

***TRDP*****Train Real Time Data Protocol**

TRDP System Requirement Specification

Document reference no: TCN-TRDP2-D-BOM-O03-09

| | |
|----------------|-------------------|
| Author : | Armin-Hagen Weiss |
| Organisation : | Bombardier |
| Document date: | 23 May 2013 |
| Revision: | 9 |
| Status: | issued |

| Dissemination Level | | |
|---------------------|---|--|
| PU | Public | |
| PP | Restricted to other programme participants (including the Commission Services) | |
| RE | Restricted to a group specified by the consortium (including the Commission Services) | |
| CO | Confidential, only for members of the consortium (including the Commission Services) | |

DOCUMENT SUMMARY SHEET

This software requirement specification with the focus to train-wide IP communication is applicable to the Train Control Management System (TCMS) developed TCNopen initiative.

Participants

| Name and Surname | Organisation | Role |
|----------------------|--------------|----------|
| Armin-Hagen Weiss | BOM | Lead |
| Bernd Löhr | BOM | Reviewer |
| Florian Weispfenning | BOM | Reviewer |

History

| | | | |
|----|-----------|----------------|---|
| V1 | 14 Nov 11 | Armin-H. Weiss | First issue |
| V2 | 10 Dec 12 | Armin-H. Weiss | Document format changed Requirements from IEC61375-2-3 taken over |
| V3 | 12 Dec 12 | Armin-H. Weiss | Editorial changes |
| V4 | 23 Jan 13 | Armin-H. Weiss | TRDP1007: CRC shall be transmitted in little endian format |
| V5 | 06 Feb 13 | Armin-H. Weiss | TRDP1006: Usage of well known ports clarified Missing/changed requirements for MD handling added |
| V6 | 15 Mar 13 | Armin-H. Weiss | TRDP4203,4206 adapted, TCP specific requirements moved to chapter 3.4.2 |
| V7 | 06 May 13 | Armin-H. Weiss | Figure 1 updated. Issued |
| V8 | 12 May 13 | Armin-H. Weiss | Referenced documents corrected, Table of Contents updated |
| V9 | 23 May 13 | Armin-H. Weiss | Referenced documents corrected, Chapter 1.5 added |

Table of Contents

| | |
|---|-----------|
| TABLE OF CONTENTS | 3 |
| TABLE OF FIGURES | 4 |
| TABLE OF TABLES | 4 |
| 1. INTRODUCTION | 5 |
| 1.1. PURPOSE | 5 |
| 1.2. INTENDED AUDIENCE | 5 |
| 1.3. REFERENCES/RELATED DOCUMENTS..... | 5 |
| 1.4. ABBREVIATIONS AND DEFINITIONS | 5 |
| 1.5. CONVENTIONS | 7 |
| 2. TRDP DESCRIPTION | 8 |
| 2.1. USE CASES..... | 9 |
| 2.1.1. <i>Low End ED's (TRDP Light)</i> | 9 |
| 2.1.2. <i>Standard ED's</i> | 9 |
| 2.1.3. <i>High End ED's</i> | 10 |
| 3. SPECIFIC REQUIREMENTS..... | 11 |
| 3.1. GENERAL | 11 |
| 3.2. TRDP API..... | 12 |
| 3.3. PROCESS DATA TRANSMISSION | 13 |
| 3.4. MESSAGE DATA TRANSMISSION..... | 16 |
| 3.4.1. <i>UDP Message Data Transmission</i> | 22 |
| 3.4.2. <i>TCP Message Data Transmission</i> | 22 |
| 3.5. OS/HW ABSTRACTION INTERFACE | 23 |
| 3.6. UTILITY SUPPORT | 23 |
| 3.6.1. <i>TRDP XML - XML Configuration File Support</i> | 23 |
| 3.6.2. <i>TRDP Marshall – User Data Marshalling/Unmarshalling Support</i> | 23 |
| 3.6.3. <i>TRDP ADR - URI-IP Address Translation</i> | 24 |
| 3.6.4. <i>TRDP TTI - Train Topology Information Access</i> | 24 |
| 3.6.5. <i>TRDP SDT - Safe Data Transmission Support</i> | 24 |
| 3.6.6. <i>TRDP SDL - SW Download Support</i> | 24 |

Table of Figures

| | |
|--------------------------------------|---|
| Figure 1 TRDP Modular Structure..... | 8 |
|--------------------------------------|---|

Table of Tables

| | |
|--|---|
| Table 1: References..... | 5 |
| Table 2: Abbreviations and Definitions | 6 |

1. Introduction

1.1. Purpose

The purpose of this document is the definition of use cases, functional and non functional requirements concerning the IP communication stack (TRDP) with respect to create a flexible, modular software which can be easily ported, adapted and downsized. These requirements shall be the basis for the TRDP stack development by the TCNOpen initiative.

1.2. Intended Audience

This document is intended for TCMS engineers, system engineers and engineers responsible for the software development within the TCNOpen initiative.

1.3. References/Related Documents

| Reference | Number | Title |
|-----------|--------------|-------------------------|
| [Wire] | IEC61375-2-3 | TRDP Protocol (Annex A) |
| | | |
| | | |

Table 1: References

1.4. Abbreviations and Definitions

| Abbreviation. | Definition |
|---------------|-------------------------------------|
| API | Application Programming Interface |
| CCU | Central Computing Unit |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| ED | End Device |
| ETBN | Ethernet Train Backbone Node |
| FRG | Functional Redundancy Group |
| HMI | Human Machine Interface |
| ID | Identifier |
| IGMP | Internet Group Management Protocol |
| IP | Internet Protocol |
| MCG | Mobile Communication Gateway |
| MD | Message Data |

| Abbreviation. | Definition |
|---------------|---|
| NRTOS | Non-RealTime Operating System |
| PD | Process Data |
| QoS | Quality of Service |
| RTOS | Real Time Operating System |
| SDT | Safe Data Transmission |
| SNMP | Simple Network Management Protocol |
| TCN | Train Communication Network |
| TOS | Type Of Service |
| UDP | User Datagram Protocol |
| UIC | Union Internationale de Chemins de Fer (railway standardisation body) |
| URI | Universal Resource Identifier |
| XML | eXtensible Markup Language |
| API | Application Programming Interface |
| CCU | Central Computing Unit |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| ED | End Device |
| ETBN | Ethernet Train Backbone Node |
| FRG | Functional Redundancy Group |
| HMI | Human Machine Interface |
| ID | Identifier |
| IGMP | Internet Group Management Protocol |
| IP | Internet Protocol |
| MCG | Mobile Communication Gateway |
| MD | Message Data |
| NRTOS | Non-RealTime Operating System |
| PD | Process Data |
| QoS | Quality of Service |
| RTOS | Real Time Operating System |
| SDT | Safe Data Transmission |
| SNMP | Simple Network Management Protocol |
| TCN | Train Communication Network |
| TOS | Type Of Service |
| UDP | User Datagram Protocol |
| UIC | Union Internationale de Chemins de Fer (railway standardisation body) |
| URI | Universal Resource Identifier |
| XML | eXtensible Markup Language |

Table 2: Abbreviations and Definitions

1.5. Conventions

| | |
|----------------------------|--|
| shall or must | strictly required, mandatory |
| should | desired, recommended (dispensable if well justified) |
| can, could | possible |
| may | optional, allowed, acceptable |
| must not, shall not | strictly forbidden |
| should not | unwanted (only acceptable if well justified) |
| can not | impossible (physically or technically) |

2. TRDP Description

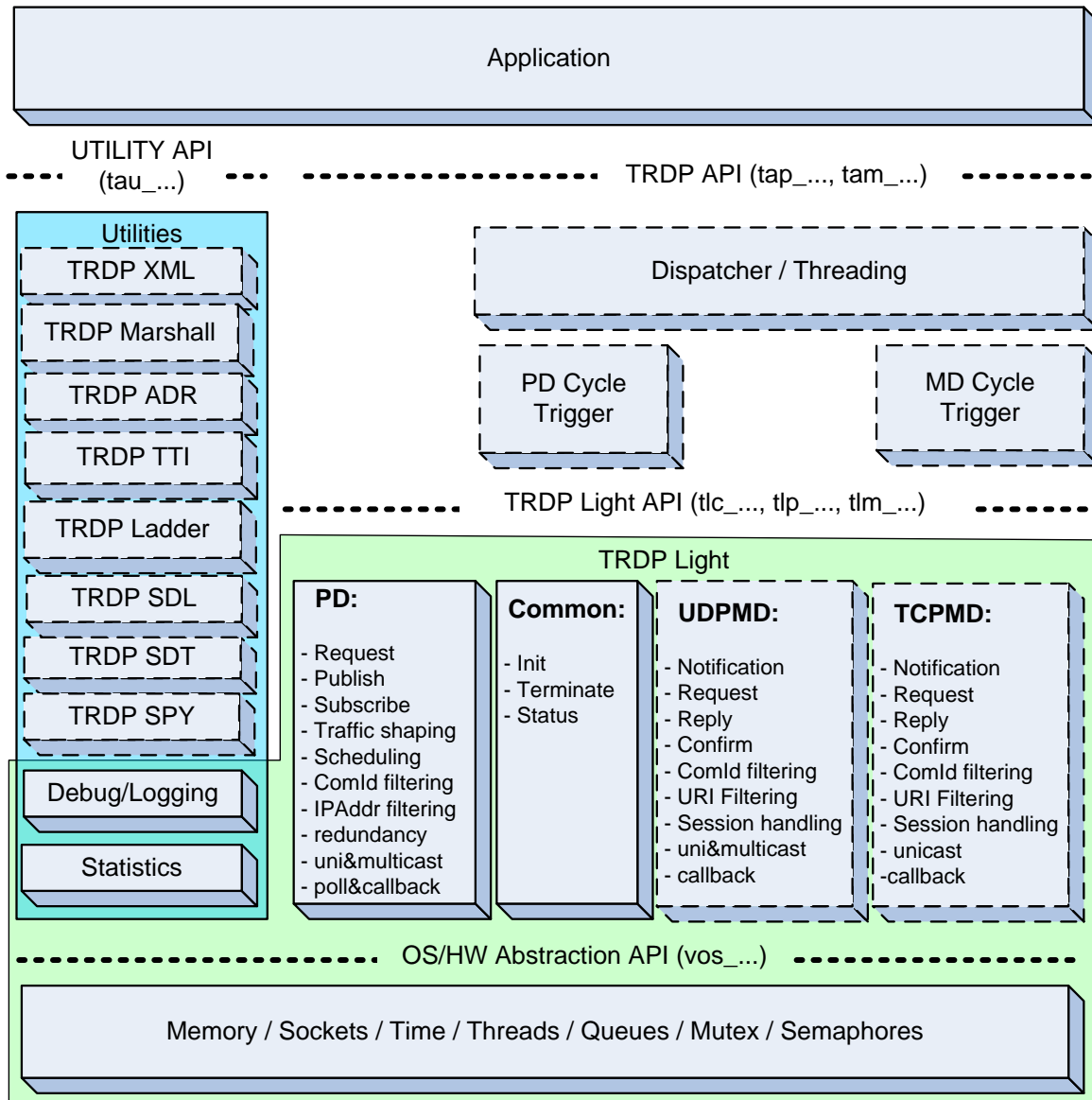


Figure 1 TRDP Modular Structure

2.1. Use Cases

In the following chapters the functionalities on the different end device (differentiated by their performance) are described.

2.1.1. Low End ED's (TRDP Light)

- PD (unicast, multicast, request)
- PD scheduling, shaping, timeout supervision
- PD base cycle triggering from application
- optional MD
 - MD base cycle triggering from application
 - MD using only call back interface
- Only IP addressing
- Only ECN (no ETB) communication
(ETB communication only done by standard and high end ED's !)
- Basic TRDP status & statistics provision via PD
- Optional extended TRDP status & statistics provision via MD
- time synchronisation via PD
- SW download by TFTP, manual check if files transferred completely and correctly
- No file system (only binary code)
- Optional SDT receive
- Necessary marshalling/unmarshalling to be done in application

2.1.2. Standard ED's

- PD (unicast, multicast, request)
- PD cycle handling
- MD (unicast and multicast)
- MD session handling
- Queue usage for MD

- Redundancy support
- URI addressing
- ECN & ETB communication
- Marshalling/Unmarshalling
- Basic TRDP status & statistics provision via PD
- Extended TRDP status & statistics provision via MD
- XML configuration file
- time synchronisation via PD
- SW download by specific protocol to ensure completeness and correctness of files
- Optional SDT
- Optional train to wayside communication

2.1.3. High End ED's

Like Standard ED's but with the following features added

- multithreading support
- time synchronisation via NTP
- Web services

3. Specific Requirements

3.1. General

| ID | Requirement |
|----------|--|
| TRDP1001 | TRDP shall have a modular structure that allows to leave out MD communication as well as one or more of the optional components without source code changes. |
| TRDP1002 | TRDP shall support optional a configuration via an XML-File. |
| TRDP1003 | TRDP shall use an OS/HW abstraction API to separate all OS/HW depending interfaces and to allow so easy portability and adaptability. |
| TRDP1004 | TRDP shall use optional PHY multicast address filtering. This is necessary to prevent overload of simple devices in networks without ring switches. |
| TRDP1005 | TRDP used well-known ports shall be configurable at stack initialisation time. As TRDP default ports shall be used the ports defined in IEC61375-2-3. |
| TRDP1006 | The ports shall be used in the following way: <ul style="list-style-type: none">• For receiving any process data telegrams and for receiving UDP message data notification, request and confirm telegrams the well-known port shall be used.• For receiving UDP message data reply telegrams the port the related request was sent from shall be used.• For sending any process data telegrams and for sending UDP message data notification, request and confirm telegrams a private source port different from the well-known port shall be used.• For sending UDP message data reply telegrams any source port different from the one the request was received from can be used.• TCP connections shall be established between a source port different from the well-known port and the well-known port as destination. |
| TRDP1007 | As frame check sequence the CRC32 according IEEE802.3 shall be used. The CRC calculation shall be done on the wire data – little endian format and byte alignment. Start value shall be 0xffffffff. The CRC itself shall be big endian. |
| TRDP1008 | TRDP shall support cyclic data transmission (PD). Specific requirements see Fehler! Verweisquelle konnte nicht gefunden werden.. |
| TRDP1009 | TRDP shall support optional event data transmission (MD). Specific requirements see Fehler! Verweisquelle konnte nicht gefunden werden.. |
| TRDP1010 | TRDP shall support optional safe data transmission. |
| TRDP1012 | TRDP shall provide optional PD interface for status and statistics access |
| TRDP1013 | TRDP shall provide optional MD interface for status and statistics access |
| TRDP1014 | TRDP shall provide API to access status and statistics |
| TRDP1015 | TRDP shall support a utility set to be used by the application. Specific requirements see 3.6 |
| TRDP1016 | Marshalling/unmarshalling shall be configurable as option for each telegram |
| | |

3.2. *TRDP API*

| ID | Requirement |
|----------|-------------|
| TRDP0201 | |

3.3. Process Data Transmission

| ID | Requirement |
|-----------|---|
| TRDP3001 | TRDP PD shall support push communication pattern. |
| TRDP3001 | TRDP PD push shall support point to point communication, cyclic without acknowledge, source knows the sink |
| TRDP3002 | TRDP PD push shall support point to multipoint communication, cyclic without acknowledge, source knows the sink, e.g.redundancy groups |
| TRDP3003 | TRDP PD push shall support point to multipoint communication, cyclic without acknowledge, source doesn't know the sink, e.g.redundancy groups |
| TRDP3004 | TRDP PD shall support pull communication pattern. |
| TRDP3005 | TRDP PD pull shall support point to point communication, without acknowledge, source knows the sink |
| TRDP3006 | TRDP PD pull shall support point to point communication, without acknowledge, source doesn't know the sink |
| TRDP3007 | TRDP PD pull shall support point to multipoint communication, without acknowledge, source knows the sink, e.g.redundancy groups |
| TRDP3008 | TRDP PD pull shall support point to multipoint communication, without acknowledge, source doesn't know the sink, e.g.redundancy groups |
| TRDP3009 | A publisher/subscriber shall use an IP unicast address for addressing a known subscriber/publisher. |
| TRDP3010 | A publisher/subscriber shall use an IP multicast address for addressing groups of known subscribers/publishers (e.g. redundancy groups). |
| TRDP3011 | A publisher/subscriber shall use an IP multicast address for addressing unknown subscribers/publishers. |
| TRDP3012 | TRDP shall use PD-PDU like defined in IEC61375-2-3 |
| TRDP3013 | The sequence counter: <ul style="list-style-type: none">• Shall be managed for sending process datagrams per ComId/MsgType at each requester/publisher.• Shall be managed (stored) for received process datagrams per SourceIPAddr/ComId/MsgType at each publisher/subscriber.• Shall be incremented with each sending of the process datagram. Datagrams sent in parallel via different subnets are sent with the same sequence counter to detect duplication at receiver side.• Can be used for the surveillance if the communication layer is still sending the PD. |
| TRDP3014 | The protocol version shall consist of: <ul style="list-style-type: none">• Higher significant octet: Version, incremented for incompatible changes• Lower significant octet: Release, incremented for compatible changes |
| TRDP3015 | MsgType shall define the type of the datagram. 'Pr' = PD Request 'Pp'=PD Reply 'Pd' = PD Data 'Pe' = PD Data (Error) |

| | |
|----------|---|
| | <p>Coded in ASCII for better reading with open protocol analyzer tools (e.g. Wireshark)</p> <p>NOTE: MsgType 'Pe' shall be used only for notification use and must not be sent over the bus</p> |
| TRDP3016 | <p>The ComId forms together with the source IP address an unique identifier within the train</p> <p>The ComId is application dependent but should be standardized in the application profiles. It should be the reference towards the following parameters of the PD datagram:</p> <ul style="list-style-type: none"> • structure (data set) (one-to-one relation) • source URI • sink URI • validity behavior (behavior in case of invalid data / timeout) • timeout • cycle time • redundancy (source is redundant) <p>(Note: source IP address might be changed by R-NAT)</p> |
| TRDP3017 | <p>The topology counter:</p> <ul style="list-style-type: none"> • shall be used (train addressing) as defined in standard part IEC 61375-2-5 • shall be set by the user • shall be set to 0 (zero) if no ETB is present • should be set to 0 (zero) for consist internal communication |
| TRDP3018 | <p>The dataset length:</p> <ul style="list-style-type: none"> • shall be the length of the user data set in number of octets without padding octets and FCS. • shall be the primary information about the user data size |
| TRDP3019 | <p>Reserved:</p> <ul style="list-style-type: none"> • shall be set to 0 |
| TRDP3020 | <p>The requested ComId:</p> <ul style="list-style-type: none"> • shall be used only in a PD request • shall be used as ComId in the reply • If set to 0, the ComId of the request shall be used for the reply |
| TRDP3021 | <p>The reply IP address:</p> <ul style="list-style-type: none"> • shall be used only in a PD request • shall be used as destination IP address in the reply • If set to 0, the source IP address of this request shall be used for the reply. |
| TRDP3022 | <p>The header frame check sequence:</p> <ul style="list-style-type: none"> • shall be calculated for the PD-PDU header • shall not include the HeaderFCS itself. |
| TRDP3023 | <p>If the user data length is not a multiple of 4 octets, octets with a value of 0 (zero) shall be appended until a multiple of 4 octets is reached (padding</p> |

| | |
|----------|---|
| | bytes). |
| TRDP3024 | The user data frame check sequence: <ul style="list-style-type: none">• shall be calculated for the PD-PDU user data set• shall not include the padding bytes nor FCS.• shall be the last 32-bit word of the telegram after the user dataset on a 4 byte alignment.• shall be calculated only if DatasetLength is > 0 |
| TRDP3025 | TRDP PD shall provide a poll mechanism to retrieve received PD |
| TRDP3026 | TRDP PD shall provide a callback mechanism to retrieve received PD |
| TRDP3027 | Process data are prepared cyclically by the publisher and shall be given to the TRDP layer calling the PD.prepare primitive. |
| TRDP3028 | To receive a request, the publisher shall subscribe for it. |
| TRDP3029 | Until receiving the first datagram for request matching to the filter criteria the related data shall be marked as invalid. |
| TRDP3030 | Also a request for already cyclically published PDU shall be possible. |
| TRDP3031 | Each incoming request datagram shall be answered by the TRDP layer using the available process datagram and the in the request datagram given IP address. |
| TRDP3032 | To receive the related reply for a request, the publisher needs to subscribe for it. Destination IP address and source IP address of the process datagram as possible filter criteria. |
| TRDP3033 | Until receiving the first request datagram matching to the filter criteria of the subscription, the related data shall be marked as invalid. |
| TRDP3034 | Before sending the request datagram the related reply data at subscriber TRDP layer shall be set to invalid. |
| TRDP3036 | Timeout supervision at subscriber TRDP layer shall be restarted after sending the request datagram. |
| | The TRDP PD request datagram can contain user data. |
| TRDP3037 | TRDP PD shall be only received after a previous subscription |
| TRDP3038 | Process data are prepared cyclically by the publisher and shall be given to the TRDP layer calling the prepare primitive. |
| TRDP3039 | The publisher TRDP layer shall send the data in the configured cycle to the configured address. |
| TRDP3040 | Any subscriber can subscribe for the process datagram using ComID, destination IP address and source IP address of the process datagram as possible filter criteria. |
| TRDP3041 | Timeout supervision at subscriber TRDP layer shall be started after subscription. |
| TRDP3042 | Timeout supervision at subscriber TRDP layer shall be restarted after receiving the related PD PDU. |
| TRDP3043 | Until receiving the first datagram matching to the filter criteria the data shall be marked as invalid. |
| TRDP3044 | TRDP PD shall support redundancy |
| TRDP3045 | The service primitives for redundancy handling shall be used in the same way for the publisher of the pull and the push pattern. |

| | |
|----------|--|
| TRDP3046 | If a redundant device enters the master state it shall call PD.activate to start publishing the redundant process data. |
| TRDP3047 | If a redundant device enters the slave state it shall call PD.deactivate to stop publishing the redundant PD. |
| TRDP3048 | Starting TRDP the redundant ComIds shall be initialized in a way that they are not published. |
| | TRDP PD shall optional support PD redundancy groups |
| TRDP3049 | To not to overload subscribers, the publisher of pushed PD-PDU shall apply a traffic shaping mechanism for equal distribution of the PD-PDU's over the time. |
| TRDP3050 | TRDP PD shall provide publish of one or more PD-PDU |
| TRDP3051 | TRDP PD shall provide unpublish of one or more PD-PDU |
| TRDP3052 | TRDP PD shall provide subscribe to one or more PD-PDU defined by its ComId |
| TRDP3053 | TRDP PD shall provide unsubscribe to one or more PD-PDU defined by its ComId |
| TRDP3054 | TRDP PD subscription shall support optional source address filtering |
| TRDP3055 | TRDP PD source address filtering should use PHY multicast filtering if available |
| | |
| | |

3.4. Message Data Transmission

| ID | Requirement |
|----------|--|
| TRDP4001 | TRDP MD transfer is limited to 64kByte |
| | A caller shall be able to define by the request type whether a reply is expected or not. |
| | A replier shall be able to define by the reply type whether a confirmation of its reply is expected or not. |
| TRDP4002 | TRDP MD shall support three MD transfer options: <ul style="list-style-type: none"> • request without reply ('notification') • request with reply but without confirmation ('request without confirmation') • request with reply and confirmation ('request with confirmation') |
| TRDP4003 | TRDP shall provide two mechanisms to transfer message data (via UDP and via TCP) the different service primitives of the two possibilities must not be mixed. |
| TRDP4004 | TRDP MD shall support push communication pattern. |
| TRDP4005 | TRDP MD push shall support point to point communication, sporadic with acknowledge, source knows the sink |
| TRDP4006 | TRDP MD push shall support point to point communication, sporadic without acknowledge, source knows the sink |
| TRDP4007 | TRDP MD push shall support point to multipoint communication, sporadic with acknowledge, source knows the sink |

| | |
|----------|--|
| TRDP4008 | TRDP MD push shall support point to multipoint communication, sporadic with acknowledge, source knows the sink |
| TRDP4009 | TRDP MD push shall support point to multipoint communication, sporadic without acknowledge, source doesn't know the sink |
| TRDP4010 | TRDP MD push shall support point to multipoint communication, sporadic with acknowledge, source doesn't know the sink |
| TRDP4011 | TRDP MD shall support pull communication pattern. |
| TRDP4012 | TRDP MD pull shall support point to point communication, sporadic with acknowledge, source knows the sink |
| TRDP4013 | TRDP MD pull shall support point to point communication, sporadic without acknowledge, source knows the sink |
| TRDP4014 | TRDP MD pull shall support point to multipoint communication, sporadic with acknowledge, source knows the sink |
| TRDP4015 | TRDP MD pull shall support point to multipoint communication, sporadic with acknowledge, source knows the sink |
| TRDP4016 | TRDP MD pull shall support point to multipoint communication, sporadic without acknowledge, source doesn't know the sink |
| TRDP4017 | TRDP MD pull shall support point to multipoint communication, sporadic with acknowledge, source doesn't know the sink |
| TRDP4018 | A caller shall use an IP unicast address or an IP multicast address for addressing known replier(s). |
| TRDP4019 | A caller shall use an IP multicast address for addressing unknown repliers. |
| TRDP4020 | A caller may use an IP multicast address for addressing a known replier redundancy group. |
| TRDP4021 | A replier shall respond to the caller's unicast address. |
| TRDP4022 | TRDP shall use PD-PDU like defined in IEC61375-2-3 |
| TRDP4023 | The TRDP MD sequence counter <ul style="list-style-type: none">• shall be incremented with each repetition of the message• shall be returned with the reply message shall start with value: 0 |
| TRDP4024 | The protocol version shall consist of: <ul style="list-style-type: none">• Higher significant octet: Version, incremented for incompatible changes• Lower significant octet: Release, incremented for compatible changes |
| TRDP4025 | MsgType shall define the type of the telegram. <ul style="list-style-type: none">• 'Mn' = Notification (Request without reply)• 'Mr' = MD Request with reply• 'Mp' = MD Reply without confirmation• 'Mq' = MD Reply with confirmation• 'Mc' = MD Confirm |

| | |
|----------|---|
| | <ul style="list-style-type: none"> • ‘Me’= MD error <p>NOTE; Coded in ASCII for better reading with open protocol analyzer tools (e.g. Wireshark)</p> <p>NOTE: MsgType ‘Me’ is only for notification use and must not be sent over the bus</p> |
| TRDP4026 | The ComId shall form together with the source IP address an unique identifier within the train. |
| TRDP4027 | <p>The ComId is application depending but should be standardized in the application profiles. It should be the reference for the following parameters of the MD telegram:</p> <ul style="list-style-type: none"> • structure (data set) • source URI • sink URI • timeout <p>(Note: source IP address might be changed by R-NAT)</p> |
| TRDP4028 | TRDP MD shall support unspecified MD (ComId = 0). |
| TRDP4029 | <p>The topology counter:</p> <ul style="list-style-type: none"> • shall be used (train addressing) as defined in standard part IEC 61375-2-5 • shall be set by the user • shall be set to 0 (zero) if no ETB is present • should be set to 0 (zero) for consist internal communication. |
| TRDP4030 | The DatasetLength shall contain the length of the user data set in number of octets without padding octets and FCS. |
| TRDP4031 | <p>The ReplyStatus value shall be set by the replier to report the execution result of a request message or by the caller sending a confirmation. The execution result is supplied by the replying application and transmitted to the requesting application in addition to the response message itself.</p> <p>In case of a TRDP error reply (‘Me’) the value is supplied by TRDP.</p> <p>< 0x0000 NOK, 0x0000 OK, > 0x0000 user reply status</p> <ul style="list-style-type: none"> • 1 - timeout • 2 - session abort • 5 – no memory • 7 – data invalid |

| | |
|----------|--|
| TRDP4032 | <p>The sessionId:</p> <ul style="list-style-type: none">• shall identify a “request-reply” or a “request-reply-confirm” session which is composed of a call session and a reply session.• shall be computed at caller side and reused at replier (listener) side.• shall be used at caller side to relate a reply message to the original request message• shall be used at replier side to identify a retransmission of the request in case the reply message was not received and to identify the confirm message• shall be a UUID according to RFC 4122, time based version |
| TRDP4033 | <p>The reply timeout shall be a μs value used in a request / reply session.</p> |
| TRDP4034 | <p>The source URI:</p> <ul style="list-style-type: none">• shall be the used for functional addressing• shall be a null terminated string• filling bytes at the end shall be set to 0• shall contain only the “user part” without “host part” and “@” |
| TRDP4035 | <p>The destination URI:</p> <ul style="list-style-type: none">• shall be used for functional addressing• shall be a null terminated string• filling bytes at the end shall be set to 0• shall contain only the “user part” without “host part” and “@” |
| TRDP4036 | <p>The header frame check sequence:</p> <ul style="list-style-type: none">• shall be calculated for the MD header• shall not include the HeaderFCS itself |
| TRDP4037 | <p>If the user data set length is not a multiple of 4 octets, octets with a value of 0 (zero) shall be appended until a multiple of 4 octets is reached (padding bytes).</p> |
| TRDP4038 | <p>The user data frame check sequence:</p> <ul style="list-style-type: none">• shall be calculated for the MD user dataset• shall not include the padding bytes nor FCS itself• shall be the last 32-bit word of the telegram after the user dataset on 4 byte alignment |

| | |
|----------|---|
| | <ul style="list-style-type: none"> shall be calculated only if DatasetLength is > 0 |
| TRDP4039 | If a TRDP MD user (caller) sends a notification, a notification message (MsgType 'Mn') shall be sent. The session identifier of the notification message shall be set to 0. |
| TRDP4040 | If a TRDP MD user (caller) sends a request message, a call session shall be opened and a request message (MsgType 'Mr') shall be sent. |
| TRDP4041 | To ensure that the session identifier is unique, each request-reply/request-reply-confirm session shall be identified by a 16 byte UUID according to RFC 4122, time based version. The session identifier is calculated at caller side in the TRDP layer, transmitted within each message and used at caller and replier side to identify the related call and reply session. |
| TRDP4042 | A timeout value for the reply message(s) shall be defined by the TRDP user (caller). |
| TRDP4043 | During the call session, the TRDP layer shall wait for incoming reply messages (MsgType 'Mp' or 'Mq'). All incoming reply messages related to the call session (identified by the received session id) shall be given immediately to the TRDP user. |
| TRDP4044 | If the number of expected incoming replies is reached and there are no more outstanding confirmations from TRDP user (caller), the call session shall be closed. |
| TRDP4045 | If the replier requests a confirmation (MsgType 'Mq'), the TRDP layer shall start a timer with the given confirm timeout time of the reply and wait for the confirmation from the TRDP user (caller). |
| TRDP4046 | The confirmation shall contain in the destination URI the source URI received in the reply query. |
| TRDP4047 | After getting the confirmation from TRDP user (caller), the TRDP layer shall send a confirmation message (MsgType 'Mc') to the replier. |
| TRDP4048 | A confirmation message must not contain user data and shall be sent only as unicast. |
| TRDP4049 | The TRDP user (caller) shall take care to provide the confirmation in time as indicated by the ReplyTimeOut parameter of the reply. |
| TRDP4050 | If the number of incoming replies is reached and the confirm timeout timer expires while waiting for outstanding confirmations from TRDP user (caller), the TRDP user (caller) shall be notified that confirmations are missing and the call session shall be closed. |
| TRDP4051 | If the timer of the call session expires because of a missing reply and the number of expected repliers is 1, the TRDP layer shall repeat the request up to two times before it notifies the TRDP user (caller) about the missing reply and closes the call session. |
| TRDP4052 | If TCP is used, there shall be no retransmission. If for TCP retransmission or continuing of the transmission is wished, this shall be done in a higher level service or in the application and not in TRDP. TRDP shall provide the interface to continue a transmission after the connection was lost. |

| | |
|----------|--|
| TRDP4053 | If the timer expires and the number of expected repliers is greater than 1 and less than the expected replies have been received, the TRDP Layer shall notify its TRDP user (caller) about the number of missing repliers. The call session shall be closed when all requested confirmations are sent or the confirmation timeout timer has timed out. |
| TRDP4054 | If the number of repliers is not known (parameter NoOfRepliers = 0 and the timer expires, the total number of replies shall be indicated to the TRDP user (caller) and the call session shall be closed. |
| TRDP4055 | If the TRDP layer receives a reply message (MsgType 'Mp' or 'Mq') without having opened a call session for the indicated session id, the message shall be discarded. |
| TRDP4056 | Each TRDP user (replier) that wants to receive MD shall register as listener for MD sent to a specific URI (multicast or unicast) or for MD of a specific ComId. |
| TRDP4057 | All incoming messages shall be checked against registered listeners. Any message to a not registered listener shall be discarded. |
| TRDP4058 | If a notification message (MsgType 'Mn') is received, the message shall be passed to the related listener. |
| TRDP4059 | If a request message (MsgType 'Mr') is received, a reply session shall be opened using the received session id, a timeout timer with the received 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. |
| TRDP4060 | If a request message (MsgType 'Mr') is received for an open reply session before the TRDP layer received the MD.reply from the listener, the request message shall be discarded. |
| TRDP4061 | After receiving the MD.reply from the TRDP user (listener), the TRDP layer shall 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. |
| TRDP4062 | A reply message shall be sent only as unicast. |
| TRDP4063 | If a request message (MsgType 'Mr') is received for an already open reply session and a different sequence counter after the TRDP layer received the MD.reply from the TRDP user (replier), the TRDP layer shall repeat sending the reply message (MsgType = 'Mq' or 'Mp') to the caller with an incremented sequence counter. After resending the reply message the reply timeout timer shall be restarted. |
| TRDP4064 | If a request message (MsgType 'Mr') is received for an already open reply session and the same sequence counter like received before, the request message shall be discarded. |
| TRDP4065 | The MD.reply of a listener shall use an IP address unique URI as source URI. |
| TRDP4066 | If a confirmation message (MsgType 'Mc') is received for an open reply session with the received session id the related listener shall be notified and the session shall be closed. |

| | |
|----------|---|
| TRDP4066 | If a confirmation message (MsgType 'Mc') is received without having an open reply session with the received session id, the message shall be discarded. |
| TRDP4067 | If the reply timeout timer of a reply session expires, the reply session shall be closed. In case of a timeout because of a missing confirmation or a missing MD.reply of the listener, the listener shall be notified. |
| TRDP4068 | TRDP MD interface shall provide ComId addressing/listening |
| TRDP4069 | TRDP MD interface shall provide user part addressing/listening |
| TRDP4070 | TRDP MD shall optional support redundancy/FRG |
| | |
| | |
| | |

3.4.1. UDP Message Data Transmission

| ID | Requirement |
|----------|---|
| TRDP4101 | UDP MD transmission shall support unicast and multicast addressing. |

3.4.2. TCP Message Data Transmission

| ID | Requirement |
|----------|--|
| TRDP4201 | TCP MD transmission shall support unicast addressing. |
| TRDP4202 | Using TCP for data transmission requires an established TCP connection between the caller and the replier device. So the first call to a specified device shall open a connection if there was no connection established before. |
| TRDP4203 | As long as a TCP connection to another device in the required direction is open, it should be reused for all further calls to this device. |
| TRDP4204 | <p>The caller shall close an existing TCP connection (active end) in the following cases:</p> <ul style="list-style-type: none"> • 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 was not used anymore for a defined time. |
| TRDP4205 | The replier shall use the connection opened by the caller |
| TRDP4206 | <p>The replier shall close an existing TCP connection (passive end) in the following cases:</p> <ul style="list-style-type: none"> • 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. |

3.5. OS/HW Abstraction Interface

| ID | Requirement |
|----------|---|
| TRDP5001 | OS/HW abstraction interface shall include socket handling |
| TRDP5002 | OS/HW abstraction interface shall include memory handling |
| TRDP5003 | OS/HW abstraction interface shall include timer handling |
| TRDP5004 | OS/HW abstraction interface shall optional include task handling |
| TRDP5005 | OS/HW abstraction interface shall optional include mutex/semaphore handling |
| TRDP5006 | OS/HW abstraction interface shall optional include mailbox handling |
| | |

3.6. Utility support

| ID | Requirement |
|----------|--|
| TRDP6001 | Communication framework shall include optional utility to read in XML configuration file(s) controlled by application. |
| TRDP6002 | Communication framework shall include optional utility for marshalling/unmarshalling |
| TRDP6003 | Communication framework shall include optional utility for IP-URI, URI-IP address translation |
| TRDP6004 | Communication framework shall include optional utility for SDT |
| TRDP6005 | Communication framework shall include optional debug/logging utility |
| TRDP6006 | Communication framework shall include payload FCS calculation utility |
| TRDP6007 | Communication framework shall include optional file upload/download utility |
| TRDP6008 | Communication framework shall include optional SW download utility |
| | |

3.6.1. TRDP XML - XML Configuration File Support

| ID | Requirement |
|----------|---|
| TRDP6101 | The configuration via a XML file is optional |
| TRDP6102 | XML file configuration shall support all data necessary for the configuration of TRDP, SDT and marshalling. |
| TRDP6104 | XML file configuration shall support multiple interfaces |
| TRDP6104 | XML file configuration shall support mapped devices |
| | |

3.6.2. TRDP Marshall – User Data Marshalling/Unmarshalling Support

| ID | Requirement |
|----------|--|
| TRDP6201 | The marshalling/unmarshalling is optional |
| TRDP6202 | The marshalling/unmarshalling function shall be supported for ComId's and for DataSetId's. |
| TRDP6203 | It should be possible to change the marshalling/unmarshalling data base during run time |

3.6.3. TRDP ADR - URI-IP Address Translation

| ID | Requirement |
|----------|-------------|
| TRDP6301 | |

3.6.4. TRDP TTI - Train Topology Information Access

| ID | Requirement |
|----------|-------------|
| TRDP6401 | |

3.6.5. TRDP SDT - Safe Data Transmission Support

| ID | Requirement |
|----------|--|
| TRDP6501 | TRDP MD interface shall meet the requirements stated in [wire] |

3.6.6. TRDP SDL - SW Download Support

| ID | Requirement |
|----------|-------------|
| TRDP6701 | |