# LoRaWAN Network Server Demonstration:
# Gateway to Server Interface Definition

## 1    History

| Revision | Modification / Remarks / Motive | Author |
|---|---|---|
| 1.0 | Document created | DRo |

## 2    Introduction

This document defines the protocol by which the Semtech LoRa gateway communicates with the Semtech LoRa network server.

The protocol authenticates neither the gateway nor the server.

Neither participant retransmits unacknowledged messages.

## 3    Protocol stack
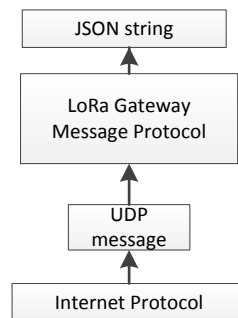


*Figure 1: Semtech LoRa gateway to Semtech LoRa network server protocol stack*

## 4    UDP

All communication between a gateway and a network server is over UDP [1].  The UDP source and destination port numbers shall be as listed in Table 1.

The GWMP message occupies the entire user data area of the UDP packet.  The gateway shall periodically transmit a PULL_DATA message to the network server, in order to keep any intervening firewall open.

| GWMP message type | Direction | Gateway UDP source port | Network server UDP port | Gateway UDP destination port |
|---|---|---|---|---|
| PUSH_DATA | To server | Arbitrary | | |
| PUSH_ACK | To gateway | | | The source port of the PUSH_DATA message to which this message is an acknowledgement |
| PULL_DATA | To server | Arbitrary | Server GWMP port (default 1700) | |
| PULL_ACK | To gateway | | | The source port of the PULL_DATA message to which this message is an acknowledgement |
| PULL_RESP | To gateway | | | The source port of the most recent PULL_DATA message received by the network server from the destination gateway |

*Table 1: UDP port used to send and receive GWMP messages*

# 5 Gateway message protocol

## 5.1 Sequence diagrams

### 5.1.1 Gateway PULL_DATA

The gateway sends a PULL_DATA message in order to keep open any firewall protecting the network server.

The period between the transmissions of PULL_DATA messages is configured in the gateway.

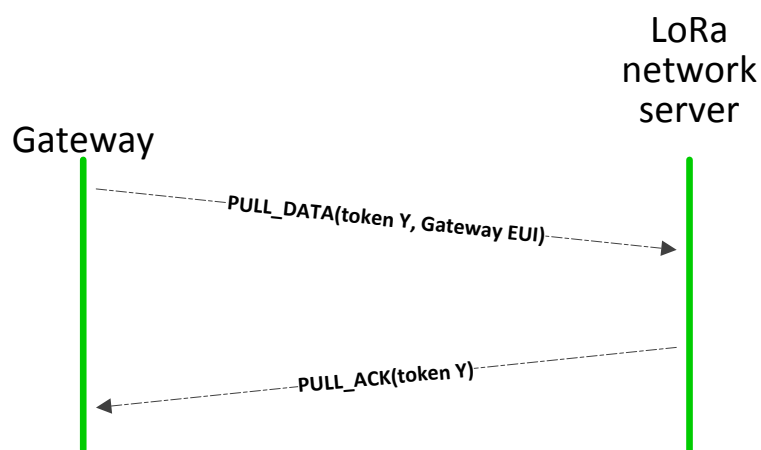The gateway does not react to a missing PULL_ACK message.



*Figure 2: Gateway PULL_DATA*

### 5.1.2  Upstream message flow

The gateway sends a PUSH_DATA message, containing the data being transferred to the network server.  The network server responds with a PUSH_ACK message, containing the sequence number contained in the PUSH_DATA message.

The gateway does not react to a missing PUSH_ACK message.
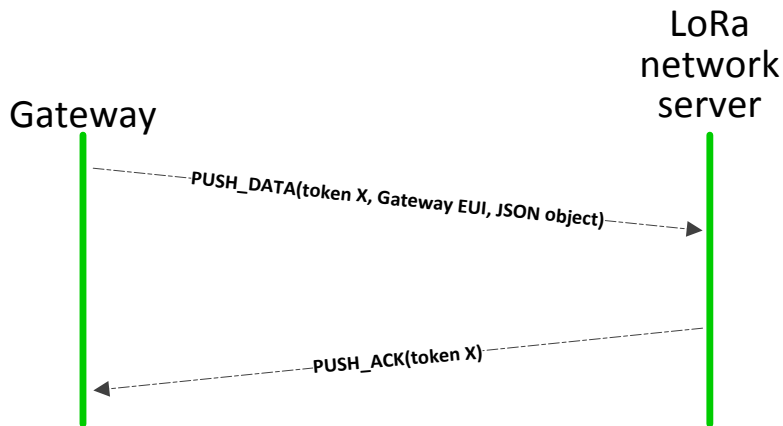


*Figure 3: Upstream GWMP sequence diagram*

### 5.1.3  Downstream message flow

The network server sends a PULL_RESP message, containing the data being transferred to the gateway.
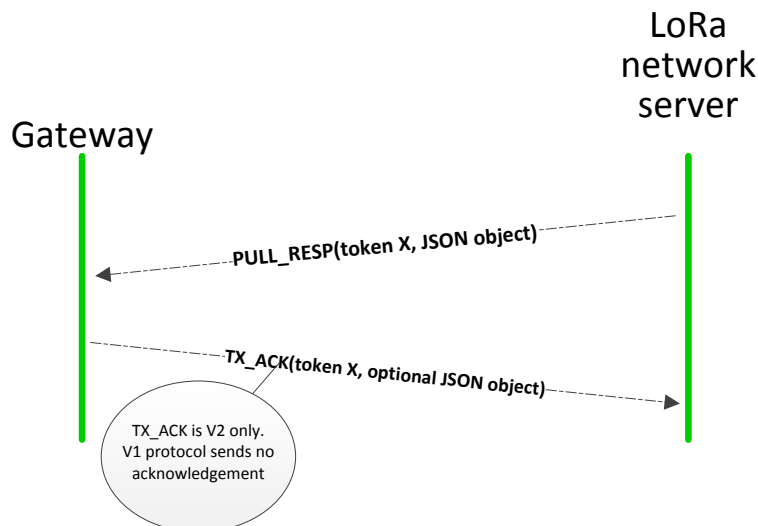


*Figure 4: Downstream GWMP sequence diagram*

## 5.2 Message formats

### 5.2.1 PUSH_DATA message

The PUSH_DATA message transports its payload, a JSON object, from the LoRa gateway to the LoRa network server.

The length of a PUSH_DATA message shall not exceed 2408 octets.

| Offset (from start) | Number of octets | Function | Value or description |
|---|---|---|---|
| 0 | 1 | Protocol version | 0x01 or 0x02 |
| 1 | 2 | Token | Arbitrary value set by Gateway |
| 3 | 1 | PUSH_DATA identifier | 0x00 |
| 4 | 8 | Gateway EUI | |
| 12 | | JSON object | |

*Table 2: PUSH_DATA message format*

### 5.2.2 PUSH_ACK message

The PUSH_ACK message is transmitted by the network server to acknowledge receipt of a PUSH_DATA message.

The server shall transmit a PUSH_ACK message immediately on receipt of a PUSH_DATA message.

| Offset (from start) | Number of octets | Function | Value or description |
|---|---|---|---|
| 0 | 1 | Protocol version | 0x01 or 0x02 |
| 1 | 2 | Token | The value of the token of the PUSH_DATA message that is being acknowledged |
| 3 | 1 | PUSH_ACK identifier | 0x01 |

*Table 3: PUSH_ACK message format*

### 5.2.3 PULL_DATA message

The PULL_DATA messages are periodically transmitted to the LoRa network server in order to inform the server of the UDP port number to which the network server should send any PULL_RESP message.

The PULL_DATA message also keeps open any firewall that protects the LoRa gateway.

| Offset (from start) | Number of octets | Function | Value or description |
|---|---|---|---|
| 0 | 1 | Protocol version | 0x01 or 0x02 |
| 1 | 2 | Token | Arbitrary value set by Gateway |
| 3 | 1 | PULL_DATA identifier | 0x02 |
| 4 | 8 | Gateway EUI | |

*Table 4: PULL_DATA message format*

## 5.2.4 PULL_ACK message

The PULL_ACK message is used by the network server to acknowledge receipt of a PULL_DATA message.

The server shall transmit a PULL_ACK message immediately on receipt of a PULL_DATA message.

| Offset (from start) | Number of octets | Function | Value or description |
|---|---|---|---|
| 0 | 1 | Protocol version | 0x01 or 0x02 |
| 1 | 2 | Token | The value of the token of the PULL_DATA message that is being acknowledged |
| 3 | 1 | PULL_ACK identifier | 0x04 |
| 4 | 8 | Gateway EUI | |

*Table 5: PULL_ACK message format*

## 5.2.5 PULL_RESP message

The PUSH_DATA message transports its payload, a JSON object, from the LoRa network server to the LoRa gateway.

The length of a PUSH_RESP message shall not exceed 1000 octets.

| Offset (from start) | Number of octets | Function | Value or description |
|---|---|---|---|
| 0 | 1 | Protocol version | 0x01 or 0x02 |
| 1 | 2 | Unused (V1)/ Token (V2) | If protocol version is 1, transmit as zero, ignore on receipt. If protocol version is 2, the |
| 3 | 1 | PULL_RESP identifier | 0x03 |
| 4 | | Payload | |

*Table 6: PULL_RESP message format*

## 5.2.6  TX_ACK message (Version 2 only)

The TX_ACK message is sent by a gateway to the network server as an acknowledgement to a received PULL_RESP message.

| Offset (from start) | Number of octets | Function | Value or description |
|---|---|---|---|
| 0 | 1 | Protocol version | 0x02 |
| 1 | 2 | Token | If protocol version is 1, transmit as zero, ignore on receipt. If protocol version is 2, the |
| 3 | 1 | TX_ACK identifier | 0x05 |
| 4 | | Payload | If no error is reported, the 'Payload' field comprises one octet of value '\0'. If an error is reported, the field contains a JSON "error" object. |

*Table 7: TX_ACK message format*

The JSON "error" object is described in Section 6.1.2

# 6   JSON protocol

The JSON protocol is defined by [2].  Within the LoRa system, JSON objects shall comprise only ASCII characters.

No JSON object shall contain white-space that is outside quoted text (i.e. names and text values). The examples given in this section are, however, printed with white-space to aid the reader.

The top level JSON objects contain many lower level JSON objects, of which many are not required. Some of the descriptions of these object rely on this.  For example, the description of optional Boolean object as 'not true' means either its value is 'false' or the object is not present.

## 6.1   Top level

### 6.1.1   Upstream

The root JSON object shall contain zero or more "rxpk" objects, zero or one "stat" object and zero or more other, unknown, JSON objects.  The "rxpk" objects may or may not be the elements of an array.

```
{
    "rxpk":[ {...}, ...],
    "rxpk":{...},
    "stat":{…},
    "other":{…},
    "other1":value
}
```

### 6.1.2   Upstream TX_ACK messages only

A TX_ACK message may contain a root JSON object.  The object, if present, shall contain a single "error" object.

### 6.1.3   Downstream

The root JSON object shall contain zero or more "txpk" objects.

```
{
    "txpk":{...}
}
```

## 6.2 Components

### 6.2.1 stat (upstream)

| Name | Required | Type | Function |
|---|---|---|---|
| time | No | string | UTC system time of the gateway. The precision is one second. The format is ISO 8601 [3] 'expanded' format |
| lati | No | float, containing up to 5 decimal places | The latitude of the gateway's position in units of degrees North of the equator. |
| long | No | float, containing up to 5 decimal places | The longitude of the gateway's position in units of degrees East of the prime meridian. |
| alti | No | signed integer | The altitude of the gateway's position in units of metres above sea level (as defined by the United States' GPS system). |
| rxnb | No | unsigned integer | The number of radio frames received since gateway start |
| rxok | No | unsigned integer | The number of radio frames received with correct CRC since gateway start |
| rwfw | No | unsigned integer | The number of radio frames forwarded to the gateway's network server since gateway start |
| ackr | No | unsigned integer | The proportion of radio frames that were forwarded to the gateway's network server and acknowledged by the server since gateway start. The proportion is expressed as a percentage. |
| dwnb | No | unsigned integer | The number of radio frames received (from the network server) for transmission since gateway start |
| txnb | No | unsigned integer | The number of radio frames transmitted since gateway start |

*Table 8: "stat" elements*

### 6.2.1.1 Example

White-spaces, indentation and newlines are added to aid comprehension

```
"stat":
{
    "time":"2014-01-12 08:59:28 GMT",
    "lati":46.24000,
    "long":3.25230,
    "alti":145,
    "rxnb":2,
    "rxok":2,
    "rxfw":2,
    "ackr":100.0,
    "dwnb":2,
    "txnb":2
}
```

## 6.2.2 rxpk (upstream)

| Name | Required | Type | Function |
|---|---|---|---|
| time | No | string | UTC time of receipt of the LoRa frame. The precision is one microsecond. The format is ISO 8601 'compact' format ( [3]). The object is present only when the receiving gateway has a source of accurate time. |
| tmst | Yes | unsigned integer < $2^{32}$ | The value of the gateway internal time counter at the instant the LoRa frame was received, with microsecond granularity. The value will rollover approximately every 72 minutes.<br><br>The timestamp values generated by different gateways are unrelated. |
| freq | Yes | unsigned float, Hz precision | The centre frequency of the received signal in units of MHz. |
| chan | Yes | unsigned integer | Concentrator "IF" channel on which the frame was received |
| rfch | Yes | unsigned integer | Concentrator radio frequency chain on which the frame was received |
| stat | Yes | signed integer | The result of the gateway's CRC test on the frame.<br>• 1 = correct<br>• -1 = incorrect<br>• 0 = no CRC test was performed |
| modu | Yes | string | The modulation technique used:<br>• "LORA", representing LoRa modulation<br>• "FSK", representing FSK modulation |
| datr | Yes | string | Datarate identifier.<br>When "modu" equals "LORA", "datr" comprises "SFnBWm", where 'n' is an integer representing the frame's 'spreading factor' and 'm' is an integer representing the frame's bandwidth in units of kHz.<br>When "modu" equals "FSK" "datr" comprises an integer representing the frame's bit rate in Hz |
| codr | Yes, if "modu" equals "LoRa" | string | ECC code rate. "codr" comprises the string "k/n", where 'k' represents the carried bits and 'n' the total number of bits received, including those used by the error checking/correction algorithm. |
| rssi | Yes | signed integer | The measured received signal strength in units of dBm. |
| lsnr | Yes | signed float, containing a maximum of 1 decimal place | The measured received signal to noise ratio in units of dB. |
| size | No | unsigned integer | The number of octets in the received frame. |
| data | Yes | string | The frame payload, encoded into Base64, [4]. The Base64 padding characters shall not be added. |

*Table 9: "rxpk" elements*

## 6.2.2.1 Example

White-spaces, indentation and newlines are added to aid comprehension

```
"rxpk":
[
    {
        "time":"2013-03-31T16:21:17.528002Z",
        "tmst":3512348611,
        "chan":2,
        "rfch":0,
        "freq":866.349812,
        "stat":1,
        "modu":"LORA",
        "datr":"SF7BW125",
        "codr":"4/6",
        "rssi":-35,
        "lsnr":5.1,
        "size":32,
        "data":"-DS4CGaDCdG+48eJNM3Vai-zDpsR71Pn9CPA9uCON84"
    },
    {
        "time":"2013-03-31T16:21:17.532038Z",
        "tmst":3316387610,
        "chan":0,
        "rfch":0,
        "freq":863.00981,
        "stat":1,
        "modu":"LORA",
        "datr":"SF10BW125",
        "codr":"4/7",
        "rssi":-38,
        "lsnr":5.5,
        "size":32,
        "data":"ysgRl452xNLep9S1NTIg2lomKDxUgn3DJ7DE+b00Ass"
    }
]
```

## 6.2.3  error (upstream)

The value of the object is a description of the cause of the error.  Expected values include:

| Text | Description |
|---|---|
| TOO_LATE | Rejected because it was already too late to program this packet for downlink |
| TOO_EARLY | Rejected because downlink packet timestamp was received by the gateway too long before the scheduled transmission time |
| COLLISION_PACKET | Rejected because there was already a packet programmed in requested timeframe |
| COLLISION_BEACON | Rejected because there was already a beacon planned in requested timeframe |
| TX_FREQ | Rejected because requested frequency is not supported by TX RF chain |
| TX_POWER | Rejected because requested power is not supported by gateway |
| GPS_UNLOCKED | Rejected because GPS is unlocked, so GPS timestamp cannot be used |

*Table 10: Description of TX_ACK error values*

## 6.2.3.1  Example

"error":"TOO_LATE"

## 6.2.4  txpk (downstream)

| Name | Required | Type | Function |
|---|---|---|---|
| imme | No | Boolean | If true, the gateway is commanded to transmit the frame immediately |
| tmst | No | unsigned integer < $2^{32}$ | If "imme" is not true and "tmst" is present, the gateway is commanded to transmit the frame when its internal timestamp counter equals the value of "tmst". Section 6.2.2 contains a description of the gateway timestamp counter. |
| time | Y | string | UTC time.  The precision is one microsecond.  The format is ISO 8601 ( [3]) 'compact' format. If "imme" is false or not present and "tmst" is not present, the gateway is commanded to transmit the frame at this time. |
| freq | N | unsigned float, Hz precision | The centre frequency on when the frame is to be transmitted in units of MHz. |
| rfch | Y | unsigned integer | The antenna on which the gateway is commanded to transmit the frame. |
| powe | N | signed integer | The output power which what the gateway is commanded to transmit the frame |
| modu | N | string | The modulation technique to be used:<br>• "LORA", representing LoRa modulation<br>• "FSK", representing FSK modulation |
| datr | N | string | Datarate identifier. When "modu" equals "LORA", "datr" comprises "SFnBWm", where 'n' is an integer representing the frame's 'spreading factor' and 'm' is an integer representing the frame's bandwidth in units of kHz. When "modu" equals "FSK" "datr" comprises an integer representing the frame's bit rate in units of Hz |
| codr | Yes, if "modu" equals "LoRa" | string | ECC code rate.  "codr" comprises the string "k/n", where 'k' represents the carried bits and 'n' the total number of bits transmitted, including those added by the error checking/correction algorithm. Transmitted only when "modu" equals "LORA" |
| ipol | Y | bool | If true, commands gateway to invert the polarity of the transmitted bits.  LoRa Server sets value to true when "modu" equals "LORA", otherwise the value is omitted. |
| size | N | unsigned integer | The number of octets in the received frame. |
| data | N | string | The frame payload, encoded into Base64, [4]. Base64 padding characters shall be not be not added. |
| ncrc | N | bool | If not false, disable physical layer CRC generation by the transmitter. |

*Table 11: "txpk" elements*

## 6.2.4.1  Example

```
"txpk":
{
    "imme":true,
    "freq":864.123456,
    "rfch":0,
    "powe":14,
    "modu":"LORA",
    "datr":"SF11BW125",
    "codr":"4/6",
    "ipol":false,
    "size":32,
    "data":"H3P3N2i9qc4yt7rK7ldqoeCVJGBybzPY5h1Dd7P7p8v"
}

"txpk":
{
    "imme":true,
    "freq":861.3,
    "rfch":0,
    "powe":12,
    "modu":"FSK",
    "datr":50000,
    "fdev":3000,
    "size":32,
    "data":"H3P3N2i9qc4yt7rK7ldqoeCVJGBybzPY5h1Dd7P7p8v"
}
```

# 7  Glossary

| | |
|---|---|
| ASCII: | American Standard Code for Information Interchange.  A widely used standard for representing Latin text, Arabic numerals and punctuation as binary values. |
| Base64: | A method of encoding binary data into ASCII text.  The LoRa system uses Base64 to transport LoRa frames in JSON objects. Base64 is defined by IETF RFC 4648 [4]. |
| CRC: | Cyclic Redundancy Check |
| Cyclic Redundancy Check: | A method of detecting transmission errors.  CRC is particularly easy to implement and is robust to corruption from inversion of many nearby bits. |
| dB: | decibel; a logarithmic ratio of power.  Defined by Bell Laboratories |
| dBm | A logarithmic measure of power, decibel, relative to 1mW |
| Downstream: | Toward the mote |
| End-device: | Synonymous with 'mote' |
| EUI: | Extended Unique Identifier.  In this document 'EUI' refers to a value from the 'EUI-64' number space managed by the IEEE. |
| Firewall: | A firewall is a network security system that controls the incoming and outgoing network traffic based on an applied rule set. A firewall establishes a barrier between a trusted, secure, internal network and another network (e.g., the Internet) that is assumed not to be secure and trusted. |
| FSK: | Frequency shift keying.  FSK is a modulation technique that encodes each frame bit value using a shift of the carrier frequency. |
| Gateway: | A LoRa gateway transmits LoRa frames to, and receives LoRa frames from, LoRa motes |
| GWMP: | Gateway message protocol.  The protocol used to transport JSON objects between the network server and the gateways. GWMP is defined by this document. |
| IEEE: | Institution of Electrical and Electronic Engineers (www.ieee.org). |
| IETF: | Internet Engineering Task Force (www.ietf.org). |
| IP: | Internet Protocol |
| IP port address | An IP address or host name and either a UDP or a TCP port number. This document represents a port address in the form <IP address>:<port number> or <host name>:<port number>. E.g. 1.2.3.4:4500 or a.com:4500. |

| | |
|---|---|
| JSON: | JavaScript Object Notation. JSON is a textual based method of representing name, value pairs. The value of an object may itself be a JSON object. Within LoRa, JSON objects contain only ASCII characters. |
| JSON object | A JSON name, value pair. |
| LoRa: | Long Range. Defined by the LoRa Alliance |
| LoRa Alliance: | The industry body that defines the LoRaWAN protocol. |
| LoRaWAN: | The protocol by which a LoRa mote communicates with a LoRa gateway. LoRaWAN is defined by the LoRa Alliance [5] |
| Metadata: | LoRa Metadata refers to information about the transmission or reception of a LoRa frame. |
| Mote: | A LoRa end device. A LoRa mote communicates with a LoRa Gateway using the LoRa MAC or LoRa WAN protocol. |
| NS: | The LoRa network Server |
| RSSI: | Received Signal Strength Indication. The power of the received signal, normally measured in dBm. |
| Rx: | Receive |
| Signal quality: | The signal quality is normally measured in dBm and is the sum of the SNR (measured in dB) and the RSSI (measured in dBm). |
| SNR: | Ratio of signal power to noise power, normally measured in dB. |
| Spreading factor: | A parameter of a LoRa transmission. Two to the power of 'spreading factor' 'on the air' bits are transmitted to represent each frame bit. |
| Thread: | An independent path of execution within a process. The threads of a process share access to memory within the process. |
| Tx: | Transmit |
| UDP: | User Datagram protocol: a simple protocol for transporting data messages. Delivery is not guaranteed. In addition the order of receipt is not necessarily the same as the order of transmission. |
| upstream: | Away from the mote |
| UTC | Co-ordinated Universal Time; also known as Greenwich Mean Time and Zulu |

# 8   References

Each trademark is the property of its owner.

[1] J. Postel, "User Datagram Protocol RFC 768," 28 August 1980. [Online]. Available: https://www.ietf.org/rfc/rfc768.txt.

[2] ECMA International, The JSON Data Interchange Format, 2013.

[3] ISO, *ISO 8601:2004 Data elements and interchange formats -- Information interchange -- Representation of dates and times,* 2004.

[4] IETF, "The Base16, Base32, and Base64 Data Encodings," October 2006. [Online]. Available: https://www.ietf.org/rfc/rfc4648.txt.

[5] LoRa Alliance, "LoRaWAN Specification," LoRa Alliance, 2015.

**Contact Information**

**Semtech Corporation**
**Wireless Sensing and Timing Products Division**
**200 Flynn Road, Camarillo, CA 93012**
**Phone: (805) 498-2111 Fax: (805) 498-3804**
**E-mail: support_rf_na@semtech.com**
**Internet: http://www.semtech.com**