

**ANNEXURE A**  
**TETRA Database server Interface specification**



*City of Cape Town*  
*Tetra Data Switch Protocol*

*6 October 2008*  
*Revision 1.0*

# Table of Contents

Table of Contents .....	2
Overview.....	3
Applicable Documents .....	3
Glossary of Terms .....	4
System Description.....	5
Telegram Routing .....	6
User Routing .....	7
Switch Routing.....	7
Combined Routing .....	7
Data Flow .....	7
Port Mapping.....	9
Rules.....	9
Telegram .....	10
Telegram Structure .....	10
Client Routing.....	11
Switch Routing.....	11
Delimiter .....	11
Telegram Header .....	11
Subscriber Identifier.....	11
Port Identifier.....	11
Examples.....	12
Telegram Body.....	12
Telegram Tail .....	12
Sending .....	12
Receiving .....	12
Vehicle Tracking Data.....	13
Field Delimiter.....	13
Identifier.....	13
Msg Type.....	13
Date.....	13
Time .....	13
Latitude.....	13
Longitude.....	14
Speed .....	14
Ignition Status.....	14
Status .....	14
Over the Air Transmission.....	14
Third Party Application Test.....	15
Third Party Application Test Result.....	16

# Overview

The City of Cape Town has installed a Data Switch on the Tetra Radio Network at SPEARS. The primary function of the Switch is to facilitate the transfer of Data (Datagrams) between Computer Servers and Terminals connected to the City IP Network and Mobile Users affiliated to the TETRA Radio network.

The information contained in this document describes the datagram message protocol and the message transfer processes that can be used.

The contents of this document are subject to change and revision from time to time.

## Applicable Documents

### TCP / IP Network Protocols

IP Internet Protocol	RFC 791
ICMP Internet Control Message Protocol	RFC 792
UDP User Datagram Protocol	RFC 768
TCP Transmission Control Protocol	RFC 793
RIP2 RIP Version 2	RFC 56
OSPF2 OSPF Version 2	RFC 54

### ETSI specifications for ETSI TETRA subscriber equipment -

- ETS 300 392
- ETS 300 394
- ETS 300 395
- ETS 300 396

### ETSI specifications for ETSI GSM subscriber equipment -

- ETSI TS 100 559
- ETSI TS 100 585
- ETSI TS 101 344
- ETSI TS 101 348
- ETSI TS 101 356
- GSM 03 04
- GSM 07 07

### Iridium Short Burst Data

- Iridium Short Burst Data Developer Service Guide v11.2.1

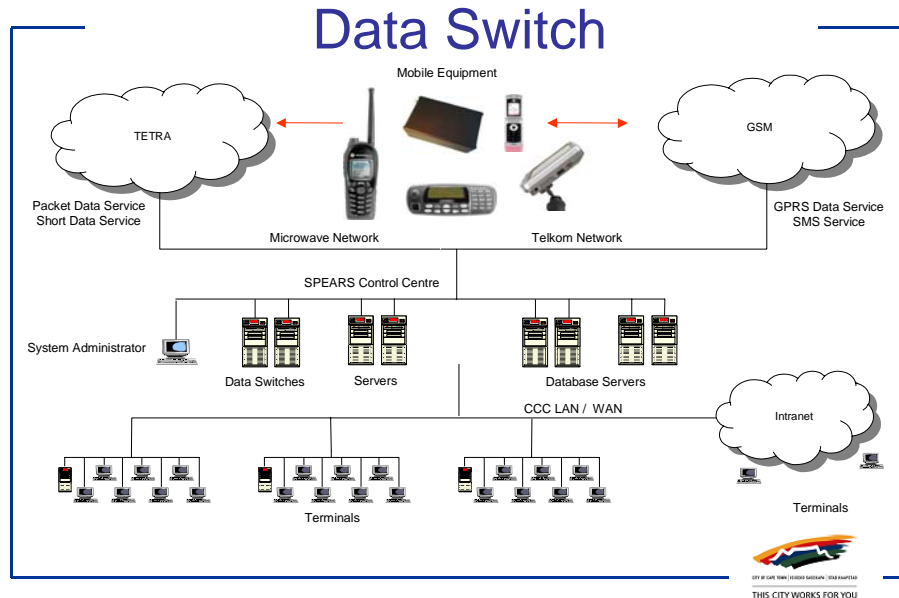
This document is a comprehensive description of the requirements and methodology required to use the Data System.

It is assumed that users wishing to develop applications that run across the Tetra Network are familiar with TCP/IP, Tetra and GSM Networking Protocols and datagram transfer conventions.

## Glossary of Terms

Term	Description
Data Switch	Server Application that marshals inbound and out bound data messages from multiple Gateways
Datagram	A packet of user definable Data
Encapsulation	The technique of adding information around data to effecting its transfer via a network
Gateway	Application running on a terminal that manages a wireless or IP service connection
Host	Terminating point, identified by an address, on an IP / Wireless Network. Servers and Terminals are Applications running on hosts. A host can have more than one server or terminal application running concurrently.
Inbound	Direction in which a data message is traveling with respect to a Gateway
Mobile Equipment	Equipment that connects to a wireless network. Where Laptop Computers, Mobile Data Terminals and the like are connected to Radio's or Phones, the combined equipment is referred to as Mobile Equipment
OBCU	An embedded computer controlling a GPS receiver and a GPRS modem
Outbound	Direction in which a data message is traveling with respect to a Gateway
Server	A Server Application running on a host
TCP	Transmission Control Protocol
Telegram	A packet of data that has been encapsulated in the prescribed format for transport via the Data Switch to different Networks
Terminal	A stand alone / client Application running on a host
UDP	User Datagram Protocol

# System Description



This document describes the protocol and method that is used by the City of Cape Town to transfer data between users connected to the City's IP and Tetra Radio Networks. Data transfer to other wireless Networks (GSM or Iridium) can also be incorporated.

Datagrams can be transported via a combination of services provided by different network types. For example -

- Tetra Short Data Messages can be forwarded to UDP Servers on an IP Network
- A TCP Server can send a datagram to a UDP client using the Tetra Packet Data Service

- Tetra Short Data Messages can be forwarded as GSM SMS Messages
- GSM SMS Messages can be forwarded to TCP Servers on an IP Network
- An email message can be converted to a Short Data Message and forwarded to a Mobile Radio

In order for datagrams to be transported across the different network types and services supported by those networks, it is necessary to use a protocol that the Data Switch used by the City of Cape Town supports.

With reference to the diagram above, the Data Switch is a Server Application that sits between Mobile Equipment operating on the Tetra and or GSM Wireless Networks and computer Servers or Terminals connected to the CCC TCP/IP network (LAN/WAN).

The Data Switch performs the following functions -

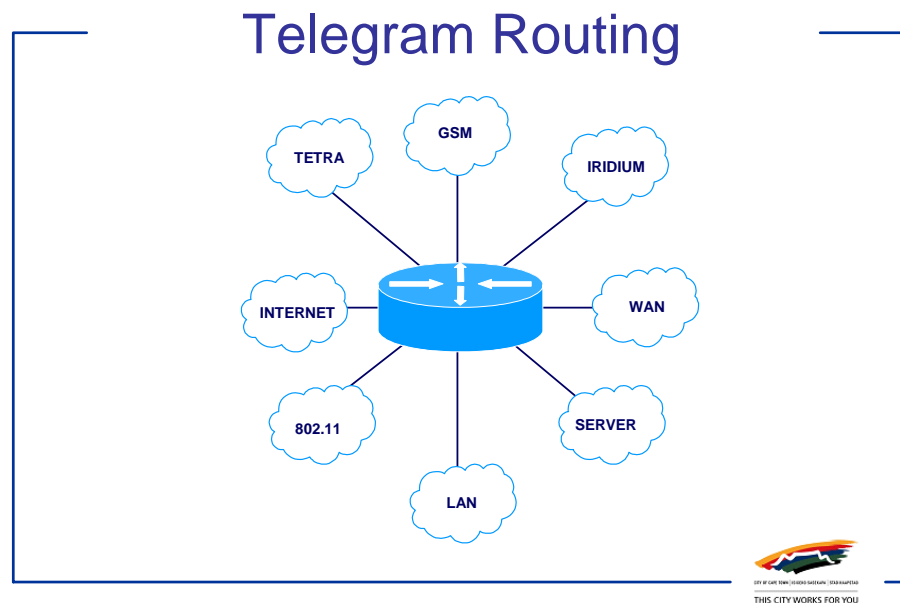
- Switches data (datagrams) from -

- Mobile Equipment to Applications on the LAN/WAN
- Applications on the LAN/WAN to Mobile Equipment
- Mobile Equipment Applications to Applications on the LAN/WAN
- Applications on the LAN/WAN to Applications on the Mobile Equipment

Monitors the datagram traffic through the Switch  
 Controls the routing of datagrams through the Switch

Application or Third Party Developers, wishing to make use of the transport mechanism incorporated in the Data Switch are required to develop datagram handlers that use the protocol described in this document.

## Telegram Routing



The function of the Data Switch is to enable the transfer of **user data** that has been broken up into small units, called **datagrams** or **packets**, across different network services.

The method used to transport datagrams via the Data Switch is called encapsulation. Each datagram has to be encapsulated in an **envelope**, in order to pass through the Data Switch and be delivered to its designated destination. The envelope contains the routing instructions for the datagram. A datagram in its encapsulated format is called a **telegram**.

With reference to the above diagram, a datagram originating on a LAN host can be delivered to a Tetra Mobile using the Short Data Service by encapsulating the datagram and forwarding the latter to the Data Switch.

In like fashion, a Tetra Mobile user can send a Short Data Message to a TCP Server, located on a remote IP Network host. Two routing methods are available -

- User Routing
- Switch Routing

## User Routing

The user (source), originating the telegram must provide the routing information for forwarding the datagram to the destination.

The advantages of this routing method are that -

- a user has the flexibility to communicate with more than one destination

## Switch Routing

The user (source), originating the telegram does not need to provide the routing information for forwarding the datagram to the destination. The routing is done by the Data Switch, based on pre-configured routing tables.

The advantages of this routing method are that -

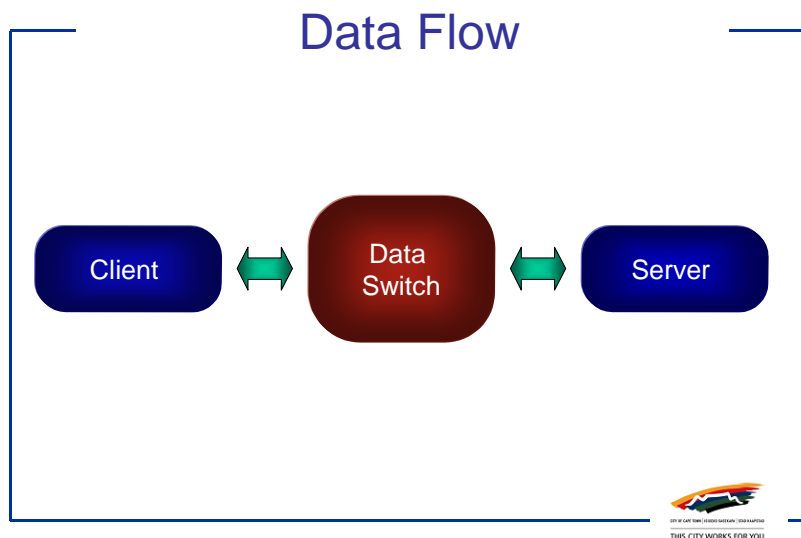
- the telegram does not contain routing information and consequently has less transmission overhead across the network. More efficient utilization of limited bandwidth or datagram, where the latter has a fixed size.
- A single telegram sent to the Data Switch can be replicated and forwarded to multiple destinations. Again there is more efficient utilization of limited bandwidth.

## Combined Routing

User and Switch Routing are not mutually exclusive. Both methods can be used at the same time.

For example a Mobile Client can use Switch Routing to one or more Servers. The servers can in turn reply using Switch Routing.

## Data Flow



Data transfer between Client and Server, or combination of the latter, can be performed

using any combination of the following data transfer protocols (**services**), provided that the connected networks support the latter -

- UDP User Datagram Protocol
- TCP Transmission Control Protocol
- SMS GSM Short Message Service
- SDS Tetra Short Data Service
- SBD Iridium Short Burst Data Service

Both the Tetra and GSM Networks support UDP and TCP services using the Tetra Packet Data Service and the GSMGPRS Services respectively.

## WHY NOT TO USE TCP

This appendix will give a short introduction to how TCP behaves thus serving as an explanation of why TCP is ***not the recommended protocol*** for the Dimetra packet data service. As also stated in the "Introduction" UDP is recommended because this is basically a WYSIWYG (What You See Is What You Get) protocol. Contrary to TCP, UDP leaves everything under the programmer's control. The build-in behaviours of TCP can lead to poor performance in an environment with large variations in delay, and the use of TCP may not always be byte efficient.

Applications that need to communicate 'Over-the-Air' via TCP will only be allowed to use the system in exceptional circumstances.

### Port Mapping

Each Application using the Data Switch service will be allocated at least one TCP / UDP port number that will be assigned for its exclusive use. Subscribers using that Application will be required to use the assigned ports.

Listeners (***Gateways***) will be opened on the Data Switch to direct the Application data traffic via the assigned ports. All un-allocated ports will be blocked to prevent unauthorized user access.

The same logic applies to the allocation of PI's on the Tetra Short Data Service.

### Rules

The following data flow rules have been applied -

- All data flow between clients and servers (or combinations of the latter) is routed via the Data Switch. A minimum of two transactions take place. The Telegram is first sent to the Data Switch and then forwarded to one or more destinations.
- There are no active connections between client and server. Each telegram sent and received is regarded as a discrete transaction having no association with the previous or subsequent telegram transfer. No mechanism for sequencing, or detection of lost telegrams is provided.
- There is no session maintenance between the client / server and the Data Switch. TCP transmissions are terminated on receipt of the telegram by the Data Switch. The Client / Server initiating the data transfer is required to re-establish the connection with the Data Switch for each transmission. Similarly the Data Switch does not maintain active sessions with the destinations of the data transfer.
- There is no retry of data transmission. Data delivery is based on a best effort module. Delivery is not guaranteed. Applications requiring confirmation of delivery must provide their own end-to-end acknowledgement / receipt of delivery mechanism. The latter must

conform to the bandwidth constraints of the respective wireless services being used.

- There is no data integrity checking. The protocol used for telegram transfer does not include an integrity and consistency check. Applications requiring such a check must provide the latter as part of the data element of the telegram.
- The ICMP protocol is not supported. Utilities like ping and trace route cannot be used to discover hosts and or routes on networks / bridged by the Data Switch.
- Telegram transfer is managed on a FIFO (First in, First Out) basis. Telegrams are queued until the transport service is available to accept delivery.
- Silent discard. Any telegram that cannot be routed or delivered is silently discarded. The source is not given any indication as to where the failure or problem occurred.

## Telegram

**Datagrams** (Small packets of data) are transferred between Terminal points using TCP/IP and wireless network protocols. Datagrams are **encapsulated** with routing information that enables the Data Switch to direct the latter to the chosen destination terminal/s. The encapsulated datagram is called a **telegram**. In other words - a telegram is a datagram being transmitted across a TCP/IP or wireless network. The information inside the telegram enables it to be routed via the Data Switch.

Telegram

This document does not deal with the encapsulation of the communication network (Wireless or IP) that may be used. In other words -

Network Header

Telegram

Network Tail

The Telegram and Datagram structures remain unchanged - irrespective of the transport protocol (**service**).

### Telegram Structure

Each telegram has three parts, header, body and tail.

Header

Body

Tail

The Header contains the routing information. The body contains the user data. The tail is optional. The tail can be used for what ever control information that a developer

wishes to use to manage the delivery and integrity of the data transfer across the networks. Typically - a length check and or CRC (Cyclic Redundancy Check).

## Client Routing

The Header is compulsory when the Client Routing method is used.

## Switch Routing

The Header is not required when the Switch Routing method is used. The source subscriber identifier is used to identify the source user and the Routing Table on the Data Switch is used to lookup the destination subscriber/s. If the source and the destination are not registered at the Data Switch the telegram is silently discarded.

## Delimiter

The Header is separated from the Body using the '>' character (Hex 0x3E or Decimal # 62). The body may be separated from the Tail using any character preferred by the Application developer.

## Telegram Header

The Header is used to determine the destination of the telegram. It comprises two parts, encapsulated in a '<>' (Hex 0x3C or Decimal # 60 to start and Hex 0x3E or Decimal # 62 to end) pair of characters -

- Subscriber identifier
- Port Identifier

The Subscriber Identifier and the Port Identifier are separated by a single ' ' (Hex 0x20 or Decimal # 32).

## Subscriber Identifier

The Subscriber Identifier is the Network Address of the destination or the source host in 'text' format (ie not binary).

For example -

Ethernet IP Network (UDP or TCP Service)	10.1.243.12	IPv4 Address
Tetra Network (Short Data Service)	7034	ISSI Address
GSM Network (SMS)	27836364567	MSISDN Address
Iridium Network	30002345768	IMEI Address

## Port Identifier

The Port Identifier is the Port Number that the Data Switch Administrator has assigned to the Application or Group of Users.

For example -

Ethernet IP Network (UDP or TCP Service)	6400
Tetra Network (Short Data Service)	131

GSM Network (SMS)	100
Iridium Network	300

## Examples

Examples of full headers -

Ethernet IP Network (UDP or TCP Service)	<10.1.243.12 6400>
Tetra Network (Short Data Service)	<7034 131>
GSM Network (SMS)	<27836364567 100>
Iridium Network	<30002345768 300>

## Telegram Body

The body is the holding space for an encapsulated datagram. There is no restriction on content or format of the datagram. The latter may be binary or ascii data.

The Body has a variable length. Check the maximum length for each service being used.

- **Note 1.** Short Data and SMS services have considerably shorter packet lengths than IP, which can also be restricted to a fixed length by designated routes.
- **Note 2.** Remember to deduct the length of the header from the maximum data length.

## Telegram Tail

There is no specification for a telegram tail. Developers are required to add tail information to perform checks that they may require on the integrity and consistency of data delivery.

## Sending

User routed telegrams must contain the Destination Address in the Header.

If the Header Information is incorrect, the Data Switch will attempt to deliver to the latter or silently discard the telegram.

## Receiving

User and Switch routed telegrams will contain the Source Address in the Header.

The Data Switch removes the Destination Address from the telegram it receives and replaces the latter with the Address of the source (sender).

# Vehicle Tracking Data

Vehicle Tracking Information can be made available to Applications that can read the following data format -

Identifier, Msg Type, Date , Time , Latitude , Longitude , Speed , Ignition Status, Status 2.

## Field Delimiter

Comma. All fields are separated by a Comma

## Identifier

The identifier field is any value that the vehicle tracking system may use to identify the tracking unit uniquely. For example - a serial number.

## Msg Type

Each vendor will be allocated a Msg Type code (or multiple codes) for each message type that is generated. The difference between Message types, being the additional fields that are added to the data string.

## Date

The data shall be expressed in the following format - `yyyymmdd`  
where

<code>yyyy</code>	year
<code>mm</code>	month
<code>dd</code>	day

## Time

The time shall be expressed in UTC (UTM) in the following format - `hhnnss`  
where

<code>hh</code>	hours, using a 24 hour clock
<code>nn</code>	minutes
<code>ss</code>	seconds

## Latitude

Latitude shall be expressed in decimal degrees to 5 decimal places, using the following convention -

Northern Hemisphere	Positive co-ordinate value
Southern Hemisphere	Negative co-ordinate value

## **Longitude**

Longitude shall be expressed in decimal degrees to 5 decimal places, using the following convention -

Eastern Hemisphere	Positive co-ordinate value
Western Hemisphere	Negative co-ordinate value

## **Speed**

Speed shall be expressed in km/hour to one decimal place

## **Ignition Status**

The status of the tracking unit ignition system shall be represented in the following way -

N	Not Available
T	Ignition Status is ON
F	Ignition Status is OFF

## **Status ....**

The status formats of additional status which may be available are to be defined in consultation with the SPEARS administrators.

## **Over the Air Transmission**

It is acknowledged that Vehicle Tracking System Providers use proprietary data formats for over-the-air transmission. Vendors intending to operate their systems on the Cape Town City Council network are required to provide handlers (libraries) that can be integrated into the Data Switch for the unpacking of their vehicle data into the above format.

With respect to the Dimetra Packet Data Developer Guide -

## **B. AIR INTERFACE CONSIDERATIONS**

Each message is sent across the air interface using the TETRA SMDCP service. The underlying TETRA layer 2 splits the message up into segments which maps to slots. Since each segment carry a fixed amount of bits, this means that the used air interface re-sources increases in jumps on some message length boundaries.

The “cost” of sending an IP datagram is therefore a basic signalling cost, plus a “variable cost” depending on the size of the IP datagram.

The boundaries where the “variable cost” changes are :-

Table 9: Uplink Slot Count to IP datagram Size

Additional uplink slots	Max IP datagram size (bytes)	Additional uplink slots	Max IP datagram size (bytes)	Additional uplink slots	Max IP datagram size (bytes)	Additional uplink slots	Max IP datagram size (bytes)
	50	14	449	28	848	42	1247
1	79	15	478	29	877	43	1276
2	107	16	506	30	905	44	1304
3	136	17	535	31	934	45	1333
4	164	18	563	32	962	46	1361
5	193	19	592	33	991	47	1390
6	221	20	620	34	1019	48	1418
7	250	21	649	35	1048	49	1447
8	278	22	677	36	1076	50	1475
9	307	23	706	37	1105	51	1500
10	335	24	734	38	1133		
11	364	25	763	39	1162		
12	392	26	791	40	1190		
13	421	27	820	41	1219		

Note that the IP datagram size includes the IP and transport protocol header. For UDP, the IP and UDP overhead is 28 bytes. That is, to send e.g. 100 bytes of user data, an IP datagram of length 128 bytes must be sent (assuming 28 bytes of UDP/IP overhead) - this requires 3 additional slots (3 slots = 132 bytes = smallest number larger than 128).

Note that normal application does not need to worry about this. This information is relevant only for applications using a **very large number of messages**.

Vehicle tracking applications that use telegrams larger than 40 bytes, to transfer data 'Over-the-Air', will be required to re-structure their data format to conform to this requirement.

## Third Party Application Test

Any Third Party Application using the Data Switch should be able to pass the following acceptance tests -

1. Send a Telegram from an Application on the City of Cape Town LAN to Mobile Radio Equipment
2. Send a Telegram from Mobile Radio Equipment to an Application on the City of Cape Town LAN
3. Send a Telegram from an Application on the City of Cape Town LAN to an Application on a Terminal connected to Mobile Radio Equipment
4. Send a Telegram to an Application on a Terminal connected to Mobile Radio Equipment to an Application on the City of Cape Town LAN

# Third Party Application Test Result

<b>PROTOCOL USED</b> <b>GATEWAY USED</b> <b>PORT USED</b>  <b>DATE OF TEST</b>  <b>MOBILE SUBSCRIBER ID</b> <b>LAN SUBSCRIBER ID</b>	
<b>DESCRIPTION</b>	<b>COMPLIANCE</b>
Brief Description of the Test Application	
1. Send a Telegram from an Application on the City of Cape Town LAN to Mobile Radio Equipment	
2. Send a Telegram from Mobile Radio Equipment to an Application on the City of Cape Town LAN	
3. Send a Telegram from an Application on the City of Cape Town LAN to an Application on a Terminal connected to Mobile Radio Equipment	
4. Send a Telegram to an Application on a Terminal connected to Mobile Radio Equipment to an Application on the City of Cape Town LAN	

The results of this test are accepted / rejected by \_\_\_\_\_ on behalf of SPEARS

\_\_\_\_\_  
Signature

Date \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_  
Witness