# Introduction

This document sets forth the interface between the Temporary Speed Restrictions Management System(TSRMS) – Central Intelligent Kavach Network Monitoring System (CIKMS).

# Scope

This document provide the interface technical details for the TSRMS to CIKMS systems.

# Interface and system overview

Temporary Speed Restrictions Management System (TSRMS) will be the central server for each division control to implement and remove the Temporary Speed Restrictions(TSR) for the particular route or section due to any maintenance work, accidents and so on.

CIKMS will be the central Monitoring system (CMS) of Kavach subsystems which will be logs all the kavach field input and provide live monitoring of railway functioning, enables to playback options and also provides the Report generations in various formats.

The TSRMS shall update all the events functions to CIKMS such as,

1. All TSR Information,

2. Diagnostic information,

3. Fault Information.

# Constraints and limitations

1. The communcation interface with the TSRMS to CIKMS will be the log purpose only so if communication link failure and no acknowledge received from the CIKMS no action taken.

2. If communication link failure then data loss will happened.

# **Transaction processing and design flow**

# The transaction flow

# The message structure and protocols

| **Message Type** | **Value** | **Purpose** |
| --- | --- | --- |
| TSR Information message | 0x11 | TSR information message from TSRMS to NMS to report the TSR addition and delete statuses |
| TSRMS diagnostic event message | 0x12 | TSRMS diagnostic event message from TSRMS to NMS to report the events in system |
| TSRMS fault information message | 0x13 | TSRMS fault information message from TSRMS to NMS to report the faults in TSRMS System |
| NMS data acknowledge message | 0x1F | TSRMS fault information message from TSRMS to NMS to report the faults in |

# The message fields

**All TSRMS Information Message**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte No** | **Element Name** | **Size in Bytes** | **Description** | | | | |
| 00..01 | Start of Frame | (2-byte binary) | Value: (0xAAAAA) | | | | |
| 02 | Message Type | (1 byte binary) | Value: (0x11) | | | | |
| 03..04 | Message Length | (2-byte binary) | Length from " Message Type” to "CRC" field (inclusive both) | | | | |
| 05..06 | Sequence Number | (2-byte binary) | Value :( 0- 65535) | | | | |
| 07..08 | TSRMS Subsystem ID | (2-byte binary) | 1 - 65533 | | | | |
| 09..10 | CIKMS Subsystem ID | (2-byte binary) | 1-65533 | | | | |
| 11 | TSRMS System Version | (1 byte binary) | 1 to 7 | | | | |
| 12..14 | IST Date | (3-byte binary) | Decimal | | | | |
| 15..17 | IST Time | (3-byte binary) | Decimal | | | | |
| 18..19 | Section Identifier | (2-byte binary) | 1 - 65530 | | | | |
| 20..21 | Section - 1 Identifier | (2-byte binary) | 1 - 65530 | | | | |
| 22..23 | Section - 2 Identifier | (2-byte binary) | 1 - 65530 | | | | |
| 24 | TSR Identify (per section) | (1 Byte binary) | Values: 1 to 250 | | | | |
| 25..27 | Active TSR start date | (3-byte binary) | YY:MM:DD(IST) DD->MM->YY | | | | |
| 28..30 | Active TSR start time | (3-byte binary) | HH:MM:SS(IST) SS->MM->HH | | | | |
| 31..33 | Active TSR End date | (3 Bytes binary) | YY:MM:DD (IST) DD->MM->YY Applicable only when TSR type is intermittent | | | | |
| 34..36 | Active TSR end time | (3-byte binary) | HH:MM:SS(IST) SS->MM->HH | | | | |
| 34..35 | TSR starting Route Identifier | (2-byte binary) | 1 - 65530 | | | | |
| 36 | TSR direction | (1 byte binary) | 0:Not-used 1:Nominal-direction 2:Reverse-direction 3.Reserved | | | | |
| 37..38 | TSR Starting distance from start of the route | (2-byte binary) | In meter | | | | |
| 39..40 | Length of the TSR | (2-byte binary) | In meter | | | | |
| 41 | TSR applicable class of trains | (1 byte binary) | 0x01-Universal-Speed 0x02- Classified Speed | | | | |
| 42 | Universal TSR | (1 byte binary) | Universal – 0 to 200Kmph.  Value: | | | | |
| **Value** | | | | **Value** |
| 0 | | | | 0 Kmph |
| 1 | | | | 5 Kmph |
| 40 | | | | 200 Kmph |
| 41 – 253 | | | | Reserved |
| 254 | | | | Special value |
| 255 | | | | Not used |
| 43 | TSR for Class A Trains (Passenger Trains) | (1 byte binary) | Class A – 0 to 200Kmph | | | | |
| **Value** | | | **TSR** | |
| 0 | | | 0 Kmph | |
| 1 | | | 5 Kmph | |
| 40 | | | 200 Kmph | |
| 41 – 253 | | | Reserved | |
| 254 | | | Special Value | |
| 255 | | | Not | |
| 44 | TSR for Class B Trains (Loaded goods) | (1 byte binary) | Class B – 0 to 200Kmph | | | | |
| Value | | TSR | | |
| 0 | | 0 Kmph | | |
| 1 | | 5 Kmph | | |
| 40 | | 200 Kmph | | |
| 41 – 253 | | Reserved | | |
| 254 | | Special value | | |
| 255 | | Not used | | |
| 45 | TSR for Class C” Trains (Empty goods or special) | (1byte binary) | Class C – 0 to 200Kmph | | | | |
| Value | TSR | | | |
| 0 | 0 Kmph | | | |
| 1 | 5 Kmph | | | |
| 40 | 200 Kmph | | | |
| 41 – 253 | Reserved | | | |
| 254 | Special value | | | |
| 255 | Not used | | | |
| 46 | Whistle code | (1byte binary) | 0x00 indicates no whistle 0x01 indicates whistle should blow | | | | |
| 47 | TSR Status | (1byte binary) | 1-active 2-inactive 3- Deleted | | | | |
| 48..51 | Message CRC | (4byte binary) | 32-bit CCITT CRC | | | | |

# **TSRMS Diagnostic Packet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte No** | **Element Name** | **Size in Bytes** | **Description** |
| 00..01 | Start of Frame | (2-byte binary) | Value: [0xAAAA] |
| 02 | Message type | (1 byte binary) | Value: [0x12] |
| 03..04 | Message Length | (2-byte binary) | Length from “Message Type” to “CRC” field (inclusive both) |
| 05..06 | Sequence Number | (2-byte binary) | value: [0-65535) |
| 07..08 | TSRMS subsystem ID | (2-byte binary) | 1 -66533 |
| 09..10 | CIKMS subsystem ID | (2-byte binary) | 1 -65533 |
| 11 | TSRMS System | (1 byte binary) | 1-7 |
| 12 | Event Count (E) | (1 byte binary) | 1 - 10 |
| 13..15 | IST Date | (3-byte binary) | Decimal |
| 16..18 | IST Time | (3-byte binary) | Decimal |
| 19..20 | Event Identifier | (2-byte binary) | Refer the“TSRMS diagnostic event data” table |
| 21..21+ (m-1) | Event Data(m) | (m byte binary) | Refer the“TSRMS diagnostic event data” table |
| Last 4 bytes | Message CRC | (4-byte binary) | 32-bit CC ITT CRC |

**TSRMS Fault Message**

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte No** | **Element Name** | **Size in Bytes** | **Description** |
| 00..01 | Start of Frame | (2-byte binary) | Value: (0xAAAA) |
| 2 | Message Type | (1 byte binary) | Value: (0x13) |
| 03..04 | Message Length | (2-byte binary) | Length from " Message Type" to "CRC" field (inclusive both) |
| 05..06 | Sequence Number | (2-byte binary) | Value: (0 - 65533) |
| 07..08 | TSRMS subsystem ID | (2-byte binary) | 0 - 65533 |
| 09..10 | CIKMS Subsystem ID | (2-byte binary) | 1 - 65533 |
| 11 | TSRMS System Version | (1 byte binary) | 1 - 7 |
| 12 | Type of KAVACH subsystem (0x33 - TSRMS | (1 byte binary) | (0X33 - TSRMS) |
| 13 | Total Fault codes (F) | (1 byte binary) | 32 - bit CCITT CRC |
| 14..16 | IST Date | (3-byte binary) | Maximum number of faults will be 10 |
| 17..19 | IST Time | (3-byte binary) | Decimal |
| 20..21 | Fault Code | (2-byte binary) | Decimal |
| Last 4 bytes | Message CRC | (4-byte binary) | 32 - bit CCITT CRC |

**TSRMS ACK Message**

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte No** | **Element Name** | **Size in Bytes** | **Description** |
| 00..01 | Start of Frame | (2-byte binary) | Value: (0x AAAA) |
| 2 | Message Type | (1 byte binary) | Value: (0x1F) |
| 03..04 | Message Length | (2-byte binary) | Length from " Message Type" to "CRC" field (inclusive both) |
| 05..06 | Last received message sequence number | (2-byte binary) | Value:(0 -65535) |
| 07..08 | CIKMS Subsystem ID | (2-byte binary) | 1 - 65533 |
| 09..10 | TSRMS subsystem ID | (2-byte binary) | 1 - 65533 |
| 11 | TSRMS System Version | (1 byte binary) | 1 - 7 |
| 12 | Type-  of-KAVACH-subsystem-(0x33- TSRMS) | (1 byte binary) | (0X33 - TSRMS) |
| 13..16 | Message CRC | 1. byte binary) | 32 - bit CCITT CRC |

# **Communications and data transfer**

The communication will be handled through the wired communication using UDP Protocol with the CIKMS.

The packet structures are based on events taken placed in the TSRMS will be logged to the CIKMS.

The TSRMS will initiate the communication between CIKMS on any event occurs by addition and deletion of tsr , diagnostic information event,

**Class**

**CIKMS\_ADAPTOR class will connect udp socket and detects events using send function by accessing database for eventdata using CommDB for storing and receiving data.**

**Received acknowledgement data packet using receive funcion will be authenticated using crc validation and last received sequence number compared to store in database.**