CLC/FprTS 50568-9

**ELECTRICITY METERING DATA EXCHANGE –
THE SMART METERING INFORMATION TABLES AND PROTOCOLS (SMITP) SUITE**

Part 9: IP profile on public telecommunication network

Following document is the Cenelec version (CLC/FprTS 50568-9) of “Meters and More protocol suite - part 9 - IP profile on public telecommunication network”

CLC/FprTS 50568-9
Electricity metering data exchange –
The Smart Metering Information Tables and Protocols (SMITP) suite –
Part 9: IP profile on public telecommunication network

Contents

Page

List of Figures

Foreword

To be updated before FV.

The European Committee for Electrotechnical Standardization (CENELEC) draws attention to the fact that it is claimed that compliance with this International Standard may involve the use of a maintenance service concerning the stack of protocols on which the present Technical Specification prEN/TS 5zzzz is based.

The CENELEC takes no position concerning the evidence, validity and scope of this maintenance service.

The provider of the maintenance service has assured the CENELEC that he is willing to provide services under reasonable and non-discriminatory terms and conditions for applicants throughout the world. In this respect, the statement of the provider of the maintenance service is registered with the CENELEC. Information may be obtained from:

Meters and More Open Technologies
Brussels/Belgium
www.metersandmore.eu

Introduction

This Technical Specification is based on the results of the European OPEN Meter project, Topic Energy 2008.7.1.1, Project no.: 226369, www.openmeter.com.

According to the structure of the CLC/FprTS 50568 documentation, this document is positioned as highlighted in the following figure:

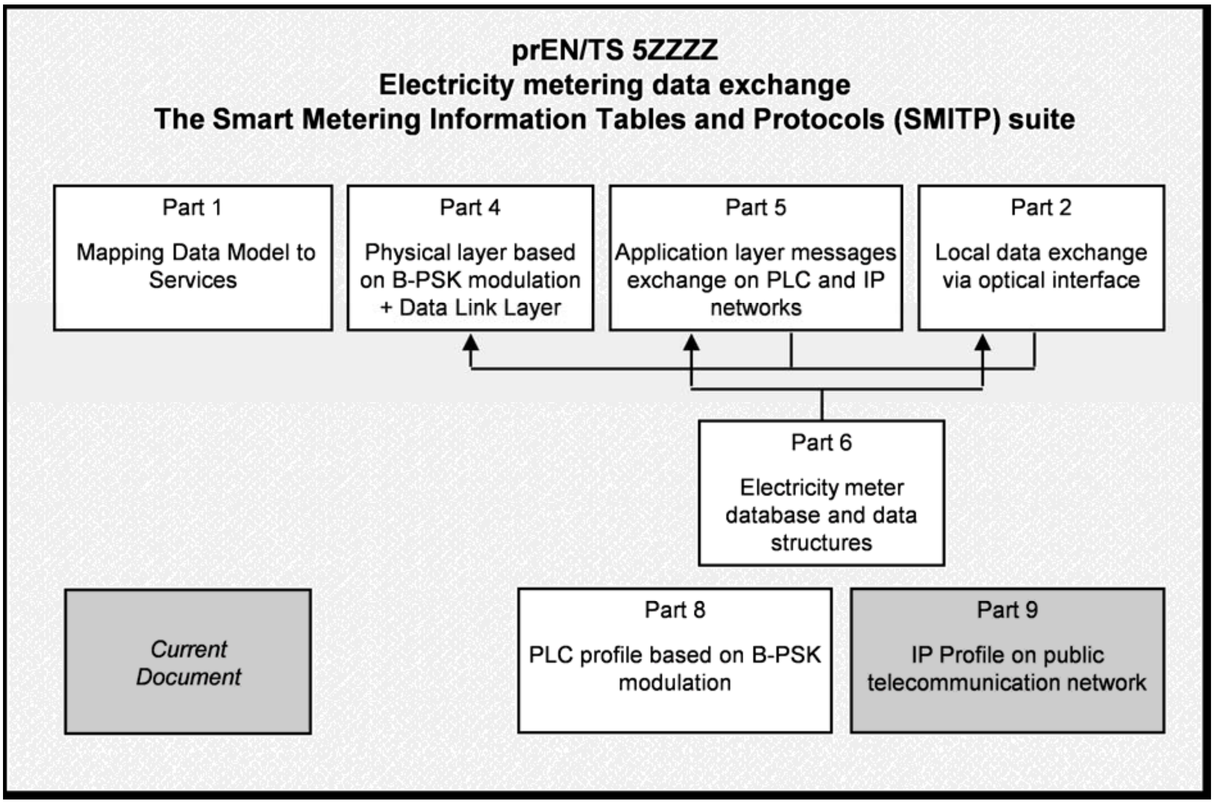

Figure 1 – Organization of the documentation

1 Scope

This Technical Specification specifies the IP profile based on public telecommunication network.

It uses the standards defined in the Electricity Metering Data Exchange – SMITP suite.

2 Normative references

CLC/FprTS 50568-4, *Electricity metering data exchange — The Smart Metering Information Tables and Protocols (SMITP) suite — Part 4: Physical layer based on B-PSK modulation + Data Link Layer*

CLC/FprTS 50568-5, *Electricity metering data exchange — The Smart Metering Information Tables and Protocols (SMITP) suite —Part 5: Application layer messages exchange on DLC and IP networks*

CLC/FprTS 50568-6, *Electricity metering data exchange — The Smart Metering Information Tables and Protocols (SMITP) suite — Part 6: Meter database and data structures*

STD0005 (1981)Internet Protocol. Also: RFC0791, RFC0792, RFC0919, RFC0922, RFC0950, RFC1112

STD0006 (1980)User Datagram Protocol. Also: RFC0768

STD0007 (1981)Transmission Control Protocol. Also: RFC0793

3 Terms and abbreviations

B-PSK:Binary Phase Shift Keying

DCS:Digital Cross-connect System

GPRS:General Packet Radio Service

GSM:Group Special Mobile

IP: Internet Protocol

LV: Low Voltage

NN:Neighbourhood Networks

NNAP:Neighbourhood Network Access Points

PLC:Power Line Carrier

PSTN:Public Switched Telephone Network

TCP:Transmission Control Protocol

ULP:Upper Layer Protocol

UMTS:Universal Mobile Telecommunications System

WAN:Wide Area Network

4 Communication environments

The PLC profile based on B-PSK modulation is used for remote data exchange on Neighbourhood Networks (NN) between Neighbourhood Network Access Points (NNAP) and End Devices using B-PSK power line technology over the low voltage electricity distribution network as a communication medium.

End devices – typically electricity meters – comprise application function and communication functions. They may be connected directly to the NNAP via the C interface.

The NNAP device, also known as concentrator, is connected upstream to the Metering Head End System (HES) through interface G, via IP enabled networks.

The reference architecture is shown in Figure 2.

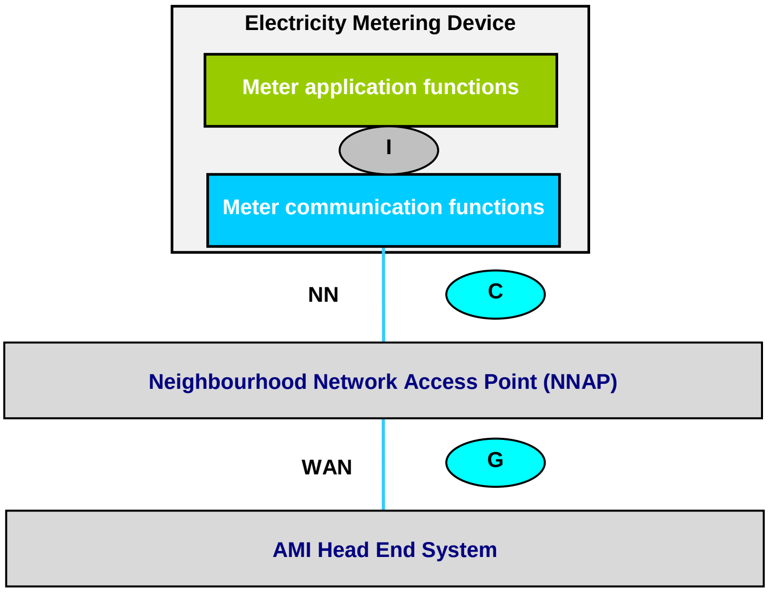

Figure 2 – Reference architecture

5 Structure of profile

Figure 3 shows the SMITP protocol stack on IP profile.

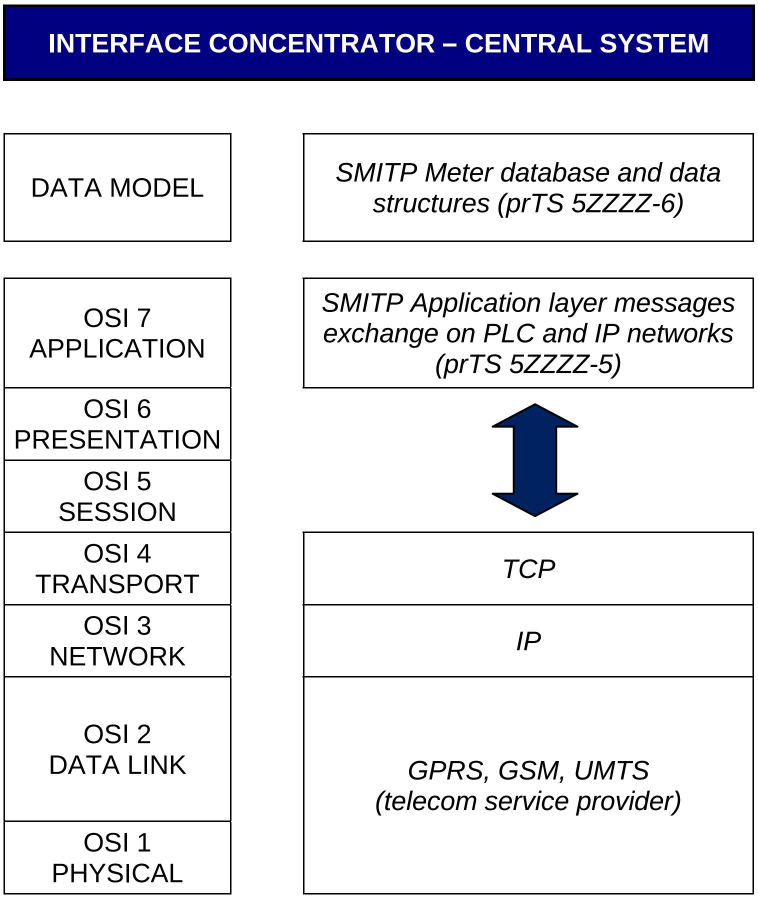

Figure 3 – Protocol stack for SMITP on IP

6 Physical and data link layers

The lower layers for this interface provide the physical medium to communicate the central system and the concentrator, the medium access control and the addressing.

Due to the geographical dispersion of the concentrators, and the lack of private communications networks among them, public telecommunication networks will be used where available: GPRS, UMTS, GSM, PSTN.

The services provided by these layers will be those supported by the public networks used, mainly error free data transportation.

7 Transport and network layers

7.1 General

The SMITP Application Layer uses the transport layer always in a connection-oriented mode, based on the connection-oriented Internet transport protocol, called Transmission Control Protocol or TCP STD0005, STD0006 and STD0007.

TCP is an end-to-end reliable protocol. This reliability is ensured by a conceptual “virtual circuit”, using a method called “Positive Acknowledgement with Retransmission” or PAR. It provides acknowledged data delivery, error detection, data re-transmission after an acknowledgement time-out, etc., therefore is dealing with lost, delayed, duplicated or erroneous data packets. In addition, TCP offers an efficient flow control mechanism and fullduplex operation, too.

TCP, as a connection-oriented transfer protocol, involves three phases: connection establishment, data exchange and connection release. These phases are performed by a logical TCP management entity that uses standard ULP/TCP services. Any message of the SMITP Application Layer to be encapsulated in TCP packets is managed by this entity. This task could be implemented through the use of any standard TCP management library. Annex A reports an example explaining the TCP/IP encapsulation of a SMITP TB Application layer message.

In the connection procedure, the client (backoffice) must to send the requests toward the server (Concentrator) TCP port 50000, and the server (Concentrator) must send the responses using its TCP port 50001.

7.2 TCP Service Request Primitives:

7.2.1 UNSPECIFIED-PASSIVE-OPEN.request

Parameters: source-port, [timeout], [timeout-action], [precedence], [security-range]

Description: this primitive is used to listen for connection attempt at specified security range and precedence from any remote destination.

7.2.2 FULLY-SPECIFIED-PASSIVE-OPEN.request

Parameters: source-port, destination-port, destination-address, [timeout], [timeout-action], [precedence], [security-range]

Description: this primitive is used to listen for connection attempt at specified security range and precedence from specified destination.

7.2.3 ACTIVE-OPEN.request

Parameters: source-port, destination-port, destination-address, [timeout], [timeout-action], [precedence], [security]

Description: this primitive is used to request a connection at a particular security and precedence to a specified destination.

7.2.4 ACTIVE-OPEN-WITH-DATA.request

Parameters: source-port, destination-port, destination-address, [timeout], [timeout-action], [precedence], [security], data, data-length, PUSH-flag, URGENT-flag

Description: this primitive is used to request a connection at a particular security and precedence to a specified destination and transmit data together.

7.2.5 SEND.request

Parameters: local-connection-name, data, data-length, PUSH-flag, URGENT-flag, [timeout], [timeout-action]

Description: this primitive is used to transfer data across named connection.

7.2.6 ALLOCATE.request

Parameters: local-connection-name, data-length.

Description: this primitive is used to issue incremental allocation for data to be received.

7.2.7 CLOSE.request

Parameters: local-connection-name

Description: this primitive is used to close connection in regular way.

7.2.8 ABORT.request

Parameters: local-connection-name

Description: this primitive is used to close connection immediately.

7.2.9 Status.request

Parameters: local-connection-name

Description: this primitive is used to query connection status.

7.3 TCP Service Response Primitives:

7.3.1 OPEN-ID.confirm

Parameters: local-connection-name, source-port, destination-port, destination-address

Description: this primitive is used to inform TCP user about the connection name assigned to pending connection requests in an Open Primitive.

7.3.2 OPEN-FAILURE.confirm

Parameters: local-connection-name

Description: this primitive is used to report failures of an Active Open request.

7.3.3 OPEN-SUCCESS.confirm

Parameters: local-connection-name

Description: this primitive is used to report completion of a pending Open request.

7.3.4 DELIVER.indication

Parameters: local-connection-name, data, data-length, URGENT-flag

Description: this primitive is used to report arrival of data.

7.3.5 CLOSING.indication

Parameters: local-connection-name

Description: this primitive is used to report that remote TCP user has issued a Close and that all data sent by remote user have been delivered.

7.3.6 TERMINATE.confirm

Parameters: local-connection-name, description

Description: this primitive is used to report that the connection has been terminated; a description of the reason for termination is provided.

7.3.7 STATUS-RESPONSE.confirm

Parameters: local-connection-name, source-port, source-address, destination-port, destination-address, connection-state, receive-window, send-window, amount-awaiting-ACK, amount-awaiting-receipt, urgent-state, precedence, security, timeout

Description: this primitive is used to report current status of connection.

7.3.8 ERROR.indication

Parameters: local-connection-name, error description

Description: this primitive is used to report service-request or internal error.

7.4 Example of primitives working

The following figures describe the use of TCP-ULP primitives in case of the ULP of an TCP connection opening request by a client (see Figure 4), a data exchange (see Figure 5) and a TCP connection closing request by a server (see Figure 6). The client is always the backoffice and the server is the concentrator.

NOTE Primitives between TCP and IP protocols are inserted in order to detail the behaviour within the system but they are out of scope in this document.

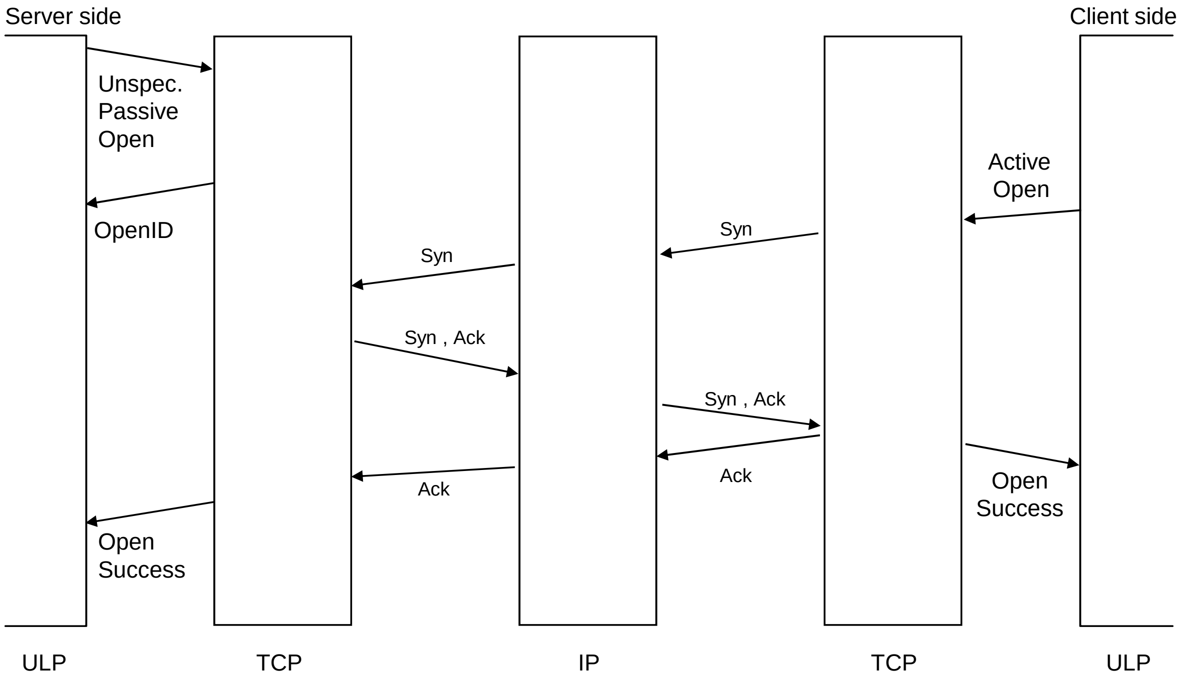

Figure 4 – Connection Opening

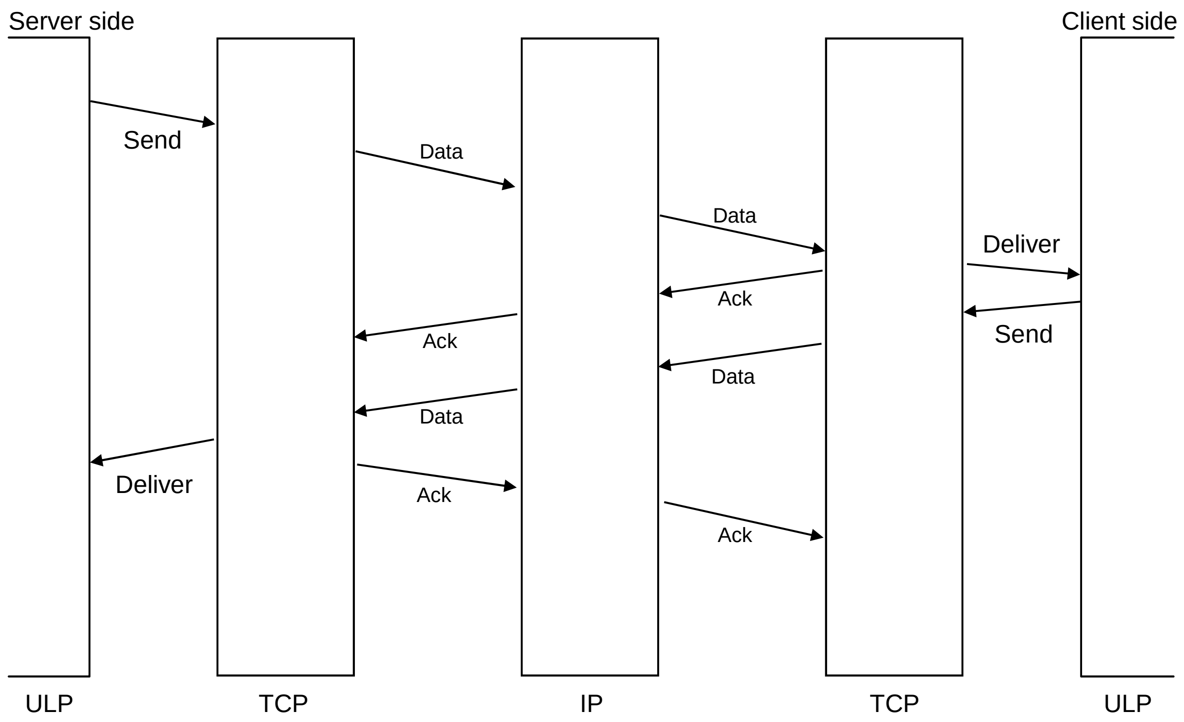

Figure 5 – Data exchange

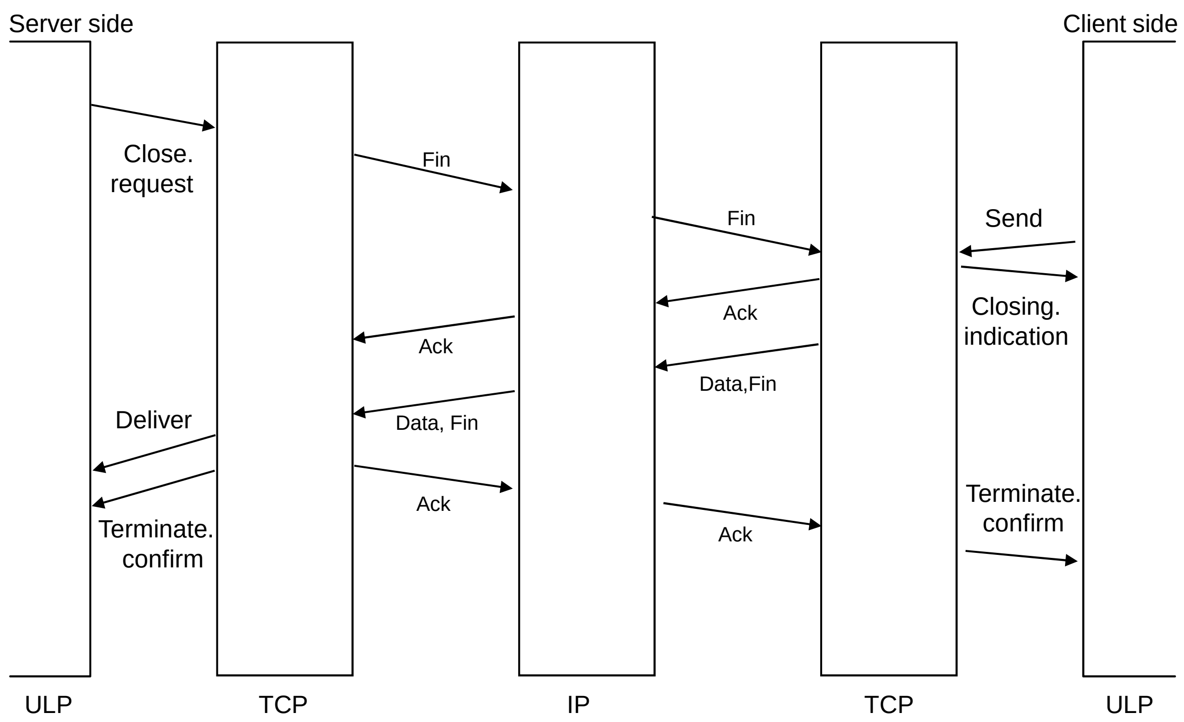

Figure 6 – Connection Closing

8 Application layer

The *SMITP Application layer* is described in CLC/FprTS 50568-5.

It provides the set of messages exchanged between back office system and the concentrator, through the telecommunication networks (GSM, DCS, PSTN, GPRS etc).

It provides the services to handle LV network pattern and concentrator management.

These services are:

- Requests to concentrators to manage meters and end customer devices (programming, data exchange, software download, etc.);

- Concentrator and network management (programming, synchronization, software download, command execution, acknowledgements, events handling, etc).

9 Data model

SMITP Meter database and data structures are described in CLC/FprTS 50568-6.

In the SMITP data model, data are grouped in tables, each table representing an homogeneous set of parameters. Each parameter can be writable/not writable and readable/not readable depending of the nature of the data.

These data can be retrieved individually, by entire tables or by user defined sets of data (custom tables), allowing a fast and flexible scheme of data access.

Annex A
(informative)

TCP/IP encapsulation example of SMITP TB Application layer message

A.1 Overview

In this clause the log of a reading operation that the Back Office performed on a specified meter through a Public communication network is shown. Back Office requested concentrator to read the meter's status word and the TCP/IP packets, cointaining the TB messages, exchanged between Back Office and Concentrator are detailed. See STD0007 for TCP header format and see STD0005 for IP header format. See subclause 5.3 in CLC/FprTS 50568-5 for SMITP TB Application layer messages format.

A.2 Back Office to Concentrator message

Reading operation of the meter's status word is performed through a ReadTab.Req (006) command sent from Back Office to the Concentrator. Below the entire hexadecimal coding of the TCP/IP part of the message is shown:

45|00|00|3a|dd|8c|40|00|80|06|c8|c9|0a|49|34|17|0a|3b|0b|cd|08|b6|c3|50|ac|87|b4|fd|fb|9b|bf|ee|50|18|ff|ed|2
4|ae|00|00|02|06|00|0b|08|01|01|00|a8|04|0a|1e|89|53|06|16|01|02|

Below the byte explanation of the IP Header section is shown:

45|Version + IHL

00|Type of Service

00|3a|Total Length

dd|8c|Identification

40|00|Flags + Fragment Offset

80|Time to Live

06|Protocol

c8|c9|Header Checksum

0a|49|34|17|Source Address (10.73.52.23)

0a|3b|0b|cd|Destination Address (10.59.11.205)

Below the byte explanation of the TCP Header section is shown:

08|b6|Source Port (2230)

c3|50|Destination Port (50000)

ac|87|b4|fd|Sequence Number

fb|9b|bf|ee|Acknowledgment Number

50|18|Data offset – Reserved – Control bits ff|ed|Window

24|ae|Checksum

00|00|Urgent Pointer

Below the byte explanation of the TB application message section is shown:

02|TM (Message type)

06|CM (Message code)

00|0b|LCD (Data Field Length)

08|01|01|Transaction ID

00|Prot

a8|04|0a|1e|89|53|Meter ID

06|Action

16|01|02|Data:

16|Table ID

01|Register1 ID

02|Register2 ID

A.3 Concentrator to Back Office message

The Concentrator responds to Back Office sending a ReadTab.Resp (007) command carrying requested data. The following hexadecimal coding represents the entire TCP/IP part of the received message:

45|00|00|3b|b9|1c|00|00|37|06|76|39|0a|3b|0b|cd|0a|49|34|17|c3|51|08|b5|fb|ae|95|9c|43|b0|08|eb|50|18|16|d0
|45|fc|00|00|02|07|00|0c|08|01|01|a8|04|0a|1e|89|53|07|16|80|c0|c0|fc

Below the byte explanation of the IP Header section is shown:

45|Version + IHL

00|Type of Service

00|3b|Total Length

b9|1c|Identification

00|00|Flags + Fragment Offset

37|Time to Live

06|Protocol

76|39|Header Checksum

0a|3b|0b|cd|Source Address (10.59.11.205)

0a|49|34|17|Destination Address (10.73.52.23)

Below the byte explanation of the TCP Header section is shown:

c3|51|Source Port (50001)

08|b5|Destination Port (2229)

fb|ae|95|9cSequence Number

43|b0|08|ebAcknowledgment Number

50|18|Data offset – Reserved – Control bits

16|d0|Window

45|fc|Checksum

00|00|Urgent Pointer

Below the byte explanation of the TB application message section is shown:

02|TM (Message type)

07|CM (Message code)

00|0c|LCD (Data Field Length)

08|01|01|Transaction ID

a8|04|0a|1e|89|53|Meter ID

07|Action

16|80|c0|c0|fcData:

16|Table ID

80|c0|Register1 value (Status Word byte 1 and 2)

c0|fc|Register2 value (Status Word byte 3 and 4)