SIEMENS

PLC management

Muslimovic jasmine
05/17/2018

Contents

List of figures and tables

1 Data types

The Siemens header has a dynamic structure, so the information in the header depends on the Head type. In Tables 1 and 2 are the defined data types with corresponding values.

Table 1: *Request data types*

HEAD END SYSTEM → GW	
Data type	Value
DLMS_REQ	0
PING_REQ	2
ROUTE_REQ	4
DLMS_MULTICAST_REQ	8

Table 1: *Response and acknowledgment data types*

GW → HEAD END SYSTEM	
Data type	Value
DLMS_RSP	1
PING_RSP	3
ROUTE_RSP	5
ACK	6
NACK	7

In next chapters are each head type is explained in more detail.

2 DLMS data

Two data types are defined: DLMS_REQ and DLMS_RSP. Data passing through TCP in both directions (HEAD END system → GW and GW → HEAD END system), which include the Siemens headers and the DLMS data:

data = Siemens header (DLMS_REQ or DLMS_RSP) + DLMS data (wrapper header + data)

Unlike the previous version of data communication, the GW Proto header is not embedded between the wrapper header and DLMS data any more. The information from the GW Proto Header or the counter address is implemented in the new Siemens header.

2.1 DLMS Request

DLMS_REQ value = 0

Table 2 *DLMS Data Request*

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	DLMS_REQ
Packet ID	2	UINT16
Meter EUI64	8	EUI64
Data Len	2	UINT16
Data	Data Len	DLMS Wrapper + DLMS Data

2.2 DLMS Response

DLMS_RSP value = 1

Table 3 *DLMS Data Request*

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	DLMS_RSP
LQI	1	UINT8
Meter EUI64	8	EUI64
Data Len	2	UINT16
Data	Data Len	DLMS Wrapper + DLMS Data

LQI - link quality indicator

2.3 Flow diagram

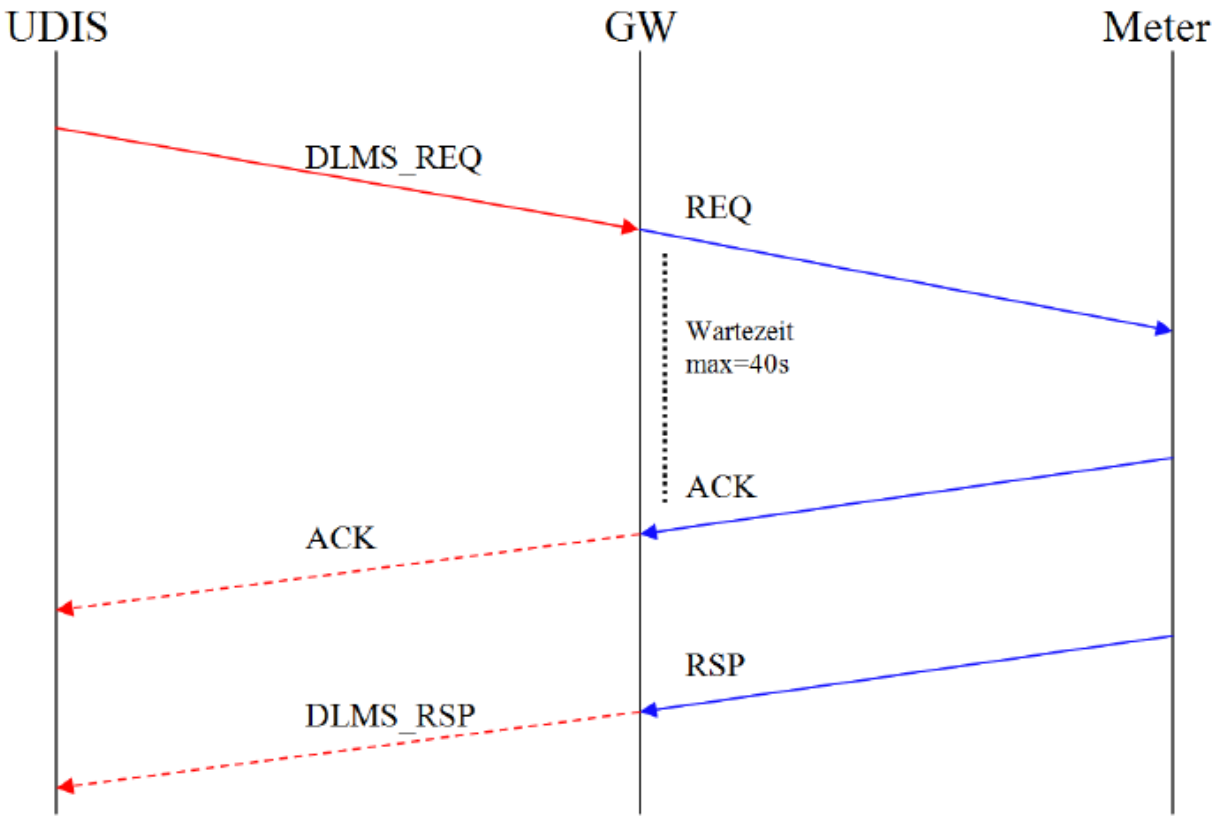

Figure 1: *DLMS Data flow diagram*

3 Ping

Two data types are defined: PING_REQ and PING_RSP. Data passing through TCP in both directions (HEAD END System → GW and GW → HEAD END System) include the Siemens headers and the ping data:

data = Siemens Header (PING_REQ or PING_RSP) + PING data

The ping data is used to build an ICMPv6 packet and send it to the meter via PLC. In this Version of PLC Management the type of ICMPV6 packet is set to "ECHO Request" with the value 128. If necessary, other types (129 - 155) can also be set.

3.1 Ping request

PING_REQ value = 2

Table 4: *Ping Request*

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	PING_REQ
Packet ID	2	UINT16
Meter EUI64	8	EUI64
Data Len	2	UINT16
Data	Data Len	Ping Data

3.2 Ping Response

PING_RSP value = 3

Table 5: Ping response

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	PING_RSP
Meter EUI64	8	EUI64
Data Len	2	UINT16
Data	Data Len	Ping Data

The meter also responds with an ICMPv6 packet and from this packet becomes the response Ping data unpacked and sent back to the HEAD END system

3.3 Flow diagram

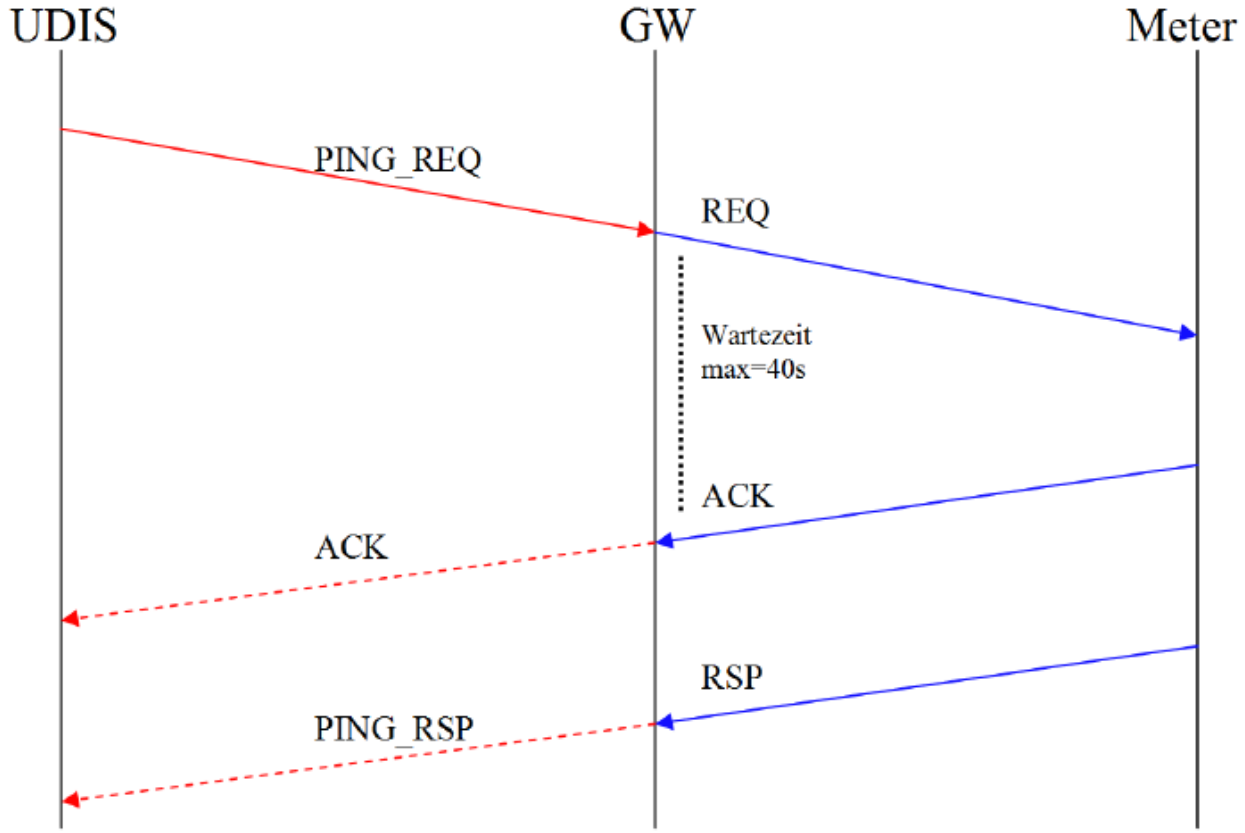

Figure 2: *Ping Request Diagram*

4 PLC Route

There are two types defined for this data type: ROUTE_REQ and ROUTE_RSP. Data via TCP sent from the HEAD END system to the GW contain the Siemens header only:

data = Siemens Header (ROUTE_REQ)

and data returned from the GW to the HEAD END system includes the response Siemens headers and the routing tables:

data = Siemens Header (ROUTE_RSP) + Routing Tables (all meters)

4.1 Route request

ROUTE_REQ value = 4

Table 6: *Route Request*

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	ROUTE_REQ
Packet ID	2	UINT16

4.2 Route response

ROUTE_RSP value = 5

Table 7: *Route Response*

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	ROUTE_RSP
Packet ID	2	UINT16
Data Len	2	UINT16
Data	Data Len	PLC PAN Routing Tables

data = dictionary

keys = EUI64 meter address

values = {

‘destAddr’: “ ”,

‘nextHopAddr’: “ ”,

‘routeCost’: “ ”,

‘hopCount’: “ “,

‘weakLinks’: “ “,

‘validTime’: “ “

}

4.3 Flow diagram

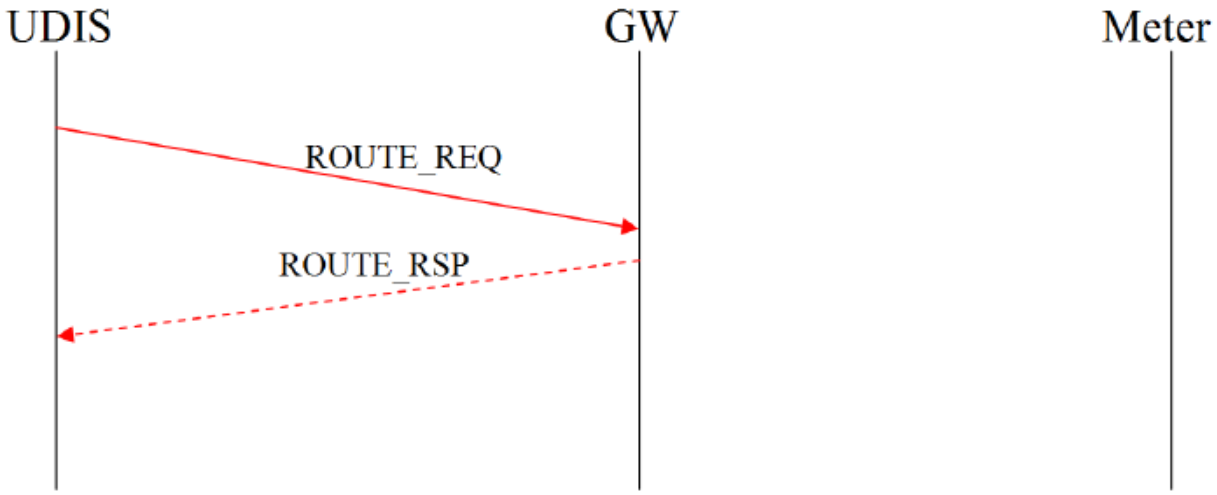

Figure 3: Routing Table diagram

5 DLMS Multicast request

DLMS_MULTICAST_REQ value = 8

Table 8: *DLMS Multicast Request*

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	DLMS_MULTICAST_REQ
Packet ID	2	UINT16
Group ID Len	2	UINIT16
Group ID	Group ID Len	Group number
Data Len	2	UINT16
Data	Data Len	DLMS Wrapper + DLMS Data

All meters that are set as members of this group are queried with this request.
Based on the group ID, a multicast address consisting of a 2-byte prefix and 14-byte Group ID composed, constructed:

multicast address = "\ xFF \ x02" + group id

and the data will be sent to this address.
The length of the group ID in the header can be a minimum of 1-byte and a maximum of 14-byte. In the case, that the groups ID e.g. is 1, the length is 1-byte and the multicast address is set as the following:

multicast address = "\xFF\x02\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01"

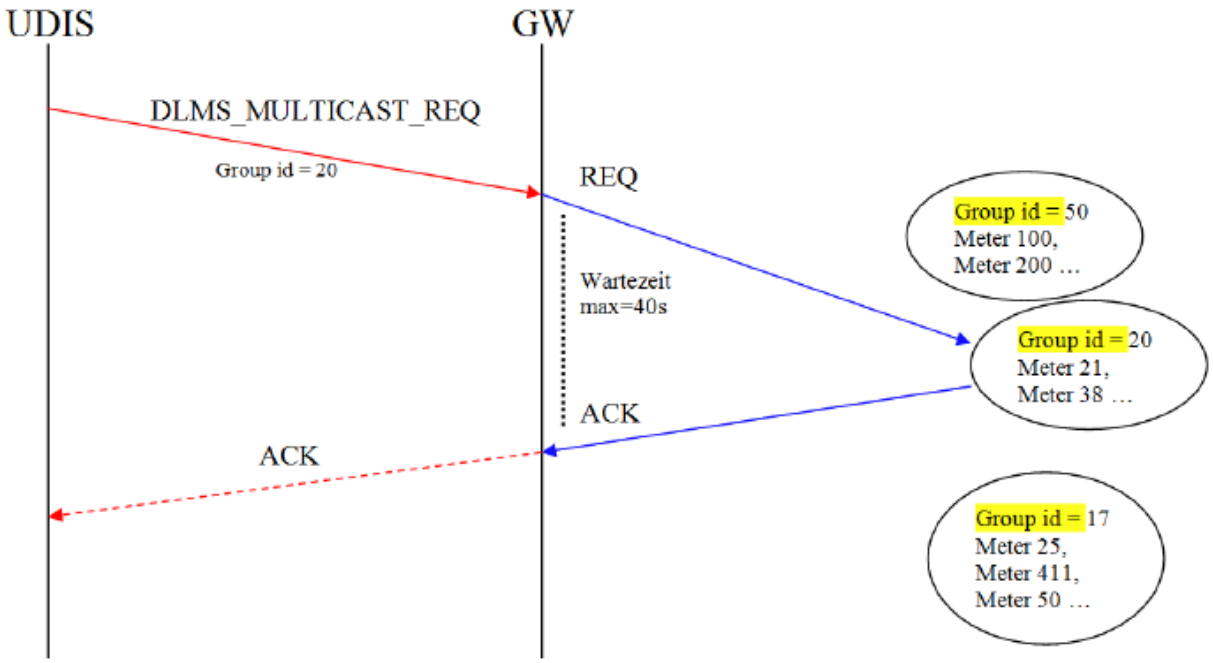

Figure 4: DLMS Multicast Diagram

6 ACK

A positive acknowledgment is returned whenever a request delivered successfully and reached the (first) meter. An ACK is sent to HEAD END system only in the case of a DLMS_REQ, PING_REQ and DLMS_MULTICAST_REQ data type only.
ACK Value = 6

Table 9: ACK

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	ACK
Packet ID	2	UINT16
Data Len	2	UINT16
Data	Data Len	Meter Path

The data contains the path of the requested meter. This information is sent at this point because the requested meter can communicate over multiple counters and the ACK
Is sent immediately after the first meter has been reached

7 NACK

A negative acknowledgment can be sent to the HEAD END system for a variety of reasons. These reasons are built into the header under the Reasons field.
NACK Value = 7

Table 10: *NACK*

Description	Len (Byte)	Type
SYNC	1	0x55
SYNC	1	0x55
SYNC	1	0x55
Version	1	UINT8
Data Type	1	NACK
Packet ID	2	UINT16
Reason	1	NACK Reason

There are 4 reasons for a NACK for this version of PLC Management. The reasons can be extended anytime. All errors that occur in the log, see Table 12, are recorded through a reason with value up to 99.

Table 11: *NACK Reasons*

NACK Reasons	Value	Description
BUSY	0	
UNKNOWN_MET	1	Meter is not registered, unknown
ROUTE_ERR	2	Route to the meter could not be found
PROTOCOL_ERROR	99	Error in protocol e.g.: - Unknown data type - - no enough data received - - data length is not correct - - etc.

7.1 Diagram Reason=Protocol Error

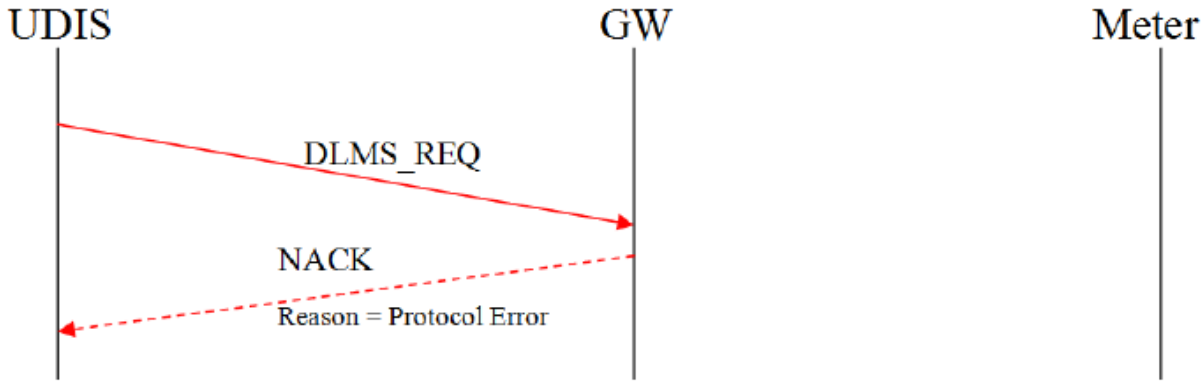

Figure 5: NACK mit Reason = Protocol Error Ablaufdiagramm

Figure 5: *NACK with Reason = Protocol Error Diagram*

7.2 Diagram Reason=Busy

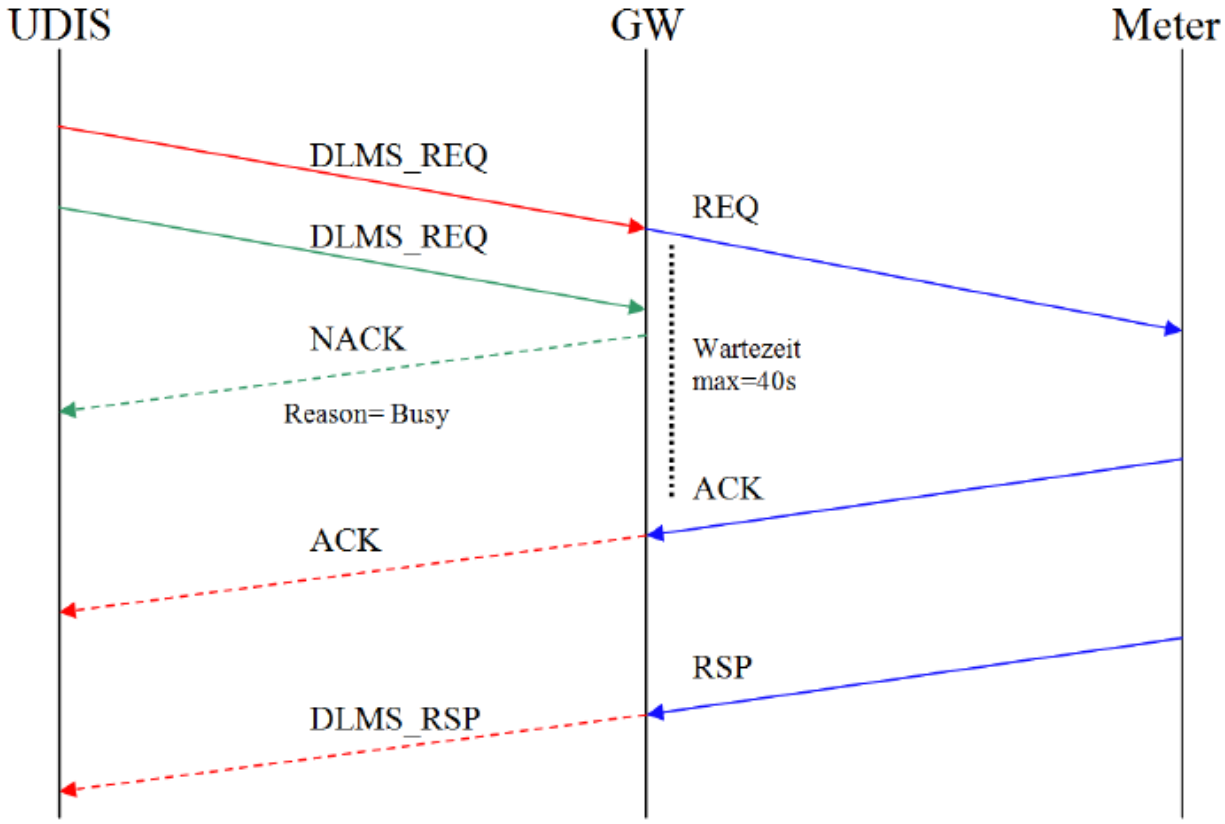

Figure 6: *NACK with Reason = Busy Diagram*

7.3 Diagram Reason= Unknown Meter

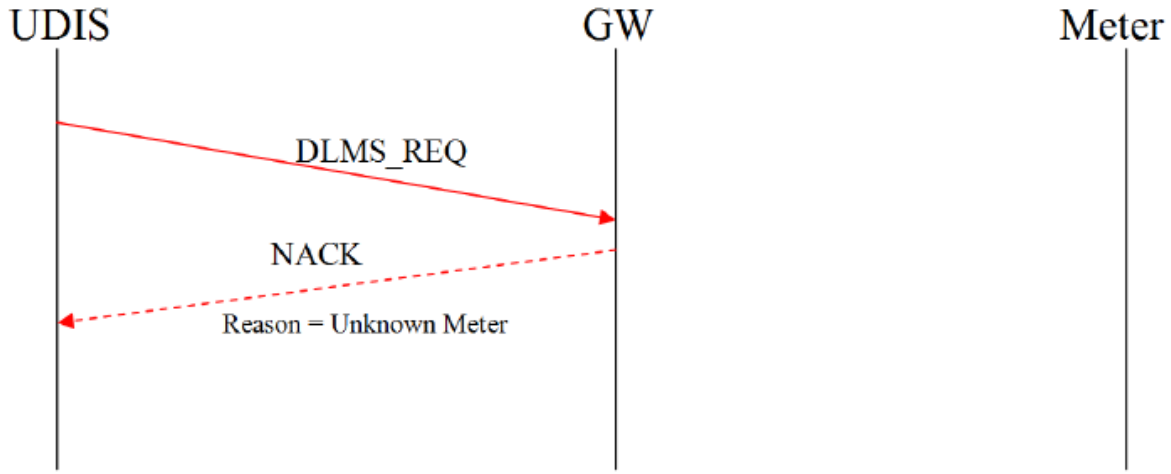

Figure 7: *NACK with Reason = Unknown Meter Diagram*

7.4 Diagram Reason= Route Error

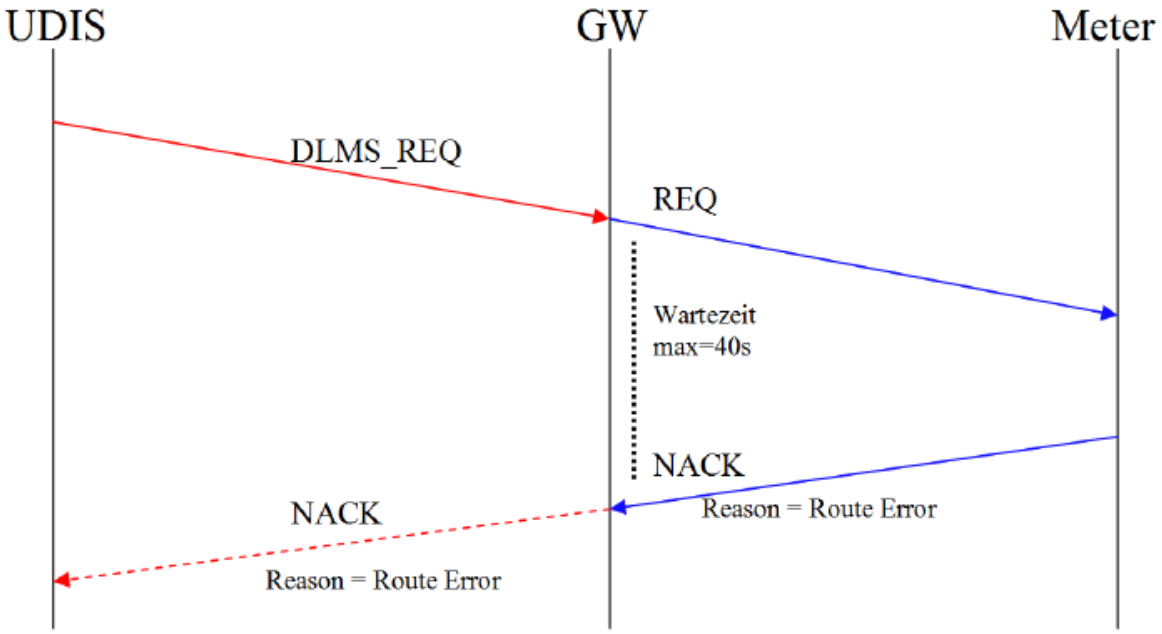

Figure 8: *NACK with Reason = Route Error Diagram*