Application Note: AN1200.94

LoRaWAN® Theory for the One-Channel Hub

Table Of Contents

1 Introduction

The One Channel Hub is a low cost LoRaWAN® Access Point reference design, based on the ESP32-S3 and SX126x shield (also works with LLCC68 and LR1121). It only supports one channel and one data rate at any moment. It uses Wi-Fi as backhaul and can be configured from a web UI.

The Hub cannot replace a traditional 8-channel LoRaWAN gateway because of its limited capacity. It can however be a good substitute or complement for situations where only a few end-devices are within range of the access point.

Before a Hub can operate in a LoRaWAN network, the devices and LNS it communicates with need customizing:

- End-device
  - Personalize to use a single data rate, and transmit only on the same channel as its parent Hub
- LoRa® Network Server (LNS)
  - Disable the Adaptive Data Rate (ADR) for end-devices communicating with a Hub
  - Send a CFList, with just the parent Hub channel enabled, in response to the Join-Request frame
  - Send a LinkADRReq MAC Command to limit the end-device to just 1 data rate.

In this application note, we explain why such changes are necessary or helpful to optimize the operation of the Hub, and some possible ways to implement them.

2 End-Device Customization

2.1 The Join Process and How to Speed It Up

In the general case, when a device wants to join to a LoRaWAN network, the Join-Request frame it sends out may be transmitted using any valid data rate (controlled by the Spreading Factor) and follow a random frequency-hopping sequence across specified Join channels. A generic LoRaWAN gateway would listen on 8 different channels and support all those data rates.

However, if a Hub is used to route messages to a LoRaWAN network, depending on the region of operation it may take some time for the Hub to receive the Join-Request frame, because the Hub only supports a single channel and a single data rate. To speed up this join process, the end-device firmware should be modified as follows:

1. Restrict the end-device to use just the Hub's single uplink channel as its Join channel.
2. Restrict the end-device to use just the Hub's single data rate.

The Hub could then receive a Join-Request frame from any device that is in range and that respects the LoRaWAN Standard Join process. End-devices that do not respect the Join process defined in the LoRaWAN Regional Parameters specification may not be able to obtain LoRaWAN certification, despite being compatible with LoRaWAN networks.

2.2 Example: Customizing An End-Device Running LoRa Basics Modem® (LBM)

LBM's advanced APIs can optimize the Join duration and ensure channel and data rate alignment between a Hub and an end-device running LBM. The following commands indicate to LBM which data rate to use (according to the Hub configuration) to send the next uplinks.

Before calling the `smtc_modem_join_network()`, the user application can call:

```
- smtc_modem_adr_set_join_distribution(id, adr_custom_list)

- smtc_modem_adr_set_profile(id, SMTC_MODEM_ADR_PROFILE_CUSTOM, adr_custom_list)
```

With:

```
static const uint8_t adr_custom_list[16] = \

{ \

       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5 \

};
```

With this configuration, the end-device running LBM will still try to join on all mandatory join channels of the region, but will quickly get on the channel of the Hub using the correct data rate.

After this, the LNS sends a downlink back to the end-device to set the channel mask to the selected channel; so the end-device has just a single channel and a single data rate enabled, which match those of the Hub, allowing seamless operation thereafter.

3 LoRa Network Server Customization

When the LoRa network server (LNS) receives the Join-Request frame, it responds with a Join-Accept frame. This frame may contain some network parameters (called CFList) for the network that the end-device is onboarding to. The CFList replaces all the channels that are stored in the end-device, apart from the default channels defined by the LoRaWAN Regional Parameters. The newly defined channels are immediately enabled and usable by the end-device for communication.

To control any end-devices that communicate with the Hub, the LNS must send a customized CFList field appended to the Join Accept frame, to guide the end-device to only use one specific channel; later, the LNS also needs to send the LinkADRReq MAC command to limit the end-device data rate to a single SF.

There are two ways for the LNS to distinguish the end-devices needing customization from those communicating with standard gateways:

- If the end-device was specified during registration as operating with a Hub or standard gateway
- If the access point was specified as a Hub or a standard gateway

Some LNSs (such as TTN) use the first way by providing methods to add new frequency plans when registering with the end-device and the gateway. Other LNSs (such as ChirpStack) don't treat end-devices differently at the registration, so they follow the second solution, explained in the following sections.

3.1 How Can the Hub Be Recognized?

If all end-devices are treated the same, then when LNS receives a Join-Request frame, it needs to know if it comes from a generic LoRaWAN gateway or a Hub, which means it needs a way to distinguish between these two types of access points.

Possible solutions:

- The LNS provides an option for that feature in the access point registration step (for example, add a radio button in the registration web page), and applies the correct profile accordingly.
- The access point distinguishes itself by using a specific UDP port, and uplinks packets before being forward to the LNS.

3.2 Example: ChirpStack

Currently ChirpStack only supports the second solution of specifying a UDP port.

Let's assume the Hub is in the AS923 channel plan region and listens on 923.4 MHz and at SF7. The step of "tagging" is handled by ChirpStack-gateway-bridge.

The following sub-sections show how to implement this purpose.

3.2.1 Customizing ChirpStack Gateway Bridge

1. Create a new profile named "chirpstack-gateway-bridge-as923-1ch.toml", put under folder "/etc/chirpstack-gateway-bridge" (It may also need to be the user owner of "gatewaybridge" and group owner of "gatewaybridge"). Let this profile listen on port 1717.

   Below is a configure example:

```
[backend]
type="semtech_udp"
  [backend.semtech_udp]
  udp_bind = "0.0.0.0:1717"
[integration]
marshaler="protobuf"
  [integration.mqtt]
  event_topic_template="as923_1ch/gateway/{{ .GatewayID }}/event/{{ .EventType }}"
  command_topic_template="as923_1ch/gateway/{{ .GatewayID }}/command/#"
  [integration.mqtt.auth]
  type="generic"
    [integration.mqtt.auth.generic]
    server="tcp://127.0.0.1:1883"
    username=""
    password=""
```

2. Restart chirpstack-gateway-bridge.

3.2.2 Customizing ChirpStack

3. For ChirpStack NS, create a profile "/etc/chirpstack/region_as923_1ch.toml" (change user owner and group owner to be "chirpstack").

   An example could be as follows:

```
[[regions]]
  id="as923_1ch"
  description="AS923_1ch"
  common_name="AS923"
  [regions.gateway]
    force_gws_private=false
    [regions.gateway.backend]
      enabled="mqtt"
      [regions.gateway.backend.mqtt]
        topic_prefix="as923"
        share_name="chirpstack"
        server="tcp://localhost:1883"
```

```
        username=""
        password=""
        qos=0
        clean_session=false
        client_id=""
        keep_alive_interval="30s"
        ca_cert=""
        tls_cert=""
        tls_key=""
    [regions.network]
      installation_margin=10
      rx_window=0
      rx1_delay=1
      rx1_dr_offset=0
      rx2_dr=2
      rx2_frequency=923200000
      rx2_prefer_on_rx1_dr_lt=0
      rx2_prefer_on_link_budget=false
      downlink_tx_power=-1
      adr_disabled=true
      min_dr=5
      max_dr=5
      enabled_uplink_channels=[1]
```

Note: The last 4 lines are the critical settings for the Hub to work: Disable ADR, set data rate to SF7 and enable only one uplink channel.

4. Add this profile to the "**enabled_regions**" list:

```
enabled_regions=[
  …,
  "as923_1ch",
  …
]
```

5. Restart Chirpstack.

3.2.3 Using the One Channel Hub with The Things Network

In order to operate a local LoRaWAN network using The Things Network (thethingsnetwork.org) with one (or more) 1-Channel Hub(s), it is necessary to register the hubs and end-devices using the proper frequency plan to indicate to the TTN LNS that a hub is being used.

The one channel Hub has to be registered as a regular gateway, but a specific frequency plan indicating that this gateway has a single channel must be used:

- EU868: Europe 868.1 MHz
- US915: United States 903.0 MHz
- CN490: China 470.3 MHz

These dedicated frequency plans are made available on the TTN LNS.

Similarly, the end-device registration has to specify the same frequency plan as the one used for the hub.

Also, it is recommended to disable the Adaptative Datarate (ADR) for the registered end-device in order to ensure that the end-device keeps using the datarate configured on the hub.

This can be done on with the TTN user interface with the following steps:

- end-device -> Settings
- Network layer -> press the expand button
- Advanced MAC settings
- Adaptative data rate (ADR) -> select Disabled or Static Mode with ADR data rate index configured to the one used by the hub, and ADR number of transmissions set as wanted.

4 References

1. TS001-1.0.4 LoRaWAN® L2 1.0.4 Specification

5 Glossary

Term	Description
ADR	Adaptive Data Rate
LoRa®	Long Range Communication The LoRa® Mark is a registered trademark of the Semtech Corporation
LoRaWAN®	LoRa® Wide Area Network Standard The LoRaWAN® mark is used under license from the LoRa Alliance®
LNS	LoRa Network Server
SF	LoRa® modulation Spreading Factor

6 Revision History

Version	ECO	Date	Changes and/or Modifications
01.00	ECO-072823	Sept. 2024	Initial Release

Important Notice

Information relating to this product and the application or design described herein is believed to be reliable, however such information is provided as a guide only and Semtech assumes no liability for any errors in this document, or for the application or design described herein.

Semtech reserves the right to make changes to the product or this document at any time without notice. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Semtech warrants performance of its products to the specifications applicable at the time of sale, and all sales are made in accordance with Semtech's standard terms and conditions of sale.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS, OR IN NUCLEAR APPLICATIONS IN WHICH THE FAILURE COULD BE REASONABLY EXPECTED TO RESULT IN PERSONAL INJURY, LOSS OF LIFE OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK.

Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

The Semtech name and logo are registered trademarks of the Semtech Corporation. All other trademarks and trade names mentioned may be marks and names of Semtech or their respective companies. Semtech reserves the right to make changes to, or discontinue any products described in this document without further notice. Semtech makes no warranty, representation or guarantee, express or implied, regarding the suitability of its products for any particular purpose. All rights reserved.

© Semtech 2024