31 consists in the ETB network. Each consist has only two ETBN nodes which are mutual redundant and there are two consist networks in the consist.

In the SUT, N equals to 2.

Test steps:

- 1) The simulator simulates 30 consists. Each consist has two ETBN nodes which are mutual redundant and there are two CNs in the consist.

The ETBN of the SUT is in a separate consist and the SUT connects two CNs too. The direction 1 of the consist points to simulator 1.

Acceptance criteria: OK if:

- The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame are consistent with the values calculated by logical topology of the IUT.

- 2) After the successful inauguration, simulate a node is missing.

Acceptance criteria: OK if:

- After the successful re-inauguration, there is no missing node in the logical topology. The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame have changed and they are consistent with the values calculated by logical topology of the IUT.

- 3) After the successful re-inauguration, simulate a node joined.

Acceptance criteria: OK if:

- After the successful re-inauguration, there exists joined nodes in the logical topology. The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame have changed and they are consistent with the values calculated by logical topology of the IUT.

Test 3: ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList for “max ETBN, Consist, CN” topology

Initial Condition: There are 63 consists in the ETB network. Each consist has only one ETBN node.

In the SUT, N equals to 1.

Test steps:

- 1) The simulator simulates 62 consists. Each consist has only one ETBN node and each node is connected to a CN.

The IUT is in a separate consist and the IUT connects a CN too.

The direction 1 of the consist points to simulator 1.

Acceptance criteria: OK if:

- The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame are consistent with the values calculated by logical topology of the IUT.

- 2) After the successful inauguration, simulate that a node has been missed.

Acceptance criteria: OK if:

- After the successful re-inauguration, there is no missing node in the logical topology. The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame have changed and are consistent with the values calculated by logical topology of the IUT.

- 3) After the successful re-inauguration, simulate that a node has joined.

Acceptance criteria: OK if:

- After the successful re-inauguration, the joined node is in the logical topology. The values of ownEtbnNb, nEtbnCst, nCnCst, cnToEtbnList in the TTDP TOPOLOGY frame have changed and they are consistent with the values calculated by logical topology of the IUT.

8.3.11 Test Suite: Failing/loss of node(s)

8.3.11.1 Purpose

Test scenarios where a train topology has been inaugurated and acknowledged (InaugInhibition TRUE), and then loses one of the ETBNs. It is assumed that at least one of the remaining ETBNs in the acknowledged topology continues to announce etbInhibition TRUE.

(TRANS-INHIBIT-2): The remaining ETBNs should keep their configuration (no new inauguration).

(TRANS-INHIBIT-3): If an end-node is lost, its neighbour should detect it has become an end-node and set the extremity ports in DISCARDING. The new end-node shall also evaluate if it should set its “shortening” or “lengthening” flags.

Special end-node cases:

- Lengthening: If an end-node (say the right end-node) fails when there is another external consist ETBN on its right side (lengthening), the internal (left) neighbour should determine it is a new end-node and put its right port(s) in DISCARDING, possibly after a delay. It considers itself to be an end-node when the TTDP topology frames from all known (internal) neighbours on the right side has timed out, and it does this even if it hears TTDP HELLO on those ports from the new, external neighbour.
- Unknown intermediate: After a topology has been acknowledged inaugurated and acknowledged (InaugInhibition TRUE), an ETBN of an intermediate (unknown) consist may come up (see “late ETBN insertion” in a later section). If an end-node is lost in such a situation, an ETBN of the acknowledged topology may determine itself to be end-node solely based on TOPOLOGY frame timeout on that side.

8.3.11.2 Test Cases: Failing/loss of intermediate node(s)

If one or more intermediate ETBNs (redundant or non-redundant) are lost, the remaining ETBNs should keep their configuration as inauguration is inhibited. The test should verify that intermediate nodes should keep their ports in FORWARDING, and End-nodes should keep their extremity ports in DISCARDING and other ports in FORWARDING.

Case 1: Losing single and multiple intermediate nodes/consists

PICS: TRANS-INHIBIT-2.

Test Setup: TTS2 (Figure 13) with ETBNs A, B1, B2, C D as shown in Figure 42.

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBNs B1 and B2 (SUT) as redundant ETBNs/single CN (8.3.2.5)

8.3.2.2)

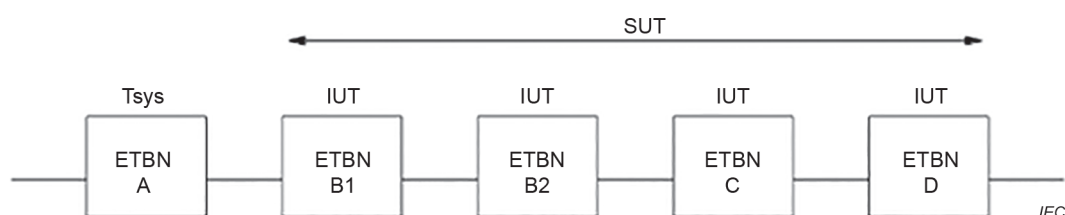


Figure 42 – Testing loss of intermediate ETBNs/consists when inauguration is inhibited

Test steps:

- 1) Bring up all ETBNs (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D which has ETB port(s) in DISCARDING (extremity).
- 3) Power off B1
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D which has ETB port(s) in DISCARDING (extremity). Verify that no new inauguration occurred.
- 4) Power off B2
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D which has ETB port(s) in DISCARDING (extremity). Verify that no new inauguration occurred.

Acceptance criteria: Successful verification as described in test steps.

Case 2: Losing intermediate node in case of “unknown intermediate consist”

PICS: TRANS-INHIBIT-3 (negative test of this requirement).

Test Setup: TTS2 (Figure 13) with ETBNs A, B, C, D, E as shown in Figure 43.

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
8.3.2.2).

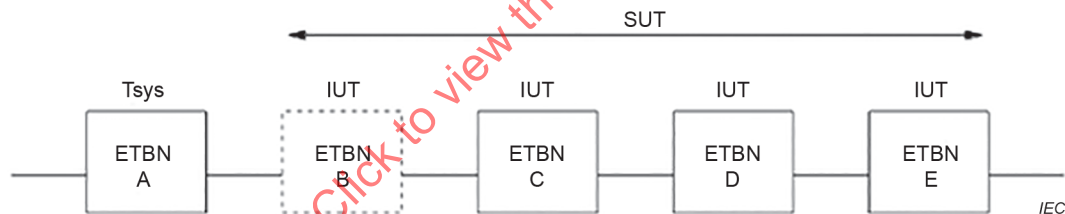


Figure 43 – Test of losing intermediate ETBN in existence of “unknown intermediate”

Test steps:

- 1) Bring up A, C, D and E (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN E which has ETB port(s) in DISCARDING (extremity)
- 3) Bring Up B (unknown intermediate)

B will not be able to inaugurate.

 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN E which has ETB port(s) in DISCARDING (extremity). Verify that no new inauguration occurred.
- 4) Power off C

D has now got a new ETBN neighbour on its left side (B), but B is not part of the topology used for etbTopoCnt. But as A is also present on the left side, D should not consider it to be entremity node towards its left side.

- Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN E which has ETB port(s) in DISCARDING (extremity). Verify that no new inauguration occurred.

Acceptance criteria: Successful verification as described in test steps.

8.3.11.3 Test Cases: Failing/loss of end node(s)

If the end-node is lost, another ETBN from the inaugurated topology should take over as end-node. Typically, it is the former neighbour ETBN of the lost end-node, although there are exceptions as when that node is a late ETBN is an “unknown intermediate”.

Case 1: Losing single or more end nodes/consists:

PICS: TRANS-INHIBIT-3.

Test Setup: TTS2 (Figure 13) with ETBNs A, B, C, D1, D2 as shown in Figure 44.

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBN B, and C (SUT) each as single ETBN/CN (8.3.2.2)
- ETBN D1, and D2 (SUT) each as single ETBN/CN (8.3.2.5).

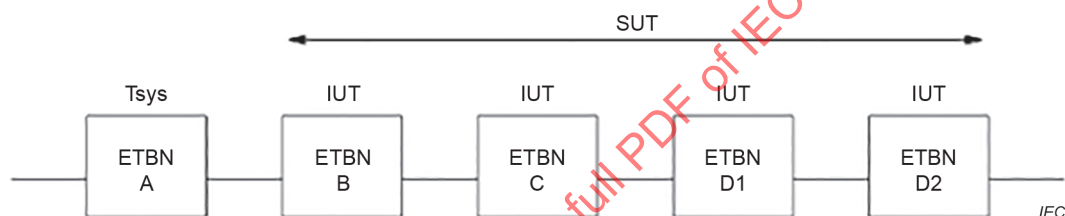


Figure 44 – Test ability to handle loss of end node

Test steps:

- 1) Bring up all ETBNs (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D2 which has ETB port(s) in DISCARDING (extremity).
- 3) Power off D2
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D1 which has ETB port(s) in DISCARDING (extremity).
- 4) Power off D1
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN C which has ETB port(s) in DISCARDING (extremity).

Acceptance criteria: Successful verification as described in test steps.

Case 2: Losing end node during train lengthening:

PICS: TRANS-INHIBIT-3.

Test Setup: TTS2 (Figure 13) with ETBNs A, B, C, D, E as shown in Figure 45.

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBN B, C, D and E (SUT) each as single ETBN/CN (8.3.2.2)

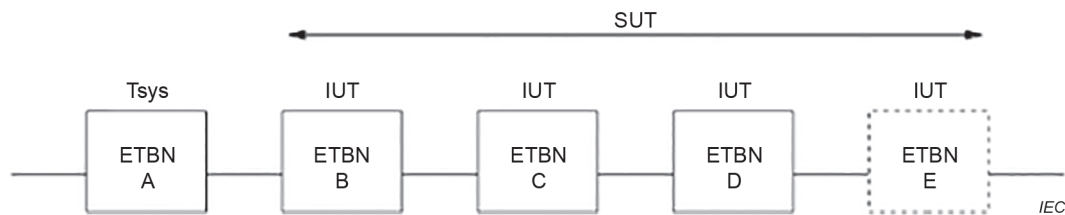


Figure 45 – Test ability to handle loss of end node during lengthening

Test steps:

- 1) Bring up A, B, C and D (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D which has ETB port(s) in DISCARDING (extremity).
- 3) Power on E (lengthening)
 - Verify that D keeps its role as extremity with the port(s) on its right side in DISCARDING. All other SUT ETB ports are in FORWARDING. E is “outside” with its own etbTopoCnt.
- 4) Power off D
 - Verify that C takes over as extremity with the port(s) on its right side in DISCARDING. All other SUT ETB ports are in FORWARDING. E is still “outside” with its own etbTopoCnt.

Acceptance criteria: Successful verification as described in test steps.

Case 3: Losing end node during in case of late intermediate consist:

Test steps:

- 1) Set inhibition (A),
 - check DISCARDING (E:right extremity), and FORWARDING (others) (C in “NOT_INAUGURATED” state)
- 2) Bring up C (unknown intermediate)
 - check DISCARDING (E:right extremity), and FORWARDING (others)
- 3) Power off E
 - check DISCARDING (D:right extremity), and FORWARDING (others)
- 4) Power off D
 - check DISCARDING (B:right extremity), and FORWARDING (others); (C in “INAUGURATED” state with other etbTopoCnt)

PICS: TRANS-INHIBIT-3.

Test Setup: TTS2 (Figure 13) with ETBNs A, B, C, D, E as shown in Figure 46.

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBN B, C, D and E (SUT) each as single ETBN/CN (8.3.2.2)

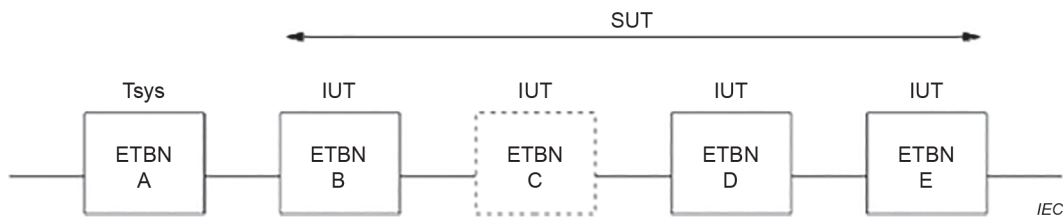


Figure 46 – Test ability to handle loss of end node when “late intermediate” is present

Test steps:

- 1) Bring up A, B, D and E (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN E which has ETB port(s) in DISCARDING (extremity).
- 3) Bring up C (unknown intermediate)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN E which has ETB port(s) in DISCARDING (extremity).
- 4) Power off E
 - Verify that D takes over as extremity with the port(s) on its right side in DISCARDING. All other SUT ETB ports are in FORWARDING.
- 5) Power off D
 - Verify that B takes over as extremity with the port(s) on its right side in DISCARDING. All other SUT ETB ports are in FORWARDING. C is “outside” with its own etbTopoCnt.

Acceptance criteria: Successful verification as described in test steps.

8.3.12 Test Suite: Late/recovered node(s)

8.3.12.1 Purpose

Test behaviour to manage scenarios with “late ETBN” or “recovered ETBN” when InaugInhibition is TRUE. The following two cases are handled separately:

- Intermediate: the late or recovered ETBN is an intermediate node, internal to the inaugurated topology (TRANS-INHIBIT-4, TRANS-INHIBIT-5).
- External: the late or recovered ETBN is an external node (external to the inaugurated topology) This case involves “moving” the extremity port, a feature which is **optional** to support. (TRANS-INHIBIT-1, TRANS-INHIBIT-6, TRANS-INHIBIT-7).

8.3.12.2 Test Cases: Late/Recovered Intermediate Node

A late or recovered intermediate ETBN can typically be included in the inaugurated topology even when InaugInhibition is TRUE (NodeRecovery in Figure 33 [2-5]). The exceptions are when the late ETBN belongs to an unknown consist, or when the late or recovered ETBN of some other reason cannot compute the etbTopoCnt corresponding to the acknowledged topology.

Case 1: Late “known” intermediate node (in redundant consist)

PICS: TRANS-INHIBIT-4 (late node).

Test Setup: TTS2 (Figure 13) with ETBNs A, B1, B2, C

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)

- ETBNs B1 and B2 (SUT) as redundant ETBNs/single CN (8.3.2.5)
- ETBN C (SUT) as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up A, B1, C (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN C, which has ETB port(s) in DISCARDING (extremity).
- 3) Bring up B2 (redundant intermediate ETBN)
 - Verify that B2 is able to inaugurate and compute correct etbTopoCnt (same as before). Also verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN C, which has ETB port(s) in DISCARDING (extremity).

Acceptance criteria: Successful verification as described in test steps.

Case 2: Recovery of single ETBN

PICS: TRANS-INHIBIT-4 (recovered node).

Test Setup: TTS2 (Figure 13) with ETBNs A, B, C, D

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBNs B, C and D (SUT) each as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up A, B, C, D (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
- 3) Power off C.
- 4) Power on C
 - Verify that C is able to inaugurate and compute correct etbTopoCnt (same as before). Also verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).

Acceptance criteria: Successful verification as described in test steps.

Case 3: Recovery of multiple intermediate consists (recovered consists, complete or incomplete)

PICS: TRANS-INHIBIT-4 (recovered node), TRANS-INHIBIT-5.

Test Setup: TTS2 (Figure 13) with ETBNs A, B1, B2, C, D

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBNs B1 and B2 (SUT) as redundant ETBNs/single CN (8.3.2.5)
- ETBN C and (SUT) each as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up all ETBNs (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.

- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN C, which has ETB port(s) in DISCARDING (extremity).
- 3) Power off B2 and C.
- 4) Bring up B2

Node B2 is expected to end up in state “not inaugurated”, computing a different etbTopoCnt as it does not “know” about C. Still, the test does not prohibit implementations where B2 do reach inaugurated state with same etbTopoCnt as other nodes (it could use opportunistic methods based on stored information).

 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
 - Verify that B2 either stays in state “not inaugurated” with different etbTopoCnt than the rest, or that B2 reaches “inaugurated” state with same etbTopoCnt as the rest.
- 5) Power on C
 - Verify that C and B2 are able to inaugurate and compute correct etbTopoCnt (same as before). Also verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).

Acceptance criteria: Successful verification as described in test.

Case 4: Late “unknown” intermediate node (late consist)

PICS: TRANS-INHIBIT-5.

Test Setup: TTS2 (Figure 13) with ETBNs A, B, C, D

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBNs B, C and D (SUT) each as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up A, B, D (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
- 3) Power on C
 - Verify that C stays in state “not inaugurated” (advertising different etbTopoCnt than the others) Also verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).

Acceptance criteria: Successful verification as described in test steps.

Case 5: Recovery or late ETBN when “unknown intermediate” is present

PICS: TRANS-INHIBIT-5.

Test Setup: TTS2 (Figure 13) with ETBNs A, B1, B2, C, D

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBNs B1 and B2 (SUT) as redundant ETBNs/single CN (8.3.2.5)
- ETBNs C and D (SUT) each as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up A, B1, B2, D (inhibition false)
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.

- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
- 3) Power on C
 - Verify that C stays in state “not inaugurated” (advertising different etbTopoCnt than the others) Also verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
- 4) Power off B2
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
- 5) Power on B2

Node B2 is expected to end up in state “not inaugurated”, computing a different etbTopoCnt due to the presence of ETBN C. Still, the test does not prohibit implementations where B2 do reach inaugurated state with same etbTopoCnt as other nodes (it could use opportunistic methods based on stored information or disregard C of other reasons).

 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity)
 - Verify that B2 either stays in state “not inaugurated” with different etbTopoCnt than the rest, or that B2 reaches “inaugurated” state with same etbTopoCnt as the rest.

Acceptance criteria: Successful verification as described in test.

8.3.12.3 Test Cases: Late Recovered End-node(s)

This test covers functionality described in 8.1.1.3, and it is optional to support inclusion of late or recovered end-nodes (TRANS-INHIBIT-6). Dedicated test for TRANS-INHIBIT-7 is missing (it is tested implicitly in some of the listed test, but that assumes support for TRANS-INHIBIT-6 too).

Case 1: Late “known” end node (in redundant consist)

PICS: TRANS-INHIBIT-6:

Test Setup: TTS2 (Figure 13) with ETBNs: A, B, C1, C2

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBN B (SUT) as single ETBN/CN (8.3.2.2)
- ETBNs C1 and C2 (SUT) as redundant ETBNs/single CN (8.3.2.5)

Test steps

- 1) Bring up A, B, and C1
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN C1, which has ETB port(s) in DISCARDING (extremity).
- 3) Power on C2 (redundant “external” ETBN)
 - Verify that C2 is included in the inaugareted trainset and compute correct etbTopoCnt (as the others). Also verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN C2, which has ETB port(s) in DISCARDING (extremity).

Acceptance criteria: Successful verification as described in test scan steps.

Case 2: Recovered "known" end node (in non-redundant consist)

PICS: TRANS-INHIBIT-6.

Test Setup: TTS2 (Figure 13) with ETBNs: A, B, C, D

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBN B, C, and D (SUT), each as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up all ETBNs
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the "right" side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
- 3) Power off D
 - Verify that C becomes the new extremity node, with "right" ETB port(s) in DISCARDING mode, while other SUT ETB ports are in FORWARDING.
- 4) Power on D
 - Verify that D is included in the inaugurated trainset and can compute correct etbTopoCnt (as the others). Also verify that all SUT ETB ports are in FORWARDING, except the "right" side of ETBN D, which has ETB port(s) in DISCARDING (extremity).

Acceptance criteria: Successful verification as described in test steps.

Case 3: Multiple recovered end consists (complete or incomplete)

PICS: TRANS-INHIBIT-6.

Test Setup: TTS2 (Figure 13) with ETBNs: A, B, C, D

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBN B, C, and D (SUT), each as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up all ETBNs
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the "right" side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
- 3) Power off C and D
 - Verify that B becomes the new extremity node, with "right" ETB port(s) in DISCARDING mode, while other SUT ETB ports are in FORWARDING.
- 4) Power on C
 - Verify that B stays as extremity with its "right" ETB port(s) in DISCARDING. C is "outside" with its own etbTopoCnt.
- 5) Power on D
 - Verify that C and D is included in the inaugurated trainset and can compute correct etbTopoCnt (as the others). Also verify that all SUT ETB ports are in FORWARDING, except the "right" side of ETBN D, which has ETB port(s) in DISCARDING (extremity).

Acceptance criteria: Successful verification as described in test steps.

Case 4: Recovered end consist during train lengthening

This is not supposed to work, i.e., it should not be possible to include a recovered end-node during lengthening, i.e., when additional ETBNs are present exchanging TOPOLOGY messages (inhibition “false”). Thus, the test aims to verify that extremity node keeps its port(s) in DISCARDING.

PICS: TRANS-INHIBIT-6.

Test Setup: TTS2 (Figure 13) with ETBNs: A, B, C, D

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBN B, C, and D (SUT), each as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up A, B, C
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.
- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN C, which has ETB port(s) in DISCARDING (extremity).
- 3) Power off C
 - Verify that B becomes the new extremity node, with “right” ETB port(s) in DISCARDING mode, while other SUT ETB ports are in FORWARDING.
- 4) Power on D (lengthening)

D has inhibition “False” such that its TOPOLOGY messages can reach B. (There are other lengthening cases where two inaugurated and inhibited trainsets are connected; the behaviour in that scenario would differ and is not the subject in this test.).

 - Verify that B stays as extremity with its “right” ETB port(s) in DISCARDING. D is “outside” with its own etbTopoCnt.
- 5) Power on C

Now B should keep its extremity port(s) in DISCARDING. It cannot expect C to compute a correct etbTopoCnt as long as there are other ETBNs (D) present.

 - Verify that B stays as extremity with its “right” ETB port(s) in DISCARDING. C and D is “outside” with other etbTopoCnt.

Acceptance criteria: Successful verification as described in test steps.

Case 5: Recovered end consist with lost intermediate consist

This is not supposed to work, i.e., it should not be possible to include a recovered end-node when there is a consist missing in the internal topology. The current end-node should detect that there is a mismatch in the internal topology and keep the extremity port in DISCARDING, as the recovering ETBN would not be able to compute the same etbTopoCnt. Thus, the test aims to verify that extremity node keeps its port(s) in DISCARDING.

PICS: TRANS-INHIBIT-6.

Test Setup: TTS2 (Figure 13) with ETBNs: A, B, C, D

- ETBN A simulated by test system, single ETBN/CN (8.3.2.2)
- ETBN B, C, and D (SUT), each as single ETBN/CN (8.3.2.2)

Test steps:

- 1) Bring up A, B, C, D
 - Verify nodes reach inaugurated state with same etbTopoCnt, and that all ETB ports are in forwarding state.

- 2) Set inhibition true on A (test system)
 - Verify that all SUT ETB ports are in FORWARDING, except the “right” side of ETBN D, which has ETB port(s) in DISCARDING (extremity).
 - 3) Power off D
 - Verify that C becomes the new extremity node, with “right” ETB port(s) in DISCARDING mode, while other SUT ETB ports are in FORWARDING.
 - 4) Power off B (lost intermediate consist)
 - Verify that C stays as extremity node, with “right” ETB port(s) in DISCARDING mode, while other SUT ETB ports are in FORWARDING.
 - 5) Power on D

Now C should keep its extremity port(s) in DISCARDING. It cannot expect D to compute a correct etbTopoCnt as long as there are internal consists (B) missing.

 - Verify that C stays as extremity with its “right” ETB port(s) in DISCARDING. D is “outside” with other etbTopoCnt.
- Acceptance criteria: Successful verification as described in test steps.

9 Conformance test of TRDP

9.1 General

This clause defines conformance requirements for conformance testing for Train Real-Time Data Protocol (TRDP) as given in IEC 61375 2-3.

Those requirements are used to derive test suites and test cases in order to determine whether a system, in this case, an ETBN running TRDP, complies with the requirements defined for TRDP.

Figure 47 gives an overview about the process to test system running TRDP.

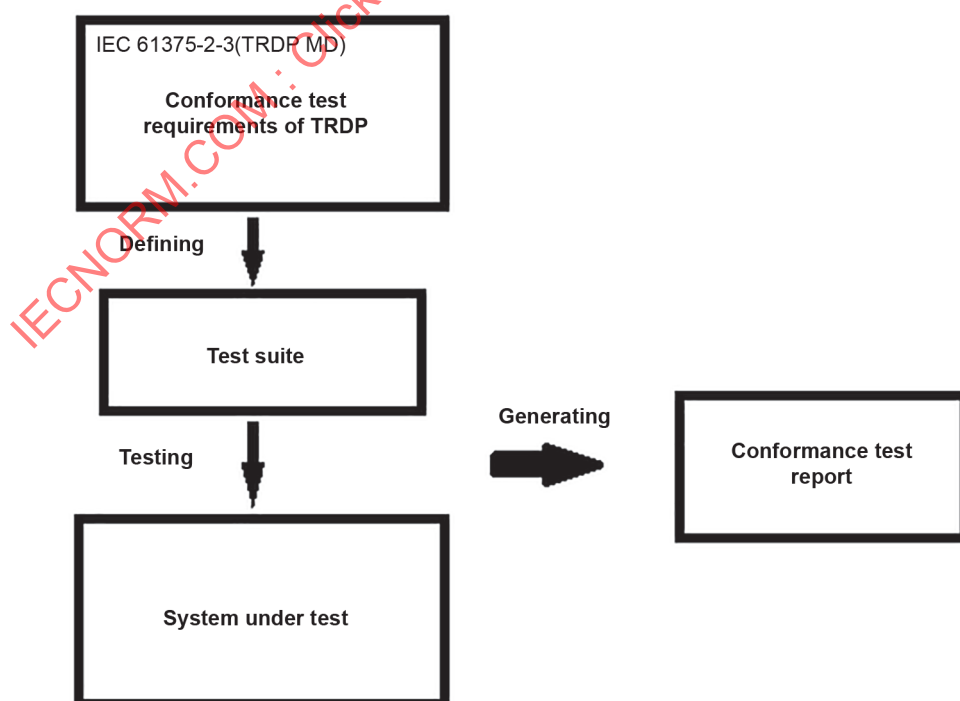


Figure 47 – Test environment

9.2 Related requirements

As given in IEC 61375 2-3:2015.

9.3 SUT and Test scope

Figure 48 shows a consist network with various ETBN and consist switch devices. The Testing Laptop shall be able to communicate with the switches and test the correctness of running TRDP service.

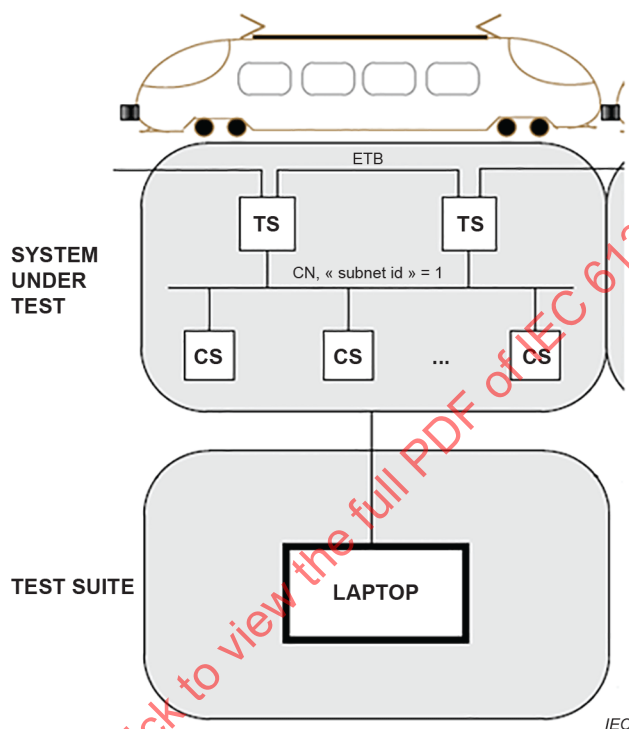


Figure 48 – System under Test

9.4 PICS Pro-forma

9.4.1 General

The PICS provides a statement of the capabilities and options which have been implemented, and any features which have been omitted, see 4.1.3.1. Instructions about filling the PICS pro-forma are given in Annex B.

9.4.2 PICS Tables

9.4.2.1 Identification

Reference	Question	Response
TRDP_ID_001	Date of statement	
TRDP_ID_002	PICS serial number	

9.4.2.2 Implementation under test

Reference	Question	Requirement	Response
TRDP_IUT_001	implementation name	M	
TRDP_IUT_002	Version number	M	
TRDP_IUT_003	Special configuration	O	
TRDP_IUT_004	Power supply voltage	M	
TRDP_IUT_005	Power supply current	M	
TRDP_IUT_006	Other information	O	
NOTE 1 Implementation name refers to the identifier of the IUT as indicated by the client. The specific conformance test is applied to the entity identified by the implementation name. NOTE 2 This is the version number of the IUT. When a version number is defined for an IUT, no subsystem which composes it can progress without a change of this figure (the architecture is frozen and constitutes a configuration). NOTE 3 Indicated if PIXIT is provided for this IUT. NOTE 4 Indicates the applicable power supply voltage. Power supply voltage is chosen amongst the values specified by IEC 60571. NOTE 5 Indicates the applicable maximum power supply current. Power supply current is chosen amongst the values specified by IEC 60571. NOTE 6 Other information the client considers relevant for IUT identification.			

9.4.2.3 IUT supplier and/or test laboratory client

Reference	Question	Requirement	Response
TRDP_supplier_001	Organisation name	M	
TRDP_supplier_002	Contact name(s)	M	
TRDP_supplier_003	Address	M	
TRDP_supplier_004	Telephone number	M	
TRDP_supplier_005	Fax number	M	
TRDP_supplier_006	e-mail address	M	
TRDP_supplier_007	Other information	O	

9.4.2.4 Identification of the standards

Reference	Question	Response
TRDP_standards_001	Specification document title	
TRDP_standards_002	Specification document IEC reference number	
TRDP_standards_003	Specification document date of publication	
TRDP_standards_004	Specification document version number	
TRDP_standards_005	Conformance document title	
TRDP_standards_006	Conformance document number	
TRDP_standards_007	Conformance document date of publication	
TRDP_standards_008	Conformance document version number	

9.4.2.5 Global statement of conformance

Reference	Question	Requirement	Implementation
TRDP_global_001	Are all mandatory capabilities implemented?	M	<u>Yes</u> []

9.4.2.6 PICS related to general TRDP

Reference	Question	Req	Ref.	Implementation
TRDP_GEN_001	Is TRDP use for the exchange of TCN PD and TCN MD over ETB?	M	A.1	<u>Yes</u> []
TRDP_GEN_002	Is TRDP process data sent with UDP?	M	A.2.3	Yes []
TRDP_GEN_003	Is the process data destination UDP port assigned to 17224?	M	A.2.3	Yes []
TRDP_GEN_004	Is the message data destination UDP/TCP port assigned to 17225?	M	A.2.3	Yes []
TRDP_GEN_005	Is the well-known port be used for receiving any process data telegrams and for receiving UDP message data notification, request and confirm telegrams?	O	A.2.3	Yes [] No []
TRDP_GEN_006	Is a private source port different from the well-known port used for sending any process data telegrams and for sending UDP message data notification, request and confirm telegrams?	O	A.2.3	Yes [] No []
TRDP_GEN_007	Is CRC32 used according to IEEE 802.3?	M	A.3	Yes []
TRDP_GEN_008	Is the CRC calculation done on the data prepared for transmission – big endian format and byte alignment?	M	A.3	Yes []
TRDP_GEN_009	Is the CRC itself appended in little endian format?	M	A.3	Yes []
TRDP_GEN_010	Is the comlds 1..999 not used by the application?	M	A.3	Yes []
TRDP_GEN_011	Comd in the header of each PDU	M	A.5	Yes []

9.4.2.7 PICS related to Process Data

Reference	Question	Req	Ref.	Implementation
TRDP_PD_001	Is PD-PDUs cyclically transmitted or transmitted on request between a publisher and one or many subscribers using a connectionless and unconfirmed TRDP service?	M	A.6.1	Yes []
TRDP_PD_002	Does Process data exchange support the following push communication pattern as defined in IEC 61375-1? a) point to point, cyclic without acknowledge, source knows the sink b) point to multipoint, cyclic without acknowledge, source knows the sink, e.g. redundancy groups c) point to multipoint, cyclic without acknowledge, source does not know the sink	O	A.6.3.1	Yes []
TRDP_PD_003	Does Process data exchange support the following pull communication pattern as defined in IEC 61375-1? – Point to point, without acknowledge, sink knows the source – Multipoint to point, without acknowledge, sink does not know the source – Point to multipoint, without acknowledge, sink knows the source. Here, one dedicated subscriber is requesting the known publisher to send its PD-PDU. – Multipoint to multipoint, without acknowledge, sink does not know the source. Here, one dedicated subscriber is requesting one or multiple unknown publisher to send their PD-PDU.	O	A.6.3.2	Yes []
TRDP_PD_004	Does a publisher/subscriber use an IP unicast address for addressing a known subscriber/publisher?	M	A.6.4	Yes []
TRDP_PD_005	Does a publisher/subscriber use an IP multicast address for addressing unknown subscribers/publishers?	M	A.6.4	Yes []
TRDP_PD_006	Does a publisher/subscriber use an IP multicast address for addressing groups of known subscribers/publishers?	M	A.6.4	Yes []
TRDP_PD_007	Is the structure of a PD-PDU defined in Figure A.11 of IEC 61375-2-3:2015 and in ASN.1 notation (additional explanation in Table A.3 of IEC 61375-2-3:2015)?	M	A.6.5	Yes []

Reference	Question	Req	Ref.	Implementation
TRDP_PD_008	Does the TRDP layer provide the service primitives PD.publish/PD.unPublish/PD.putData/PD.activateRed/PD.deactivateRed/PD.request/PD.subscribe/PD.unsubscribe/PD.indicate/PD.poll(defined in Table A.4 of IEC 61375-2-3:2015) to the TRDP user?	M	A.6.6.1	Yes []
TRDP_PD_009	Does the TRDP user have two possibilities to retrieve PD: either via a poll mechanism, typically used in cyclic user tasks, or via indication mechanism, where the TRDP layer notifies the user when new data are available or when there is a timeout?	M	A.6.6.2	Yes []
TRDP_PD_010	Are process data prepared cyclically by the publisher and given to the TRDP layer calling the PD.putData primitive?	M	A.6.6.3	Yes []
TRDP_PD_011	Does the publisher subscribe for it to receive a request?	M	A.6.6.3	Yes []
TRDP_PD_012	Is the related data in the receive buffer of the replier marked as invalid until receiving the first request telegram matching to the filter criteria of the subscription?	M	A.6.6.3	Yes []
TRDP_PD_013	Is an incoming request telegram discarded when the parameter values 'etbTopoCnt' and 'opTrnTopoCnt' different to expected (own locally stored) values?	M	A.6.6.3	Yes []
TRDP_PD_014	Does the TRDP layer using the available process data to response each incoming request?	M	A.6.6.3	Yes []
TRDP_PD_015	Is the source IP address of the request telegram used as destination IP address for the reply if the reply IP address of the request telegram is 0?	M	A.6.6.3	Yes []
TRDP_PD_016	Does the requester needs to subscribe for it to receive the related reply for a request?	M	A.6.6.3	Yes []
TRDP_PD_017	Is related data in the receive buffer of the subscriber marked as invalid until receiving the first reply telegram matching to the filter criteria of the subscription?	M	A.6.6.3	Yes []
TRDP_PD_018	Is the related reply data in the receive buffer of the subscriber set to invalid before sending the request telegram?	M	A.6.6.3	Yes []
TRDP_PD_019	Is the timeout supervision at subscriber TRDP layer restarted after sending the request telegram?		A.6.6.3	Yes []

Reference	Question	Req	Ref.	Implementation
TRDP_PD_020	Before sending out a request, does the TRDP layer check the topology counters submitted with the request against the actual topology counters?	M	A.6.6.3	Yes []
TRDP_PD_021	Are process data prepared cyclically by the publisher and given to the TRDP layer calling the PD.putData primitive?	M	A.6.6.4	Yes []
TRDP_PD_022	Does the publisher TRDP layer send the data in the configured cycle to the configured address?	M	A.6.6.4	Yes []
TRDP_PD_023	Does the publisher TRDP layer send out the data only after checking the locally stored topography counters submitted with the publish against the actual topography counters and at least one of the cases listed in Table A.5 for the topography counters be fulfilled?	M	A.6.6.4	Yes []
TRDP_PD_024	Does any subscriber subscribe for the process telegram using ComID, destination IP address and source IP address of the process telegram as possible filter criteria?	O	A.6.6.4	Yes[] No []
TRDP_PD_025	Is timeout supervision at subscriber TRDP layer started after subscription?	M	A.6.6.4	Yes []
TRDP_PD_026	Is timeout supervision at subscriber TRDP layer restarted after receiving the related PD PDU?	M	A.6.6.4	Yes []
TRDP_PD_027	Does the subscriber TRDP layer check the topography counters of the received telegram against the actual topography counters and against the topography counters submitted with the subscription and at least one of the cases listed in Table A.5 for the topography counters be fulfilled?	M	A.6.6.4	Yes []
TRDP_PD_028	Is the data marked as invalid until receiving the first valid telegram matching to the filter criteria?	M	A.6.6.4	Yes []
TRDP_PD_029	Are the service primitives for redundancy handling used in the same way by the publisher for the pull and the push pattern?	M	A.6.6.5	Yes []
TRDP_PD_030	Does a redundant device call PD.activate to start publishing process data related to ComIDs marked as redundant if it enters the redundancy leader state?	M	A.6.6.5	Yes []
TRDP_PD_031	Does a redundant device call PD.deactivate to stop publishing process data related to ComIDs marked as redundant if it enters the redundancy follower state?	M	A.6.6.5	Yes []

Reference	Question	Req	Ref.	Implementation
TRDP_PD_032	Are publishers of redundant process data (identified by ComId marked as redundant) initialized in redundancy follower mode (publishing deactivated) when starting TRDP ?	M	A.6.6.5	Yes []
TRDP_PD_033	Before sending a telegram, does the topography counters of the telegram be checked against the actual topography counters to ensure that the sending application shares the actual train backbone view and operational train view?	M	A.6.7	Yes []
TRDP_PD_034	After reception of a telegram, does the topography counter values be checked to ensure that caller and replier share an identical train backbone view and operational train view?	M	A.6.7	Yes []
TRDP_PD_035	Does a publisher publish PD-PDU(data) when a defined time cycle terminates (push pattern) and/or when it receives a request from a dedicated subscriber or an independent communication device?	M	A.6.8.1	Yes []
TRDP_PD_036	Does the publisher of pushed PD-PDU apply a traffic shaping mechanism for equal distribution of the PD-PDU's over the time not to overload subscribers?	M	A.6.8.1	Yes []
TRDP_PD_037	Does the publishing of PD-PDU process follow the state diagram shown in Figure A.15.?	O	A.6.8.1	Yes[] No []
TRDP_PD_038	Does one dedicated subscriber or an independent communication device request one or many publishers to send their PD-PDUs?	O	A.6.8.2	Yes[] No []
TRDP_PD_039	Does the requesting of PD-PDU process follow the state diagram shown in Figure A.16.?	O	A.6.8.2	Yes[] No []
TRDP_PD_040	Does the receiving of PD-PDU process follow the state diagram shown in Figure A.17.?	O	A.6.8.2	Yes[] No []

9.4.2.8 PICS related to Message Data

9.4.2.8.1 PICS related to Communication Model

Reference	Question	Req	Ref.	Implementation
TRDP_MD_001	Does MD support "notification" communication model ?	M	A.7.1	<u>Yes</u> []
TRDP_MD_002	Does MD support "request without confirmation" communication model?	M	A.7.1	<u>Yes</u> []
TRDP_MD_003	Does MD support "request with confirmation" communication model?	M	A.7.1	<u>Yes</u> []
TRDP_MD_004	Is the caller able to define by request type whether a reply is expected or not?	M	A.7.1	<u>Yes</u> []
TRDP_MD_005	Is the replier able to define by the reply type whether a confirmation of its reply is expected or not?	M	A.7.1	<u>Yes</u> []
TRDP_MD_006	Does TRDP provide two mechanisms to transfer MD(via UDP and TCP)?	M	A.7.1	<u>Yes</u> []
TRDP_MD_007	Are the different service primitives of UDP transfer and TCP transfer mixed?	M	A.7.1	<u>Yes</u> []

IECNORM.COM : Click to view the full PDF of IEC 61375-2-8:2021

9.4.2.8.2 PICS related to Communication Pattern and Addressing

Reference	Question	Req	Ref.	Implementation
TRDP_MD_008	Is MD exchange support the following push communication pattern defined in IEC 61375-1? a) point to point , sporadic with acknowledge, source knows the sink; b) point to point , sporadic without acknowledge, source knows the sink; c) point to multipoint, sporadic with acknowledge, source knows the sink; d) point to multipoint, sporadic without acknowledge, source knows the sink; e) point to multipoint, sporadic with acknowledge, source does not know the sink; f) point to multipoint, sporadic without acknowledge, source does not know the sink.	O	A.7.3.1	<u>Yes</u> []
TRDP_MD_009	Is MD exchange support the following pull communication pattern defined in IEC 61375-1? a) point to point , sporadic with acknowledge, sink knows the source; b) point to point , sporadic without acknowledge, sink knows the source; c) point to multipoint, sporadic with acknowledge, sink knows the source; d) point to multipoint, sporadic without acknowledge, sink knows the source; e) point to multipoint, sporadic on first acknowledge, sink does not know the source; f) point to multipoint, sporadic without acknowledge, sink does not know the source.	O	A.7.3.2	<u>Yes</u> []
TRDP_MD_010	Does MD caller use an IP unicast address or an IP multicast address for addressing known repplier(s)?	M	A.7.4	<u>Yes</u> []
TRDP_MD_011	Does MD caller use an IP multicast address for addressing unknown repliers?	M	A.7.4	<u>Yes</u> []
TRDP_MD_012	Does MD repplier respond to the caller's unicast address?	M	A.7.4	<u>Yes</u> []

9.4.2.8.3 PICS related to MD-PDU

Reference	Question	Req	Ref.	Implementation
TRDP_MD_013	Is the structure of a MD-PDU defined in Figure A.19 of IEC 61375-2-3:2015 and subsequently in ASN.1 notation (additional explanation in Table A.18 of IEC 61375-2-3:2015)?	M	A.7.5	<u>Yes</u> ☐

9.4.2.8.4 PICS related to TRDP Layer Service Primitives

Reference	Question	Req	Ref.	Implementation
TRDP_MD_014	Does the caller provide MD.request/MD.indicate/MD.confirm/MD.abort services primitive to TRDP user?	M	A.7.6.1	<u>Yes</u> ☐
TRDP_MD_015	Does the replier provide MD.addListener/MD.updateListener/MD.removeListener/MD.indicate/MD.reply/MD.release services primitive to TRDP user?	M	A.7.6.1	<u>Yes</u> ☐

9.4.2.8.5 PICS related to TRDP Layer Filtering Rules

Reference	Question	Req	Ref.	Implementation
TRDP_MD_016	Does the service primitive MD.addListener allows to define SourceURI and Destination URI for filtering received MD telegrams according the following rules?	O	A.7.6.3	<u>Yes</u> ☐ <u>NO</u> ☐ ?

9.4.2.8.6 PICS related to Caller TRDP Layer

Reference	Question	Req	Ref.	Implementation
TRDP_MD_019	Is the message discarded If none of the cases listed in Table A.21 of IEC 61375-2-3:2015 is TRUE when caller requests sending a message and check the topography counters?	M	A.7.8.1	<u>Yes</u> ☐
TRDP_MD_020	Is the session identifier of the notification message set to 0?	M	A.7.8.1	<u>Yes</u> ☐
TRDP_MD_021	Is each request-reply/request-reply-confirm session identified by a 16 byte UUID according to RFC 4122, time based version to ensure that the session identifier is unique?	M	A.7.8.1	<u>Yes</u> ☐
TRDP_MD_022	Is The session identifier calculated at caller side in the TRDP layer, transmitted within each message and used at caller and replier side to identify the related caller and replier session?	M	A.7.8.1	<u>Yes</u> ☐
TRDP_MD_023	Is a timeout value for the reply message(s) defined by the TRDP user (caller)?	M	A.7.8.1	<u>Yes</u> ☐
TRDP_MD_024	Does the TRDP layer wait for incoming reply messages (MsgType 'Mp', 'Mq' or 'Me') ?	M	A.7.8.1	<u>Yes</u> ☐

Reference	Question	Req	Ref.	Implementation
TRDP_MD_025	Are all incoming reply messages related to the caller session (identified by the received session id) given immediately to the TRDP user?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_026	Is the caller session closed if the number of expected incoming replies is reached and there are no more outstanding confirmations from TRDP user (caller)?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_027	Does the TRDP layer start a timer with the given confirm timeout time of the reply and wait for the confirmation from the TRDP user (caller) If the replier requests a confirmation MsgType 'Mq'?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_028	Does the confirmation use as destination URI the source URI received in the reply message?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_029	Does the TRDP layer send a confirmation message (MsgType 'Mc') to the replier after getting the confirmation from TRDP user (caller)?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_030	Does a confirmation message not contain user data and be sent only as unicast?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_031	Does the TRDP user (caller) take care to provide the confirmation in time as indicated by the ReplyTimeOut parameter of the reply?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_032	Is the TRDP user (caller) notified that confirmations are missing and the caller session be closed if the number of incoming replies is reached and the confirm timeout timer expires while waiting for outstanding confirmations from TRDP user (caller),	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_033	Does the TRDP layer, depending on the given parameter value 'MaxNumRetries', repeat the request up to two times before it notifies the TRDP user (caller) about the missing reply and closes the caller session if the reply timeout timer of the caller session expires because of a missing reply and the number of expected repliers is 1?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_034	Does the TRDP Layer notify its TRDP user (caller) about the timeout and the number of missing replies if the reply timeout timer expires and the number of expected repliers is greater than 1 and less than the expected replies have been received?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_035	Is the caller session closed when all requested confirmations are sent or the confirmation timeout timer has timed out?	M	A.7.8.1	<u>Yes</u> []

Reference	Question	Req	Ref.	Implementation
TRDP_MD_036	Is the total number of replies indicated to the TRDP user (caller) and the caller session closed if the number of repliers is not known (parameter NoOfRepliers = 0 and the reply timeout timer expires)?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_037	Is no retransmission used of request messages if TCP is used?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_038	Does TRDP provide the interface to continue a transmission after the connection was lost for TCP used?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_039	Is the message discarded if the TRDP layer receives a reply message (MsgType 'Mp', 'Mq' or 'Me') without having opened a caller session for the indicated session id?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_040	Is the topography counter values obtained during caller session opening used throughout the session (for request and confirmation messages)?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_041	Is the message discarded if a TRDP user receives a reply message with topography counter values different to the expected ones?	M	A.7.8.1	<u>Yes</u> []
TRDP_MD_042	Is the session identifier destroyed and the session closed and running timers stopped if a TRDP user aborts an (open) session (e.g. after a train topology change)?	M	A.7.8.1	<u>Yes</u> []

9.4.2.8.7 PICS related to Replier TRDP Layer

Reference	Question	Req	Ref.	Implementation
TRDP_MD_043	Does each TRDP user (replier) that wants to receive MD register as listener for MD sent to a specific URI (multicast or unicast) or for MD of a specific ComId?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_044	Are all incoming messages checked against the actual topography counters?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_045	Are all incoming reply messages related to the caller session (identified by the received session id) given immediately to the TRDP user?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_046	Are all incoming messages checked against registered listeners?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_047	Is any message to a not registered listener or to a listener expecting another value of the topography counters discarded?	M	A.7.8.2	<u>Yes</u> []

Reference	Question	Req	Ref.	Implementation
TRDPMMD_048	Is an error message (MsgType = 'Me') sent indicating the error with replyStatus == -3 (no replier instance) in case of a unicast request message to a not registered listener or to a listener expecting another value of the topography counters?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_049	Is the message passed to the related listener if a notification message (MsgType 'Mn') is received,	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_050	Is a timeout timer with the received reply timeout started and the message passed to the related listener and the TRDP layer wait for the MD.reply of the listener if a request message (MsgType 'Mr') is received, then a reply session opened using the received session id?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_051	Is the received message discarded and an error message (MsgType = 'Me') sent, indicating the error with replyStatus == -4 (no memory (at replier side)) due to the lack of resources, a reply session can't be opened?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_052	Is the request message discarded if a request message (MsgType 'Mr') is received for an open reply session with the received session id while waiting for the MD.reply of the listener,	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_053	Does the TRDP layer send, depending on MD.reply of the listener, a reply message without confirmation (MsgType = 'Mp') or a reply message with confirmation (MsgType = 'Mq') to the caller after receiving the MD.reply from the TRDP user (listener)?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_054	Is a reply message sent only as unicast using source IP address and source URI of the related request message as destination IP address and destination URI?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_055	Does the MD.reply of a listener use as source IP address the IP address of the listener and as source URI the unique URI (user part) of the listener?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_056	Is the reply session closed If a reply message without confirmation (MsgType = 'Mp') was sent?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_057	Is the timeout timer of the reply session restarted using the given confirm timeout value if a reply message with confirmation (MsgType = 'Mq') was sent?	M	A.7.8.2	<u>Yes</u> []

Reference	Question	Req	Ref.	Implementation
TRDP_MD_058	Does the TRDP layer repeat sending the reply message (MsgType = 'Mq') to the caller with an incremented sequence counter if a request message (MsgType 'Mr') is received with a different sequence counter for an already open reply session, after the TRDP layer has received the MD.reply from the TRDP user (replier)?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_059	Is the timeout timer of the reply session restarted using the given confirm timeout value after resending the reply message?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_060	Is the request message discarded if a request message (MsgType 'Mr') is received for an already open reply session and the same sequence counter like received before?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_061	Is the related listener notified and the session closed if a confirmation message (MsgType 'Mc') is received for an open reply session with the received session id?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_062	Is the message discarded if a confirmation message (MsgType 'Mc') is received without having an open reply session with the received session id?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_063	Is the reply session closed if the timeout timer of a reply session expires?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_064	Is the listener notified in case of a timeout because of a missing confirmation or a missing MD.reply of the listener?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_065	Are the topography counter values obtained in an MD.addListener service primitive used for upcoming reply sessions unless they are updated by the user (MD.updateListener service primitive)?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_066	Is the message discarded if a TRDP user receives a request message with topography counter values different to the expected ones (see Table A.21 of IEC61375-2-3)?	M	A.7.8.2	<u>Yes</u> []
TRDP_MD_067	Are running timers stopped, the session identifier destroyed and the session closed if a TRDP user removes or updates a listener during an open session?	M	A.7.8.2	<u>Yes</u> []

9.4.2.8.8 PICS related to TCP Connection Handling

Reference	Question	Req	Ref.	Implementation
TRDP_MD_068	Does the first TCP connection between the caller and the replier device request open a connection if there was not yet a connection established before?	M	A.7.9	<u>Yes</u> []
TRDP_MD_069	Does the caller close an existing TCP connection (active end) in the following cases? – A signal that the TCP connection will be closed has been received. – TRDP shut down or re-initialization. – A timeout occurred because the TCP connection has not been used for a defined time.	M	A.7.9	<u>Yes</u> []
TRDP_MD_070	Does the replier use the connection opened by the caller?	M	A.7.9	<u>Yes</u> []
TRDP_MD_071	Does the replier close an existing TCP connection (passive end) in the following cases? – A signal that the TCP connection will be closed has been received. – TRDP shut down or re-initialization. – Another TCP connection was opened from the same caller device and the old connection is not used anymore for a defined time.	M	A.7.9	<u>Yes</u> []

9.4.2.8.9 PICS related to Message Data Echo Server

Reference	Question	Req	Ref.	Implementation
TRDP_MD_072	Does end devices with TRDP message data support provide a message data echo function?	O	A.8	<u>Yes</u> []
TRDP_MD_073	Is the Message Data Echo implemented in the TRDP layer?	O	A.8	<u>Yes</u> []
TRDP_MD_074	Is the user data of the returned message a copy of the received user data?	O	A.8	<u>Yes</u> []
TRDP_MD_075	Is an echo message created and returned to the source when message data arrive with the echo ComId?	O	A.8	<u>Yes</u> []

9.5 Message Data test**9.5.1 Testing framework**

A testing system with Ethernet connection to the ETBN shall be used. The TRDP library is integrated in the ETBN control software. The testing system needs to be able to send/receive UDP/TCP telegrams to/from the ETBN.

It is necessary to have an "Ethernet bus analyzer tool" to test the communication between the testing system and the ETBN. "Ethernet bus analyzer tool" could be defined as a software tool to capture Ethernet frames and dissect and analyze them. For example: "Wireshark".

9.5.2 Communication Model Test

9.5.2.1 Test purpose

Test the correctness of TRDP Message Data communication model. The communication model is formed by a Caller and a Replier.

9.5.2.2 TRDP Message Data supports notification (request without reply)

IDENTIFIER	TC23TRDPM01
TITLE	TRDP Message Data supports notification
DESCRIPTION	The next test case will ensure that: The testing system receives correctly notification messages of the ETBN.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are received from port=17225. Step 4: Check that MsgType field of received message is equal to '4D6E'H ('Mn'), "Request without reply" telegram type. Step 5: Check that no reply messages are sent.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A frame is received with correct MsgType (notification) identifier. As it is a notification message, no reply message is sent.

9.5.2.3 TRDP Message Data supports request without confirmation (request with reply but without confirmation)

IDENTIFIER	TC23TRDPM02
TITLE	TRDP Message Data supports request without confirmation
DESCRIPTION	The next test case will ensure that: The testing system sends a request and expects a reply, but without a confirmation
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that a request message is sent to the ETBN from port=17225. Step 4: Check that MsgType identifier is equal to '4D72'H ('Mr'), "Request with Reply" datagram type. Step 5: Check that a reply message is received back from the ETBN, port=17225. Step 6: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 7: Check that no confirmation message is sent to the ETBN.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request with reply datagram is sent to the ETBN and a reply without confirmation is received back. As no confirmation message is expected, no confirmation message is sent.

9.5.2.4 TRDP Message Data supports request with confirmation (request with reply and confirmation)

IDENTIFIER	TC23TRDPMMD_003
TITLE	TRDP Message Data supports request with confirmation
DESCRIPTION	The next test case will ensure that: The testing system sends a request and expects a reply. A confirmation message is needed for a successful communication.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that a request message is sent to the ETBN from port=17225. Step 4: Check that MsgType identifier is equal to '4D72'H ('Mr'), "Request with Reply" datagram type. Step 5: Check that a reply message is received back from the ETBN, port=17225. Step 6: Check that MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 7: Check that a confirmation message is sent to the ETBN, port=17225. Step 8: Check that MsgType identifier is equal to '4D63'H ('Mc'), "confirm" datagram type.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request with reply datagram is sent to the ETBN and a reply with confirmation datagram is received back. A confirmation message is sent to the ETBN to close the communication.

9.5.2.5 TRDP Message Data provides UDP mechanisms to transfer MD

IDENTIFIER	TC23TRDPM04
TITLE	TRDP Message Data provides UDP mechanisms to transfer MD
DESCRIPTION	The next test case will ensure that: The testing system receives correctly notification messages of the ETBN via UDP.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	ETBN and testing system shall be prepared for UDP communication. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus on UDP frames. Step 3: Check that message data frames are received from port=17225. Step 4: Check that MsgType field of received message is equal to '4D6E'H ('Mn'), "Request without reply" telegram type, and that it is an UDP frame.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A notification UDP frame is received from the ETBN.

9.5.2.6 TRDP Message Data provides TCP mechanisms to transfer MD

IDENTIFIER	TC23TRDPM05
TITLE	TRDP Message Data provides TCP mechanisms to transfer MD
DESCRIPTION	The next test case will ensure that: The testing system receives correctly notification messages of the ETBN via TCP.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	ETBN and testing system shall be prepared for TCP communication. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus on TCP frames. Step 3: Check that message data frames are received from port=17225. Step 4: Check that MsgType field of received message is equal to '4D6E'H ('Mn'), "Request without reply" telegram type, and that it is a TCP frame.
FINAL CONDITIONS	
ACCEPTANCE CRITERIA	The result of test is successful if: A notification TCP frame is received from the Switch.

9.5.2.7 TRDP UDP transfer and TRDP TCP transfer not mixed

IDENTIFIER	TC23TRDPMD_006
TITLE	TRDP UDP transfer and TRDP TCP transfer not mixed
DESCRIPTION	The next test case will ensure that: TRDP shall provide two mechanisms to transfer message data (via UDP and via TCP); the different service primitives of the two possibilities shall not be mixed.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	ETBN shall be prepared for TCP communication, and testing system shall be prepared for TCP communication. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus on TCP frames. Step 3: Check that message data frames are received from port=17225. Step 4: Check that a request message is sent to the ETBN from port=17225. Step 5: Check that MsgType identifier is equal to '4D72'H ('Mr'), "Request with Reply" datagram type. Step 6: Check that it is a TCP frame type. Step 7: Check that TRDP user level at ETBN is not notified about the reception of a TCP message. Step 8: Check that ETBN does not send a reply message. Step 9: Check that no retransmission of request message is done (as TCP configuration is used in the testing system). Step 10: Check that TRDP user at testing system is notified about that retransmission is not done.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: There is no retransmission of request message as it is part of UDP communication type, and primitives of different protocols are not mixed.

9.5.3 Communication Pattern and Addressing Test

9.5.3.1 Test purpose

Test the correctness of TRDP Message Data communication pattern. TRDP data exchange supports push and pull communication patterns.

In Push communication pattern, the ETBN(Caller) is the source of the message, and the Testing System(Replier) is the sink of the message data.

In Pull communication pattern, the Testing System(Caller) is the sink of the message, and the ETBN(Replier) is the source of the message data.

9.5.3.2 Point to Point, sporadic with acknowledge (reply), source knows the sink

IDENTIFIER	TC23TRDPM_D_007
TITLE	Point to Point, sporadic with acknowledge (reply), source knows the sink
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports push communication patterns, point to point, sporadic with acknowledge (reply), where source knows the sink.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and an ETBN shall be used. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are received from port=17225. Step 4: A request message is sent to the ETBN. Step 5: Check that MsgType field of sent message is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that sent DestinationURI field contains an unicast URI. Step 7: Check that destination IP address is an unicast IP address. Step 8: Check that dataSet field contains no information. Step 9: Check that a reply message is sent from the ETBN, port=17225. Step 10: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 11: Check that as DestinationURI, the source URI sent in the request message is used. Step 12: Check that the destination IP address is an unicast IP address. Step 13: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 14: Check that TRDP user is notified about the reply.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller, and a reply message is sent back, with correct fields.

9.5.3.3 Point to Point, sporadic without acknowledge (reply), source knows the sink

IDENTIFIER	TC23TRDPM_D_008
TITLE	Point to Point, sporadic without acknowledge (reply), source knows the sink
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports push communication patterns, point to point, sporadic without acknowledge (reply), where source knows the sink.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and an ETBN shall be used. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are received from port=17225. Step 4: Check that a notification message is sent from the ETBN. Step 5: Check that MsgType is equal to '4D6E'H ('Mn'), "Request without reply" telegram type. Step 6: Check that DestinationURI field contains an unicast URI. Step 7: Check that destination IP address is an unicast IP address. Step 8: Check that dataSet field contains information related to (TTDB - train directory information request, ComId = 102). Step 9: Check that no reply message is sent back to the ETBN in port=17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A notification message is sent from the caller, with correct fields, and no reply message is sent back.

9.5.3.4 Point to Multipoint, sporadic with acknowledge (reply), source knows the sink

IDENTIFIER	TC23TRDPMMD_009
TITLE	Point to Multipoint, sporadic with acknowledge (reply), source knows the sink
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports push communication patterns, point to multipoint, sporadic with acknowledge (reply), where source knows the sink.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and two ETBNs shall be used. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are received from port=17225. Step 4: Check that a request message is sent to the ETBNs. Step 5: Check that MsgType field is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that DestinationURI field contains a multicast URI. Step 7: Check that destination IP address is a multicast IP address. Step 8: Check that dataSet field contains no information. Step 9: Check that a reply message is sent from both ETBNs, port=17225. Step 10: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 11: Check that as DestinationURI, the source URI sent in the request message is used. Step 12: Check that destination IP address is an unicast IP address. Step 13: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102).
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller, with correct fields, and a confirmation message is sent back.

9.5.3.5 Point to Multipoint, sporadic without acknowledge (reply), source knows the sink

IDENTIFIER	TC23TRDPM_010
TITLE	Point to Multipoint, sporadic without acknowledge (reply), source knows the sink
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports push communication patterns, point to multipoint, sporadic without acknowledge (reply), where source knows the sink.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and two ETBNsETBNs shall be used. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are received from port=17225. Step 4: Check that a notification message is sent from the ETBNs. Step 5: Check that MsgType of sent message is equal to '4D6E'H ('Mn'), "Request without reply" telegram type. Step 6: Check that DestinationURI field contains a multicast URI. Step 7: Check that destination IP address is a multicast IP address. Step 8: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 9: Check that no reply message is sent to the ETBNs, port=17225.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A notification message is sent from the caller, with correct fields, and no reply message is sent back.

9.5.3.6 Point to Multipoint, sporadic with acknowledge (reply), source does not know the sink

IDENTIFIER	TC23TRDPM_011
TITLE	Point to Multipoint, sporadic with acknowledge, source does not know the sink
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports push communication patterns, point to multipoint, sporadic with acknowledge (reply), where source does not know the sink.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and two ETBNs shall be used. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are received from port=17225. Step 4: Check that a request message is sent to the ETBNs. Step 5: Check that MsgType of sent messages is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that DestinationURI field contains a multicast URI. Step 7: Check that destination IP address is a multicast IP address. Step 8: Check that no information is sent in the dataSet. Step 9: Check that no reply message is sent back from the ETBNs, port=17225, as source does not know the sink (IEC 61375-1 Section: 7.2.3.3). Step 10: Check that TRDP user is notified about this event.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller, with correct fields, and as source does not know the sink, no confirmation message is sent back from the repliers.

9.5.3.7 Point to Multipoint, sporadic without acknowledge (reply), source does not know the sink

IDENTIFIER	TC23TRDPM_012
TITLE	Point to Multipoint, sporadic without acknowledge (reply), source does not know the sink
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports push communication patterns, point to multipoint, sporadic without acknowledge (reply), where source does not know the sink.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and two ETBNs shall be used. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are received from port=17225. Step 4: Check that a notification message is sent from the ETBNs. Step 5: Check that MsgType field is equal to '4D6E'H ('Mn'), "Request without reply" telegram type. Step 6: Check that DestinationURI field contains a multicast URI. Step 7: Check that destination IP address is a multicast IP address. Step 8: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 9: Check that no reply message is sent to the ETBNs, port=17225.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A notification message is received from the caller, with correct fields, and no reply message is sent back.

9.5.3.8 Point to Point, sporadic with acknowledge (confirm), sink knows the source

IDENTIFIER	TC23TRDPM_D_013
TITLE	Point to Point, sporadic with acknowledge (confirm), sink knows the source
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports pull communication patterns, point to point, sporadic with acknowledge(confirm), where sink knows the source.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and an ETBN shall be used. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType field of sent message is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that sent DestinationURI field contains an unicast URI. Step 7: Check that destination IP address is an unicast IP address. Step 8: Check that a reply message is sent back from the ETBN, port=17225. Step 9: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 10: Check that MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 11: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 12: Check that TRDP user is notified about the reply. Step 13: Check that a confirmation message is sent to the ETBN, port 17225. Step 14: Check that comId identifier is equal to 0, as it is a message data confirmation message. Step 15: Check that dataSetLength field is equal to 0, as it is an acknowledge datagram. Step 16: Check that as DestinationURI, the source URI received in the reply message is used. Step 17: Check that destination IP address is an unicast IP address.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller, a reply message is sent back, and a confirmation message closes the communication, all with correct fields.

9.5.3.9 Point to Point, sporadic without acknowledge (confirm), sink knows the source

IDENTIFIER	TC23TRDPMMD_014
TITLE	Point to Point, sporadic without acknowledge (confirm), sink knows the source
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports pull communication patterns, point to point, sporadic without acknowledge (confirm), where sink knows the source.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and an ETBN shall be used. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType field of sent message is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that sent DestinationURI field contains an unicast URI. Step 7: Check that destination IP address is an unicast IP address. Step 8: Check that a reply message is sent back from the ETBN, port=17225. Step 9: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 10: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 11: Check that as DestinationURI, the source URI sent in the request message is used. Step 12: Check that destination IP address is an unicast IP address Step 13: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 14: Check that TRDP user is notified about the replies.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller and a reply message is sent back, without a confirmation message closing the communication, all with correct fields.

9.5.3.10 Point to Multipoint, sporadic with acknowledge (confirm), sink knows the source

IDENTIFIER	TC23TRDPM_015
TITLE	Point to Multipoint, sporadic with acknowledge (confirm), sink knows the source
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports pull communication patterns, point to multipoint, sporadic with acknowledge (confirm), where sink knows the source.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and two ETBNs shall be used. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBNs. Step 5: Check that MsgType field of sent message is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that sent DestinationURI field contains a multicast URI. Step 7: Check that destination IP address is a multicast IP address. Step 8: Check that a reply message is sent back from the ETBNs, port=17225. Step 9: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 10: Check that MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 11: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 12: Check that TRDP user is notified about the replies. Step 13: Check that a confirmation message is sent to the Train Swithes, port 17225. Step 14: Check that comId identifier is equal to 0, as it is a message data confirmation message. Step 15: Check that dataSetLength field is equal to 0, as it is an acknowledge datagram. Step 16: Check that as DestinationURI, the source URI received in the reply message is used. Step 17: Check that destination IP address is an unicast IP address.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller, reply messages are sent back, and confirmation messages close the communication, all with correct fields.

9.5.3.11 Point to Multipoint, sporadic without acknowledge (confirm), sink knows the source

IDENTIFIER	TC23TRDPM_D_016
TITLE	Point to Multipoint, sporadic without acknowledge (confirm), sink knows the source
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports pull communication patterns, point to multipoint, sporadic without acknowledge (confirm), where sink knows the source.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and two ETBNs shall be used. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBNs. Step 5: Check that sent messages MsgType field is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that sent DestinationURI field contains a multicast URI. Step 7: Check that destination IP address is a multicast IP address. Step 8: Check that a reply message is sent back from the ETBNs, port=17225. Step 9: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 10: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 11: Check that as DestinationURI, the source URI sent in the request message is used. Step 12: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 13: Check that TRDP user is notified about the replies. Step 14: Check that no confirmation message is sent to the ETBNs, port 17225.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller and reply messages are sent back, without a confirmation message closing the communication, all with correct fields.

9.5.3.12 Point to Multipoint, sporadic on first acknowledge (confirm), sink does not know the source

IDENTIFIER	TC23TRDPM_017
TITLE	Point to Multipoint, sporadic on first acknowledge (confirm), sink does not know the source
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports pull communication patterns, point to multipoint, sporadic on first acknowledge (confirm), where sink does not know the source.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and two ETBNs shall be used. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBNs. Step 5: Check that MsgType field of sent messages is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that sent DestinationURI field contains a multicast URI. Step 7: Check that destination IP address is a multicast IP address. Step 8: Check that a reply message is sent back from the ETBNs, port=17225. Step 9: Check that each reply message is sent within the replyTimeout value defined in the request message. Step 10: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 11: Check that as DestinationURI, the source URI sent in the request message is used. Step 12: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 13: Check that TRDP user is only notified about the first acknowledge, other replies are ignored. Step 14: Check that no confirmation message is sent to the ETBNs, port 17225, as sink does not know the source (IEC 61375-1:2012, 7.2.4.3).
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller and reply messages are sent back, but all are ignored except the first one. No confirmation message is sent as source is unknown.

9.5.3.13 Point to Multipoint, sporadic without acknowledge (confirm), sink does not know the source

IDENTIFIER	TC23TRDPMD_018
TITLE	Point to Multipoint, sporadic without acknowledge (confirm), sink does not know the source
DESCRIPTION	The next test case will ensure that: TRDP data exchange supports pull communication patterns, point to multipoint, sporadic without acknowledge (confirm), where sink does not know the source.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A network with a testing system and two ETBNs shall be used. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBNs. Step 5: Check that MsgType field of sent messages is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 6: Check that sent DestinationURI field contains a multicast URI. Step 7: Check that destination IP address is a multicast IP address. Step 8: Check that a reply message is sent back from the ETBNs, port=17225. Step 9: Check that each reply message is sent within the replyTimeout value defined in the request message. Step 10: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 11: Check that as DestinationURI, the source URI sent in the request message is used. Step 12: Check that dataSet field contains information related to (TTDB – train directory information request, ComId = 102). Step 13: Check that TRDP user is notified about the replies. Step 14: Check that no confirmation message is sent to the ETBNs, port 17225.
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: A request message is sent from the caller and a reply message are sent back, without a confirmation message closing the communication, all with correct fields.

9.5.4 MD-PDU Test

9.5.4.1 Test purpose

Test the correctness of TRDP MD-PDU structure, as defined in Figure A.19 and Table A.18 of IEC 61375-2-3:2015.

9.5.4.2 TRDP MD-PDU structure

IDENTIFIER	TC23TRDPMD_019
TITLE	TRDP MD-PDU structure
DESCRIPTION	The next test case will ensure that: TRDP MD-PDU structure complies with specification defined in Figure A.19 and Table A.18.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that captured frame contains UINT32 type SequenceCounter field, that is 0 at start. It is returned with the reply message and it is incremented with the repetition of the message. Step 5: Check that captured frame contains VERSION type protocolVersion field. MainVersion is UINT8 higher significant octet, and SubVersion is lower significant octet. Step 6: Check that captured frame contains UINT16 type msgType field. Type of telegram could be one of the following: '4D6E'H ('Mn', Notification) '4D72'H ('Mr', Request with reply) '4D70'H ('Mp', Reply without confirmation) '4D71'H ('Mq', Reply with confirmation) '4D68'H ('Mc', Confirm) '4D65'H ('Me', error) Step 7: Check that captured frame contains UINT32 type comId field, and its value complies with values of Table A.2. Step 8: Check that captured frame contains UINT32 type etbTopoCnt field. It could be set to 0 if it is not used. Step 9: Check that captured frame contains UINT32 type opTrnTopoCnt field. It could be set to 0 if it is not used. Step 10: Check that captured frame contains UINT32 type datasetLength field. Step 11: Check that captured frame contains INT32 type replyStatus field. A value equal to 0 is used for a correct reply status, and a negative value for error status. Step 12: Check that captured frame contains UINT32[4] type sessionId field. Step 13: Check that captured frame contains UINT32 type replyTimeout field. It could be set to 0 for 'Mn', 'Mp', 'Mc' or 'Me' type of telegram. Step 14: Check that captured frame contains CHAR[32] type sourceUri field. It could be a null terminated empty string. Step 15: Check that captured frame contains CHAR[32] type destinationURI field. It could be a null terminated empty string. Step 16: Check that captured frame contains UINT32 type headerFCS field, with header checksum. Step 17: Check that captured frame contains UINT8[] dataSet. DatasetLength is the length of the array.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The structure of captured frame complies with the structure defined in Figure A.19 and Table A.18.

9.5.5 TRDP Layer Service Primitives Test

9.5.5.1 Test purpose

Test that TRDP layer provides services for message data communication to the TRDP user, callers and repliers, and that the services comply with the characteristics defined in Table A.19 and Table A.20.

9.5.5.2 The caller provides MD.request, MD.indicate, MD.confirm and MD.abort services to the TRDP user

IDENTIFIER	TC23TRDPM020
TITLE	The caller provides MD.request, MD.indicate, MD.confirm and MD.abort services to the TRDP user
DESCRIPTION	The next test case will ensure that: The caller provides MD.request, MD.indicate, MD.confirm and MD.abort services to the TRDP user
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). This processing is provided by MD.request service. Step 6: Check that a reply message is sent back from the ETBN, port=17225. Step 7: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 8: Check that MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 9: Check that as DestinationURI, the source URI sent in the request message is used. Step 10: Check that SessionId identifier of received message is equal to the identifier configured in the request message. Step 11: Check that TRDP user at the Caller, is notified about that reply is received. This process is carried out by MD.indicate process. Step 12: Check that a confirmation message is sent to the ETBN, port 17225. This processing is provided by MD.confirm service. Step 13: Check that SessionId is the same as received in the reply message. This parameter is received from reply message by MD.indicate service, and then used by MD.confirm service. Step 14: Check that no more messages are sent or received using the same SessionId, as the communication is closed by MD.abort service.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The request/reply communication process is completed successfully after receiving correctly the requested information.

9.5.5.3 The replier provides MD.addListener, MD.updateListener, MD.removeListener, MD.indicate, MD.reply and MD.release services to the TRDP user

IDENTIFIER	TC23TRDPMD_021
TITLE	The replier provides MD.addListener, MD.updateListener, MD.removeListener, MD.indicate, MD.reply and MD.release services to the TRDP user
DESCRIPTION	The next test case will ensure that: The replier provides MD.addListener, MD.updateListener, MD.removeListener, MD.indicate, MD.reply and MD.release services to the TRDP user
REQUIREMENT	Mandatory
INITIAL CONDITIONS	MD.addListener is configured in replier to accept request messages with ComId equal to 102. Test Bench powered off.

IECNORM.COM : Click to view the full PDF of IEC 61375-2-8:2021

IDENTIFIER	TC23TRDPM_D_021
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that ComId is equal to 102, asking for train directory information. Step 7: TRDP user at the replier is notified about that a request message is received. This is done by MD.addListener service. Step 8: Check that a reply message is sent back from the ETBN, port=17225. This process is done by MD.reply service in the replier, after Md.indicate service creates an event for the received request. Step 9: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 10: Check that MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 11: Check that as DestinationURI, the source URI sent in the request message is used. Step 12: Check that ComId is equal to ComId sent in request message, and that DataSet field contains information for requested ComId. Replier MD.addListener service filters by ComId. Step 13: Check that SessionId identifier of received messages is equal to the identifier configured in the request message. Step 14: Check that a confirmation message is sent to the ETBN, port 17225. Step 15: Check that no more messages are sent during this request/reply process session, because replier MD.release service receives confirmation message. Step 16: Check that a new request message is sent to the ETBN. Step 17: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 18: Check that ComId is equal to 104, asking for train directory information. Step 19: Check that a reply message is sent back from the ETBN, port=17225. This is done because old MD.remListener service filtering ComId=102 was removed with MD.remListener service, and a new listening service was created to filter ComId=104. Step 20: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 21: Check that as DestinationURI, the source URI sent in the request message is used. Step 22: Check that ComId is equal to ComId sent in request message, and that DataSet field contains information for requested ComId. Step 23: Check that SessionId identifier of received messages is equal to the identifier configured in the request message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The request/reply communication processes are completed successfully after receiving correctly the requested information.

9.5.6 TRDP Layer Filtering Rules Test

9.5.6.1 Test purpose

Test the Filtering Rules defined for MD.addListener service.

9.5.6.2 Empty DestinationURI in MD telegrams

IDENTIFIER	TC23TRDPMD_022
TITLE	Empty DestinationURI in MD telegrams
DESCRIPTION	The next test case will ensure that: TRDP listeners shall receive MD telegrams with empty DestinationURI regardless of the DestinationURI parameter they may have configured.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A TRDP listener with non-empty DestinationURI filter is configured in the ETBN. DestinationURI = "grpAll.aVeh.ICst.ICITrn.ITrn" Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that DestinationURI field of sent message is empty. Step 7: Check that a reply message is sent back from the ETBN, port=17225. Step 8: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 9: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 10: Check that as DestinationURI, the source URI sent in the request message is used. Step 11: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: An empty DestinationURI MD telegram is sent to the ETBN, and a correct reply message is sent back, even though a non-empty DestinationURI filter is configured in the listener.

9.5.6.3 Empty SourceURI in MD telegrams

IDENTIFIER	TC23TRDPMMD_023
TITLE	Empty SourceURI in MD telegrams
DESCRIPTION	The next test case will ensure that: TRDP listeners shall receive MD telegrams with empty SourceURI
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A TRDP listener with non-empty SourceURI filter is configured in the ETBN. SourceURI = "IDev.IVeh.ICst.ICITrn.ITrn" Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that SourceURI field is empty. Step 7: Check that a reply message is sent back from the ETBN, port=17225. Step 8: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 9: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 10: Check that as DestinationURI, the source URI sent in the request message is used. Step 11: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: An empty SourceURI field MD telegram is sent to the ETBN, and a correct reply message is sent back, even though a non-empty SourceURI filter is configured in the listener.

9.5.6.4 Empty DestinationURI filter entry in MD.addListener service

IDENTIFIER	TC23TRDPMMD_024
TITLE	Empty DestinationURI filter entry in MD.addListener service
DESCRIPTION	The next test case will ensure that: TRDP listeners configured with empty DestinationURI filter shall respond to MD telegrams regardless of their DestinationURI field
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A TRDP listener with empty DestinationURI filter is configured in the ETBN. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that DestinationURI field is equal to "grpAll.aVeh.ICst.ICITrn.ITrn". Step 7: Check that a reply message is sent back from the ETBN, port=17225. Step 8: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 9: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 10: Check that as DestinationURI, the source URI sent in the request message is used. Step 11: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A non-empty DestinationURI field MD telegram is sent to the ETBN, and a correct reply message is sent back, even though an empty DestinationURI filter is configured in the listener.

9.5.6.5 Empty SourceURI filter entry in MD.addListener service

IDENTIFIER	TC23TRDPMMD_025
TITLE	Empty SourceURI filter entry in MD.addListener service
DESCRIPTION	The next test case will ensure that: TRDP listeners configured with empty SourceURI filter shall respond to MD telegrams regardless of their SourceURI field
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A TRDP listener with empty SourceURI filter is configured in the ETBN. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that SourceURI field of sent message is equal to "IDev.IVeh.ICst.ICITrn.ITrn". Step 7: Check that a reply message is sent back from the ETBN, port=17225. Step 8: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 9: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 10: Check that as DestinationURI, the source URI sent in the request message is used. Step 11: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A non-empty SourceURI field MD telegram is sent to the ETBN, and a correct reply message is sent back, even though an empty SourceURI field is configured in the listener.

9.5.6.6 Multiple listeners with empty filtering parameters

IDENTIFIER	TC23TRDPMO_026
TITLE	Multiple listeners with empty filtering parameters
DESCRIPTION	The next test case will ensure that: Any two listeners configured with many empty parameters, have to differ in at least one non-empty filter parameter
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Two TRDP listeners with empty SourceURI, DestinationURI, SourceIpAddress, SourceIpAddress2 and DestinationIpAddress are configured in the ETBN. They differ in the configured ComId, 102 and 104. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType field of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that DestinationURI field of sent message is empty, and that messages differ in their ComId. Step 7: Check that ComId field of sent message is equal to 102(TTDB – train directory information request). Step 8: Check that a reply message is sent back from the ETBN, port=17225. Step 9: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 10: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 11: Check that as DestinationURI, the source URI sent in the request message is used. Step 12: Check that information for asked ComID 102 is returned. Step 13: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: Two listeners are configured in the ETBN, with empty filtering parameters, except for ComId, 102 and 104. Only one reply message is sent back as the request message is for ComId=102. Listener with ComId=102 is the one that processes the request.

9.5.6.7 Reception of MD telegrams with empty DestinationURI field by several listeners with different DestinationURI filters

IDENTIFIER	TC23TRDPMMD_027
TITLE	Reception of MD telegrams with empty DestinationURI field by several listeners with different DestinationURI filters.
DESCRIPTION	The next test case will ensure that: Several TRDP listeners configured with non-empty DestinationURI filter shall process and respond to MD telegrams with empty DestinationURI field.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Several TRDP listeners with non-empty DestinationURI filter are configured in the ETBN. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that DestinationURI field is empty. Step 7: Check that a reply message is sent back from each of the ETBN listeners, port=17225. Step 8: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 9: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 10: Check that as DestinationURI, the source URI sent in the request message is used. Step 11: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: An empty DestinationURI field MD telegram is sent to the ETBN, and a correct reply message is sent back from each of the ETBN listeners, even though they are configured with a non-empty DestinationURI filter.

9.5.7 Caller TRDP Layer Test

9.5.7.1 Test purpose

Test the defined TRDP layer in the Caller.

9.5.7.2 Check Caller Topography Counters against Actual Topography Counters

IDENTIFIER	TC23TRDPMD_028
TITLE	Check Caller Topography Counters against Actual Topography Counters
DESCRIPTION	The next test case will ensure that: If a TRDP user (caller) requests sending a message, its topography counters shall be checked against actual topography counters. If none of the cases listed in Table A.21 is TRUE, the message shall be discarded.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that TRDP user at Caller is notified about the attempt to send a request message. Step 5: Check that TRDP user is notified that the message is discarded as topography counter values of the message are not the actual topography counters. Step 6: Check that the request message is not send. Step 7: Check that a new request message is sent to the ETBN. Step 8: Check that MsgType field of the sent message is equal to '4D72'H ('Mr'), "Request with reply" telegram type. Step 9: Check that a reply message is sent back from the ETBN, port=17225. Step 10: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 11: Check that the TRDP user at Caller is notified about the reception of a message and that the topography counters at reply message are no the actual topography counters. Step 12: Check that the TRDP user at Caller is notified about that the message is discarded. Step 13: Check that no confirmation message is sent to the ETBN.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The Caller sends only messages with actual topography counter values, and discards messages with non-actual topography counter values.

9.5.7.3 Set to 0 the session identifier of the notification message

IDENTIFIER	TC23TRDPMD_029
TITLE	Set to 0 the session identifier of the notification message
DESCRIPTION	The next test case will ensure that: If a TRDP user (caller) requests sending a notification, the session identifier of the notification message shall be set to 0.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a notification message is received from the ETBN. Step 5: Check that the MsgType field of sent message is equal to '4D6E'H ('Mn'), "Request without reply" telegram type. Step 6: Check that the SessionId of sent notification message is equal to 0.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A notification message is received from the ETBN, and the SessionID field is equal to 0.

IECNORM.COM : Click to view the full PDF of IEC 61375-2-8:2021

9.5.7.4 Set the SessionID identifier format as 16-byte UUID according to RFC 4122

IDENTIFIER	TC23TRDPMO_030
TITLE	Set the SessionID identifier format as a 16-byte UUID according to RFC 4122
DESCRIPTION	The next test case will ensure that: The SessionID identifier format shall be defined as a 16-byte UUID according to RFC 4122
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to 4D72'H (Mr, Request with reply). Step 6: Check that SessionID field is 16-byte length and RFC 4122 format. Step 7: Check that a reply message is sent back from the ETBN, port=17225. Step 8: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 9: Check that SessionID field is 16-byte length and RFC 4122 format. Step 10: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: SessionID Parameter complies with the requested format.

9.5.7.5 SessionID identifier calculated in the Caller and shared with the Replier

IDENTIFIER	TC23TRDPMO_031
TITLE	SessionId identifier calculated in the Caller and shared with the Replier
DESCRIPTION	The next test case will ensure that: The SessionID identifier is calculated in the Caller TRDP layer, and shared with the replier during communication
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply message is sent back from the ETBN, port=17225. Step 7: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 8: Check that the SessionId transmitted in the request message is the same to the SessionId received in the reply message. Step 9: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: SessionID parameter is the same during the request/reply communication process.

9.5.7.6 Timeout value for reply messages at caller

IDENTIFIER	TC23TRDPMO_032
TITLE	Timeout value for reply messages at caller
DESCRIPTION	The next test case will ensure that: A timeout value for the reply messages shall be defined by the TRDP user (caller)
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply message is sent back from the ETBN, port=17225. Step 7: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 8: Check that MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 9: Check that no confirmation message is sent to the ETBN, port 17225.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The required time to reply to the request message complies with the time out time defined in ReplyTimeOut field of request message.

9.5.7.7 Wait for incoming reply messages and deliver them to the TRDP user

IDENTIFIER	TC23TRDPMO_033
TITLE	Wait for incoming reply messages
DESCRIPTION	The next test case will ensure that: During the caller session, the TRDP layer shall wait for incoming reply messages (MsgType 'Mp', 'Mq' or 'Me'). All incoming reply messages related to the caller session shall be given immediately to the TRDP user. When all the expected incoming replies is reached, the session is closed.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply message is sent back from the ETBN, port=17225. Step 7: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 8: Check that the TRDP user at Caller is notified about the reception of a reply message. Step 9: Check that a confirmation message is sent to the ETBN, port 17225. Step 10: Check that MsgType of sent message is equal to '4D63'H (Mc, Confirm). Step 11: Check that a new request message is sent to the ETBN. Step 12: Check that the SessionId is different from previous request/reply communication process, as the session was closed after the reception of the last confirmation message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The TRDP layer in the caller waits for incoming reply messages and give them immediately to the TRDP user. The communication session is closed after the last confirmation message.

9.5.7.8 SourceURI received in reply message used for DestinationURI in the confirmation message

IDENTIFIER	TC23TRDPMO_034
TITLE	SourceURI received in reply message used for DestinationURI in confirmation message
DESCRIPTION	The next test case will ensure that: The confirmation message shall use as destination URI the source URI received in the reply message
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 7: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 8: Check that a confirmation message is sent to the ETBN, port 17225. Step 9: Check MsgType of sent message is equal to '4D63'H ('Mc'), "Confirm" datagram type. Step 10: Check that as DestinationURI, the source URI received in the reply message is used.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After the request/reply messages, a confirm message is sent using as destination URI the source URI received in the reply message.

9.5.7.9 A confirmation message shall be unicast, and shall not contain user data

IDENTIFIER	TC23TRDPMO_035
TITLE	A confirmation message shall be unicast, and do not contain user data
DESCRIPTION	The next test case will ensure that: The confirmation message shall not contain user data and shall be sent only as unicast.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 7: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 8: Check that a confirmation message is sent to the ETBN, port 17225. Step 9: Check MsgType of sent message is equal to '4D63'H ('Mc'), "Confirm" datagram type. Step 10: Check that DatasetLength field is equal to 0. Step 11: Check that Dataset field is empty. User data is not sent in the confirmation message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After the request/reply messages, a confirm message is sent with an unicast address, and not sending user data in the message Dataset.

9.5.7.10 Provide confirmation message in time

IDENTIFIER	TC23TRDPMO_036
TITLE	Provide confirmation message in time
DESCRIPTION	The next test case will ensure that: The TRDP caller shall take care to provide the confirmation in time as indicated by the ReplyTimeout parameter of the reply.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 7: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 8: Check that a confirmation message is sent to the ETBN, port 17225. Step 9: Check MsgType of sent message is equal to '4D63'H ('Mc'), "Confirm" datagram type. Step 10: Check that confirmation message is sent in time as indicated by the ReplyTimeout parameter of the reply message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After the request/reply messages, a confirm message is sent in time as indicated by the ReplyTimeout parameter of the reply message.

9.5.7.11 TRDP session closed in Caller after confirm timeout expires

IDENTIFIER	TC23TRDPMMD_037
TITLE	TRDP session closed in Caller after confirm timeout expires
DESCRIPTION	The next test case will ensure that: The TRDP session shall be closed if the number of incoming replies is reached and the confirm timeout timer expires while waiting for outstanding confirmations from TRDP user (caller).
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType field of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 7: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 8: Check that no confirmation message is sent, and that TRDP user receives a notification. Step 9: Check that a new request message is sent to the ETBN and that SessionId parameter of new request is different to previous request message, as the previous session was closed.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After the request/reply messages, a confirm message is not sent, and after the ReplytimeOut time, the Caller receives a notification and TRDP session is closed.

9.5.7.12 TRDP session closed in Caller after reply timeout expires. One expected replier

IDENTIFIER	TC23TRDPMD_038
TITLE	TRDP session closed in Caller after confirm timeout expires
DESCRIPTION	The next test case will ensure that: The TRDP session shall be closed if reply timeout is reached without the expected reply. The TRDP layer will repeat the request "MaxNumRetries" times.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	"MaxNumRetries" is set to 2 in the Caller, the maximum number of retries. The number of expected repliers is 1. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType parameter of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that no reply message is sent within replyTimeout time defined in the request message. Step 7: Check that Caller repeats the request. Step 8: Check that no reply message is sent within replyTimeout time defined in the request message. Step 9: Check that Caller repeats the request. Step 10: Check that no reply message is sent within replyTimeout time defined in the request message. Step 11: Check that TRDP user receives a notification, because of missing reply. Step 12: Check that a new request message is sent to the ETBN and that SessionId parameter of new request is different to previous request message, as the previous session was closed.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After the first request message, no reply message is sent within the ReplytimeOut time. After two new request retries without a reply message, the Caller receives a notification and TRDP session is closed.

9.5.7.13 TRDP session closed in Caller after reply timeout expires. More than one expected replier

IDENTIFIER	TC23TRDPMMD_039
TITLE	TRDP session closed in Caller after confirm timeout expires
DESCRIPTION	The next test case will ensure that: The TRDP session shall be closed if reply timeout is reached without the expected reply. The TRDP layer will repeat the request "MaxNumRetries" times.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	"MaxNumRetries" is set to 1 in the Caller, the maximum number of retries. The number of expected repliers is 2. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBNs. Step 5: Check that MsgType parameter of sent messages is equal to '4D72'H ('Mr' Request with reply). Step 6: Check that only one reply message is sent within the replyTimeout value defined in the request messages. Step 7: Check that Caller does not repeat the request. Step 8: Check that TRDP user receives a notification, because of missing reply. It notifies the number of missing replies. Step 9: Check that a new request message is sent to the ETBNs and that SessionId parameter of new request is different to previous request message, as the previous session was closed.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After the request message, only one reply message is sent within the ReplytimeOut time. The TRDP user at the Caller receives a notification and TRDP session is closed. It is notified about the number of missing replies.

9.5.7.14 Session closed when all requested confirmations are sent or confirmation timeout timer has timed out

IDENTIFIER	TC23TRDPMO_040
TITLE	Session closed when all requested confirmations are sent or confirmation timeout timer has timed out
DESCRIPTION	The next test case will ensure that: The TRDP session shall be closed when all requested confirmations are sent or confirmation timeout timer has timed out
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that the reply message is sent within the replyTimeout value defined in the request message. Step 7: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 8: Check that a confirmation message is sent Step 9: Check MsgType of sent message is equal to '4D63'H ('Mc'), "Confirm" datagram type. Step 10: Check that a new request message is sent to the ETBN and that SessionId parameter of new request is different to previous request message, as the previous session was closed. The previous session was closed after a successful communication process. Step 11: Check that the MsgType identifier of sent new message is equal to '4D72'H ('Mr', Request with reply). Step 12: Check that a reply message is sent within the replyTimeout value defined in the request message. Step 13: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 14: Check that no confirmation message is sent, and that TRDP user receives a notification. Step 15: Check that a new request message is sent to the ETBN and that SessionId parameter of new request is different to previous request message, as the previous session was closed. The previous session was closed because of the confirmation timeout timer timed out.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: An unsuccessful communication process, without a confirmation message within confirmation timeout, follows to a first successful request/reply communication process. The TRDP session is closed in both cases.

9.5.7.15 Close TRDP session when reply timeout timer expires and number of repliers is unknown

IDENTIFIER	TC23TRDPMMD_041
TITLE	Close TRDP session when timeout timer expires, and number of repliers is unknown
DESCRIPTION	The next test case will ensure that: The TRDP session shall be closed when reply timeout timer expires and number of repliers is unknown. The total number of replies shall be indicated to TRDP user.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	A system with two repliers is required, but caller is not aware of number of replies (NoOfReplies = 0). Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBNs. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr' Request with reply). Step 6: Check that one of the ETBNs replies a message within the replyTimeout value defined in the request message. Step 7: Check that after the reply timeout timer expires, the TRDP user is notified with the total number of received replies.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After a communication process with unknown repliers is finished, the TRDP user shall be notified with received total number of replies.

9.5.7.16 No retransmission used for request messages if TCP is used

IDENTIFIER	TC23TRDPMO_042
TITLE	No retransmission used of request messages if TCP is used
DESCRIPTION	The next test case will ensure that: There shall be no retransmission of request messages if TCP is used. This shall be done in a higher level or in the application.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	TCP is configured as communication protocol
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that no reply is received from the ETBN. Step 7: Check that the request is not retransmitted.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: In a TCP communication where the reply is not received, the request message is not retransmitted.

9.5.7.17 TRDP provides the interface to continue a transmission after the connection was lost for TCP user

IDENTIFIER	TC23TRDPMMD_043
TITLE	TRDP provides the interface to continue a transmission after the connection was lost for TCP user
DESCRIPTION	The next test case will ensure that: TRDP shall provide the interface to continue a transmission after the connection was lost for TCP user.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	The TRDP caller and replier shall be configured to work with TCP. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that no reply is received from the ETBN. Step 7: Check that TRDP user is asked about its willingness to send a new request message. Step 8: Check that user accepts to send a new request message, and that a new request message is sent to the ETBN. Step 9: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 10: Check that a reply is received from the ETBN. Step 11: Check that the MsgType identifier of sent message is equal to '4D70'H ('Mp', Reply without confirmation).
FINAL CONDITIONS	Test Bench powered off
ACCEPTANCE CRITERIA	The result of test is successful if: After an unsuccessful request/reply communication process, the TRDP user is asked to retransmit the request message. As far as the user accepts the retransmission, a new request message is sent to the ETBN.

9.5.7.18 Discard TRDP messages ('Mp', 'Mq', 'Me') of an unknown session ID

IDENTIFIER	TC23TRDPMMD_044
TITLE	Discard TRDP messages of an unknown session ID
DESCRIPTION	The next test case will ensure that: If the TRDP layer receives a reply message without having opened a caller session for the indicated session id, the message shall be discarded.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a reply message is received from the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D71'H ('Mq', Reply with confirmation). Step 6: Check that the SessionId parameter of received message is an unknown id. Step 7: Check that no confirmation message is sent to the ETBN for the received reply with confirmation message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The reply message with an unknown SessionId will be discarded.

9.5.7.19 Use caller topography counter values throughout the session

IDENTIFIER	TC23TRDPMO_045
TITLE	Use caller topography counter values throughout the session
DESCRIPTION	The next test case will ensure that: The topography counter values obtained during caller session shall be used throughout the communication session.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply message is sent back from the ETBN, port=17225. Step 7: Check that the MsgType Identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 8: Check that etbTopoCnt and opTrnTopoCnt values are the same in request and reply messages.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: Topography counter values are the same in the caller and the replier during the request/reply communication process.

9.5.7.20 Discard messages with topography counter values different in caller and replier

IDENTIFIER	TC23TRDPMMD_046
TITLE	Discard messages with topography counter values different in caller and replier
DESCRIPTION	The next test case will ensure that: If a TRDP user receives a reply message with topography counter values different to the expected ones, the message shall be discarded.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply message is sent back from the ETBN, port=17225. Step 7: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 8: Check that etbTopoCnt and opTrnTopoCnt values are different in request and reply messages. Step 9: Check that no confirmation message is sent back to the ETBN, as reply message is discarded.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: Topography counter values are different in the caller and the replier during the request/reply communication process. The reply message is discarded.

9.5.7.21 Destroy session parameters after TRDP abort

IDENTIFIER	TC23TRDPMMD_047
TITLE	Destroy session parameters after TRDP abort
DESCRIPTION	The next test case will ensure that: If a TRDP user aborts an (open) session, running timers shall be stopped, the session identifier shall be destroyed, and the session shall be closed.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	An environment with two ETBNs and a testing system is needed. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBNs with a correct IP address. Check that the communication between the systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBNs. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN 1. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply message is sent back from the ETBN 1, port=17225. Step 7: Check that the MsgType identifier is equal to '4D71'H ('Mq'), "Reply with confirmation" datagram type. Step 8: Check that etbTopoCnt and opTrnTopoCnt values are equal in request and reply messages. Step 9: Check that a confirmation messages is sent back to the ETBN 1. Step 10: Check that a topology change is produced after ETBN 2 is disconnected from ETBN 1. Step 11: Check that a request message is sent to the ETBN 1. Step 12: Check that Session ID is different from previous TRDP session, as previous session was aborted. And check that topology counters are also different.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The TRDP session is aborted when a topology change happens.

9.5.8 Replier TRDP Layer Test

9.5.8.1 Test purpose

Test the defined TRDP layer in the Replier.

9.5.8.2 Register replier as listener for MD

IDENTIFIER	TC23TRDPMMD_048
TITLE	Register replier as listener for MD
DESCRIPTION	The next test case will ensure that: Each TRDP user (replier) that wants to receive MD shall register as listener for MD, to a specific URI (multicast or unicast) or for MD of a specific ComId.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	MD replier shall be registered as listener for MD of ComId 102. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that ComId parameter of request message is equal to 102 (TTDB – train directory information request) Step 7: Check that a reply message is sent back from the ETBN, port=17225. Step 8: Check that the MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 9: Check that reply message returns information for ComId=102. Step 10: Check that a new request message is sent to the ETBN. Step 11: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 12: Check that ComId parameter of request message is equal to 104 (TTDB – consist information request) Step 13: Check that no reply message is received from the ETBN.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: TRDP replier has a listener subscribed to ComId 102, so it responds to request messages for ComId = 102

9.5.8.3 Incoming messages shall be checked against the actual topography counters

IDENTIFIER	TC23TRDPMD_049
TITLE	Incoming messages shall be checked against the actual topography counters
DESCRIPTION	The next test case will ensure that: All incoming messages shall be checked against the actual topography counters. Messages not fitting to the actual topography counters shall be discarded. Messages related to the caller session are given immediately to the TRDP user.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply message is sent back from the ETBN, port=17225. Step 7: Check that the MsgType identifier is equal to '4D70'H ('Mp'), "Reply without confirmation" datagram type. Step 8: Check that topography counter values are the same in the request and reply messages. Step 9: Check that the SessionId identifier is the same for request and reply message. Step 9: Check that a new request message is sent to the ETBN. Step 10: Check that MsgType of sent message is equal to '4D72'H ('Mr', Request with reply). Step 11: Check that the topography counter values have changed from previous request/reply TRDP communication. Step 12: Check that no reply message is received from the ETBN, as topography counters are different in the caller and the replier.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: Messages not fitting to the actual topography counters are discarded.

9.5.8.4 “No replier instance” error when sending message to a not registered listener

IDENTIFIER	TC23TRDPMO_050
TITLE	No replier instance error when sending message to a not registered listener
DESCRIPTION	The next test case will ensure that: In case of a unicast request message to a not registered listener or to a listener expecting another value of the topography counters an error message shall be sent indicating the error with replyStatus == -3.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	ETBN is not registered as MD listener. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the “Ethernet bus analyzer tool” in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN, with an unicast address. Step 5: Check that the MsgType field of sent message is equal to ‘4D72’H (‘Mr’, Request with reply). Step 6: Check that an error message is sent back from the ETBN, port=17225. Step 7: Check that the MsgType identifier is equal to ‘4D65’H (‘Me’), “MD error” datagram type. Step 8: Check that ReplyStatus field is equal to -3, no replier instance (at replier side).
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: No replier instance error is returned when sending a unicast request message to a not registered ETBN.

9.5.8.5 Notification message passed to the related listener

IDENTIFIER	TC23TRDPMMD_051
TITLE	Notification message passed to the related listener
DESCRIPTION	The next test case will ensure that: If a notification message (MsgType 'Mn') is received, the message shall be passed to the related listener
REQUIREMENT	Mandatory
INITIAL CONDITIONS	An environment with two ETBNs and a testing system is needed. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a notification message is sent from ETBN 1 to the ETBN 2. Step 5: Check that the MsgType field of sent message is equal to '4D6E'H (Mn, Request without reply). Step 6: Check that TRDP layer in ETBN 2 notifies the reception of a TRDP notification message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A notification message is processed correctly in the TRDP Layer of the ETBN, and the TRDP user is notified about the reception.

9.5.8.6 Reply session opened using the received session id

IDENTIFIER	TC23TRDPMMD_052
TITLE	Reply session opened using the received session id
DESCRIPTION	The next test case will ensure that: If a request message (MsgType 'Mr') is received, a reply session shall be opened using the received session id, a timeout timer with the reply timeout shall be started, the message shall be passed to the related listener and the TRDP layer shall wait for the MD.reply of the listener.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply is received from the ETBN. Step 7: Check that the MsgType identifier of received message is equal to '4D70'H (Mp, Reply without confirmation). Step 8: Check that same SessionId identifier is used in the request/reply communication process. Step 9: Check that reply message is sent within ReplyTimeOut value defined in request message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A reply session is opened to manage the request message received from the caller. A reply messages is sent from replier to caller using session parameters received in the request message.

9.5.8.7 Received message discarded due to lack of resources

IDENTIFIER	TC23TRDPMMD_053
TITLE	Received message discarded due to Lack of Resources
DESCRIPTION	The next test case will ensure that: If, due to the lack of resources, a reply session can't be opened, the received message shall be discarded and an error message (MsgType = 'Me') shall be sent, indicating the error with replyStatus == -4 (no memory at replier side)
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that an error message is received from the ETBN. Error due to lack of resources. Step 7: Check that the MsgType identifier of received message is equal to '4D65'H (Me, error). Step 8: Check that ReplyStatus parameter is equal to -4 (no memory at replier side). Step 9: Check that no other reply message is received.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After lack of resources happens, the caller receives an error message from the replier as response to a request message.

9.5.8.8 Request message discarded for an open reply session

IDENTIFIER	TC23TRDPMO_054
TITLE	Request message discarded for an open reply session
DESCRIPTION	The next test case will ensure that: If a request message is received for an open reply session with the received session id while waiting for the MD.reply of the listener, the request message shall be discarded
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Listener at Replier is subscribed to process messages with ComId=102 and ComId=104. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that ComId parameter is equal to 102 (TTDB – train directory information request) Step 7: Check that a new request message is sent to the ETBN. Step 8: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 9: Check that ComId parameter is equal to 104 (TTDB – consist information request) Step 10: Check that SessionId is the same in both request messages. Step 11: Check that a reply is received from the ETBN. Step 12: Check that the MsgType identifier of received message is equal to '4D70'H ('Mp', Reply without confirmation). Step 13: Check that ComId identifier of received message is equal to 102. Step 14: Check that same SessionId identifier is used in the request/reply communication process. Step 15: Check that no reply message is received for request message with ComId=104.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The request message with ComId=104 is discarded because a reply message for request message with ComId=102 is pending.

9.5.8.9 A reply message shall be sent as unicast using information of request message and information of replier

IDENTIFIER	TC23TRDPMD_055
TITLE	A reply message shall be sent as unicast using information of request message and information of replier
DESCRIPTION	The next test case will ensure that: A reply message shall be sent as unicast using source IP address and source URI of the related request message as destination IP address and destination URI
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply message is received from the ETBN. Step 7: Check that the MsgType identifier of received message is equal to '4D70'H ('Mp', Reply without confirmation). Step 8: Check that the destination IP address of the reply message is equal to the source IP address of the request message. Step 9: Check that the DestinationURI of the reply message is equal to the SourceURI of the request message. Step 10: Check that the source IP address of the reply message is equal to the IP address of listener at the replier. Step 11: Check that the SourceURI of the reply message is equal to the unique URI of the listener at the replier.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The destination IP address and DestinationURI fields of reply message are defined with the source IP address and SourceURI fields of the request message. The source ip address and SourceURI fields are defined with the information of the replier.

9.5.8.10 Reply session closed because of missing confirmation

IDENTIFIER	TC23TRDPMO_056
TITLE	Reply session closed because of missing confirmation
DESCRIPTION	The next test case will ensure that: If a reply message without confirmation was sent, the reply session shall be closed.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply is received from the ETBN. Step 7: Check that the MsgType identifier of received message is equal to '4D71'H ('Mq', Reply with confirmation). Step 8: Check that no confirmation is sent to the ETBN within the given confirm timeout. Step 9: Check that a new request message is sent to the ETBN. Step 10: Check that the SessionId of new request is different from SessionId of previous request, as previous request was closed after confirmation timeout expired.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A confirmation timeout is started in a "reply message with confirmation" message type and reply session is closed when no confirmation is received after that time.

9.5.8.11 Message sequence number increase

IDENTIFIER	TC23TRDPMO_057
TITLE	Message sequence number increase
DESCRIPTION	The next test case will ensure that: If a request message is received with a different sequence counter for an already open reply session, after the TRDP layer has received the MD.reply from the TRDP user, the TRDP layer shall repeat sending the reply message to the caller with an incremented sequence counter. A confirm timeout value shall be restarted and the session shall be closed after that time.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that a reply is received from the ETBN. Step 7: Check that the MsgType identifier of received message is equal to '4D71'H ('Mq', Reply with confirmation). Step 8: Check that a new request message is sent to the ETBN. Step 9: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 10: Check that sequence counter of new request is smaller than sequence counter of first request. Step 11: Check that the ETBN resends the reply but with an incremented sequence number comparing to the previous reply. Step 12: Check that the dataset of the last reply message is equal to the dataset of first reply message. Step 13: Check that the MsgType identifier of received message is equal to '4D71'H ('Mq', Reply with confirmation). Step 14: Check that no confirmation message is sent to the ETBN. Step 15: Check that previous session is closed after confirmation timeout expired. Step 16: Check that new request is sent using new session id.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A reply message is repeated with an incremented sequence counter after a request with a decremented sequence counter.

9.5.8.12 Discard request message with same sequence counter

IDENTIFIER	TC23TRDPMO_058
TITLE	Discard request message with same sequence counter
DESCRIPTION	The next test case will ensure that: If a request message is received for an already open reply session and the same sequence counter as received before, the request message shall be discarded.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	ETBN listeners shall be subscribed to listen to request messages with ComId=102 or ComId=104. Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that the ComId parameter is equal to 102. Step 7: Check that a new request message is sent to the ETBN. Step 8: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 9: Check that the ComId parameter of new request is equal to 104. Step 10: Check that sequence counter of new request is the same to the sequence counter of first request. Step 11: Check ETBN sends a reply message only for the first request, ComId=102. Step 12: Check that the MsgType identifier of received message is equal to '4D70'H ('Mp', Reply without confirmation).
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: Only a reply message is sent for ComId=102 as it is the reply for the first of two requests with same sequence counter value.

9.5.8.13 Confirmation message management

IDENTIFIER	TC23TRDPMO_059
TITLE	Confirmation message management
DESCRIPTION	The next test case will ensure that: If a confirmation message is received without having an open reply session with the received session id, the message shall be discarded, otherwise the related listener shall be notified, and the session shall be closed.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check ETBN sends a reply message. Step 7: Check that the MsgType identifier of received message is equal to '4D74'H ('Mq', Reply with confirmation). Step 8: Check that a confirmation message is sent to the ETBN. Step 9: Check that the SessionId parameter is different to the SessionId parameter of the request/reply session. Step 10: Check that the TRDP user is not notified about the session closing. Step 11: Check that another confirmation message is sent to the ETBN. Step 12: Check that the SessionId parameter is the one used in the request/reply session. Step 13: Check that the TRDP user is notified about the session closing
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A confirmation message with a different SessionId is discarded, but a message with same SessionId as used in request/reply process is used to close the session.

9.5.8.14 Reply session closed after timeout timer expires

IDENTIFIER	TC23TRDPMO_060
TITLE	Confirmation message management
DESCRIPTION	The next test case will ensure that: If the timeout timer of a reply session expires, the reply session shall be closed. In case of a timeout because of a missing MD.reply of the listener, the listener shall be closed.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that ETBN does not send a reply message. Step 7: Check that the TRDP user at replier is notified about this event.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: TRDP session is closed and listener notified after the reply timeout timer expires.

9.5.8.15 Topography counter values used over the request/reply session

IDENTIFIER	TC23TRDPMO_061
TITLE	Topography counter values used over the request/reply session
DESCRIPTION	The next test case will ensure that: The topography counter values obtained in an MD.addListener service primitives shall be used for upcoming reply sessions unless they are updated by the user. If a TRDP user receives a request message with topography counter values different to the expected ones, the message shall be discarded.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check ETBN sends a reply message. Step 7: Check that the MsgType identifier of received message is equal to '4D70'H ('Mp', Reply without confirmation). Step 8: Check that topography counter values, etbTopoCnt and opTrnTopoCnt, are the same in the request and reply messages. Step 9: Check that a new request message is sent to the ETBN. Step 10: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 11: Check that the ETBN does not send a reply message. Step 12: Check that topography counter values have changed from previous request/reply process to the last request process.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: There are discarded the requests with topography counters different to the expected ones.

9.5.8.16 Session closed after a TRDP user updates or removes a listener

IDENTIFIER	TC23TRDPM062
TITLE	Session closed after a TRDP user updates or removes a listener
DESCRIPTION	The next test case will ensure that: If a TRDP user removes or updates a listener during an open session, running timers shall be stopped, the session identifier shall be destroyed and the session shall be closed.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	The ETBN has a listener subscribed to ComId 102. Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that ComId parameter of the message is equal to 102. Step 7: Check that TRDP user at replier is notified about the TRDP listener for ComId 102 is deleted. Step 8: Check that the ETBN does not send a reply message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The request/reply session is closed after the listener is deleted.

9.5.9 TCP Connection Handling Test**9.5.9.1 Test purpose**

Test the defined TRDP layer in the Replier.

9.5.9.2 Open a TCP connection

IDENTIFIER	TC23TRDPMO_063
TITLE	Open a TCP connection
DESCRIPTION	The next test case will ensure that: Using TCP for data transmission requires an established TCP connection between the caller and the replier device. The first request to a specified device shall open a connection if there was not yet a connection established before.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	There was not a previous communication between the caller and the replier. Test Bench powered off.
STEPS	Step 1: Ensure that there is not a tcp port used for TRDP. Execute "netstat -putane". Step 2: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 3: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 4: Check that message data frames are sent from port=17225. Step 5: Check that a notification message is sent to the ETBN. Step 6: Check that the MsgType identifier of sent message is equal to '4D6E'H (Mn, Request without reply). Step 7: Check that there is a tcp port used for TRDP. Execute "netstat -putane".
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A TCP port is opened for TRDP communication.

9.5.9.3 Close a TCP connection in the Caller because of a TCP closing signal reception

IDENTIFIER	TC23TRDPM_D_064
TITLE	Close a TCP connection in the Caller because of a TCP closing signal reception
DESCRIPTION	The next test case will ensure that: The caller shall close an existing TCP connection when a TCP closing signal is received.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a notification message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D6E'H (Mn, Request without reply). Step 6: Check that there is a new TCP socket for TRDP. Execute "netstat -putane". Step 7: Check that a TCP closing signal is received. Step 8: Ensure that the opened TRDP socket is closed. Execute "netstat -putane".
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A TCP connection is closed in the caller when TCP closing signal is received.

9.5.9.4 Close a TCP connection in the Caller because of a TCP shutdown

IDENTIFIER	TC23TRDPMO_065
TITLE	Close a TCP connection in the Caller because of a TCP shutdown
DESCRIPTION	The next test case will ensure that: The caller shall close an existing TCP connection when a TRDP shut down event happens.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off.
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a notification message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D6E'H (Mn, Request without reply). Step 6: Check that there is a new TCP socket for TRDP. Execute "netstat -putane". Step 7: Check that a trdp "tlc_terminate" event happens in the caller. Step 8: Ensure that the opened TCP socket for TRDP is closed. Execute "netstat -putane".
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A TCP connection is closed in the Caller when TRDP "tlc_terminate" event happens.

9.5.9.5 Close a TCP connection because of connection inactivity

IDENTIFIER	TC23TRDPMO_066
TITLE	Close a TCP connection because of connection inactivity
DESCRIPTION	The next test case will ensure that: The caller shall close an existing TCP connection when a timeout occurred because the TCP connection has not been used for a defined time
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a notification message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D6E'H ('Mn', Request without reply). Step 6: Check that there is a new TCP socket for TRDP. Execute "netstat -putane". Step 7: Check that TCP connection inactivity timer expires. Step 8: Ensure that opened TCP socket for TRDP is closed. Execute "netstat -putane".
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A TCP connection is closed when TCP connection inactivity timer expires.

9.5.9.6 Replier uses existing connection

IDENTIFIER	TC23TRDPMO_067
TITLE	Replier uses existing connection
DESCRIPTION	The next test case will ensure that: The replier shall use the connection opened by the caller
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that the ETBN sends a reply using the same TCP connection (same socket). Step 7: Check that the MsgType identifier of sent message is equal to '4D70'H ('Mp', Reply without confirmation).
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: Replier sends a reply to the caller using the TCP connection, opened by the caller.

9.5.9.7 Close a TCP connection in the Replier because of a TCP closing signal reception

IDENTIFIER	TC23TRDPMO_068
TITLE	Close a TCP connection in the Replier because of a TCP closing signal reception
DESCRIPTION	The next test case will ensure that: The replier shall close an existing TCP connection when a TCP closing signal is received.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a notification message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D6E'H ('Mn', Request without reply). Step 6: Check that there is a new TCP socket used for TRDP. Execute "netstat -putane". Step 7: Check that a TCP closing signal is received. Step 8: Ensure that the opened TCP socket for TRDP is closed. Execute "netstat -putane".
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A TCP socket is closed in the replier when TCP closing signal is received.

9.5.9.8 Close a TCP connection in the Replier because of a TRDP shutdown

IDENTIFIER	TC23TRDPMO_069
TITLE	Close a TCP connection in the Replier because of a signal reception
DESCRIPTION	The next test case will ensure that: The replier shall close an existing TCP connection when a TRDP shut down event happens.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a notification message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D6E'H ('Mn', Request without reply). Step 6: Check that there is a new TCP socket for TRDP. Execute "netstat -putane". Step 7: Check that a trdp "tlc_terminate" event happens in the caller. Step 8: Ensure that the opened TCP socket for TRDP is closed. Execute "netstat -putane".
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: A TCP port is closed in the replier when TRDP "tlc_terminate" event happens.

9.5.9.9 Close a TCP connection in the Replier because another TCP connection was opened from the same caller

IDENTIFIER	TC23TRDPM_070
TITLE	Close a TCP connection in the Replier because another TCP connection was opened from the same caller
DESCRIPTION	The next test case will ensure that: If another TCP connection was opened from the caller device and the old connection is not used anymore for a defined time, the replier closes the old connection.
REQUIREMENT	Mandatory
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a notification message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D6E'H ('Mn', Request without reply). Step 6: Check that there is a new TCP socket for TRDP. Execute "netstat -putane". Step 7: Check that another request message is sent to the ETBN Step 8: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mn', Request with reply). Step 9: Check that there is a new TCP socket for TRDP. Execute "netstat -putane". Step 10: Check that after the defined time, the first opened TCP socket for TRDP is closed. Execute "netstat -putane".
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: After a defined time, the first opened TCP is closed.

9.5.10 Message Data Echo Server Test

9.5.10.1 Test purpose

Test the optionally provided TRDP message data echo function.

9.5.10.2 Returned Message Data in TRDP Message Data Echo Function

IDENTIFIER	TC23TRDPMMD_071
TITLE	Returned Message Data in TRDP Message Data Echo Function
DESCRIPTION	The next test case will ensure that: Message Data Echo is implemented in TRDP layer to provide message data echo function. The user data of the returned message shall be a copy of the received user data.
REQUIREMENT	Optional
INITIAL CONDITIONS	Test Bench powered off
STEPS	Step 1: Ensure that the testing system is connected to the ETBN with a correct IP address. Check that the communication between both systems is good. Step 2: Open the "Ethernet bus analyzer tool" in the testing system and capture the frames of the ETBN. Configure the tool to focus in TCP frames. Step 3: Check that message data frames are sent from port=17225. Step 4: Check that a request message is sent to the ETBN. Step 5: Check that the MsgType identifier of sent message is equal to '4D72'H ('Mr', Request with reply). Step 6: Check that the ComId parameter is equal to 86 (echo request message). Step 7: Check that the ETBN sends a reply message. Step 8: Check that the MsgType identifier of sent message is equal to '4D70'H ('Mp', Reply without confirmation). Step 9: Check that the ComId parameter is equal to 87 (echo reply message). Step 10: Check that the dataset of request message is equal to the dataset of request message.
FINAL CONDITIONS	Test Bench powered off.
ACCEPTANCE CRITERIA	The result of test is successful if: The dataset of reply message is equal to the dataset of request message.

9.6 Process Data test

9.6.1 General

Process Data test is used to evaluate TRDP Process Datafunction and performance of the IUT.

9.6.2 Test setup

The test system should be able to monitor the network communication of the IUT, to analyze the data of TRDP and to evaluate the consistency of IUT communication. The test set-up as shown in Figure 49 should satisfy the following requirement during testing: The train communication device running TRDP (IUT) is connected to the test system. The test system (with a monitor and a TRDP simulator inside the system and a connection network outside the system) shall analyze and record the network port of the IUT.

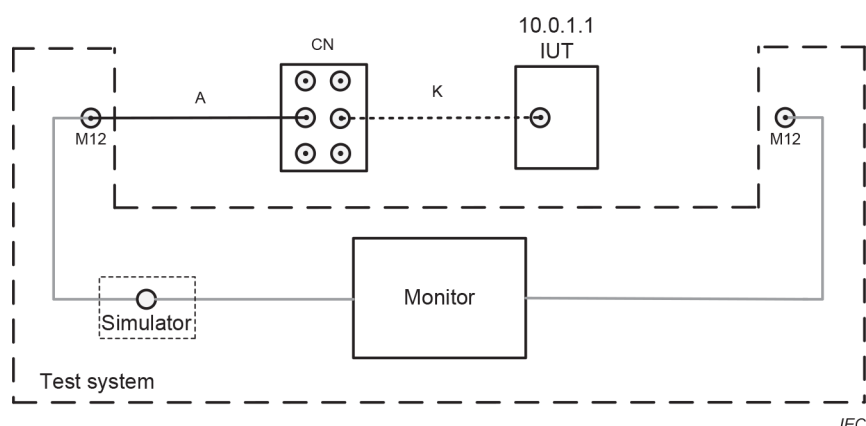


Figure 49 – TTS, IUT connected to the test system

During the TRDP testing, the port of IUT should connect to the testing network, and adding a switch shown in Figure 50 in the communication link of test topology to simulate the network link disconnection.

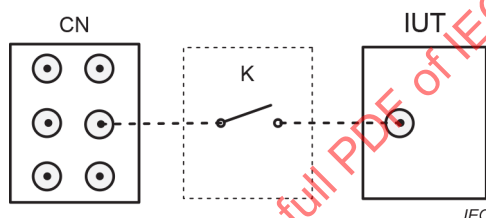


Figure 50 – The switch in test topology

9.6.3 PD push pattern test cases

9.6.3.1 Push pattern telegram test

Testing topology: TTS.

The test process and criterion are shown in Table 47.

Table 47 – Push pattern telegram test

Step	Action	Expected result
1	Start IUT, IUT send TRDP Process Data with ComId of 300 and period of 1000 ms	The format of received telegram matches the push pattern telegram format of Table 41 (All telegrams from the first one shall match)

Figure 51 shows TRDP telegram format specified by IEC 61375-2-3:2015, Clause A.6, the former 40 bytes are TRDP PD header, and the next 0 to 1432 bytes are TRDP application data. Table 48 shows the description and value of the TRDP PD header.

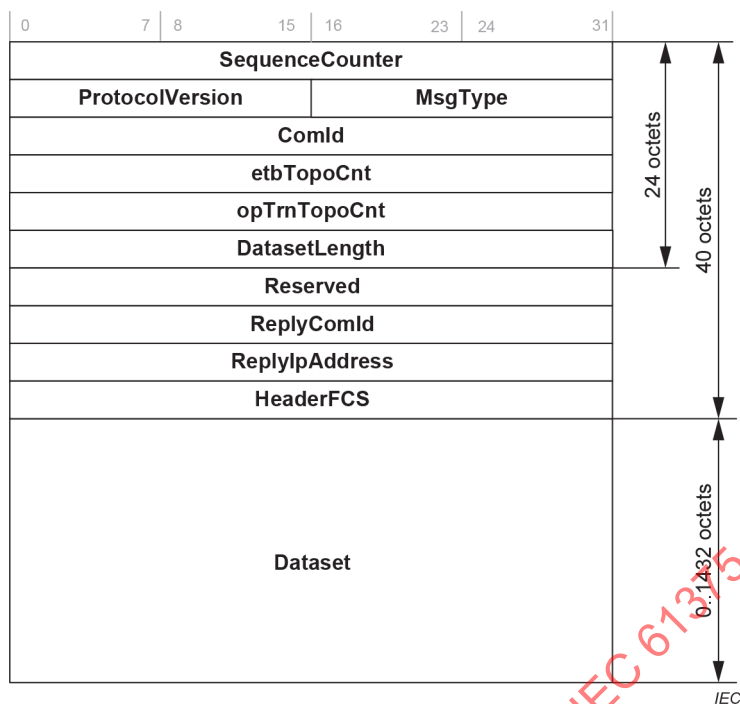


Figure 51 – TRDP telegram format

Table 48 – Push pattern telegram format content

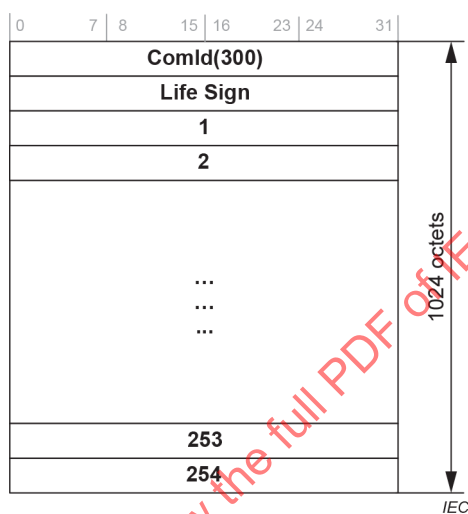
No.	Item in TRDP header	Expected Result
1	sequenceCounter: The sequence counter	Shall be incremented by one with each sending of the process telegram.
2	protocolVersion: The version of protocol	Same with the statement in TRDP PICS, example with 0x0100 or 0x0102
3	msgType: The type of telegram	Push pattern: fixed with 0x5064
4	ComId: The identifier of communication	300
5	etbTopoCnt: The ETB topography counter	Fixed with 0 when communicating within the consist
6	opTrnTopoCnt: The operational train topography counter	Fixed with 0 when communicating within the consist
7	datasetLength: The dataset length	1024 bytes
8	reserved01: Reserved for extension	Fixed with 0
9	replyComId: The requested ComId	Push pattern: fixed with 0
10	replyIpAddress: The reply IP address	Push pattern: fixed with 0
11	headerFCS: Header checksum	Calculated checksum of header according to IEC 61375-2-3:2015, Clause A.3 (checksum self is not included)

Telegram parameter as in Table 49.

Table 49 – Telegram parameter of push pattern test

Telegram type	'Pd'
ComId	300
Dataset	CONFTEST_PUSH_FORMAT
Dataset length	1024 Bytes
Destination IP	239.255.3.0
Cycle Time	1000 ms

Telegram data definition as in Figure 52.

**Figure 52 – Dataset of CONFTEST_PUSH_FORMAT**

9.6.3.2 Destination UDP port test

Test topology: TTS.

The test process and criterion as shown in Table 50.

Table 50 – Destination UDP port test

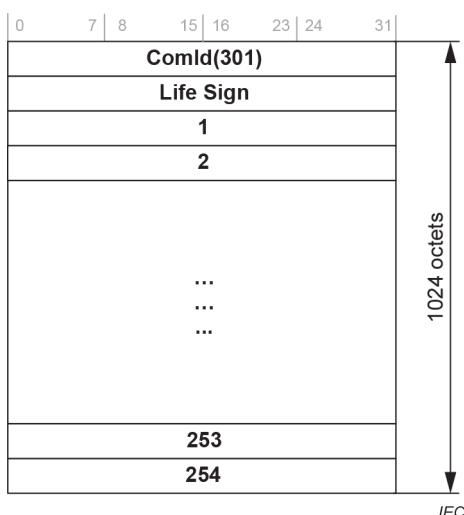
Step	Action	Expected Result
1	Start IUT with the destination UDP port default value 17224, and send TRDP process data with ComId of 301 and period of 1 000 ms and Simulator receives associated ComId data with the UDP port default value 17224	The data received by Simulator are same with the data sent by IUT
2	After step 1, keep the destination UDP port of IUT same, and modify the destination UDP port of Simulator to another value 17000, IUT send TRDP process data with ComId of 301 and period of 1 000 ms and Simulator receives associated ComId data	No TRDP PD received by Simulator, but a UDP port 17224 data will be received.
3	After step 1, keep the destination UDP port of Simulator same, and modify the destination UDP port of IUT to another value 17000, restart IUT, IUT send TRDP process data with ComId of 301 and period of 1 000 ms and Simulator subscribe associated ComId data	No data received by Simulator, but a UDP port 17000 data will be received.
4	Modify destination UDP port of both IUT and Simulator to non-default value 17000, and restart IUT, IUT send TRDP process data with ComId of 301 and period of 1 000 ms and Simulator subscribe associated ComId data	The data received by Simulator are same with the data sent by IUT

Telegram parameter as in Table 51.

Table 51 – Destination UDP port test telegram parameter

Telegram type	'Pd'
ComId	301
Dataset	CONFTEST_UDP_PORT
Dataset length	1024 Bytes
Destination IP	239.255.3.1
Cycle time	1000 ms

Telegram data are defined as in Figure 53.

**Figure 53 – Dataset of CONFTEST_UDP_PORT**

9.6.3.3 FCS check test

Test topology: TTS.

Simulator sends TRDP process data with ComId of 302. IUT receives the data from Simulator and replies a telegram to Simulator with ComId of 303, and sets check value of reply telegram 0x55555555 if received, sets check value 0xAAAAAAAA if timeout.

The test process and criterion as shown in Table 52.

Table 52 – FCS check test

Step	Action	Expected Result
1	Start IUT, Simulator send TRDP process data ComId of 302 with period of 1 000 ms and with correct FCS of TRDP PD header	The check value of reply telegram with ComId of 303 shall be 0x55555555
2	Simulator send TRDP process data ComId of 302 with period of 1 000 ms and with incorrect FCS of TRDP PD header	The check value of reply telegram with ComId of 303 shall be 0xAAAAAAAA

Telegram parameter as in Table 53 and Table 54.

Table 53 – FCS check test telegram parameter

Telegram type	'Pd'
ComId	302
Dataset	CONFTEST_FCS
Dataset length	1024 Bytes
Destination IP	239.255.3.2
Cycle time	1 000 ms

Table 54 – FCS check test reply telegram parameter

Telegram type	'Pd'
ComId	303
Dataset	CONFTEST_FCS_REPLY
Dataset length	1024 Bytes
Destination IP	239.255.3.3
Cycle time	1 000 ms

Telegram data are defined as in Figure 54 and Figure 55.

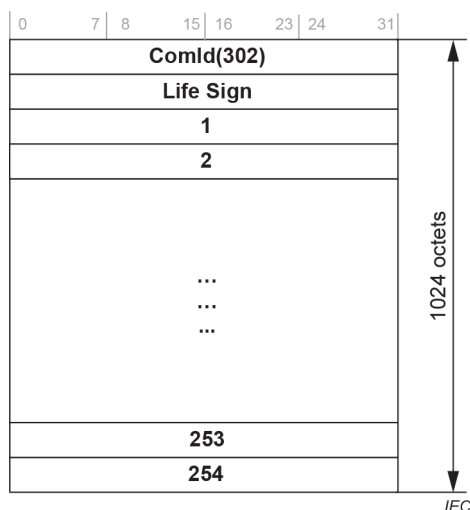


Figure 54 – Dataset of CONFTEST_FCS

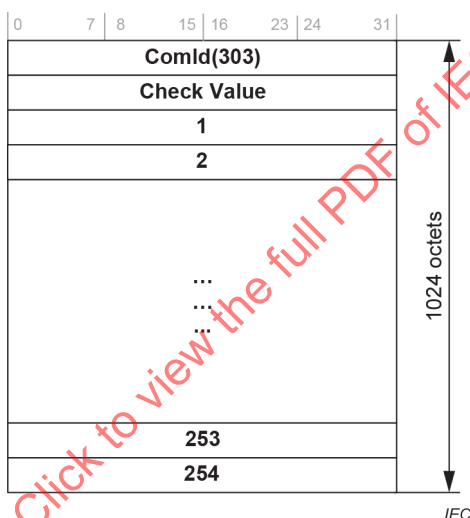


Figure 55 – Dataset of CONFTEST_FCS_REPLY

9.6.3.4 Protocol version test

Test topology: TTS.

Simulator send TRDP process data with ComId of 304 IUT receive the data from Simulator and reply a telegram to Simulator with ComId of 305 and set check value of reply telegram 0x55555555 if received, set check value 0xAAAAAAAA if timeout.

Test process and criterion as shown in Table 55.

Table 55 – Protocol version test

Step	Action	Expected Result
1	Start IUT, the TRDP high version H2 is 1, low version L2 is 2. Simulator send TRDP process data with ComId of 304 and period of 1 000 ms, the TRDP high version H1 is 1 (same as H2) and low version L1 is 1 (no specific requirement)	The check value of reply telegram with ComId of 305 shall be 0x55555555
2	Restart IUT, the TRDP high version H2 is 2, low version L2 is 1. Simulator send TRDP process data with ComId of 304 and period of 1 000 ms, the TRDP high version H1 is 1 (different from H2) and low version L1 is 1 (no specific requirement)	The check value of reply telegram with ComId of 305 shall be 0xAAAAAAAA

Telegram parameter as in Table 56 and Table 57.

Table 56 – Protocol version test telegram parameter

Telegram type	'Pd'
ComId	304
Dataset	CONFTEST_PROTO_VERSION
Dataset length	1 024 Bytes
Destination IP	239.255.3.4
Cycle time	1 000 ms

Table 57 – Protocol version test reply telegram parameter

Telegram type	'Pd'
ComId	305
Dataset	CONFTEST_PROTO_VERSION_REPLY
Dataset length	1 024 Bytes
Destination IP	239.255.3.5
Cycle time	1 000 ms

Telegram data definition as in Figure 56 and Figure 57.

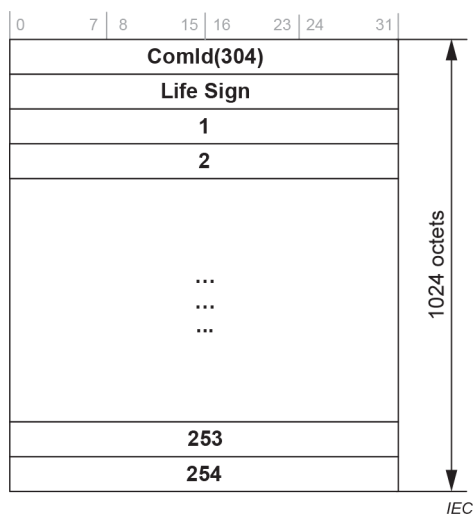


Figure 56 – Dataset of CONFTEST_PROTO_VERSION

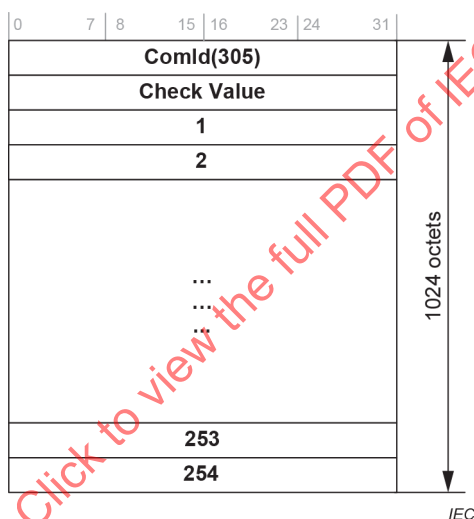


Figure 57 – Dataset of CONFTEST_PROTO_VERSION_REPLY

9.6.3.5 Topology counter test

Test topology: TTS.

Test process and criterion as shown in Table 58.

Table 58 – Topology counter test

Step	Action	Expected Result
1	Start IUT, Simulator publish TRDP process data with ComId of 306 with the etbTopoCnt=0 and opTrnTopoCnt=0 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in IUT, IUT publish TRDP process data with ComId of 315 with the etbTopoCnt=0 and opTrnTopoCnt=0 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in Simulator. IUT receive TRDP process data from Simulator and then copy the data to the sending buffer of ComId 315.	The data of ComId 315 received by Simulator are same with the data of ComId 306
2	Restart IUT, Simulator publish TRDP process data with ComId of 307 with the etbTopoCnt=0 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in IUT, IUT publish TRDP process data with ComId of 316 with the etbTopoCnt=0 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in Simulator. IUT receive TRDP process data from Simulator and then copy the data to the sending buffer of ComId 316.	The data of ComId 316 received by Simulator are same with the data of ComId 307
3	Restart IUT, Simulator publish TRDP process data with ComId of 308 with the etbTopoCnt=0 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x66778899 in IUT, IUT publish TRDP process data with ComId of 317 with the etbTopoCnt=0 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x66778899 in Simulator.	No data received by Simulator and no data received by IUT
4	Restart IUT, Simulator publish TRDP process data with ComId of 309 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in IUT, IUT publish TRDP process data with ComId of 318 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in Simulator. IUT receive TRDP process data from Simulator and then copy the data to the sending buffer of ComId 318.	The data of ComId 318 received by Simulator are same with the data of ComId 309
5	Restart IUT, Simulator publish TRDP process data with ComId of 310 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0 in the telegram, while the actual etbTopoCnt=0x22334455 and opTrnTopoCnt=0x55667788 in IUT, IUT publish TRDP process data with ComId of 319 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0 in the telegram, while the actual etbTopoCnt=0x22334455 and opTrnTopoCnt=0x55667788 in Simulator.	No data received by Simulator and no data received by IUT

Step	Action	Expected Result
6	Restart IUT, Simulator publish TRDP process data with ComId of 311 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in IUT, IUT publish TRDP process data with ComId of 320 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in Simulator. IUT receive TRDP process data from Simulator and then copy the data to the sending buffer of ComId 318.	The data of ComId 320 received by Simulator are same with the data of ComId 311
7	Restart IUT, Simulator publish TRDP process data with ComId of 312 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x66778899 in IUT, IUT publish TRDP process data with ComId of 321 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x11223344 and opTrnTopoCnt=0x66778899 in Simulator.	No data received by Simulator and no data received by IUT
8	Restart IUT, Simulator publish TRDP process data with ComId of 313 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x22334455 and opTrnTopoCnt=0x55667788 in IUT, IUT publish TRDP process data with ComId of 322 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x22334455 and opTrnTopoCnt=0x55667788 in Simulator.	No data received by Simulator and no data received by IUT
9	Restart IUT, Simulator publish TRDP process data with ComId of 314 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x22334455 and opTrnTopoCnt=0x66778899 in IUT, IUT publish TRDP process data with ComId of 323 with the etbTopoCnt=0x11223344 and opTrnTopoCnt=0x55667788 in the telegram, while the actual etbTopoCnt=0x22334455 and opTrnTopoCnt=0x66778899 in Simulator.	No data received by Simulator and no data received by IUT

Telegram parameter as in Table 59.

Table 59 – Simulator sending telegram parameter

Telegram type	'Pd'
ComId	306~314
Dataset	CONFTEST_SIMU_TOPO
Dataset length	1 024 Bytes
Destination IP	239.255.3.6~239.255.3.14
Cycle time	1 000 ms

Telegram data are defined in Figure 58 and Table 60.

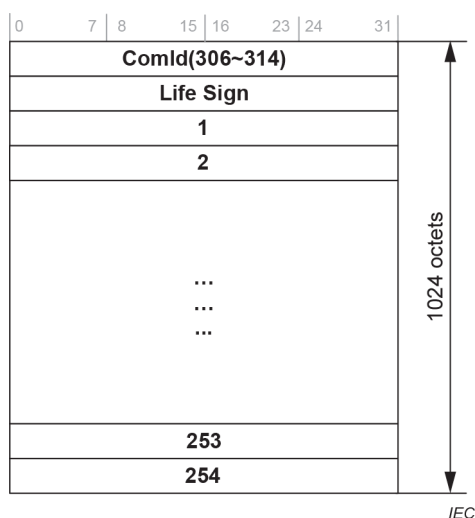


Figure 58 – Dataset of CONFTEST_SIMU_TOPO

Table 60 – IUT sending telegram parameter

Telegram type	'Pd'
ComId	315~323
Dataset	CONFTEST_IUT_TOPO
Dataset length	1 024 Bytes
Destination IP	239.255.3.15~239.255.3.23
Cycle time	1 000 ms

Telegram data are defined in Figure 59.

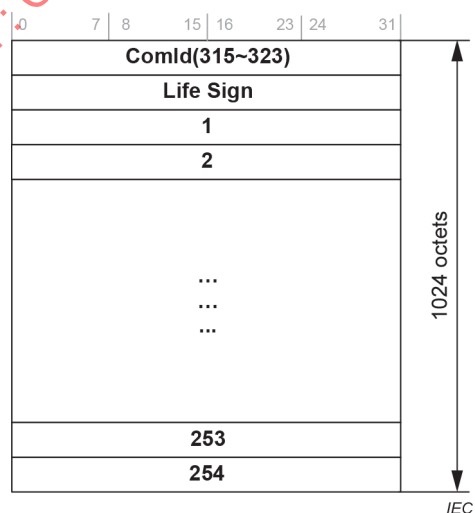


Figure 59 – Dataset of CONFTEST_IDU_TOPO

9.6.3.6 Data length test

Test topology: TTS.

The test procedure and criteria are shown in Table 61.

Table 61 – Data length test

Step	Action	Expected Result
1	Start IUT, IUT send TRDP Process Data with ComId of 324 and period of 1 000 ms, the data length is 1 432 bytes by the configuration file.	The data received by Simulator are same with the data sent by IUT, and data length is 1 432 bytes
2	Start IUT, IUT send TRDP Process Data with ComId of 325 and period of 1000 ms, the data length is 0 bytes by the configuration file.	No data received by Simulator, data length is 0 byte
3	Start IUT, IUT send TRDP Process Data with ComId of 326 and period of 1000 ms, the data length is 512 bytes by the configuration file.	The data received by Simulator are same with the data sent by IUT, and data length is 512 bytes

Telegram Parameters are defined in Table 62, Table 63 and Table 64.

Table 62 – Data length test of 1 432 bytes length telegram parameters

Telegram type	'Pd'
ComId	324
Dataset	CONFTEST_DATA_LEN_1432
Dataset length	1 432 Bytes
Destination IP	239.255.3.24
Cycle time	1 000 ms

Table 63 – Data length test of 0 byte length telegram parameters

Telegram type	'Pd'
ComId	325
Dataset	CONFTEST_DATA_LEN_0
Dataset length	0 Bytes
Destination IP	239.255.3.25
Cycle time	1 000 ms

Table 64 – Data length test of 512 bytes length telegram parameters

Telegram type	'Pd'
ComId	326
Dataset	CONFTEST_DATA_LEN_512
Dataset length	512 Bytes
Destination IP	239.255.3.26
Cycle time	1 000 ms

Telegram data are defined in Figure 60, Figure 61 and Figure 62.

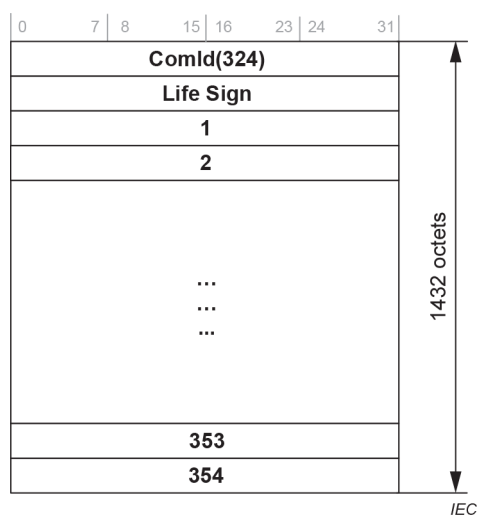


Figure 60 – Dataset of CONFTEST_DATA_LEN_1432

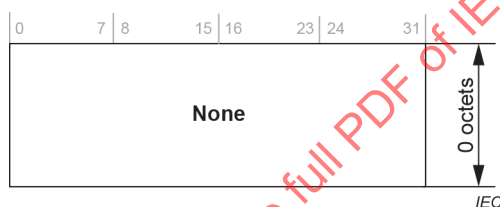


Figure 61 – Dataset of CONFTEST_DATA_LEN_0

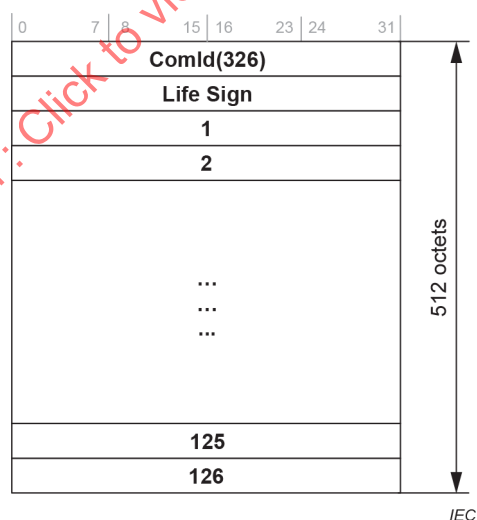


Figure 62 – Dataset of CONFTEST_DATA_LEN_512

9.6.3.7 Byte alignment test

Test topology: TTS.

The test procedure and criteria are shown in Table 65.

Table 65 – Byte alignment test

Step	Action	Expected Result
1	Start IUT, IUT send TRDP Process Data with ComId of 327 and period of 1000 ms, the data length is 441 (not a multiple of 4) bytes by the configuration file.	The data length received by Simulator is 444, 3 bytes of zero appended
2	Start IUT, IUT send TRDP Process Data with ComId of 328 and period of 1 000 ms, the data length is 442 (not a multiple of 4) bytes by the configuration file.	The data length received by Simulator is 444, 2 bytes of zero appended
3	Start IUT, IUT send TRDP Process Data with ComId of 329 and period of 1 000 ms, the data length is 443 (not a multiple of 4) bytes by the configuration file.	The data length received by Simulator is 444, 1 bytes of zero appended

Telegram parameters are defined in Table 66, Table 67 and Table 68.

Table 66 – Byte alignment test of 441 bytes length telegram parameters

Telegram type	'Pd'
ComId	327
Dataset	CONFTEST_BYTE_ALIGN_441
Dataset length	441 Bytes
Destination IP	239.255.3.27
Cycletime	1 000 ms

Table 67 – Byte alignment test of 442 bytes length telegram parameters

Telegram type	'Pd'
ComId	328
Dataset	CONFTEST_BYTE_ALIGN_442
Dataset length	442 Bytes
Destination IP	239.255.3.28
Cycle time	1 000 ms

Table 68 – Byte alignment test of 443 bytes length telegram parameters

Telegram type	'Pd'
ComId	329
Dataset	CONFTEST_BYTE_ALIGN_443
Dataset length	443 Bytes
Destination IP	239.255.3.29
Cycle time	1 000 ms

Telegram data are defined in Figure 63, Figure 64 and Figure 65.

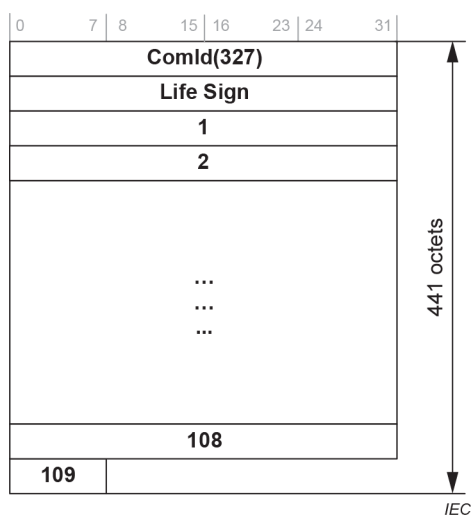


Figure 63 – Byte alignment test of 441 bytes length telegram data

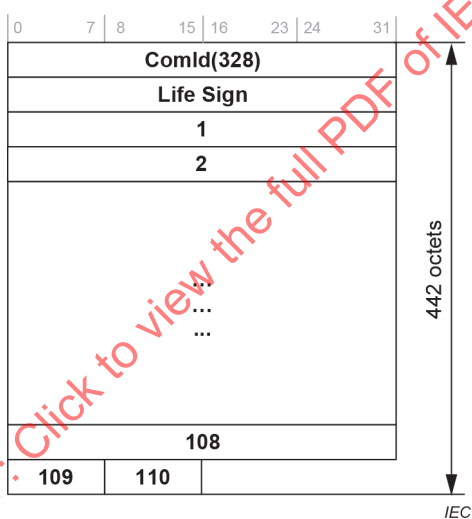


Figure 64 – Byte alignment test of 442 bytes length telegram data

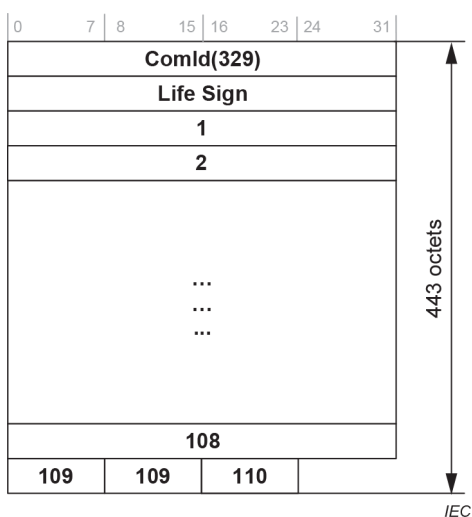


Figure 65 – Byte alignment test of 443 bytes length telegram data

9.6.3.8 Timeout and validity test

Test topology: TTS.

Simulator send TRDP process data with ComId of 330. IUT receive the data from Simulator and reply a telegram to Simulator with ComId of 331 or 332, set reply data zero (validity is ZERO mode) if receive timeout, set reply data last received (validity is KEEP mode) if receive timeout.

The test procedure and criteria are shown in Table 69.

Table 69 – Timeout and validity test

Step	Action	Expected Result
1	Start IUT, Simulator send TRDP process data with ComId of 330 and period of 1 000 ms	
2	Start IUT, set validity is ZERO mode and receive TRDP process data from Simulator	
3	Simulator stop sending TRDP process data through disconnect the cable or other ways	The reply telegram data with ComId of 331 shall be zero
4	Simulator continues sending TRDP process data. Restart IUT, set validity is KEEP mode and receive TRDP process data from Simulator	
5	Simulator stop sending TRDP process data again through disconnect the cable or other ways	The life sign of ComId 332 will be the same value with the last received telegram of ComId 330

Telegram parameters are defined in Table 70, Table 71 and Table 72.

Table 70 – Timeout and validity test telegram parameters

Telegram type	'Pd'
ComId	330
Dataset	CONFTEST_TIMEOUT_BEHAVIOR
Dataset length	1 024 Bytes
Destination IP	239.255.3.30
Cycle time	1 000 ms
Timeout value	5 000 ms

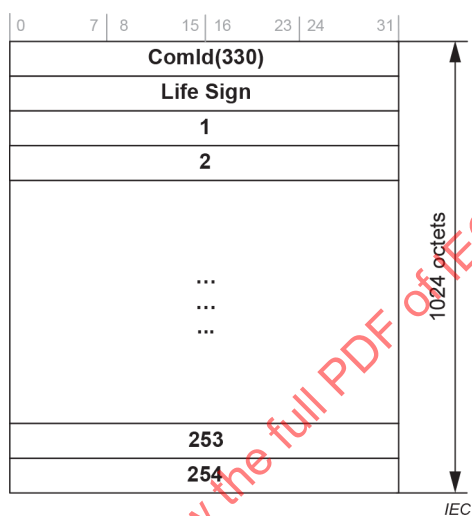
Table 71 – Timeout and validity test reply telegram parameters(ZERO mode)

Telegram type	'Pd'
ComId	331
Dataset	CONFTEST_TIMEOUT_BEHAVIOR_ZERO
Dataset length	1 024 Bytes
Destination IP	239.255.3.31
Cycle time	1 000 ms

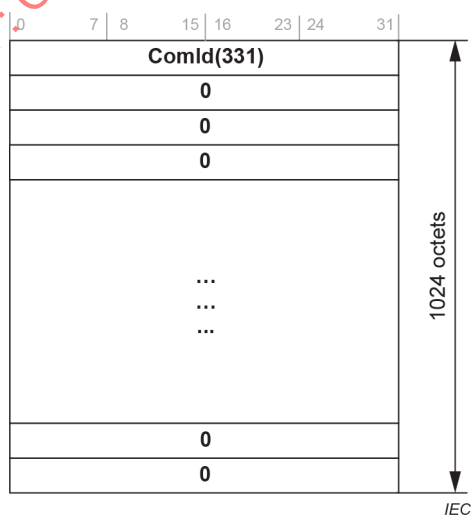
Table 72 – Timeout and validity test reply telegram parameters(KEEP mode)

Telegram type	'Pd'
ComId	332
Dataset	CONFTEST_TIMEOUT_BEHAVIOR_KEEP
Dataset length	1024 Bytes
Destination IP	239.255.3.32
Cycle time	1 000 ms

Telegram data definition as in Figure 66.

**Figure 66 – Dataset of CONFTEST_PUSH_FORMAT**

Reply telegram data (ZERO mode) are defined in Figure 67.

**Figure 67 – Dataset of CONFTEST_TIMEOUT_BEHAVIOR_ZERO**

Reply telegram data (KEEP mode) are defined in Figure 68.

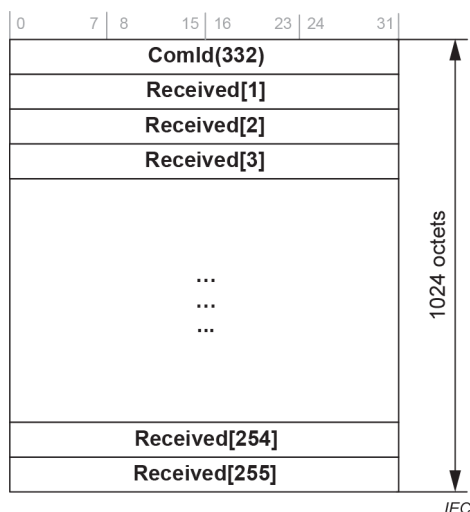


Figure 68 – Dataset of CONFTEST_TIMEOUT_BEHAVIOR_KEEP

9.6.3.9 Receiving port IP filtering test (Option)

Test topology: TTS.

Simulator send TRDP process data with ComId of 333. IUT receive the data from Simulator and reply a telegram to Simulator with ComId of 334, and set received flag of reply telegram 0x55555555 if received, set received flag 0xAAAAAAAA if timeout.

The test procedure and criteria are shown in Table 73.

Table 73 – Receiving port IP filtering test

Step	Action	Expected Result
1	Simulator send TRDP process data with ComId of 333 and period of 1 000 ms	
2	Start IUT, receive the data from Simulator with no filtering IP	The received flag of reply telegram with ComId of 334 shall be 0x55555555
3	ReStart IUT, receive the data from Simulator with filtering IP same with the IP address of Simulator	The received flag of reply telegram with ComId of 334 shall be 0x55555555
4	ReStart IUT, receive the data from Simulator with filtering IP different from the IP address of Simulator	The received flag of reply telegram with ComId of 334 shall be 0xAAAAAAAA

Telegram parameters are defined in Table 74 and Table 75.

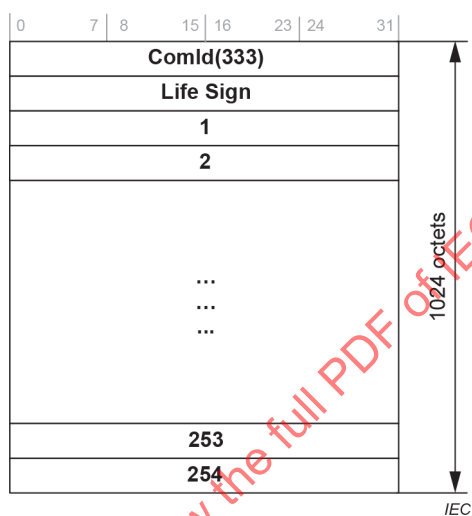
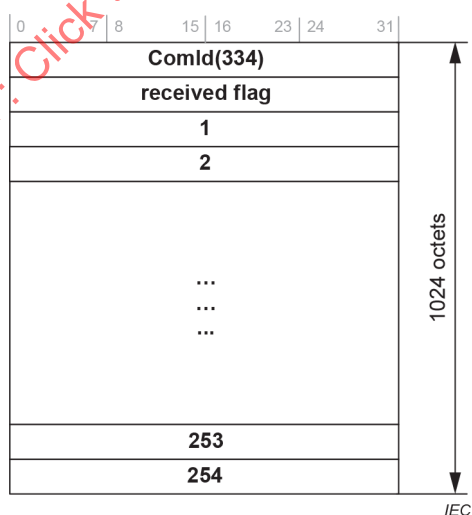
Table 74 – Receiving port IP filtering test telegram parameters

Telegram type	'Pd'
ComId	333
Dataset	CONFTEST_IP_FILTER
Dataset length	1 024 Bytes
Destination IP	239.255.3.33
Cycle time	1 000 ms

Table 75 – Receiving port IP filtering test reply telegram parameters

Telegram type	'Pd'
ComId	334
Dataset	CONFTEST_IP_FILTER_REPLY
Dataset length	1 024 Bytes
Destination IP	239.255.3.34
Cycle time	1 000 ms

Telegram data are defined in Figure 69 and Figure 70.

**Figure 69 – Dataset of CONFTEST_IP_FILTER****Figure 70 – Dataset of CONFTEST_IP_FILTER_REPLY**

9.6.3.10 DSCP(QOS) and TTL test (Option)

Test topology: TTS.

The test procedure and criteria are shown in Table 76.

Table 76 – DSCP(QoS) and TTL test

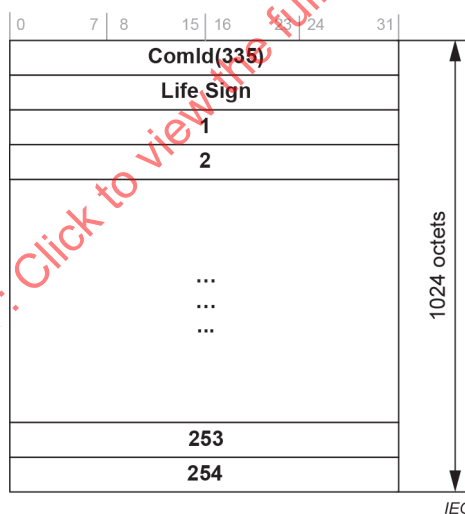
Step	Action	Expected Result
1	Start IUT, IUT send TRDP Process Data with ComId of 335 and period of 1 000 ms, set the QoS=5, and TTL=64	Monitor detect that the TOS segment of the telegram is 0xA0, and the TTL of the telegram is 64.

Telegram parameters are defined in Table 77.

Table 77 – DSCP(QoS) and TTL test telegram parameters

Telegram type	'Pd'
ComId	335
Dataset	CONFTEST_DSCP_TTL
Dataset length	1 024 Bytes
Destination IP	239.255.3.35
Cycle time	1 000 ms
QoS	5
TTL	64

Telegram data are defined in Figure 71.

**Figure 71 – DSCP(QoS) and TTL test telegram data**

9.6.3.11 Redundancy test (Option)

Test topology: TTS.

Test process and criterion as shown in Table 78.

Table 78 – Redundancy test

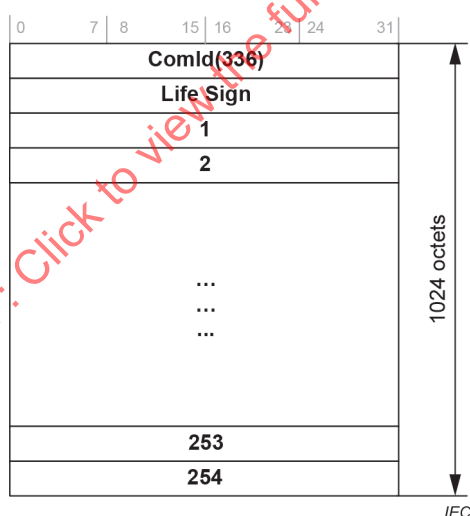
Step	Action	Expected Result
1	Start IUT, and set IUT as a master, then it will send TRDP process data with ComId of 336 and period of 1 000 ms	The data received by Simulator are same with the data sent by IUT
2	Set IUT as a slave, it will not send TRDP process data with ComId of 336 and period of 1 000 ms	No data package received by Simulator

Telegram parameter as in Table 79.

Table 79 – Redundancy test telegram parameter

Telegram type	'Pd'
ComId	336
Dataset	CONFTEST_REDUNDANT
Dataset length	1 024 Bytes
Destination IP	239.255.3.36
Cycle time	1 000 ms

Telegram data definition as in Figure 72.

**Figure 72 – Dataset of CONFTEST_REDUNDANT**

9.6.3.12 Point to point test

Test topology: TTS.

Test process and criterion as shown in Table 80.

Table 80 – point to point test of push pattern

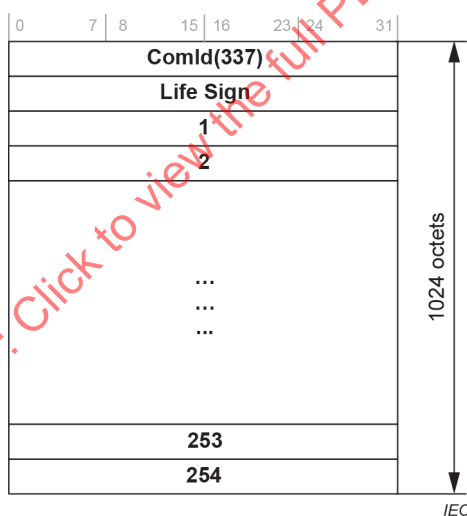
Step	Action	Expected Result
1	Start IUT, set the IUT IP address of 10.0.1.1 and set the Simulator IP address of 10.0.1.2	
2	IUT send TRDP process data with ComId 337	The Simulator will receive the telegrams from IUT with ComId 337

Telegram parameter as in Table 81.

Table 81 – point to point test telegram parameter

Telegram type	'Pd'
ComId	337
Dataset	CONFTEST_UNICAST
Dataset length	1 024 Bytes
Destination IP	10.0.1.2
Cycle time	1 000 ms

Telegram data definition as in Figure 73.

**Figure 73 – Dataset of CONFTEST_UNICAST**

9.6.4 PD pull pattern test cases

9.6.4.1 Pull pattern telegram test

Test topology: TTS.

Test process and criterion as shown in Table 82, Table 83 and Table 84.

Table 82 – Pull pattern telegram test

Step	Action	Expected Result
1	Start IUT, Simulator send TRDP request with ComId of 400 and period of 1 000 ms to IUT	The format of request telegram matches the push patternrequest telegram format of Table 75 (All telegrams from the first one shall matches)
2	IUT send the reply with ComId of 401 to Simulator when received the request from Simulator	The format of reply telegram matches the push pattern telegram format of Table 76 (All telegrams from the first one shall match)
3	Monitor receiving 10 000 reply packets from IUT	The period of data received should be 900 ms~1 100 ms

Table 83 – Pull pattern request telegram format content

No.	Item in TRDP header	Expected Result
1	sequenceCounter: The sequence counter	Shall be incremented with each sending of the process telegram, start value 0.
2	protocolVersion: The version of protocol	Same with the statement in TRDP PICS, example with 0x0100 or 0x0102.
3	msgType: The type of telegram	Request in pull pattern: fixed with 0x5072
4	ComId: The identifier of communication	400
5	etbTopoCnt: The ETB topography counter	Fixed with 0 when communicating within the consist
6	opTrnTopoCnt: The operational train topography counter	Fixed with 0 when communicating within the consist
7	datasetLength: The dataset length	1 024 bytes
8	reserved01: Reserved for extension	Fixed with 0
9	replyComId: The requested ComId	Request in pull pattern: 401
10	replyIpAddress: The reply IP address	Request in pull pattern: 0
11	headerFCS: Header checksum	Calculated checksum of header according to IEC 61375-2-3:2015, Clause A.3 (checksum itself is not included)

Table 84 – Pull pattern reply telegram format content

No.	Item in TRDP header	Expected Result
1	sequenceCounter: The sequence counter	Shall be incremented with each sending of the process telegram, start value 0.
2	protocolVersion: The version of protocol	Same with the statment in TRDP PICS, example with 0x0100 or 0x0102.
3	msgType: The type of telegram	Reply in pull pattern: fixed with 0x5070
4	ComId: The identifier of communication	401
5	etbTopoCnt: The ETB topography counter	Fixed with 0 when communicating within the consist
6	opTrnTopoCnt: The operational train topography counter	Fixed with 0 when communicating within the consist
7	datasetLength: The dataset length	1 024 bytes
8	reserved01: Reserved for extension	Fixed with 0
9	replyComId: The requested ComId	Reply in pull pattern: fixed with 0
10	replyIpAddress: The reply IP address	Reply in pull pattern: fixed with 0
11	headerFCS: Header checksum	Calculated checksum of header according to IEC 61375-2-3:2015, Clause A.3 (checksum itself is not included)

Telegram parameter as in Table 85 and Table 86.

Table 85 – Request telegram parameter of pull pattern test

Telegram type	'Pr'
ComId	400
Dataset	CONFTEST_PULL_FORMAT_REQUEST
Dataset length	1 024 Bytes
Destination IP	239.255.4.0
Reply ComId	0
Reply IP Address	0
Cycle time	1 000 ms

Table 86 – Reply telegram parameter of pull pattern test

Telegram type	'Pp'
ComId	401
Dataset	CONFTEST_PULL_FORMAT_REPLY
Dataset length	1 024 Bytes
Destination IP	239.255.4.1

Telegram data definition as in Figure 74 and Figure 75.

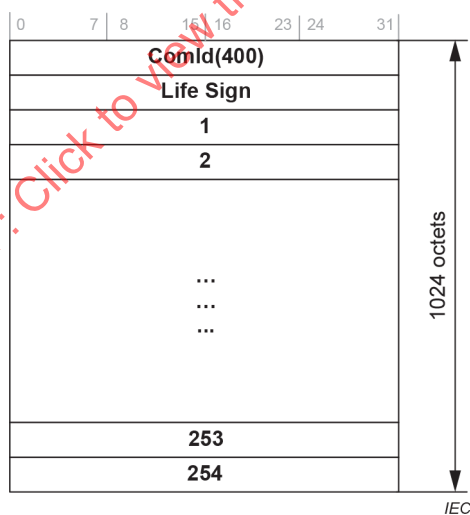


Figure 74 – Dataset of CONFTEST_PULL_FORMAT_REQUEST

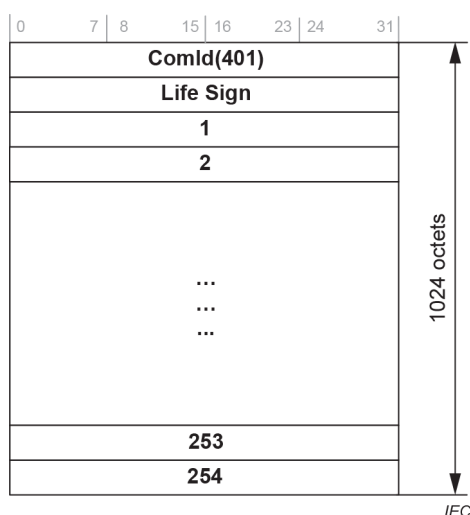


Figure 75 – Dataset of CONFTEST_PULL_FORMAT_REPLY

9.6.4.2 FCS check test

Test topology: TTS.

Simulator send TRDP request with ComId of 402. IUT receive the request from Simulator and then send a TRDP reply to Simulator with ComId of 403 if the FCS of TRDP PD header is correct, and set check value of reply telegram 0x55555555 if received.

Test process and criterion as shown in Table 87.

Table 87 – FCS check test

Step	Action	Expected Result
1	Start IUT, Simulator send TRDP request with ComId of 402 and period of 1000 ms and with correct FCS of TRDP PD header	The check value of TRDP reply telegram with ComId of 403 shall be 0x55555555
2	Simulator send TRDP request with ComId of 402 and period of 1000 ms and with incorrect FCS of TRDP PD header	IUT will discard the request telegram and not reply the telegram with ComId of 403 because of incorrect FCS.

Telegram parameter as in Table 88 and Table 89.

Table 88 – FCS check test request parameter

Telegram type	'Pr'
ComId	402
Dataset	CONFTEST_FCS_REQUEST
Dataset length	1 024 Bytes
Destination IP	239.255.4.2
Reply ComId	0
Reply IP Address	0
Cycle time	1 000 ms

Table 89 – FCS check test reply parameter

Telegram type	'Pp'
ComId	403
Dataset	CONFTEST_FCS_REPLY
Dataset length	1 024 Bytes
Destination IP	239.255.4.3
Cycle time	Depends on the period of request

Telegram data see Figure 76 and Figure 77.

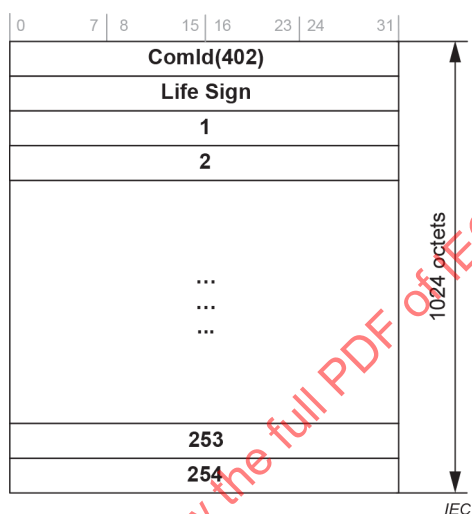


Figure 76 – Dataset of CONFTEST_FCS_REQUEST

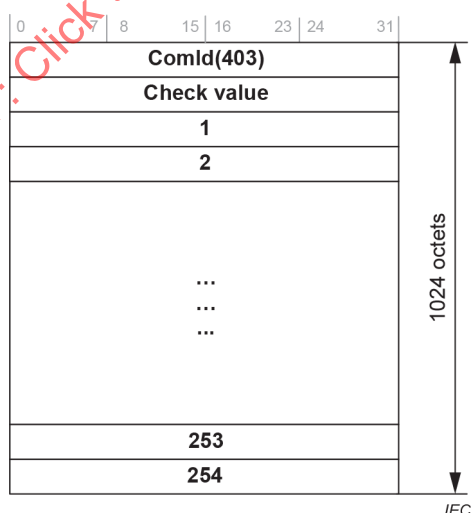


Figure 77 – Dataset of CONFTEST_FCS_REPLY

9.6.4.3 Protocol version test

Test topology: TTS.

Simulator send TRDP request with ComId of 404, IUT receive the request from Simulator and send a TRDP reply to Simulator with ComId of 405 and set check value of reply telegram 0x55555555 if received, set check value 0xAAAAAAAA if timeout.

Test process and criterion as shown in Table 90.

Table 90 – Protocol version test

Step	Action	Expected Result
1	Start IUT, the TRDP high version H2 is 1, low version L2 is 2: Simulator send TRDP request with ComId of 404 and period of 1 000 ms, the TRDP high version H1 is 1 (same as H2) and low version L1 is 1 (no specific requirement)	The check value of TRDP reply telegram with ComId of 405 shall be 0x55555555
2	Restart IUT, the TRDP high version H2 is 2, low version L2 is 1: Simulator send TRDP request with ComId of 404 and period of 1 000 ms, the TRDP high version H1 is 1 (different from H2) and low version L1 is 1 (no specific requirement)	IUT will discard the request telegram and not reply the telegram with ComId of 405

Telegram parameter as in Table 91 and Table 92.

Table 91 – Protocol version test request telegram parameter

Telegram type	'Pr'
ComId	404
Dataset	CONFTEST_PROTO_VERSION
Dataset length	1 024 Bytes
Destination IP	239.255.4.4
Reply ComId	0
Reply IP Address	0
Cycle time	1 000 ms

Table 92 – Protocol version test reply telegram parameter

Telegram type	'Pp'
ComId	405
Dataset	CONFTEST_PROTO_VERSION
Dataset length	1 024 Bytes
Destination IP	239.255.4.5

Telegram data definition as in Figure 78 and Figure 79.

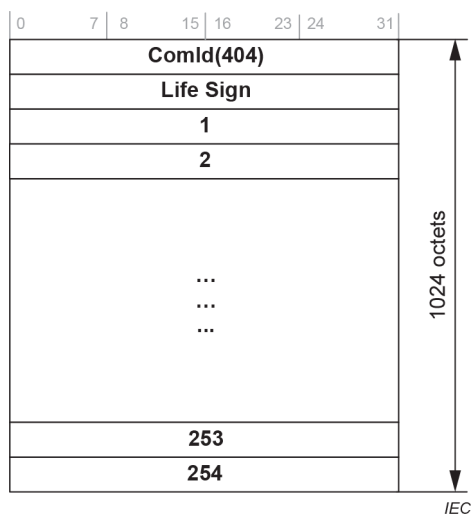


Figure 78 – Dataset of CONFTEST_PROTO_VERSION_REQUEST

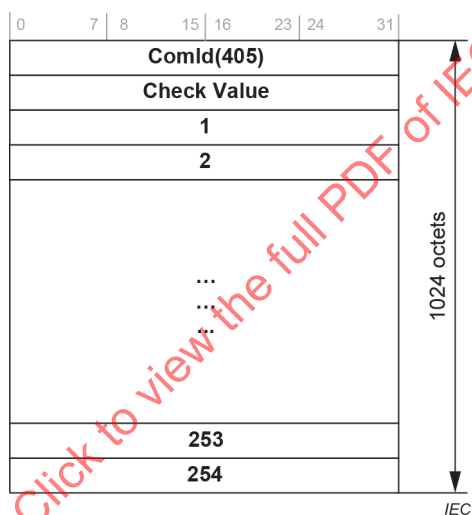


Figure 79 – Dataset of CONFTEST_PROTO_VERSION_REPLY

9.6.4.4 Topology counter test

Test topology: TTS.

Test process and criterion as shown in Table 93.