TRANSNET
freight rail

TECHNOLOGY MANAGEMENT

SPECIFICATION

TRITON DATA COMMUNICATION PROTOCOL

Author:	Senior Engineer Technology Management	N. Gobhozi	
Approved:	Senior Manager IC&TM	R. Oosthuizen	
Authorised:	Principal Engineer Technology Management	B. Steyn	

Date: 04 November 2009

Circulation Restricted To:

Transnet Freight Rail

Transnet and Relevant Third Parties

Document Revision Sheet

Rev	Section(s)	Description	Date	Authorised
1.		First Release.	March 07	Dr Bennie Steyn
2.	2.3.1 2.3.2 2.3.3 2.4.2 2.4.6 2.4.7 2.4.8 2.7	New and modified diagnostic message types and description.	November 09	Dr Bennie Steyn
3.				
4.				

TABLE OF CONTENTS

DOCUMENT REVISION SHEET **2**

1 SCOPE **4**

1.1 SYSTEM OVERVIEW 4

2 SYSTEM CHARACTERISTICS **5**

2.1 DATA MESSAGE STRUCTURE 6
2.2 TELEGRAM HEADER 7
2.2.1 Header Bytes Description *7*
2.2.2 Telegram Type [1 Byte] *7*
2.2.3 Serial Number [3 Bytes in small-endian order] where default is [0x00 0x00 0x00] *7*
2.2.4 Sequence Number [- - - - 4 - - - - -] *8*
2.2.5 Message Type [- - - - - 5 - - - -] ; where default is 0x00 ..... *8*
2.2.6 Host ID [- - - - - - 6 - - -] *8*
2.2.7 Port ID [2 bytes in smallendian order] *8*
2.2.8 Gateway Message Identifier [GMI] *9*
2.3 TELEGRAM BODY 9
2.3.1 General [0 - 10] *9*
2.3.2 Application Specific [11 - 253] *10*
2.3.3 TRITON Processes [101 – 110 and 201 – 210] *10*
2.3.4 General [254 – 255] *10*
2.4 DESCRIPTION OF GENERAL DATA MESSAGE TYPES 10
2.4.1 Message Type 0x01: Locomotive Number and Locomotive Leading/ Trailing status. *10*
2.4.2 Message Type 0x02: Availability of wireless / network services *10*
2.4.3 Message Type 0x03: GPS NMEA String Broadcast - $GPRMC *11*
2.4.4 Message Type 0x04: GPS NMEA String Broadcast - $GPGGA *11*
2.4.5 Message Type 0x05: GPS NMEA String Broadcast - $GPVTG *12*
2.4.6 Message Type 0x06: Triton Track and Trace Messages *12*
2.4.7 Message Type 0x65: Triton Software Version - $STSSV *12*
2.4.8 Message Type 0x66: Triton Hardware Version - $STSHV *12*
2.4.9 Message Type 0xFE: Poll/ Message Request *13*
2.4.10 Message Type 0xFF: Open Format *13*
2.4.11 Application Specific Messages *13*
2.5 TELEGRAM TAIL [2 BYTES IN THE NORMAL BIG-ENDIAN ORDER] 13
2.6 ACKNOWLEDGEMENT 13
2.6.1 Hop Acknowledgement *13*
2.6.2 Acknowledgement format *14*
2.6.3 Delivery Acknowledgement *14*
2.7 EXAMPLES 14
2.7.1 Example 1 ..... *14*
2.7.2 Example 2 ..... *15*
2.8 SERVER TERMINAL 16

3 APPENDIX **18**

1 SCOPE

This document provides the description of the communication protocol used to interface with the Transnet Freight Rail supplied locomotive data communication system called TRITON.

1.1 System overview

Transnet Freight Rail has an integrated locomotive data communication and tracking system that was designed to fulfil its locomotive data communication and tracking requirements. The overall program to achieve this is known as the TRITON (**Tr**ain cond**i**tion and movemen**t** informati**on**) system.

The diagram in Figure 1 shows the system context diagram of the overall integrated system (TRITON). The diagrams in Figures 2 to 3 in the Appendix are conceptual with functional groupings and does not require strict physical conformance.

The context diagram shows all system components that enable proper and seamless communication with the locomotive peripheral systems. All peripheral devices on the locomotive shall utilize the communication structure for all their external communication needs. The diagram also indicates the communications structure adopted by Transnet Freight Rail for these peripheral devices on locomotives.

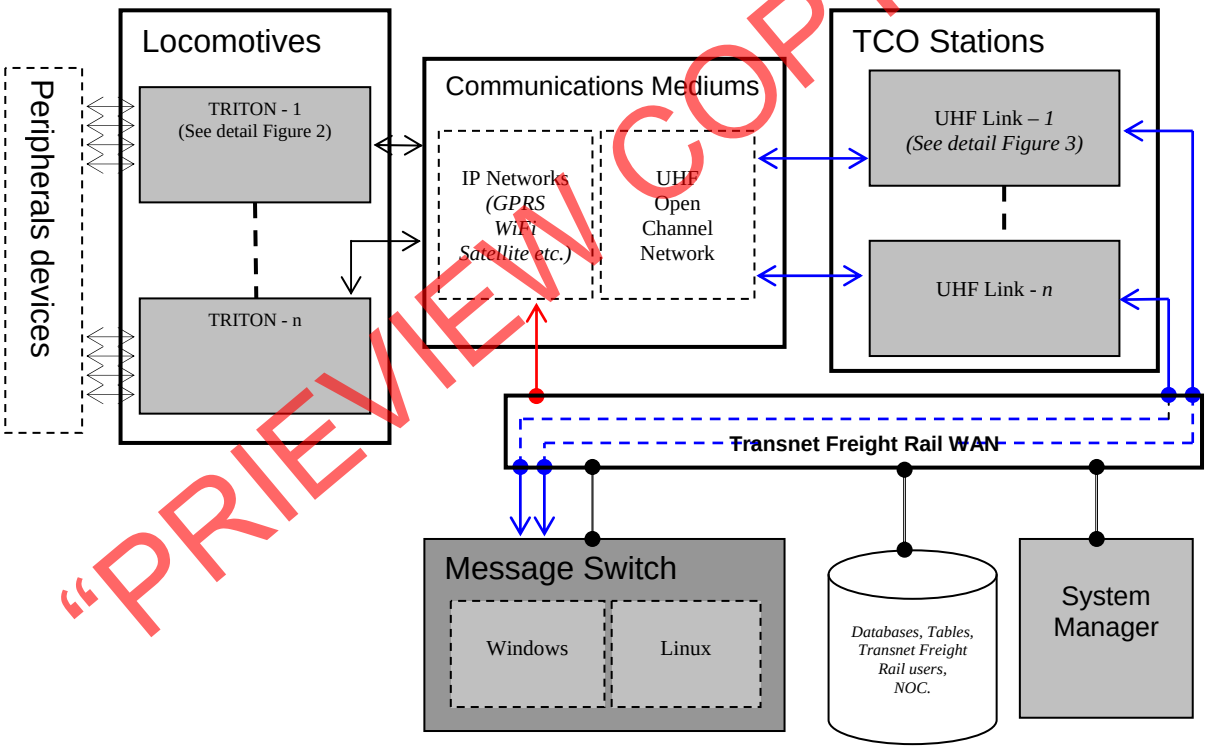


Figure 1: Locomotive Communications System detail

The communication infrastructure that Transnet Freight Rail provides entails a Local Area Network (LAN) on each locomotive. This network is connected to the Transnet Freight Rail Wide Area Network (WAN) using various data communications options. The communication units employed to connect the locomotive network to the Transnet Freight Rail network currently utilise GPRS, WiFi and UHF Radio as shown in Figure 2. If it is required Transnet Freight Rail might decide to add other technologies in conjunction with the above to connect the locomotive network to the Transnet Freight Rail land based Network.

The communications mediums are broadly labelled as IP network mediums and UHF Open Channel network mediums. The IP networks connect directly to the Transnet Freight Rail WAN and the security will be under Transnet Freight Rail control. The UHF open channel network has land based access points at various control centres in the country under the control of the Train Control Officer (TCO Stations in Figure 1). Data sent on the UHF open channel network is received by the UHF-Link software (TFE software), reconstructed into the required format and forwarded to the Transnet Freight Rail WAN.

Two servers on the Transnet Freight Rail WAN are available to execute the supplier messaging software, and are either a Linux, or Microsoft based server. If peripheral devices on the locomotive LAN need to send information to other servers, database tables or users on the Transnet Freight Rail WAN, the Triton Message Switch software shall route it to the required destinations.

The TRITON system has a GPS module and it broadcasts GPS data at defined intervals on its LAN ports.

2 System Characteristics

The Triton system provides Ethernet connectivity and a communication platform between devices in the locomotive. It consolidates all communication requirements of measurement and operation systems in the locomotive that require GPS data and/ or require a communication channel to send information from the locomotive to an application listening on the Transnet Freight Rail WAN. On the hardware layer Triton provides connectivity within systems as a LAN hub does in an office environment. As such there are network parameters that systems connecting to the network must be configured with to gain access to the network facilities. Below is a table listing the parameters that any system connected to Triton must configure, and the values that they must be configured to.

Network Gateway	192.168.0.2
Network Mask	255.255.255.0
IP Address	It is not DHCP allocated. A Triton Administrator will grant the user an IP address for a specific device that will be fixed and common to all those devices in all locomotives that it is installed in. The IP will be in the Gateway range of 192.168.0.XXX. The ranges .1 – .10 and .255 have been reserved for Triton specific applications.
Protocol Manager Port	6400, this is the port that all locomotive devices must transmit their messages to in the Triton protocol format if they intend to use Triton for sending their messages. Triton listens on this port, scans the predefined message header for validity and priority, then selects the appropriate channel to transmit it with.

All devices that are configured and have access to the Triton network have direct connection to applications on the Transnet Freight Rail WAN running on servers that have been granted access to the Wireless networks that Triton uses. Devices can implement TCP/IP, HTTP, FTP etc applications directly to their servers within the Guidelines stipulated above and those defined by the network administrators. Even though Triton provides multiple communication channels, i.e. UHF, GPRS and WiFi, there are guidelines regarding the usage of each medium and there is continuous information provided to users informing them of the availability of these channels. An example of a guideline would be that based on file size. Any large file downloads/ updates should only be executed when there is WiFi coverage because GPRS has limited bandwidth and has a cost element considering that the systems are deployed countrywide. Another example refers to importance of data. An important alarm condition can be flagged using protocols defined below, to enable transmission over UHF that has very low bandwidth and is only used when there is no other available communications channel.

The sections below provide a description of the protocol that systems configured to communicate using Triton should use to send messages. It is important to acknowledge that the protocol has to be used when transmitting a message via Triton, else other communication protocols such as TCP/IP can be used without the protocol, as briefly discussed above.

Triton provides certain common information to devices connected to it, such as GPS data. The sections below describe the exact format of this data. This information is broadcast every second to all applications that are listening on certain ports and can also be polled by applications that do not require to receive it all the time.

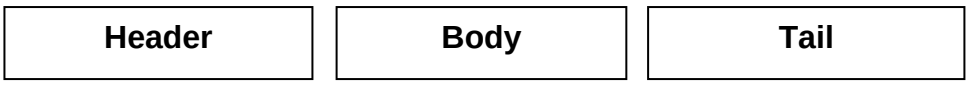
Guidelines have been defined to regulate the ports that the applications must listen on to receive broadcast information. At the time of compiling this specification, port 9009 is used to broadcast Triton and GPS information. The need for guidelines is that an application running on a certain port not using Triton broadcast might suddenly receive it and cause it to crash if another application selects the same port for their application. The system provider must indicate if they require any information that Triton provides to the Triton administrator, and must then also provide the ports that their application is listening on within the guidelines above.

2.1 Data Message Structure

Data Messages are transferred between Terminals using UDP/IP and wireless network protocols. More than one data message can be sent at the same time. The transmission of the data messages is called a 'telegram'. In other words - a telegram is a sequence of data messages being transmitted across a UDP/IP or wireless network together.

This document does not deal with the encapsulation of the communication network (Wireless or IP) that may be used. In other words, the Telegram and Data Message structures remain unchanged - irrespective of the transport protocol.

Each telegram has three parts:



The byte order of the telegram header is smallendian i.e. the least significant byte of each data field in the telegram is followed by a more significant byte.

2.2 Telegram Header

The Header has a fixed length - 10 bytes

0	1	2	3	4	5	6	7	8	9

2.2.1 Header Bytes Description

0	Telegram Type
1– 3	Serial Number - 3 bytes
4	Sequence Number
5	Message Type
6	Host ID
7 - 8	Port ID
9	Gateway Message Identifier

2.2.2 Telegram Type [1 Byte]

Byte 0 [7 6 5 4 3 2 1 0] = X + Y + Z, where

Bits 7 - 4 [7 6 5 4 - - - -] = X
Telegram Protocol

0x3--	TFR	Used as default
0xF--	Reserved	Acknowledgement

Bits 3 - 2 [- - - - 3 2 - -] = Y
Serial Number Type

0x-C-	OBCU	Reserved for Triton
0x-8-	Triton Terminal	Reserved for Triton
0x-4-	Locomotive	Reserved for Triton
0x-0-	None	used by all peripheral Devices

Bits 1 - 0 [- - - - - - 1 0] = Z
Priority

0x--3	High	Must obtain approval from TFR to use
0x--2	Medium	Message / data can be transmitted over GPRS
0x--1	Low	Message / data can only be transmitted over WiFi when available
0x--0	None	Triton selects transmission channel between Low and Medium

2.2.3 Serial Number [3 Bytes in small-endian order] ;where default is [0x00 0x00 0x00]

Bytes 1 - 3 [- 3 2 1 - - - - - -]

OBCU

RAMM Production Serial No	Defined by Triton

Triton Terminal

Triton Serial Number	Defined by Triton

2.2.4 Sequence Number [- - - - 4 - - - - -]

Byte 4

0 – 255; Each telegram built is given a sequence number by the originator of the telegram. The sequence number is set at Application level. Applications can control the message number associated with each message type. If the message sequence is broken or the data transmission service has been interrupted or the message discarded because it has timed out, then the user's message server application that receives to the messages can identify a message loss.

2.2.5 Message Type [- - - - - 5 - - - -] ;where default is 0x00

Byte 5

0 – 255 Each Application can generate data messages. To distinguish between the different data messages generated by a single application, the application can set a message type identifier. It is the responsibility of the Triton System Administrator to control the message types used by each application if the default is not used. The Message Type, Host ID and Port ID information is used by Triton Message Switches to route data messages that it receives to the intended destinations.

2.2.6 Host ID [- - - - - - 6 - - -]

Byte 6

Applications are hosted on terminals networked to the Triton Terminal on each locomotive. Data message traffic is bi-directional. Applications must be able to send (outbound) and receive (inbound) data messages. The Host ID is used by the MS Gateway to determine which Host the application resides on. The Host ID is the least significant byte of the IP address of the Applications Host e.g. [---. ---. ---. XXX], where XXX is the Host ID in binary.

It is the responsibility of the Application to set this field. The MS Gateway will check that the IP address of the UDP client sending the IP datagram is the same as the address in the header. If they differ the Triton Gateway will not ACK the Application and silently discard the Telegram. This control measure ensures that the Applications are resident on the allocated Hosts.

2.2.7 Port ID [2 bytes in smallendian order]

Byte 7 - 8 [- - - - - - - - 8 7 -]

Applications are configured to use specific ports of terminals networked to the Triton Terminal on each locomotive. The Port ID is used by the MS Gateway to determine which IP Port the Application resides on. The Port ID is the IP Port of the IP address of the Applications Host.

It is the responsibility of the Application to set this field. The MS Gateway will check that the IP Port of the UDP client sending the IP datagram is the same as the address in the header. If they differ then the MS Gateway will not ACK the Application and silently discard the Telegram. This control measure ensures that the Applications are using the correct ports.

2.2.8 Gateway Message Identifier [GMI]

Byte 9 [- - - - - - - - - 9]

The Gateway Message Identifier has three values 0, 1 or 2 and has the following meanings:

0	→	The message body does not contain a Locomotive Number i.e. Messages originating from Locomotive based applications.
1	→	The message body contains a Locomotive Number i.e. Messages transmitted to a Locomotive based application.
2	→	Reserved for Triton diagnostic tests.

If GMI = 1, the Locomotive Number is an ASCII string prefixing the data message body and is separated from the data messages in the body with a ',' delimiter. The Locomotive Number must exist in the Triton Master Switch Locomotive Register. If it does not exist, the telegram will be silently discarded.

The Triton Master Switch Server will translate the Locomotive Number into the Locomotive Identifier, strip the Locomotive Number message, repair the message header and forward the remainder of the message string using the routing logic of the Locomotive Identifier. The reverse process will occur when the Triton Master Switch Server is forwarding messages to the WAN.

2.3 Telegram Body

The Body has a variable length with a maximum of 1500 - 10 - 2 bytes = 1488 bytes. The body is the holding space for encapsulated 'data messages'. Each message type is distinct and contains data particular to the purpose of the message. Message Types are distinguished from one another by the message type byte (range 0 - 255) as discussed above. An application can send multiple messages in a single telegram, provided that it's server / listener application knows the length of each message and can identify each separately. The body of the message can contain anything that the two terminal applications understand between themselves. The data can be binary or ASCII.

It is the application's responsibility to manage concatenated messages. The Gateway will only examine the header information of the first message in a telegram in order to establish the routing of the telegram.

The following message types have been reserved for general use by all Applications -

2.3.1 General [0 - 10]

0x00	Reserved - Default value for Message Switch routing
0x01	Locomotive Numbers and Leading Loco
0x02	Availability of wireless / network services
0x03	GPS Broadcast - $GPRMC
0x04	GPS Broadcast - $GPGGA
0x05	GPS Broadcast - $GPVTG
0x06	Triton Track and Trace Messages
0x07	Triton Software Versions Message
0x08	Triton Hardware Version Message
0x09	Reserved for Triton Specific Messages
0x0A	Reserved for Triton Specific Messages

2.3.2 Application Specific [11 - 253]

0x0C Used by Transnet Freight Rail Train Definition Units

2.3.3 TRITON Processes [101 – 110 and 201 – 210]

0x65 Triton Software Revisions Messages - $STSSV
0x66 Triton Hardware Revisions Messages - $STSHV

2.3.4 General [254 – 255]

0xFE Data Poll
0xFF Open Format/ Broadcast

2.4 Description of General Data Message Types

General Data messages are messages that are useable by all Applications. These messages are broadcast on specific ports and may be polled by applications on different ports. The General Messages have a prefix, $STS => Spoornet Triton System message.

2.4.1 Message Type 0x01: Locomotive Number and Locomotive Leading/ Trailing status.

$STSTL, Triton Loco Number, Leading/Trailing where TL => Triton Loco number

1 = Locomotive Number
2 = Keyword - leading or trailing

This message is an ASCII string. The message begins with a dollar sign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited. All commas must be included as they act as markers. This message broadcasts the Locomotive number that the Triton is installed in as well as an indication of whether the Locomotive is Leading or Trailing in the train consist.

2.4.2 Message Type 0x02: Availability of wireless / network services

$STSSA,UHF,S,BO,S,AP,S,GPS,S,WF,S,OCU,S where SA => Service Availability

1 = UHF Service
2 = Y if available, N if unavailable
3 = BO (Back Office) Service
4 = Y if available, N if unavailable
5 = AP (GSM Access Point Network) Service
6 = Y if available, N if unavailable
7 = GPS Service
8 = Y if available, N if unavailable
9 = WF (WiFi) Service
10 = Y if available, N if unavailable
11 = OCU (OBCU) Service
12 = Y if available, N if unavailable

This message is an ASCII string. The message begins with a dollar sign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited. All commas must be included

as they act as markers. This message broadcasts the status and availability of TRITON's wireless communication channels and modules.

2.4.3 Message Type 0x03: GPS NMEA String Broadcast - $GPRMC

$GPRMC, hhmmss.ss, A, llll.ll, a, yyyyy.yy, a, x.x, x.x, ddmmyy, x.x, a*hh
Where RMC => Recommended Minimum Specific GPS/TRANSIT Data

1 = UTC of position fix
2 = Data status (V=navigation receiver warning)
3 = Latitude of fix
4 = N or S
5 = Longitude of fix
6 = E or W
7 = Speed over ground in knots
8 = Track made good in degrees True
9 = UT date
10 = Magnetic variation degrees (Easterly var. subtracts from true course)
11 = E or W
12 = Checksum

This message is an ASCII string conforming to the NMEA 0183 specification. The message begins with a dollar sign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited. All commas must be included as they act as markers.

2.4.4 Message Type 0x04: GPS NMEA String Broadcast - $GPGGA

$GPGGA, hhmmss.ss, llll.ll, a, yyyyy.yy, a, x, xx, x.x, x.x, M, x.x, M, x.x, xxxx*hh
Where GGA => Global Positioning System Fix Data

1 = UTC of Position
2 = Latitude
3 = N or S
4 = Longitude
5 = E or W
6 = GPS quality indicator (0=invalid; 1=GPS fix; 2=Diff. GPS fix)
7 = Number of satellites in use [not those in view]
8 = Horizontal dilution of position
9 = Antenna altitude above/below mean sea level (geoid)
10 = Meters (Antenna height unit)
11 = Geoidal separation (Diff. between WGS-84 earth ellipsoid and mean sea level. -=geoid is Below WGS-84 ellipsoid)
12 = Meters (Units of geoidal separation)
13 = Age in seconds since last update from diff. reference station
14 = Diff. reference station ID#
15 = Checksum

This message is an ASCII string conforming to the NMEA 0183 specification. The message begins with a dollar sign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited. All commas must be included as they act as markers.

2.4.5 Message Type 0x05: GPS NMEA String Broadcast - $GPVTG

$GPVTG,t,T,,,s.s,N,s.s,K*hh
Where VTG => Actual track made good and speed over ground

1 = Track made good
2 = Fixed text 'T' indicates that track made good is relative to true north
3 = not used
4 = not used
5 = Speed over ground in knots
6 = Fixed text 'N' indicates that speed over ground in knots
7 = Speed over ground in kilometres/hour
8 = Fixed text 'K' indicates that speed over ground is in hour/hour
9 = Checksum

This message is an ASCII string conforming to the NMEA 0183 specification. The message begins with a dollar sign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited. All commas must be included as they act as markers.

2.4.6 Message Type 0x06: Triton Track and Trace Messages

Triton Track and Trace messages are formatted and transmitted by the Triton TTTServer Module to a TRITON WAN server. The messages are sent from Triton to the main Switch Servers on the TFR WAN using any available communication channel.

This messages contains:

Locomotive number, GPS coordinates, date and time and speed of the locomotive.

2.4.7 Message Type 0x07: Triton Software Version - $STSSV

$STSSV,TPMxx,TSyy,PMzz,TTaa,GWbb,KGcc
Where the numerics following the software item is the last 2 digits of the version number.

1 = TPM – Triton Protocol Manager version 3.0.0.xx
2 = TS – Triton Switch version 3.0.0.yy
3 = PM – Triton Protocol Manager Server version 3.0.0.zz
4 = TT – Track and Trace Server version 3.0.0.aa
5 = GW – Gateway version 3.0.0.bb
6 = KG – Kavicomm Gateway version 3.0.0.cc

This message is an ASCII string. The message begins with a dollar sign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited. All commas must be included as they act as markers. This is a static message with its contents and broadcast interval defined in the windows registry.

2.4.8 Message Type 0x08: Triton Hardware Version - $STSHV

$STSHV,IPS,x,PS,x,FAN,x,WBIOS,y,OBCU,z
Where the numeric following the hardware item is the latest hardware version number.

1 = IPS,x - Intelligent Power Supply revision
2 = PS,x - 110v power supply revision

3 = FAN,x - Fan revision installed
4 = WBIOS,x - Wafer Bios version
5 = OBCU,x - OBCU revision

This message is an ASCII string. The message begins with a dollar sign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited. All commas must be included as they act as markers. This is a static message with its contents and broadcast interval defined in the windows registry.

2.4.9 Message Type 0xFE: Poll/ Message Request

A poll message may be sent by an Application to the Triton Protocol Manager for any of the General Message types. The body of the message is a single byte, which is the message type requested (polled).

2.4.10 Message Type 0xFF: Open Format

An open message may be sent by any Application to any other Application on the Triton WAN. This function can be used by applications that understand each other's messages and protocols.

2.4.11 Application Specific Messages

Applications may selects to identify any of their messages with a Message type number between 11 [0x0B] and 253 [0xFD]. A unique message type allows for message specific configuration of parameters such as the Message Keep Alive time. If the default message type of 0 [0x00] is not selected, then the user has to consult the Triton Administrator to obtain an available Message Type.

2.5 Telegram Tail [2 bytes in the normal Big-endian order]

The telegram tail is two bytes - a CCITT (Polynomial 0x1021) CRC check of the entire telegram.

2.6 ACKnowledgement

Data Message acknowledgement is implemented using two mechanisms:

1. Hop Acknowledgement
2. Delivery Acknowledgement

2.6.1 Hop Acknowledgement

A Hop Acknowledgement takes place between:

1. An Application communicating directly to a Gateway
2. Two Gateways communicating with one another
3. A Gateway and an Application

The purpose of the Hop Acknowledgement is to control the flow of data across a link between two nodes. The ACK indicates to the transmitting node that the receiving node has received the data message. i.e. The message has hopped across the link. The transmitting gateway then removes the message from its outbound queue and sends the next message.

If an ACK is not received the transmitting node retries to send the data message until the keep

alive expires. The receiving node is deemed to be unreachable and the data message is silently discarded. Applications must be disciplined. Retries must not occur more frequently than 10 seconds. i.e. If an ACK is not received within 10 seconds the transmitting node may retry.
There is no guarantee that the data message will reach its destination. The latter can be silently discarded on any broken link. In other words the delivery mechanism is best effort' and not guaranteed delivery.

2.6.2 Acknowledgement format

The header of the transmitted message is returned to the transmitting node by the receiving node with the protocol bits [7 6 5 4 - - - -] of byte 0, the telegram type, change to 0xF-- from 0x3-- as defined above.

2.6.3 Delivery Acknowledgement

Applications are responsible for providing delivery acknowledgement to the Application requiring the Acknowledgement. For example - an Application residing on a WAN Host may require an Application on a Locomotive Host to change its settings and acknowledge that it has received the instruction to change the settings. It is the responsibility of the Application on the Locomotive Host to respond to the Application on the WAN Host. The format of a delivery Acknowledgment is Applicant dependant and outside the scope of this specification.

2.7 Examples

Below are two examples showing messages transmitted between two applications using the Triton UDP Messaging protocol, and their acknowledgements.

The two applications have the following network parameters:

	Application 1	Application 2
IP Address	192.168.0.165	192.168.0.100
Port	9009	9000
Message Type	0	15

2.7.1 Example 1

Application 1 transmitted a message with the following Triton Protocol parameters:

Serial Number Type	-	Transnet Freight Rail
Priority	-	High
Serial Number	-	654321
Sequence Number	-	1
Message Type	-	0
Host ID	-	165
Host Port	-	9009
GMI	-	0
Message Body	-	Hello World

2.7.1.1 **Message 1**

The full telegram that was transmitted is:

33F1FB090100A531230048656C6C6F20576F726C64A0B8. Below is the analysis of the actual telegram that was transmitted to deliver the message above using the Triton Protocol. The 10 Byte header is: 33 F1 FB 09 01 00 A5 31 23 00, where:

Telegram Type OR Serial Number Type OR Priority	= 0x33	[1 byte]	
Serial Number (Small-endian order)	= 0xF1 0xFB 0x09	[3 bytes]	Note: The actual Serial Number is 09FBF1.
Sequence Number	= 0x01	[1 byte]	
Message type	= 0x00	[1 byte]	
Host ID	= 0xA5	[1 byte]	
Host Port (Small-endian byte order)	= 0x31 0x23	[2 bytes]	Note: The actual Port is 2331.
Gateway Message Identifier	= 0x00	[1 byte]	
Message Body	0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64	[11 bytes]	
CRC Check Bytes	0xA0 0xB8	[2 bytes]	

2.7.1.2 **Acknowledgement 1**

Application 2 responded with the following acknowledgment. The full acknowledgement that was sent is: **F0 F1 FB 09 01 00 A5 31 23 00 AF CE**

The Acknowledgement 10 Byte header	F0F1FB090100A5312300,	
Original message 10 Byte header	33F1FB090100A5312300 Note that the difference is the first byte that changed from 0x33 to 0xF0.	
CRC Check Bytes	0xAF 0xCE	[2 bytes]

2.7.2 Example 2.

Application 2 transmitted a message with the following Triton Protocol parameters:

Serial Number Type	-	Transnet Freight Rail
Priority	-	Medium
Serial Number	-	123456
Sequence Number	-	128
Message Type	-	15
Host ID	-	100

Host Port	-	9000
GMI	-	0
Message Body	-	1234567890

2.7.2.1 **Message 2**

The full telegram that was transmitted is:

3240E201800F64282300313233343536373839 30F898. Below is the analysis of the actual telegram that was transmitted to deliver the message above using the Triton Protocol.

The 10 Byte header is: 32 40 E2 01 80 0F 64 28 23 00, where:

Telegram Type OR Serial Number Type OR Priority	= 0x32	[1 byte]	
Serial Number (Smallendian order)	= 0x40 0xE2 0x01	[3 bytes]	Note: The actual Serial Number is 01 E2 40.
Sequence Number	= 0x80	[1 byte]	
Message type	= 0x0F	[1 byte]	
Host ID	= 0x64	[1 byte]	
Host Port (Smallendian byte order)	= 0x28 0x23]	[2 bytes	Note: The actual Port is 23 28.
Gateway Message Identifier	= 0x00	[1 byte]	

Message Body	0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x30	[10 bytes]
CRC Check Bytes	0xF8 0x98	[2 bytes]

2.7.2.2 **Acknowledgement 2**

Application 1 responded with the following acknowledgment. The full acknowledgement that was sent is: **F0 F1 FB 09 01 00 A5 31 23 00 AF CE.**

The Acknowledgement 10 Byte header		F040E201800F64282300
Original message 10 Byte header		3240E201800F64282300 Note that the difference is the first byte that changed from 0x32 to 0xF0.
CRC Check Bytes	0x31 0x10	[2 bytes]

2.8 Server Terminal

Triton sends all messages it receives from the Locomotives peripherals devices to its Master Switch on the Transnet Freight Rail WAN. The Switch manages the routing of all messages to and from Locomotive systems via Triton. The Triton Message Switch is used to manage routing tables and Locomotive identification information such that applications on the WAN only need to know the Loco number that an application resides on to communicate back to it.

Else each application would need to know all the devices' Triton IP addresses on all the communication networks. The Message switch scans the header to acquire information containing the source of the message.

The Triton system administrator has to define a route first on the Message Switch tables based on all Locomotive systems (TRITON Network) IP addresses, Port Numbers and Message Types information. Then the systems provider must provide a destination IP address when the listener for their system is installed. Once the link is defined then it is available for WAN applications as well to communicate back to Triton peripheral devices. An application on the WAN would send a message to the Message Switch in the following format:

Header (GMI = 1)	**Locomotive Number,**	**Message Body**	**Tail**

The format differs from the Locomotive to WAN format only in that there is a loco number field in ASCII and a comma before the body of the message. The Triton administrator will on request provide a Loco system provider the IP address and Port Number information of the Message Switch that their server application can communicate with.

3 Appendix

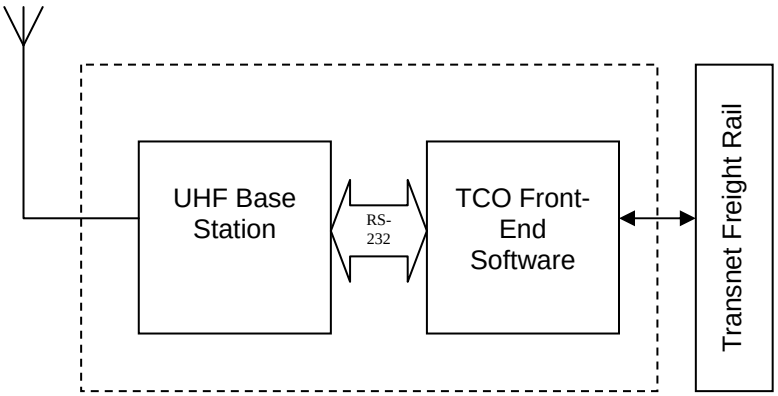


Figure 2: UHF Link detail

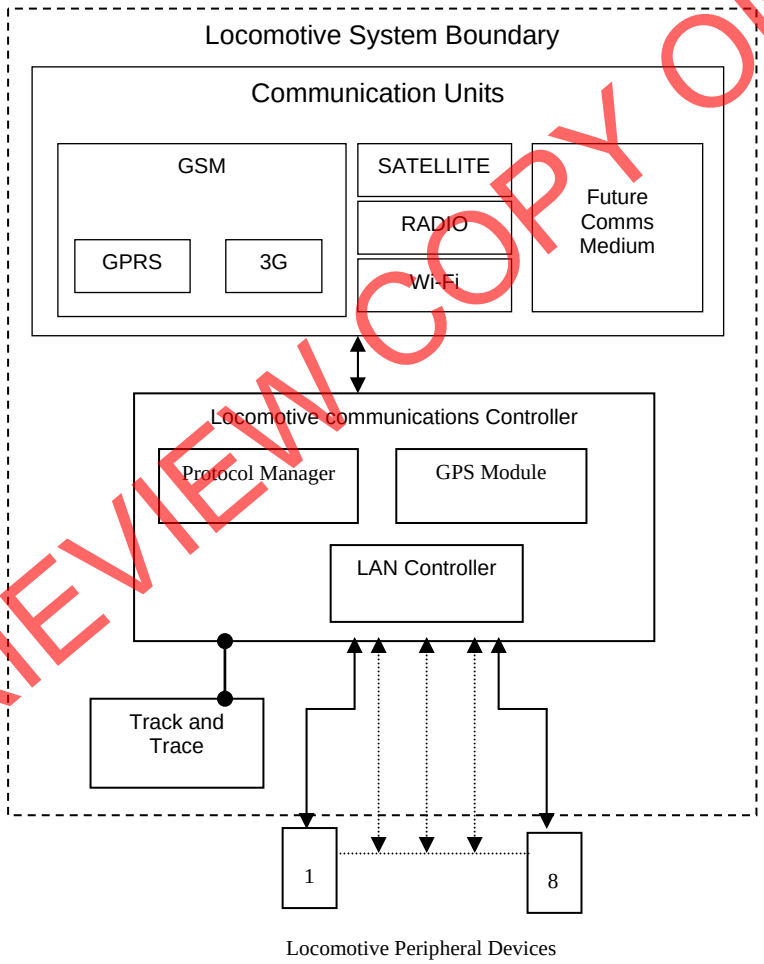


Figure 3: Locomotive Communications System detail