Conduit AEP - LoRaWAN Upgrade guide

Conduit AEP v1.4.16 / mLinux v3.3.23
LoRa Network Server – v2.0.19

Table of Contents

1 Introduction

The latest update to AEP and mLinux provides a major upgrade for the LoRa Network Server in the system and the UI.

Queries to the network server for packets, gateways and downlink queue have been added along with corresponding UI pages.

Support for multiple gateways reporting to a single network server instance has been added to increase network area or supported frequencies.

1.1 Glossary

A list of LoRaWAN acronyms and definitions can be found on multitech.net.

http://www.multitech.net/developer/software/lora/glossary/

1.2 Maximum End-Devices and Gateways

The Conduit supports a maximum 2000 end-devices and up to 10 additional gateways connected at one time. An end-device is considered connected when the LoRaWAN keys or session information is configured on the Conduit. The number of end-devices and gateways is limited by the size of storage allocated for configuration and processing power.

1.3 Existing End-Device Records

Existing end-device sessions will be retained, any joined end-devices should still be able to communicate with the Conduit after upgrade. Unique end-device DevEUI/AppKey settings will be output to a file, /home/root/whitelist.jsonlines.

1.3.1 AEP

This script can be used to import the keys into the API on an AEP installation using the API.

```
#!/bin/bash
while IFS='' read -r line || [[ -n "$line" ]]; do
    curl 127.0.0.1/api/loraNetwork/whitelist/devices -X POST --data '$line' -H
'Content-Type: application/json'
done < "/home/root/whitelist.jsonlines"
```

1.3.2 mLinux

On mLinux the whilelist can be added to /var/config/lora/lora-network-server.json whitelist section.

Commas must be added between device records. The following script can be used to ouput the whitelist in JSON array format, although the last comma must be removed.

```
#!/bin/bash
while IFS='' read -r line || [[ -n "$line" ]]; do
    echo “$line,”
done < "/home/root/whitelist.jsonlines"


/var/config/lora/lora-network-server.json

“whitelist”: {
      “devices”: [
            {
                  “deveui”: “0011223344556677”,
                  “appeui”: “0011223344556677”,
                  “appkey”: “00112233445566778899AABBCCDDEEFF”,
                  “class”: “A”
            }
      ],
      “enabled”: true
}
```

1.4 New Network Mode

Public LoRaWAN is new default mode.

Private LoRaWAN mode has been added to allow LoRaWAN compliant modules to connect to the Conduit using the private sync word 0x12.

Private MTS is the previous default, the private sync word was used and downlink frequencies in US/AU differ from LoRaWAN to separate Frequency SubBands. This mode is provided for backwards compatibility with existing end-device firmware. It is designed for an 8 channel LoRaWAN network. If support for more than 8 channels or a cloud join server is required or may be in the future Public or Private LoRaWAN modes should be used.

Network Mode, Join Delay and Rx Delay are independently configurable.

Network Mode	Private MTS	Public LoRaWAN	Private LoRaWAN
Sync Word	0x12	0x34	0x12
Join Delay	1	5	5
Rx1 Delay	1	1	1
Rx1 Freq (EU/AS/KR/IN)	Uplink Freq	Uplink Freq	Uplink Freq
Rx2 Freq (EU/AS/KR/IN)	Set by Region	Set by Region	Set by Region
Rx1 Freq (US/AU)	Uplink / 8*	Uplink % 8	Uplink % 8
Rx2 Freq (US/AU)	Depends on FSB*	923.3 MHz	923.3 MHz

* Incompatibilities with LoRaWAN protocol.

2 AEP

2.1 Services and Configuration

LoRaWAN services must be restarted after changing the AEP configuration.

The AEP configuration must be saved and the Conduit restarted for the changes to be saved to flash.

2.2 New UI Pages

A top level LoRaWAN menu group has been added. LoRa Network Settings has moved from Setup into this new group. New pages

Pages

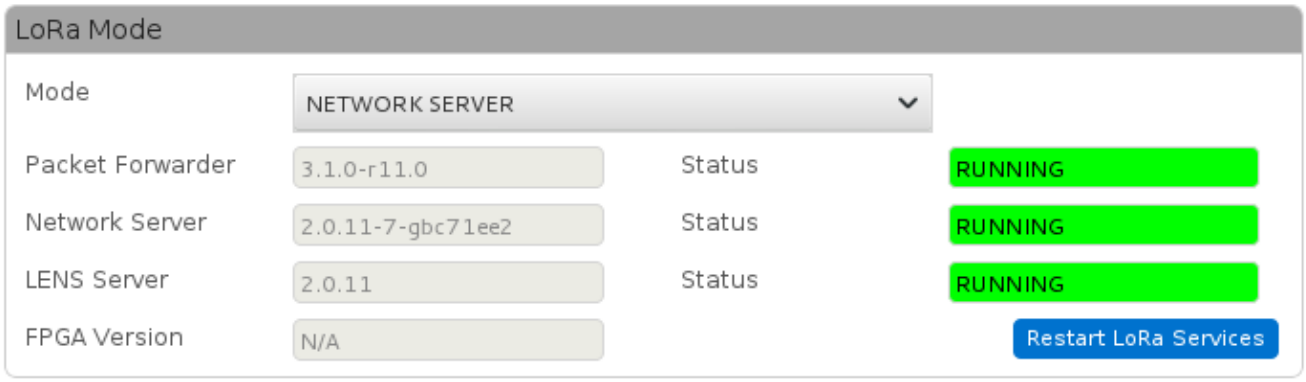
- Network Settings – Configure Network Server channels, database and logging settings
- Key Management – Configure Join Server and end-device AppKey settings
- Gateways – View connected gateway and network statistics
- Device Configuration – Configure end-device default operating class setting
- Device Sessions – View connected end-device session information
- Packets – View recent uplink, downlink, join request, and received packets
- Downlink Queue – View and queue downlink packets to be sent to end-devices

2.2.1 Network Settings

Configuration for the network server channel plan, rx window timing, database, logging, UDP ports and testing options.

Version information for installed hardware/software versions and process status has been added to the top of the page.

2.2.1.1 LoRa Mode



2.2.1.1.1.1 Fields

- **Packet Forwarder** – Version of the installed packet forwarder IPK
- **Network Server** – Version of the installed binary at /opt/lora/lora-network-server
- **Lens Server** – Version of the installed binary at /opt/lora/lora-lens-server
- **FPGA Version** – Version of the firmware installed in the LoRa gateway hardware if available, USB/SPI MTAC-LORA-1.0 cards do not have an FPGA.
  - Version 28 – Outdated firmware, will not work for Spectral Scan or Listen Before Talk
  - Version 31 – Spectral Scan enabled firmware
  - Version 33 – Listen before talk enabled firmware
- **Status**
  - DISABLED – The process is not enabled with the current configuration settings
  - STOPPED – The process has been stopped
  - RUNNING – The process is up and running
  - RESTARTED – The process id has changed since the page was loaded, could indicated an problem with configuration or hardware.
- **Restart LoRa Services** – Restart the enabled LoRa processes, this action is needed after making changes to the LoRaWAN settings.

2.2.1.2 LoRaWAN Network Server Configuration

The Frequency Band shows the supported frequency of the installed LoRa hardware. Gateway LoRa hardware has SAW filters to limit frequencies to the proper range.

The 868 hardware has a filter from 863-870 MHz, there is a drop-off at the low end so the best effective range is 865-870 MHz. Duty-cycle regulations in EU can be raised to 1% if frequencies are

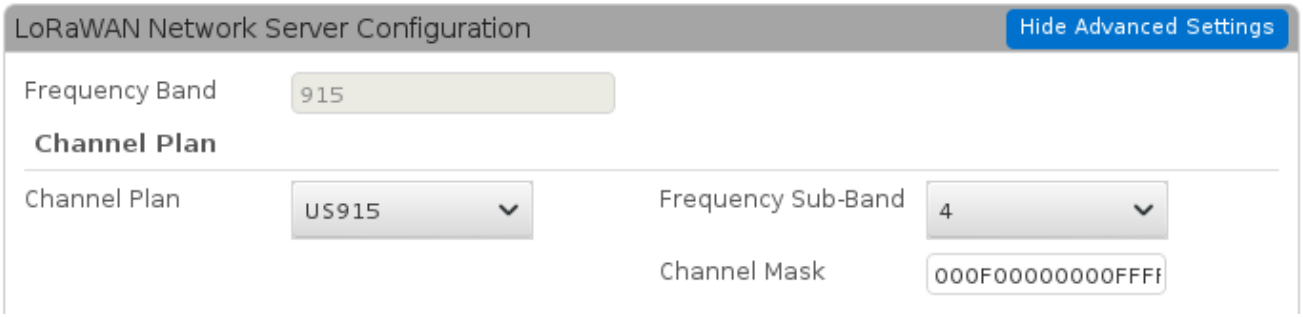
limited to 865-870 MHz.

The 915 hardware has a filter from 902-928 MHz.

2.2.1.2.1 Channel Plan

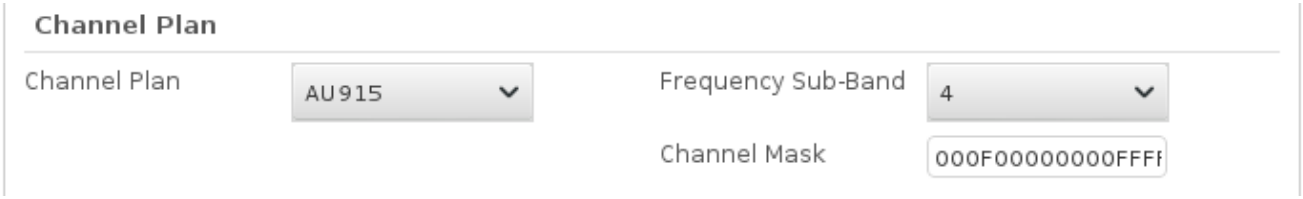
Channel plan option control the available datarates and frequencies for the gateway card to receive and transmit packet. Plans are defined by the LoRa Alliance regional specifications.

2.2.1.2.1.1 US915

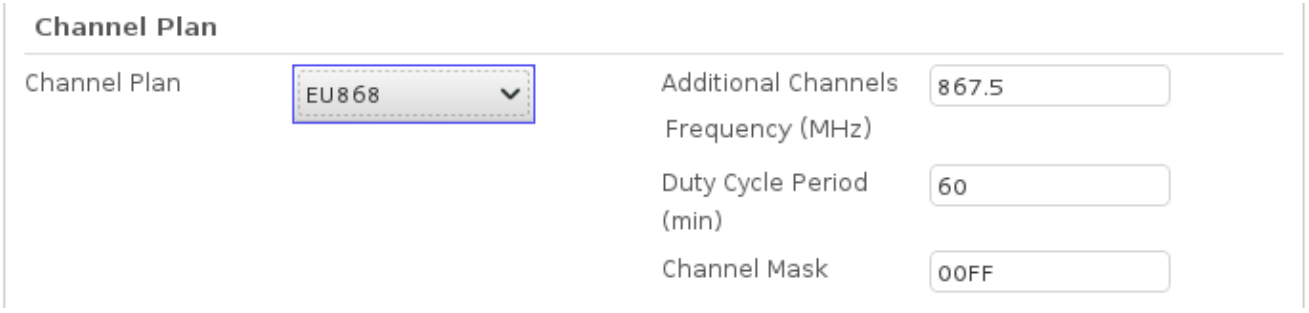


2.2.1.2.1.2

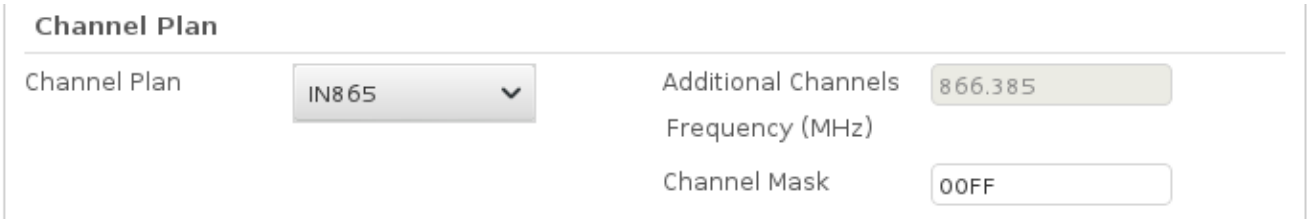
2.2.1.2.1.3 AU915



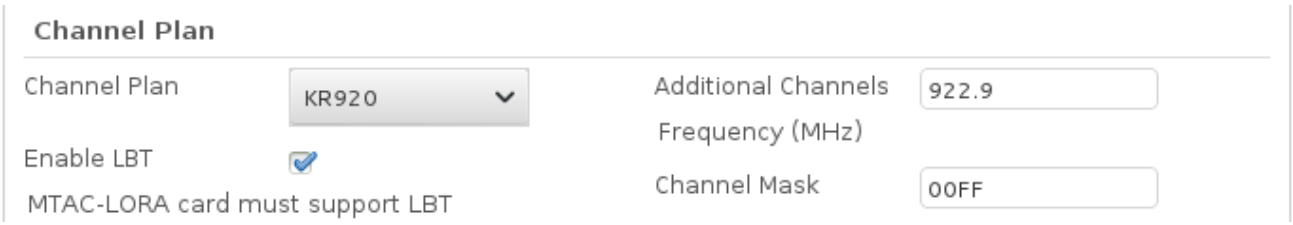
2.2.1.2.1.4 EU868



2.2.1.2.1.5 IN865



2.2.1.2.1.6 KR920



2.2.1.2.1.7 AS923

2.2.1.2.1.8 Fields

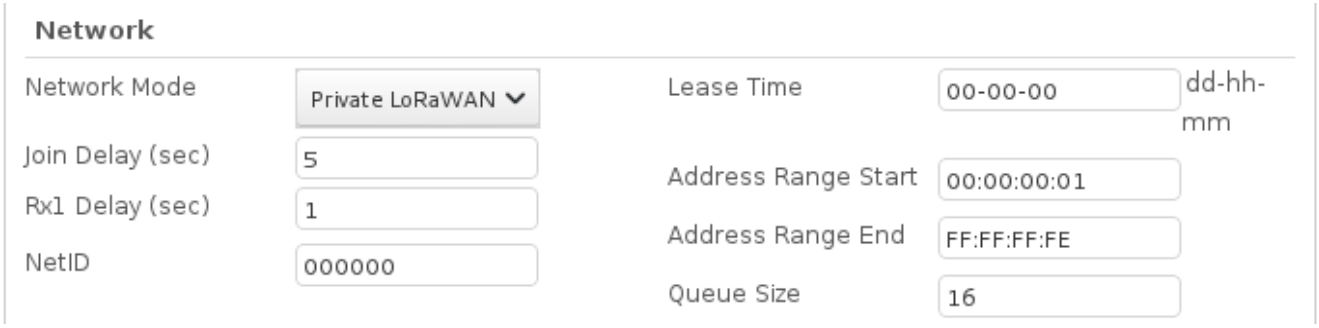
- **Channel Plan** – Select the frequency and datarate limits according to LoRaWAN regional specifications.
  - Select from EU868, IN865, US915, AU915, AS923, KR920
- **Frequency Sub-Band** (US915 and AU915)
  - Choose 8 channels for the gateway to listen for packets from the 64 supported in the regional specification. The 64 channels are divided into 8 sub-bands of 8 channels.
  - FSB1 – Channels 0-7, FSB2 – Channels 8-15, etc...

- **Channel Mask** – Select the supported channels if multiple gateways are configured.
  - Configured mask will be sent using ADR commands in first downlink following an OTAA Join event. For ABPA devices these commands will be sent on first downlink or anytime downlink and uplink counters are reset to 0.
  - US915 and AU915 (64 – 125 KHz channels + 8 – 500 KHz channels)
    - Start with “00” Channels 79-72 are not defined (1-byte), Channels 71-64 (1-byte), Channels 63-0 (8-bytes)
    - FSB1 and FSB2 – `00030000000000000FFFF`
    - FSB1 and FSB8 – `0081FF000000000000FF`
  - EU868, IN865, AS923 and KR920 (up to 16 channels)
    - Enable 8 channels - 00FF
- **Additional Channels** (EU868, IN865, AS923 and KR920)
  - Select an additional set of up to 5 channels centered on the provided frequency in MHz.
  - EU868 – Default: 867.5 MHz
    - Range: 863.5-867.5, 869.5 MHz
    - DR0-DR5 – 867.1, 867.3, 867.5, 867.7, 867.9 MHz
    - DR6 – 868.3 MHz
    - DR7 – 868.8 MHz
    - 869.5 – 868.8, 896.0, 869.525, 869.8 MHz
      - Only 4 channels are available above 868.8 due to alarm bands.
      - 869.525 MHz can be used with 10% duty-cycle
      - 869.8 can be used with 100% duty-cycle if EIRP is below +7 dBm
  - IN865 – Default: 866.385 MHz
    - Range: Fixed, only 866.385 can be configured in the allowed frequencies, 865-867 MHz
    - DR0-DR5 – 865.985, 866.185, 866.385, 866.585, 866.785 MHz
    - DR6 – 865.2 MHz
    - DR7 – 865.5 MHz
  - AS923 – Default: 922.6 MHz
    - Range: 920.5 – 922.6, 924.1 - 927.5 MHz
    - DR0-DR5 – 922.2, 922.4, 922.6, 922.8, 923.0 MHz

        - DR6 – 923.4 MHz
        - DR7 – 923.9 MHz
        - LBT is optional, depends on regional regulations
        - AS923-Japan Default Settings
            - Enable LBT: checked
            - Max EIRP: 16 dBm
            - Dwelltime Up/Down: 1
            - Rx2 Datarate: 2
            - Min Datarate: 2
            - Max Datarate: 5
    - KR920 – Default: 922.9 MHz
        - Range: 921.3 – 921.5, 922.9 MHz.
        - DR0-DR5 – 922.5, 922.7, 922.9, 923.1, 923.3 MHz
        - DR6 and DR7 – disabled, these datarates are not defined in regional specification
        - LBT must be enabled
- **Duty-Cycle period** – configure the size of sliding window used in duty-cycle limits.
    - From start of network server process time-on-air will accrue according to duty-cycle per band. The max amount that can be accrued is set by the Duty-Cycle period setting.
- **Enable LBT** – Enables listen before talk if available for the selected Channel Plans
- **Max EIRP** – Configure the maximum transmission allowed by end-devices. This setting will be transmitted to the end-device in a downlink following OTAA join.
- **Dwelltime Up/Down** – When set to one maximum payloads for each datarate are limted to 400 ms time-on-air. This setting will be transmitted to the end-device in a downlink following OTAA join.

2.2.1.2.2 Network

Network settings for the server control a gateway filter for public/private packets, delays for Rx Windows, Lease Time, Dev Addr range and downlink queue size.

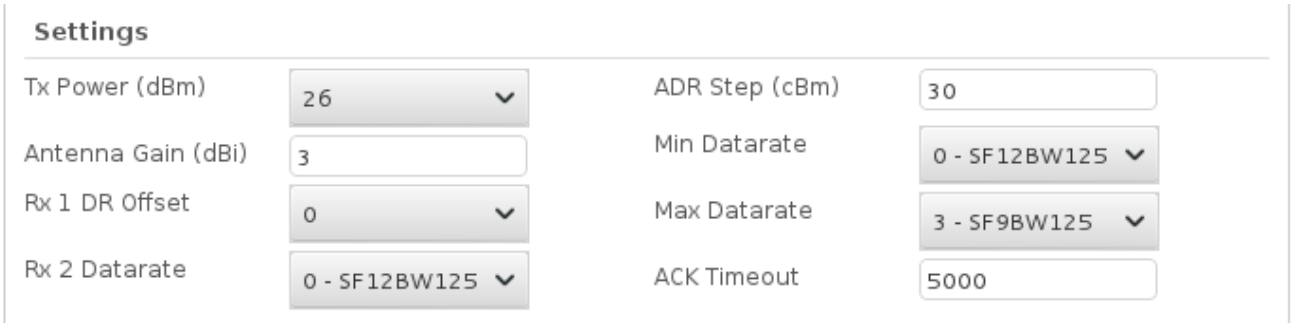


2.2.1.2.2.1 Fields

- **Network Mode** – Choose network type, Private MTS (sync word: 0x12 and US/AU Downlinks per FrequencySubBand), Public LoRaWAN (sync word: 0x34), Private LoRaWAN (sync word: 0x12)
- **Lease Time** – Time until a network session should expire after node-inactivity. The lease-time is tracked from the last received packet. Default is disabled, 00-00-00.
- **Join Delay** – Default setting for public and private network modes is five seconds, this was changed from one second for private networks to allow a round trip to a cloud Join Server during the OTAA process. This delay is used to time the end of TX of a Join Request to the beginning of the first RX window on the end-device.
- **Rx1 Delay** – Default setting is one second, this setting is sent to the end-device in the OTAA Join Accept packet. This delay may need to be extended if the latency for application message exceeds one second to allow the server application to respond in a downlink to the device.
- **NetID** – LoRaWAN NetID setting to be used in assigning network addresses. Private and test networks should use 000000 or 000001. Public networks are assigned a NetID by the LoRa Alliance. The seven least significant bits will become the seven most significant bits of a DevAddr assigned to a joining end-device. Unique values allow the networks to differentiate between network packets received over the air.
- **Address Range Start** – Start of assigned DevAddr range, seven MSB will be overwritten by NetID setting.
- **Address Range End** – End of assigned DevAddr range, seven MSB will be overwritten by NetID setting.
- **Queue Size** – Number of downlinks to hold per end-device, defaults to sixteen. Depending on the number of end-devices in the system and available disk space available this value may need to be adjusted.

2.2.1.2.3 Settings

The settings section configures transmit power of the gateways, datarates of receive windows, ACK timeouts and ADR settings.



2.2.1.2.3.1

2.2.1.2.3.2 Fields

- **Tx Power** – Maximum transmit power of the gateway. If the selected channel plan limits by EIRP then the antenna gain will be subtracted from the power sent in the downlink.
- **Antenna Gain** – The amount of gain for the installed gateway antenna. May be subtracted from max tx power for certain channel plans that regulate EIRP.
- **Rx 1 DR Offset** – Offset to adjust the datarate of downlinks in the first Rx window from the default according to received uplink packet datarate. Normally the downlink will use a certain datarate for downlink, often the same datarate as the received uplink. If more range is desired for downlinks then this setting can be adjusted.
- **Rx 2 Datarate** – Datarate to be used for downlinks in the seconds Rx window
- **ADR Step** – SNR step between datarates assigned by ADR algorithm. In LoRa modulation the theoretical step of Rx sensitivity is 2.5 dBm between two spreading factors of the same bandwidth. Lower this setting will reduce the SNR threshold needed to reach the next datarate level when ADR is enabled. Raising this setting will require a greater SNR to reach higher datarates when ADR is enabled. Default step is 30 cBm/3.0 dBm.
- **Min Datarate** – Minimum datarate to use in ADR assignment of end-device datarate.
- **Max Datarate** – Maximum datarate to use in ADR assignment of end-device datarate.
- **ACK Timeout** – Time to wait for ACK from end-device before sending a repeat packet for Class C end-devices.

2.2.1.2.4 Transmit Power

The transmit power of gateways and end-devices is determined by regional regulations. This regulation vary in the manner of limitation such as EIRP, ERP or conducted power. These variations affect how the software applies the max power and antenna gain settings.

2.2.1.2.4.1 US and AU

US and AU regulations limit the transmit power based on conducted power limits.

The AT+TXP and Tx Power settings on the Dot and Conduit represent the conducted power of the end-

device.

If the power and antenna settings combine to exceed the regulated limit the software will reduce the radio power to maintain compliance based on the antenna gain setting.

FCC dictates a maximum conducted output for a transmitter in the ISM band is +30 dBm, up to +6 dBm antenna gain is allowed. If the total EIRP exceeds +36 dBm the conducted power must be reduced to stay below the maximum.

In the case of Conduit +27 dBm can be output with the MTAC-LORA-H card. Therefore a +9 dBm antenna may be used. If a +10 dBm antenna is installed and configured, the conducted power will be reduced to +26 dBm.

2.2.1.2.4.2 EU, AS, KR and IN

EU, AS, KR and IN regions limit the EIRP of the transmitter. The software must take account the antenna gain.

The AT+TXP and Tx Power settings on the Dot and Conduit represent the EIRP of the end-device, the antenna gain will be used to reduce the power of the radio to meet the regulated limit at the selected frequency. If the power and antenna settings combine to exceed the regulated limit the software will reduce the radio power to maintain compliance based on the antenna gain setting.

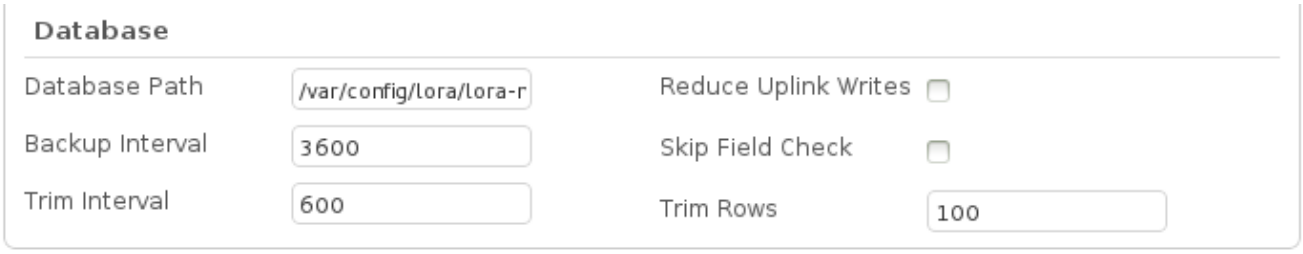
ETSI dictates maximum EIRP for a transmitter in the ISM band is +14 dBm for most of the band with exception of +27 dBm at 869.4-869.65 MHz.

In the case of Conduit +27 dBm can be output with the MTAC-LORA-H card. Therefore if +3 dBm antenna is installed and configured the conducted output power will be limited to +24 dBm when using the 869.525 MHz channel and +11 dBm at channels below 869.4 MHz or above 869.65 mHz.

AS923 channel plan can configure a MaxEIRP setting on Conduit to be sent to end-device to limit the radio output.

Operation in Japan should configure for +16 dBm EIRP for both end-devices and gateway. With a +3 dBm antennas installed the Tx Power should be set to +13 dBm and the MaxEIRP set to +16 dBm.

2.2.1.2.5 Database



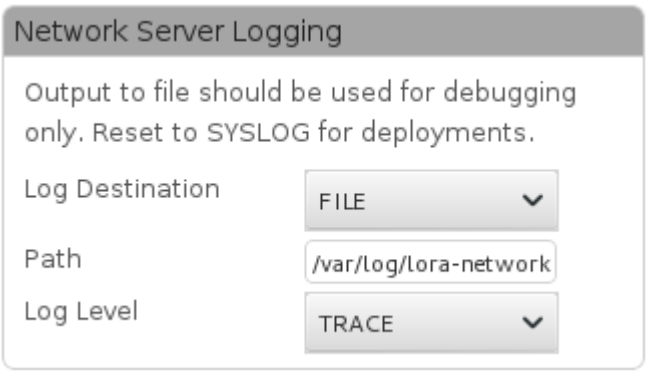
2.2.1.2.5.1 Fields

- **Database Path** – Location to store the network server database in non-volatile memory.

    - A copy of the database is held in RAM for normal operations, this database is backed-up to the database path. Only 8MB are available in /var/config for all Conduit configuration information. Changing this path can allow the database to be stored on an SD Card or a USB Flash drive. It is recommended to move the database out of /var/config if the number of connected LoRa end-devices exceeds 2000.

- **Backup Interval** – Number of seconds between database back-up to NVM. The database back-up will create a journal copy of the database in next to the file in the database path. Therefore double the expected database size is needed in the database path.
- **Trim Interval** – Number of seconds between operations to trim the packets tables. To keep the database size low, received packets are limited to a number of total rows.
- **Trim Rows** – Number of packet rows to keep in the database after trim interval.
- **Reduce Uplink Writes** – Reduce the number of writes to the database. Normally every uplink packet is written to the database to retain the frame counter. This operation can slow the receipt of uplinks. If many packets are expected from end-devices, i.e. 30/second, then this setting should be enabled to reduce system load of database writing. The received packets will be reported to the application to forward to remote server, but only one of one-hundred will saved to the database. If the Conduit is reset, the counter will fall behind the last sent from the end-device, the network server will recover but missed packets will be reported in statistics.
- **Skip Field Check** – Skip checking fields in packets received from packet forwarders. Similar to reduce uplink writes, this setting can help increase throughput of uplinks.

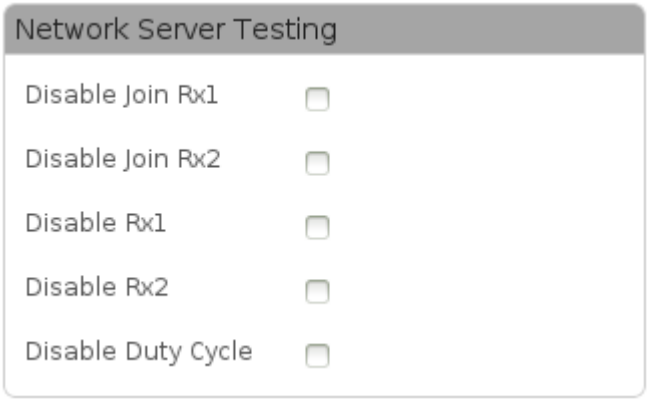
2.2.1.3 Network Server Logging



2.2.1.3.1.1 Fields

- **Log Destination** – Set logging to a file or to syslog
- **Path** – Set log file path if log to file is selected. /var/log/ is a directory in RAM and is not kept over reset.
- **Log Level** – Level of logging to send to the log file.

2.2.1.4 Network Server Testing



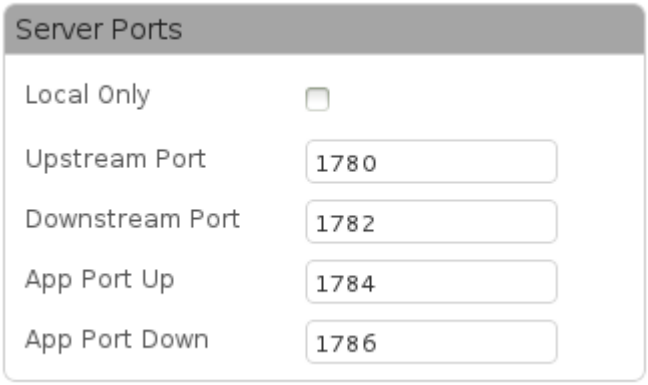
2.2.1.4.1.1 Fields

- **Disable Join Rx1** – Disable join accept downlinks sent in the first Rx window
- **Disable Join Rx2** – Disable join accept downlinks sent in the second Rx window
- **Disable Rx1** – Disable normal downlinks sent in the first Rx window, non-join packets.
- **Disable Rx2** – Disable normal downlinks sent in the second Rx window, non-join packets.
- **Disable Duty Cycle** – Disable the duty-cycle limitations for downlink packets, should be used only for testing purposes.

2.2.1.5 Advanced Settings

Advanced settings rarely need to be changed from defaults.

2.2.1.5.1 Server Ports



2.2.1.5.1.1 Fields

- **Local Only** – Open upstream and downstream port only on loop-back interface. If not checked then the ports will be opened on the external interfaces and connections. Inbound traffic must also be allowed through the firewall.

- **Upstream Port** – UDP port to accept uplink packets from packet forwarders.
- **Downstream Port** – UDP port to communicate downlink packet to packet forwarders.
- **App Port Up** – UDP port used to forward uplink packets to an application. An application can listen on a local UDP port for packets from the network server similar to the MQTT interface events.
- **App Port Down** – UDP port open to accept downlink packet from an application. An application can publish downlink packets to this port to be queued by the network server. Similar to the MQTT down topic.

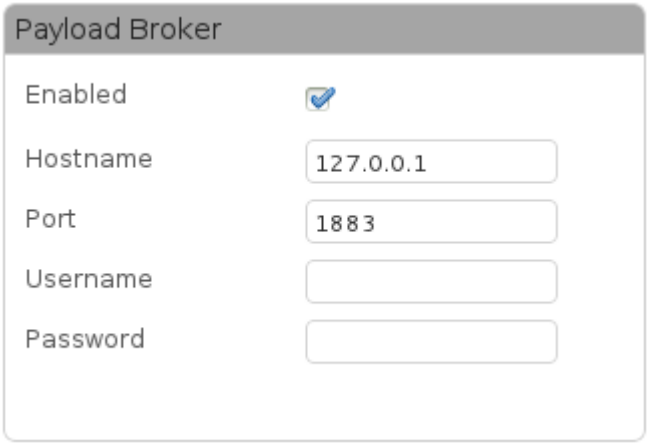
2.2.1.5.1.2 Firewall Settings

Be sure to enable connections through the firewall on the configured upstream and downstream ports when Local Only is disabled and remote gateways are expected to report to the network server.

2.2.1.5.2 Payload Broker

The settings for connecting the network server to an MQTT broker are provided.

It is recommended to leave the broker as localhost (127.0.0.1) and create a bridge application with node-red, node.js, python or c++ to forward desired topics to a remote MQTT broker rather than change this setting. That way an SSL connection can be used to the remote server without the overhead to the network server for local events.

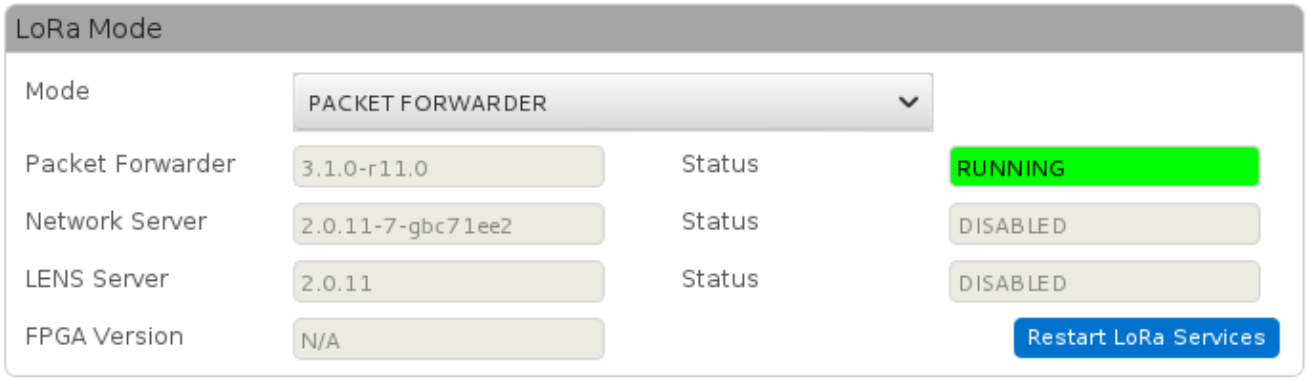


2.2.1.5.2.1 Fields

- **Enabled** – Enable or disable reporting network server events to MQTT
- **Hostname** – Post name of MQTT broker, localhost (127.0.0.1) by default.
- **Port** – Port of MQTT broker, 1883 by default.
- **Username** – Username used to connect to the MQTT broker.
- **Password** – Password used to connect to the MQTT broker.

2.2.1.6 Packet Forwarder Mode

Packet forwarder mode can be enabled to relay packets to and from a remote network server.



Packet forwarder protocol definition and source code is available on github.

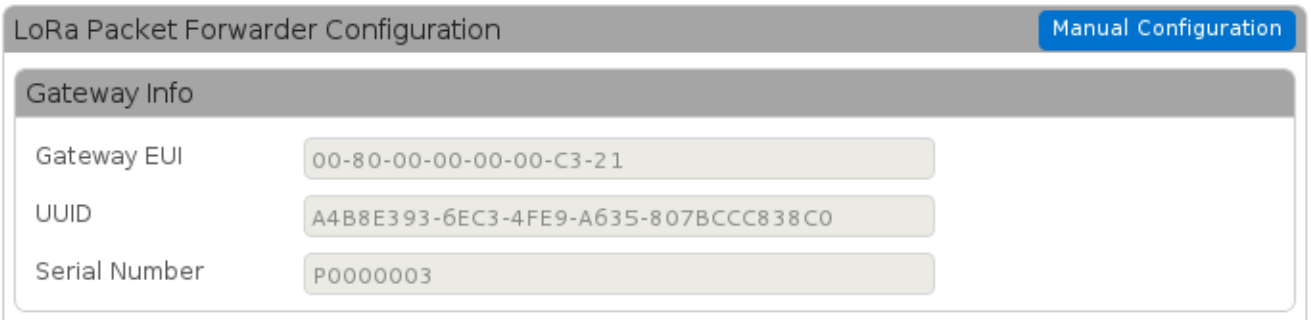
https://github.com/Lora-net/packet_forwarder/blob/master/PROTOCOL.TXT

Recipes and patches used to run the packet forwarder on Conduit can be found here.

http://git.multitech.net/cgi-bin/cgit.cgi/meta-mlinux.git/tree/recipes-connectivity/lora?h=3

2.2.1.6.1 Gateway Information

Gateway information is provided to register the Conduit in Lens. Gateways configured in packet forwarder mode reporting to another Conduit in network server mode must also be registered in Lens as packets received and gateway statistics will be reported with this Gateway EUI.



2.2.1.6.1.1 Fields

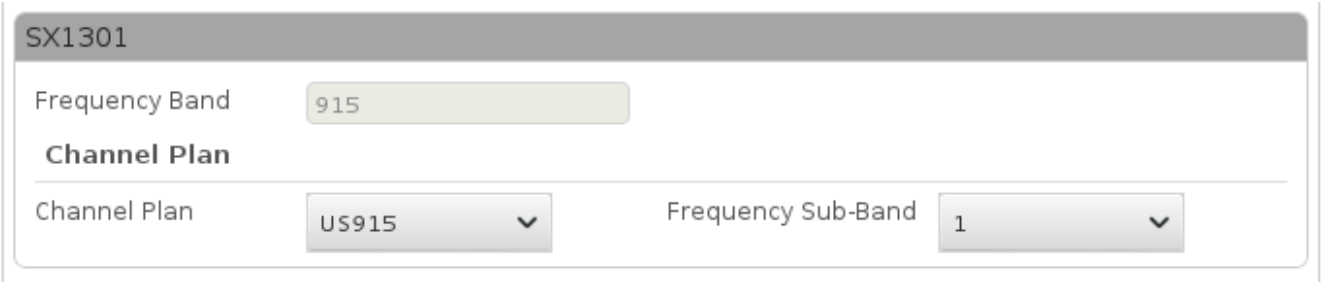
- **Gateway EUI** – Gateway identifier read from installed LoRa hardware.
- **UUID** – Unique gateway identifier used to authenticate with Lens.
- **Serial Number** – Serial number of the Conduit

2.2.1.6.2 Normal Configuration

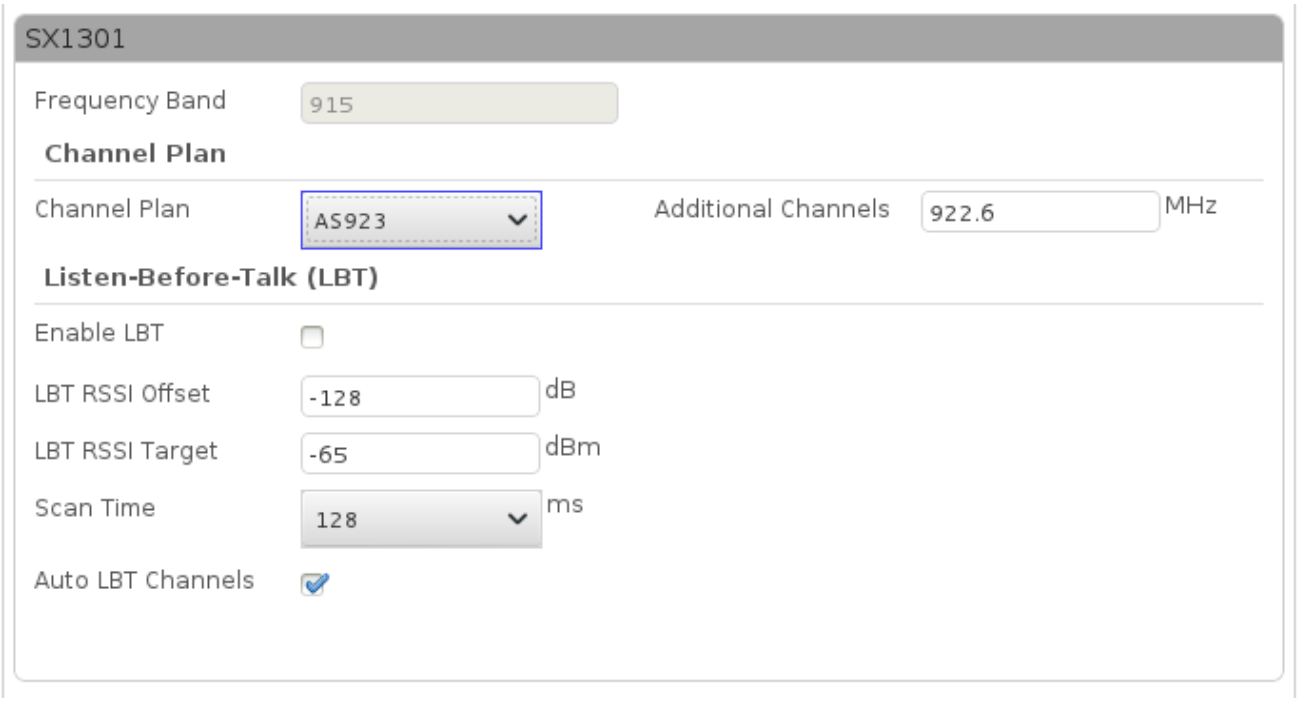
Choose settings to configure the gateway select channels to listen according to Channel Plans and enter a network server destination to forward packets to and receive downlinks from.

2.2.1.6.3 SX1301

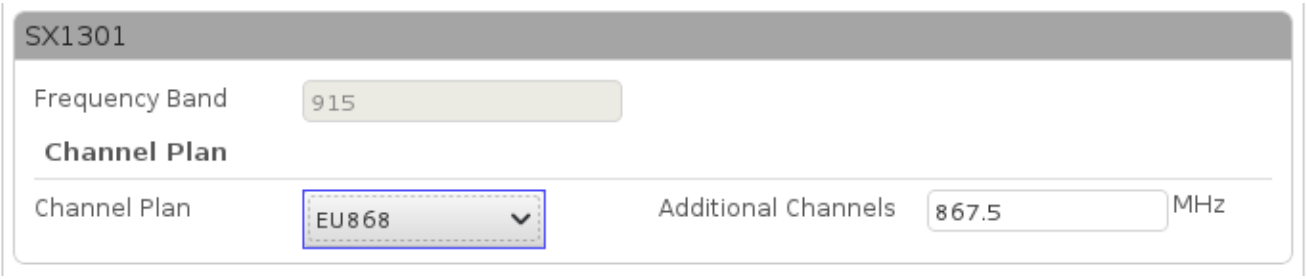
2.2.1.6.3.1 US915/AU915



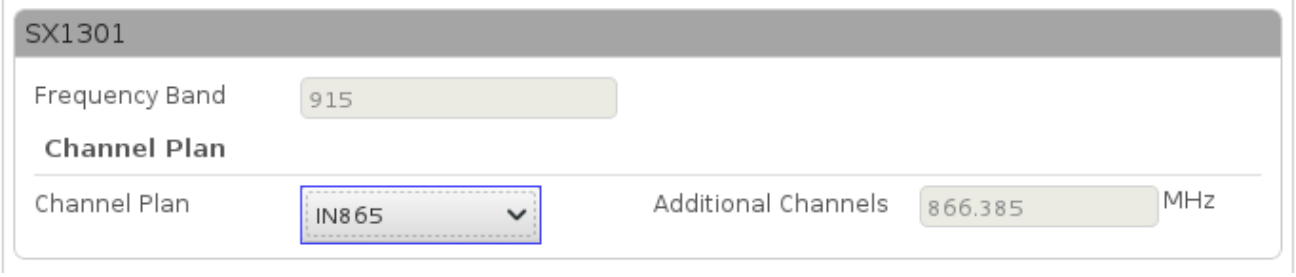
2.2.1.6.3.2 AS923/KR920



2.2.1.6.3.3 EU868



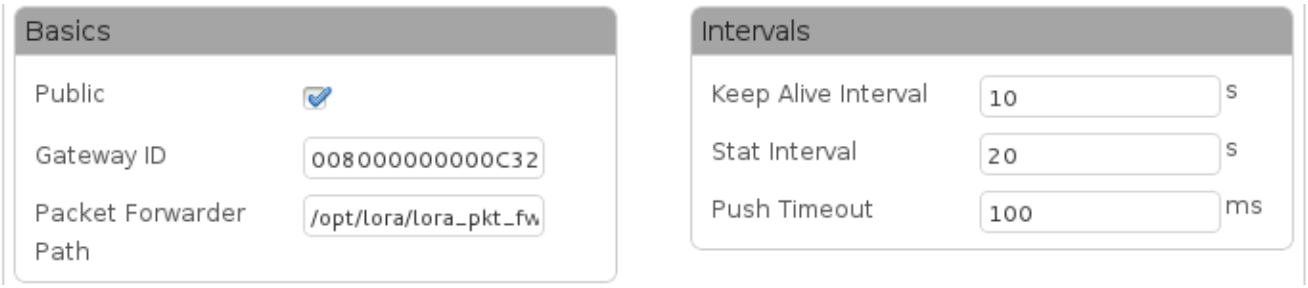
2.2.1.6.3.4 IN865



2.2.1.6.3.5 Fields

- **Frequency Sub-Band** – Choose the subset of frequencies to listen for uplinks. See network server Channel Plan Section 2.2.1.2.1 for more details.
- **Additional Channels** – Choose additional channels to listen for uplinks. See network server Channel Plan Section 2.2.1.2.1 for more details.
- **Enable LBT** – Enable listen before talk if supported by install LoRa hardware. If the hardware does not support this option, then the packet forwarder cannot be started with this setting enabled.
- **LBT RSSI Offset** – Listen before talk offset to use to adjust the RSSI from the radio.
- **LBT RSSI Target** – Listen before talk target the RSSI must be below in order to allow transmission on the selected channel
- **Scan Time** – Listen before talk amount of time in microseconds to record RSSI to determine if the channel has been used. Should be set according to regional regulations.
- **Auto LBT Channels** – Configure the Rx channels to be used for Tx of downlinks. If unchecked a set of channels can be manually configured to be allowed for downlinks. All downlinks must use a configured channel when LBT is enabled. Any downlink attempting to use any other frequency will be discarded.

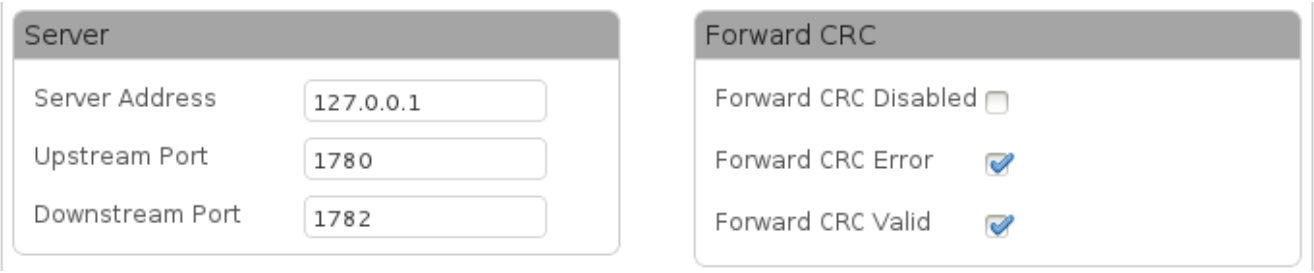
2.2.1.6.4 Basics and Intervals



2.2.1.6.4.1 Fields

- **Public** – Configure gateway for public or private network mode. Configures the LoRa sync word.
- **Gateway ID** – Gateway identifier to report to the network server.
- **Packet Forwarder Path** – Path to the packet forwarder binary to be used
- **Keep Alive Interval** – Interval to send keep alive packets to the network server
- **Stat Interval** – Interval to send gateway stats to the network server
- **Push Timeout** – Timeout for packets to the network server

2.2.1.6.5 Server and Forward CRC



2.2.1.6.5.1 Fields

- **Server Address** – IP Address or hostname of Network Server
- **Upstream Port** – IP UDP Port of Network Server for uplink packets
- **Downstream Port** – IP UDP Port of Network Server for downlink packets
- **Forward CRC Disabled** – Default unchecked, LoRaWAN requires CRC to be enabled for uplink packets.
- **Forward CRC Error** – Default checked, packets with CRC errors will be rejected by the network server without being processed. This can be disabled to save back-haul network usage without affecting performance. Some random false packets are reported by the gateway hardware due to noise. The random nature causes the length of the reported packets to range between 0-255 bytes in length.
- **Forward CRC Valid** – Default checked, forward packets that have passed the radio CRC

2.2.1.6.6 Manual Configuration

2.2.1.6.6.1 Fields

- **Gateway EUI** – Gateway identifier read from installed LoRa hardware.
- **UUID** – Unique gateway identifier used to authenticate with Lens.
- **Serial Number** – Serial number of the Conduit
- **Config** – Enter JSON configuration for Packet Forwarder
  - Examples – http://www.multitech.net/developer/software/lora/aep-lora-packet-forwarder/
  - Notes – Comments are not allowed, JSON format can be checked for errors or minimized using on-line tools such as https://codebeautify.org/jsonviewer

2.2.2 Key Management

Local and cloud join server settings.

Unique end-device AppKeys can be configured.

Local Join Server Network ID and Network Key have been moved from the network settings page.

2.2.2.1 Join Server

Two join server locations are available **Cloud Key Store** or **Local Keys**.

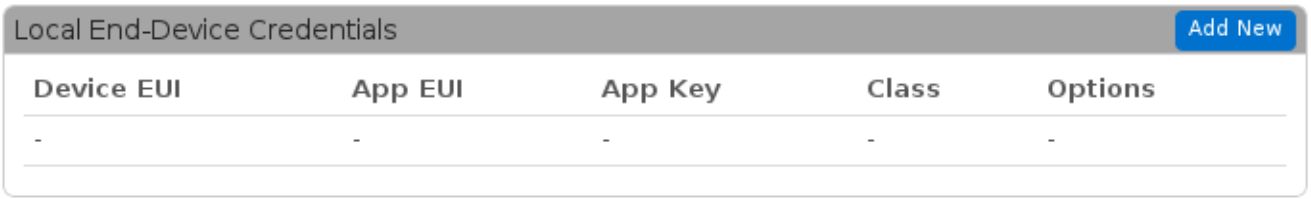
A cloud key store can be used to keep the end-device AppKeys in a remote and secure location. Only session keys for end-devices will be kept on the gateway to maintain network integrity. Any end-device in the key store can be configured to connect to a Conduit without changing configuration of the end-device or Conduit using Multitech's EnterpriseHQ Lens interface.

Local keys can be used in stand-alone deployments where Internet access to a cloud key store is not available. AppKeys must be installed on each Conduit for each end-device expected to connected to it.

2.2.2.1.1.1 Fields

- **Location** – Choose **Cloud Key Store** or **Local Keys**
  - Cloud Key Store – Requires an Internet connection and an account for Multitech's EnterpriseHQ Lens
  - Local Keys – End-device DevEUI/AppKey pairs can be stored on the gateway

2.2.3 Local End-Device Credentials

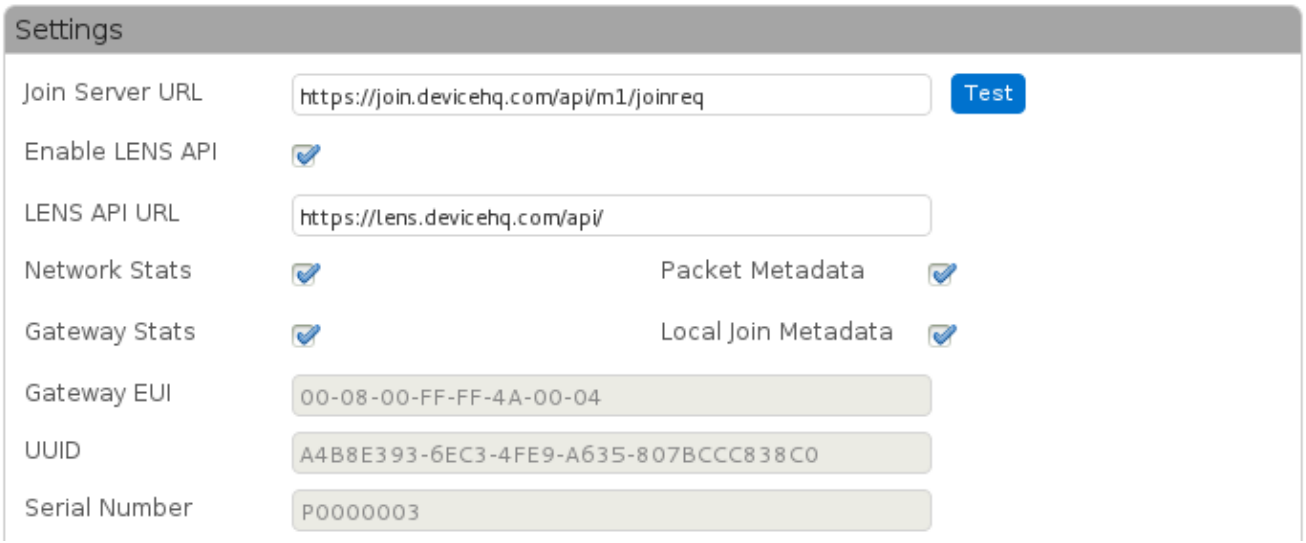


2.2.3.1.1.1 Fields

- **Dev EUI** – End device identifier, 8-byte hexadecimal string. Used to identify the end-device during OTAA join and look-up the AppKey. Uplink packets received from this end-device will have the Dev EUI attached to events from the Network Server.

- **App EUI** – Application identifier, 8-byte hexadecimal string. Uplink packets received from this end-device will have the App EUI attached to events from the Network Server.
- **App Key** – Secret pre-shared key used to authenticate the end-device during OTAA join.
- **Class** – Default operating class for the end-device, select A or C.
  - Class A – Downlink packets only possible in Rx windows following an Uplink
  - Class C – Downlink packets can be sent any time, end-device is listening when idle

2.2.3.2 Settings



2.2.3.2.1.1 Fields

- **Join Server URL** – Configure server to forward received join requests that cannot be handled by the local join server settings. URL could point to a multi-tenet or private server
- **Enable Lens API** – Send statistic and metadata to the Lens cloud service
- **Lens API URL** – Configure Lens server API to a multi-tenet or private server
- **Network Stats** – Send aggregated network statistics to the Lens cloud.
- **Gateway Stats** – Send aggregated gateway statistics to the Lens cloud.
- **Packet Metadata** – Send information about each packet to the Lens cloud. DevEUI, Frequency, datarate, RSSI, SNR, type, timestamp, header, MAC commands and size. Basically everything but the payload is captured for analysis.
- **Local Join Metadata** – Send OTAA join information about locally joined end-devices to the Lens cloud.
- **Gateway EUI** – Displays EUI of the installed MTAC lora card

- If an MTAC card is not installed an EUI will be created from the Conduit MAC address. This EUI can be registered in Lens to accept packet details, network statistics and gateway statistics from the network server.

- **UUID** – Unique identifier of Conduit used to authenticate requests through the Lens machine API, must be registered with Lens
- **Serial Number** – Serial number of the Conduit to provide to Lens

2.2.3.2.2 Local Network Settings

2.2.3.2.2.1 Fields

- **Enabled** – Enable or disable OTAA joins using the NetworkId/NetworkKey settings. Use of these settings can allow a device to test connectivity to a gateway without use of Internet.
- **Network ID** – Name or EUI, if Name is selected an EUI will be created using the Name field.
- **Name** – EUI or Name, EUI value must match the AppEUI field in the Join Request and the request must be signed using the Network Key
- **Network Key** – Passphrase or EUI, if Passphrase is selected a Key will be created using the Passphrase field.
- **Passphrase** – Key or Passphrase, Key value must match the AppKey used by the end-device to sign the OTAA Join Request. A Join Accept message will be returned encrypted with this key.
  - Sharing AppKeys among end-devices has a few quirks. If multiple end-devices attempt to join a the same time, they may all receive the same Join Accept message if the Rx window settings match. The end-device will think it is joined but has invalid session keys. Testing the session after join with a few confirmed packets is advised. If a downlink is not received the end-device should re-join the server. If the end-devices happened to pick the same DevNonce for the Join Request, then the session keys will actually be valid on two end-

devices.

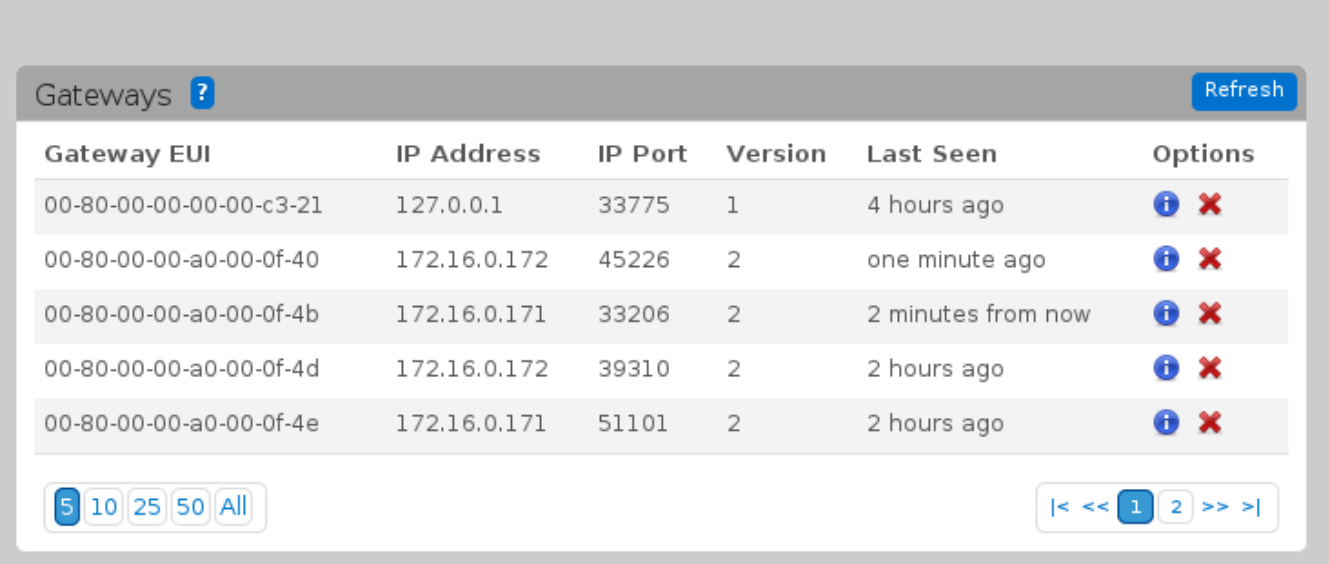
2.2.4 Gateways

Statistics from connected gateways and overall network statistics.

Received packet counts and available duty-cycle time-on-air is shown for each gateway.

Global network statistics for response time for join requests and various packet counts.

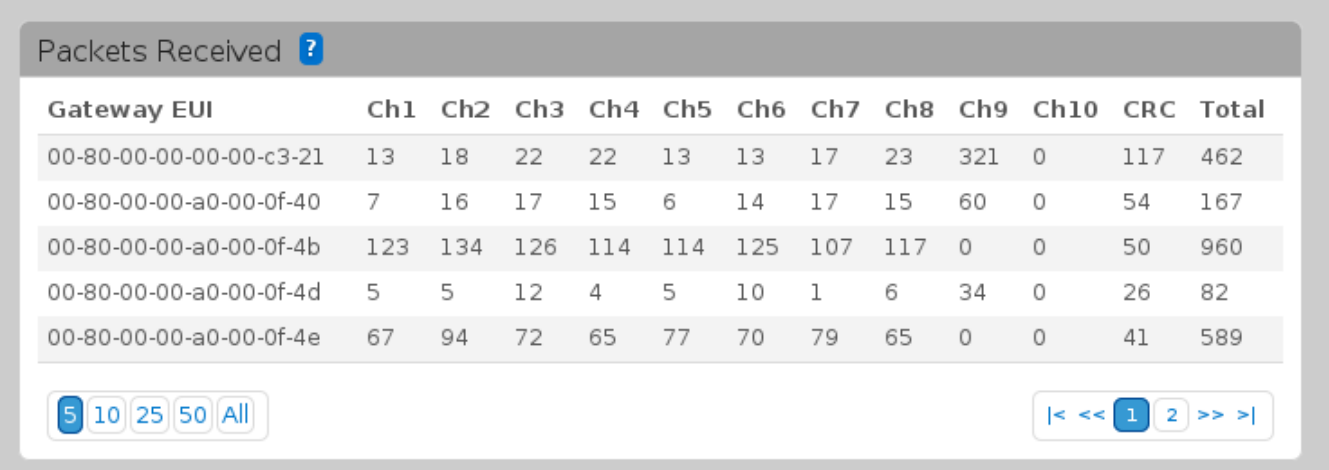
2.2.4.1 Gateways



2.2.4.1.1 Columns

- **Gateway EUI** – Gateway identifier received in UDP packet header
- **IP Address** – Address UDP packet was received
- **IP Port** – Port UDP packet was received on, used for return packets
- **Version** – Version of packet forwarder protocol from end-device
  - Version 1 packet forwarders will not send gateway statistics to the network server
    - USB MTAC cards are only supported with the version 1 packet forwarder. Suggest updating to SPI cards.
- **Last Seen** – Time the gateway was last seen or start of Network Server process.

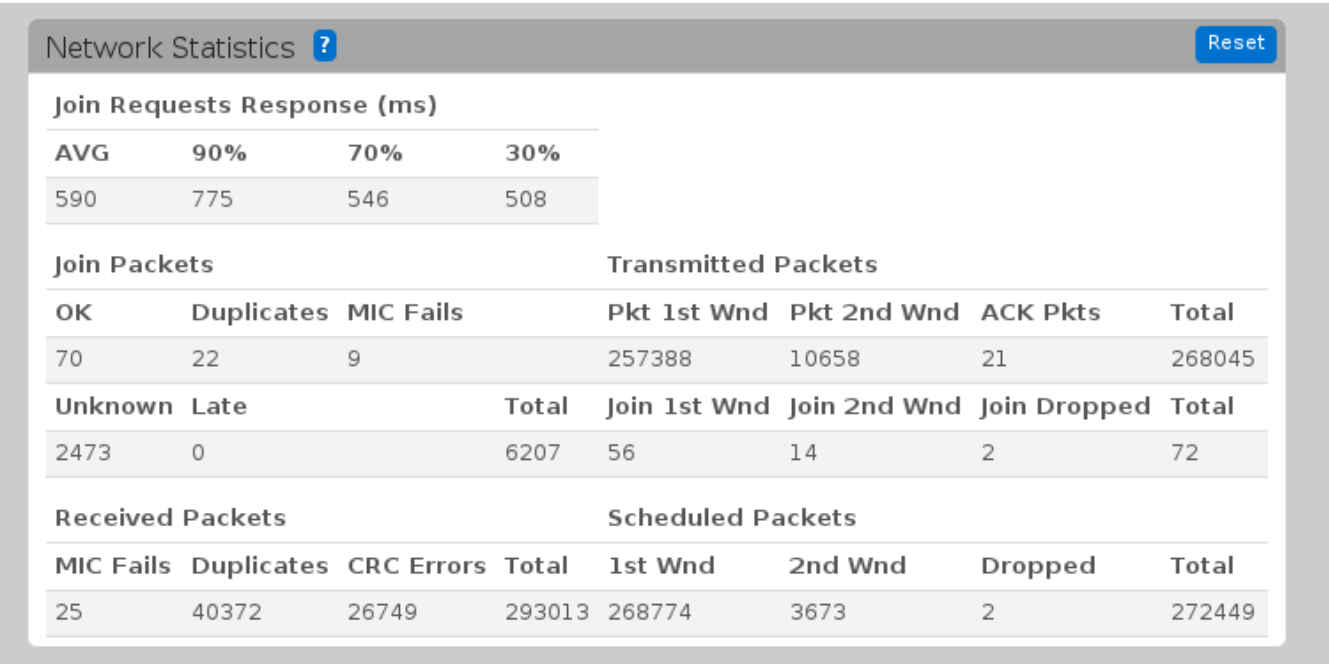
2.2.4.2 Packets Received



2.2.4.2.1 Columns

- **Gateway EUI** – Gateway identifier
- **Ch1-Ch10** – Number of packets received for each channel
- **CRC** – Total number of packet with CRC error received on all channels
- **Total** – Total number of packets received on all channels, CRC errors included.

2.2.4.3 Network Statistics



2.2.4.3.1 Fields

- **Join Request Response (ms)** – Latency from gateway to Join Server for the last 5 minutes
  - AVG – average of all join requests forwarder to Join Server in milliseconds
  - 90% - 90th percentile of latency to Join Server in milliseconds
  - 70% - 70th percentile of latency to Join Server in milliseconds
  - 30% - 30th percentile of latency to Join Server in milliseconds
- **Join Packets** – Counts of Join Request results
  - OK – Join Request was accepted by the Join Server
  - Duplicates – Join Request contained a duplicate Nonce value or received on multiple gateways
  - MIC Fails – Join Request failed to be authenticated using the AppKey
  - Unknown – End-device DevEUI was not found in Join Server end-device list
  - Late – Join Request response from the Join Server was too late to be transmitted in the Rx windows
  - Total – Total Join Requests packets received
- **Transmitted Packets** – Packets sent to packet forwarder to be transmitted
  - Pkt 1st Wnd – Packets sent for first Rx window
  - Pkt 2nd Wnd – Packets sent for second Rx window
  - ACK Pkts – Confirmed packets sent requesting ACK from end-device in next uplink
  - Total – Total packets sent in Rx1 and Rx2
  - Join 1st Wnd – Join Accept packets sent in first Rx window
  - Join 2nd Wnd – Join Accept packets sent in second Rx window
  - Join Dropped – Join Accept packets that could not be sent
  - Total – Total Join Packets sent in Rx1 and Rx2
- **Received Packets** – Counts for packets received
  - MIC Fails – Count of packets that failed MIC verification with DevAddr in packet header used to look-up the session keys
  - Duplicates – Duplicate packets received from multiple gateways or retransmitted packets from the end-device
  - CRC Errors – Packets received with CRC error, packet content cannot be trusted to provide a valid Dev Addr, MIC or payload

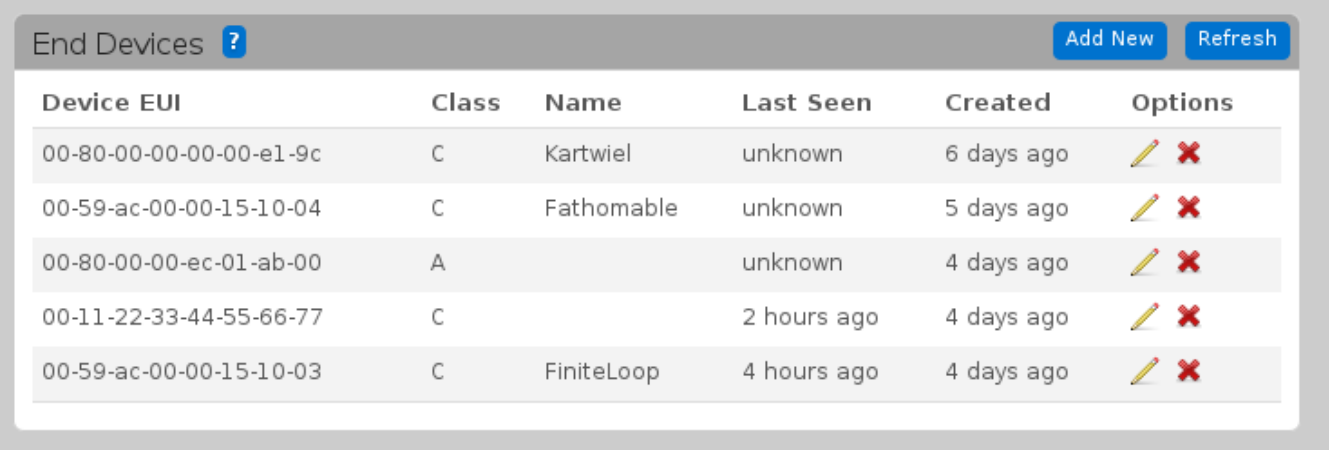
  - Total – Total packets received
- **Scheduled Packets** – Packets scheduled in each receive window
  - 1st Wnd – Packets scheduled for first Rx window
  - 2nd Wnd – Packets scheduled for second Rx window
  - Dropped – Packet that could not be scheduled for Rx1 or Rx2 due to conflicts with other scheduled packets.
  - Total – Total packets scheduled

2.2.5 Device Configuration

Device configuration allows configuration of end-device operation class A or C.

Information about hardware and firmware can be stored or updated from cloud join server.

2.2.5.1 End Devices



2.2.5.1.1.1

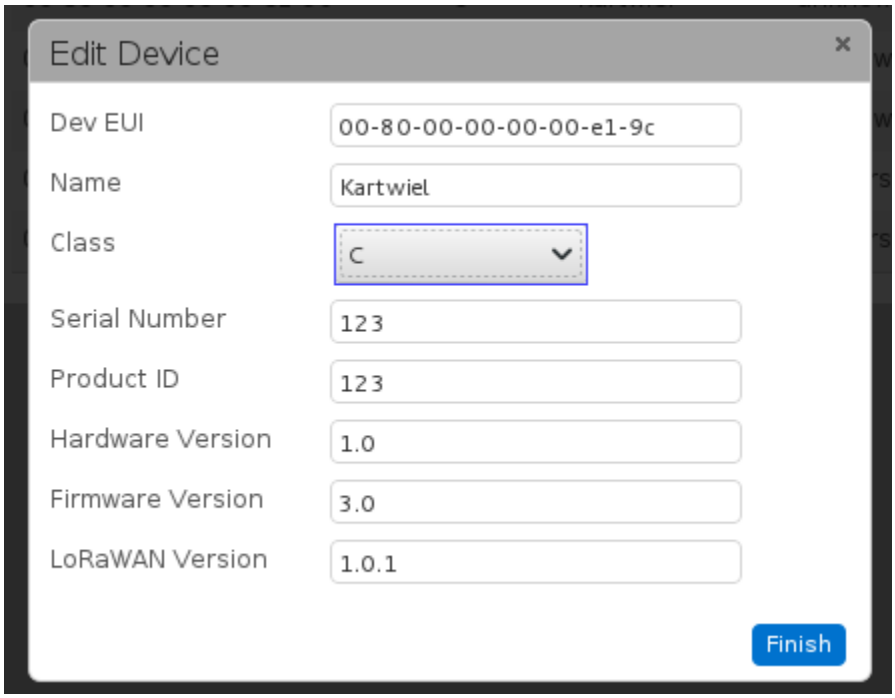
2.2.5.1.2 Columns

- **Device EUI** – End-device identifier
- **Class** – Configured operating class of end-device, settings on end-device must match and are configured out-of-band. There is no message in LoRaWAN 1.0.2 for an end-device to declare it is Class C. If the device is Joined via Cloud Join Server this information will be sent down with Join Accept message. The setting from the Cloud Join Server will override this local setting.
- **Name** – Friendly name given end-device. If the device is Joined via Cloud Join Server this information will be sent down with Join Accept message.
- **Last Seen** – Time of last uplink packet received.
- **Created** – Time end-device record was created. Time of first OTAA Join or manually

configured.

2.2.5.2 Edit End-device

These settings can be configured in the Lens cloud and sent to the Conduit from the Join Server with the Join Accept packet. Settings from the Join Server will override the local settings. If the settings are changed on Conduit a message will be sent to Lens to update the information in the cloud. All values but DevEUI, Serial Number and Product ID can be changed remotely.



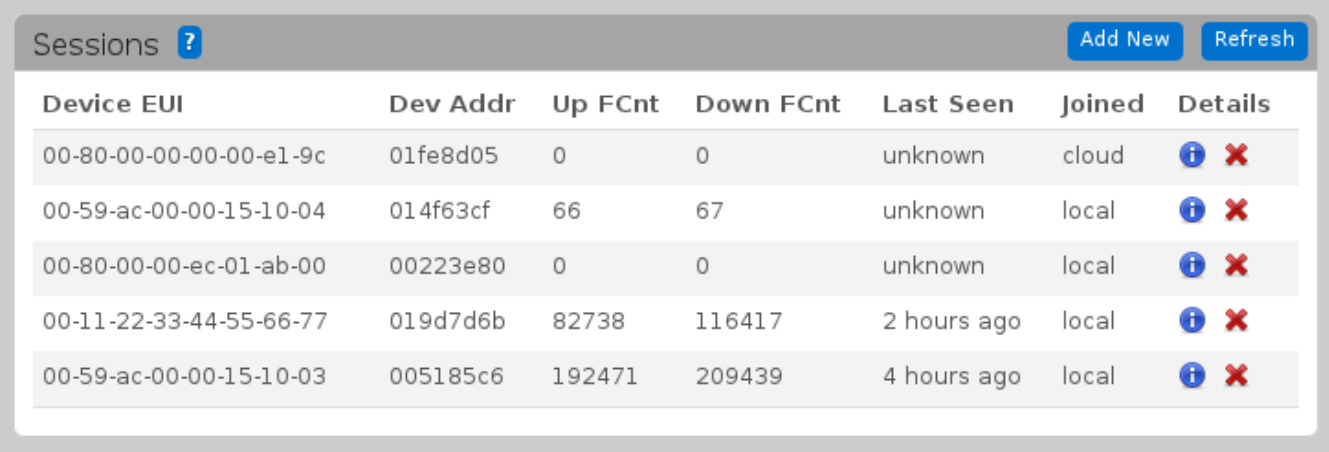
2.2.5.2.1 Fields

- **Dev EUI** – End-device identifier, 8-byte hexadecimal string
- **Name** – Friendly name of end-device
- **Class** – Operating class of end-device, must match end-device settings
- **Serial Number** – Serial number of end-devie
- **Product ID** – Product id of end-device
- **Hardware Version** – Hardware version of end-device
- **Firmware Version** – Firware version of end-device
- **LoRaWAN Version** – LoRaWAN version of end-device

2.2.6 Device Sessions

Session information for joined devices can be seen on this page. Session keys and counters are available.

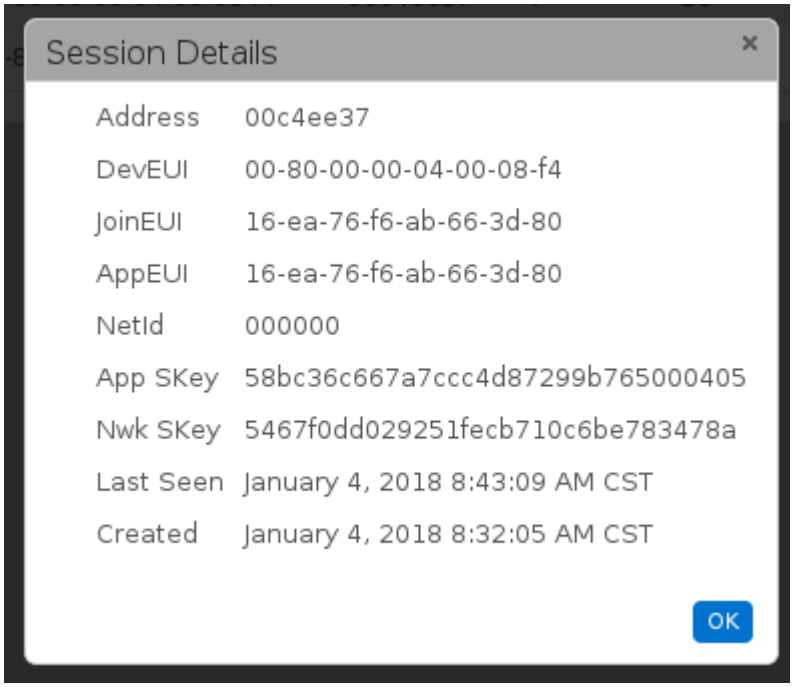
2.2.6.1 Sessions



2.2.6.1.1 Columns

- **Device EUI** – End-device identifier, 8-byte hexadecimal string
- **Dev Addr** – Address assigned by network server to be included in header of all LoRaWAN uplinks and downlinks.
- **Up FCnt** – Session counter for uplinks, any packets received with FCnt at or below this value will not be forwarded to the application. Each packet with an FCnt value will be sent to the application only once.
- **Down FCnt** – Session counter for downlinks, the end-device will reject packets at or below this value.
- **Last Seen** – Time of last uplink received from end-device
- **Joined** – Set to cloud or local to signify the Join Server used to authenticate the Join Request and author the Join Accept packet.
- **Details** – View details or delete end-device session, if the session is deleted the end-device will not be able to communicate to the Conduit until an OTAA Join. The end-device will not be notified of this action and must detect the loss of connectivity and attempt an OTAA Join.

2.2.6.1.2 Details

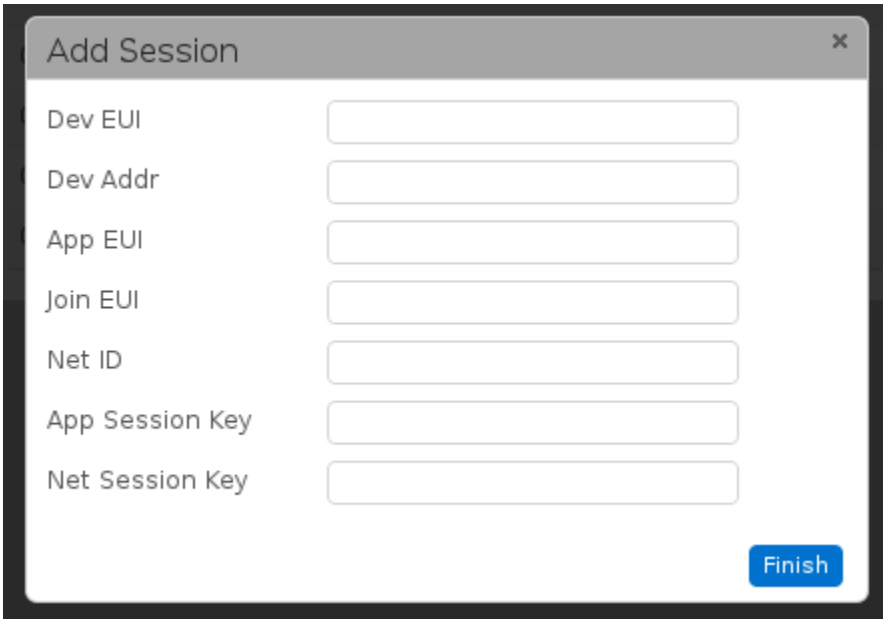


2.2.6.1.2.1 Fields

- **Address** – Dev Address assigned to the end-device session, used in the packet header to look-up session keys.
- **DevEUI** – End-device identifier
- **JoinEUI** – EUI presented in the OTAA Join Request
- **AppEUI** – EUI assigned by the Lens cloud server, identifies the joined Application Network
- **NetId** – NetID setting of Conduit network server
- **App SKey** – Application Session key for encrypting packet payloads
- **Nwk SKey** – Network Session key for generating MIC bytes for packet authentication
- **Last Seen** – Time of last uplink
- **Created** – Time session was created

2.2.6.2 Add Session

Session settings must match end-device configuration for communication to be possible.



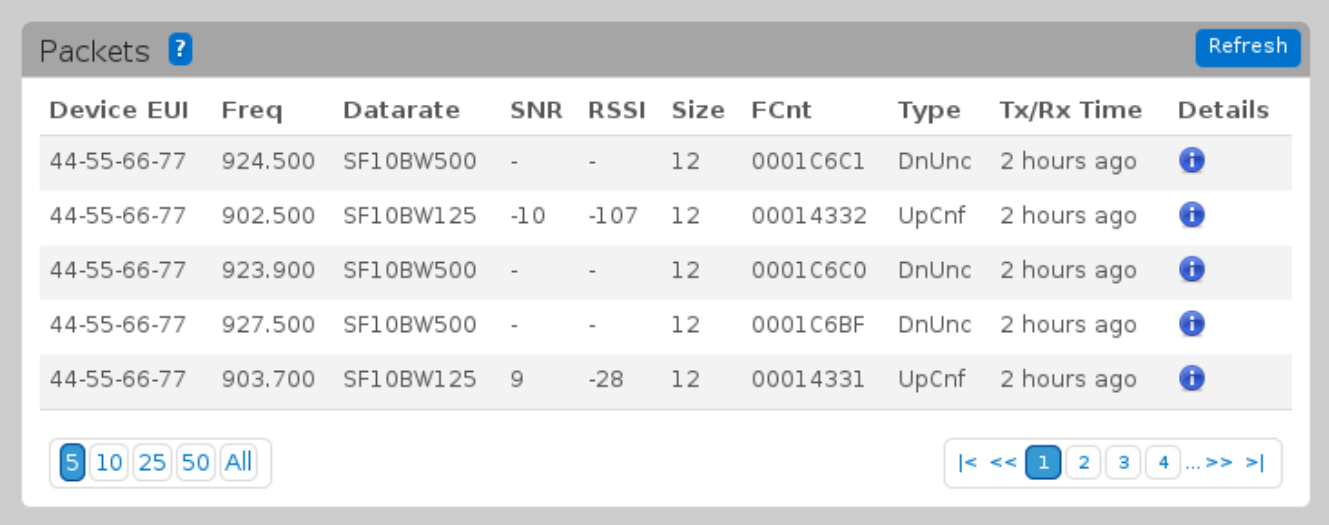
2.2.6.2.1 Fields

- **Dev EUI** – End-device identifier
- **Dev Addr** – Dev Address assigned to the end-device session, used in the packet header to look-up session keys.
- **App EUI** – EUI assigned by the Lens cloud server, identifies the joined Application Network
- **Join EUI** – EUI presented in the OTAA Join Request
- **Net ID** – NetID setting of Conduit network server
- **App Session Key** – Application Session key for encrypting packet payloads
- **Net Session Key** – Network Session key for generating MIC bytes for packet authentication

2.2.7 Packets

Received and sent end-device packets as well as recent received packets and join requests.

2.2.7.1 Packets

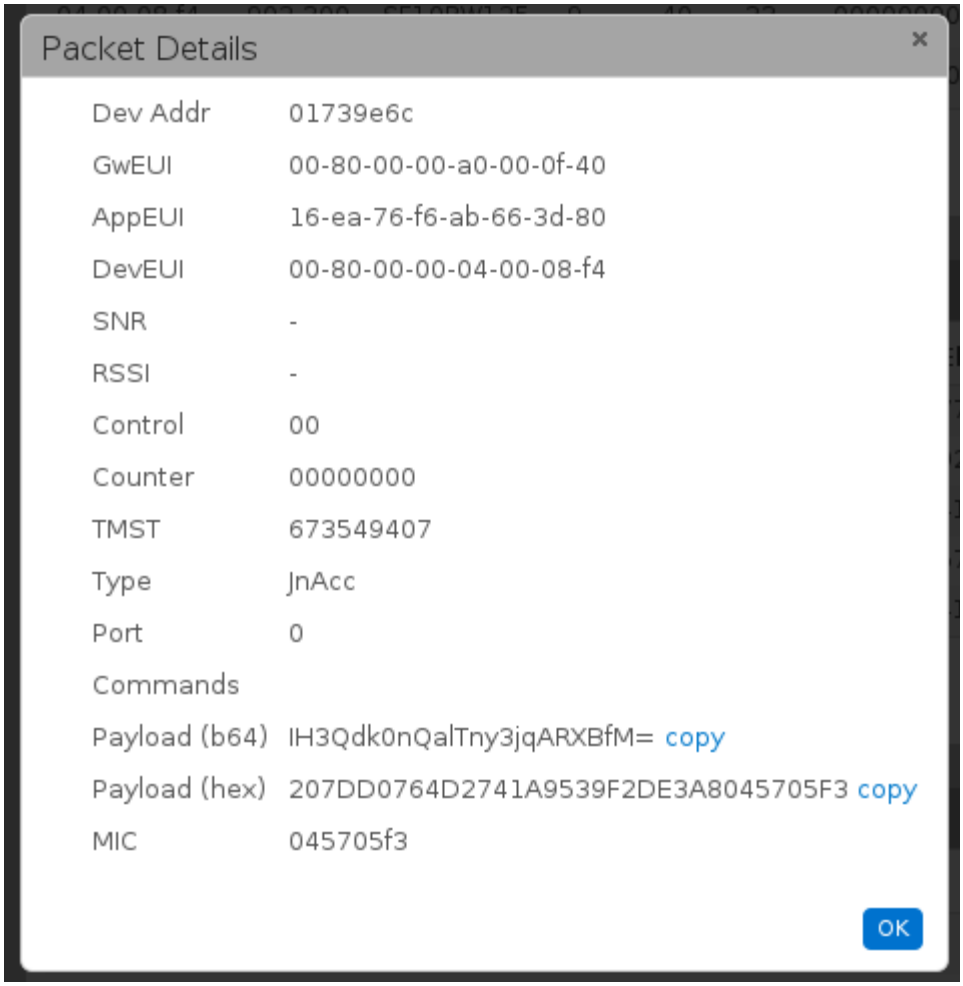


2.2.7.1.1 Columns

- **Device EUI** – End-device identifier, only 4 least-significant-bytes are shown. Full EUI is available on hover over.
- **Freq** – Frequency the packet was received or sent in MHz
- **Datarate** – Datarate used to transmit packet
- **SNR** – Signal to noise ratio of received packet -20 to 20 dB, a negative SNR notes packet was received below the noise floor. SNR is the best indicator of LoRa link quality.
- **RSSI** – Received signal strength during packet reception. Includes noise and signal strength. See SNR for the best indicator of LoRa link quality.
- **Size** – Size in bytes of packet, includes header, MAC commands, payload and MIC bytes
- **FCnt** – Uplink or downlink counter in used to authenticate the packet MIC and decrypt packet payload.
- **Type** – Type of packet
  - DnUnc – Downlink unconfirmed packet
  - DnCnf – Downlink confirmed packet, request ACK from end-device in next uplink packet
  - UpUnc – Uplink unconfirmed packet
  - UpCnf – Uplink confirmed packet, request ACK from server in next downlink packet
  - JnReq – Join Request uplink packet
  - JnAcc – Join Accept downlink packet

- **Tx/Rx Time**- Time packet was sent or received
- **Details** – Veiw details of packet

2.2.7.1.2 Details

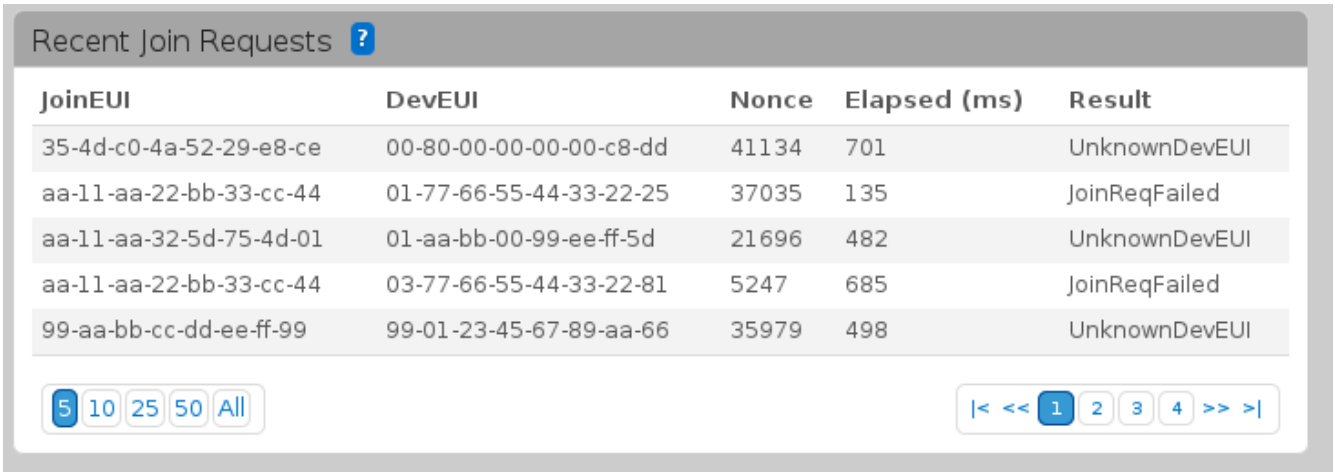


2.2.7.1.2.1 Fields

- **Dev Addr** – End-device session address present in the packet header
- **GwEUI** – Gateway identifier the packet was received from
- **AppEUI** – Application EUI assigned to session and reported with packet by network server
- **DevEUI** – Device identifier associated with Dev Addr record and authenticated using the session key
- **SNR** – Signal to noise ratio of received packet -20 to 20 dB, a negative SNR notes packet was received below the noise floor. SNR is the best indicator of LoRa link quality.

- **RSSI** – Received signal strength during packet reception. Includes noise and signal strength. See SNR for the best indicator of LoRa link quality.
- **Control** – Control byte in the LoRaWAN header, indicates ACK, ADR, Class B and length of MAC command optional bytes field
- **Counter** – Counter value used to authenticate the packet, 32-bit counter maintained by end-device and network server, only 16-bits are sent in the packet header
- **TMST** – Clock timestamp for the gateway radio to synchronize the downlink transmission with the Rx windows opened on the end-device
- **Type** – Type of packet indicated by first byte of packet
- **Port** – App Port if present in the packet.
- **Commands** – MAC commands bytes in hex if present in the packet
- **Payload (b64)** – LoRaWAN packet payload bytes in base64
- **Payload (hex)** – LoRaWAN packet payload bytes in hexadecimal
- **MIC** – message integrity check bytes of packet computed and authenticated using the network session key

2.2.7.2 Recent Join Requests



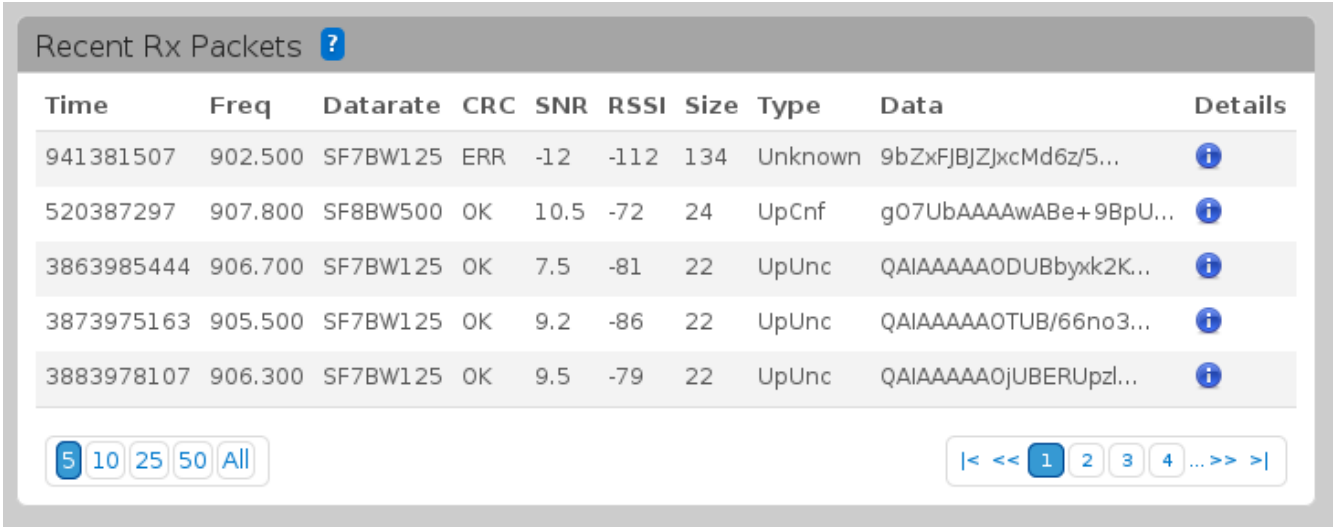
2.2.7.2.1 Columns

- **JoinEUI** – EUI from Join Request bytes 1-8
- **DevEUI** – EUI from Join Request bytes 9-16
- **Nonce** – Nonce value in Join Request bytes 17-18
- **Elapsed (ms)** – Latency time to service Join Request at local or cloud Join Server, if latency

exceeds the Join Delay setting by 750 ms, then the packet was too late to be sent in either Rx window

- **Result** – Success or failure result from Join Server
  - **MICFailed –** AppKey setting did not match the end-device record in Join Server
  - **Dropped –** Downlink packet could not be scheduled for transmit on any available gateways
  - **Duplicate Dev Nonce –** Nonce in join request has already been used
  - **JoinReq Failed –** Other server error
  - **UnknownDevEUI –** Device record was not found at Join Server
  - **Gateway Mismatch –** Join Server configuration does not allow this device to join through this gateway
  - **Server Error –** Join Server is not reachable possibly due to Internet connection settings or DNS resolution, or an error occurred at the server

2.2.7.3 Recent Rx Packets

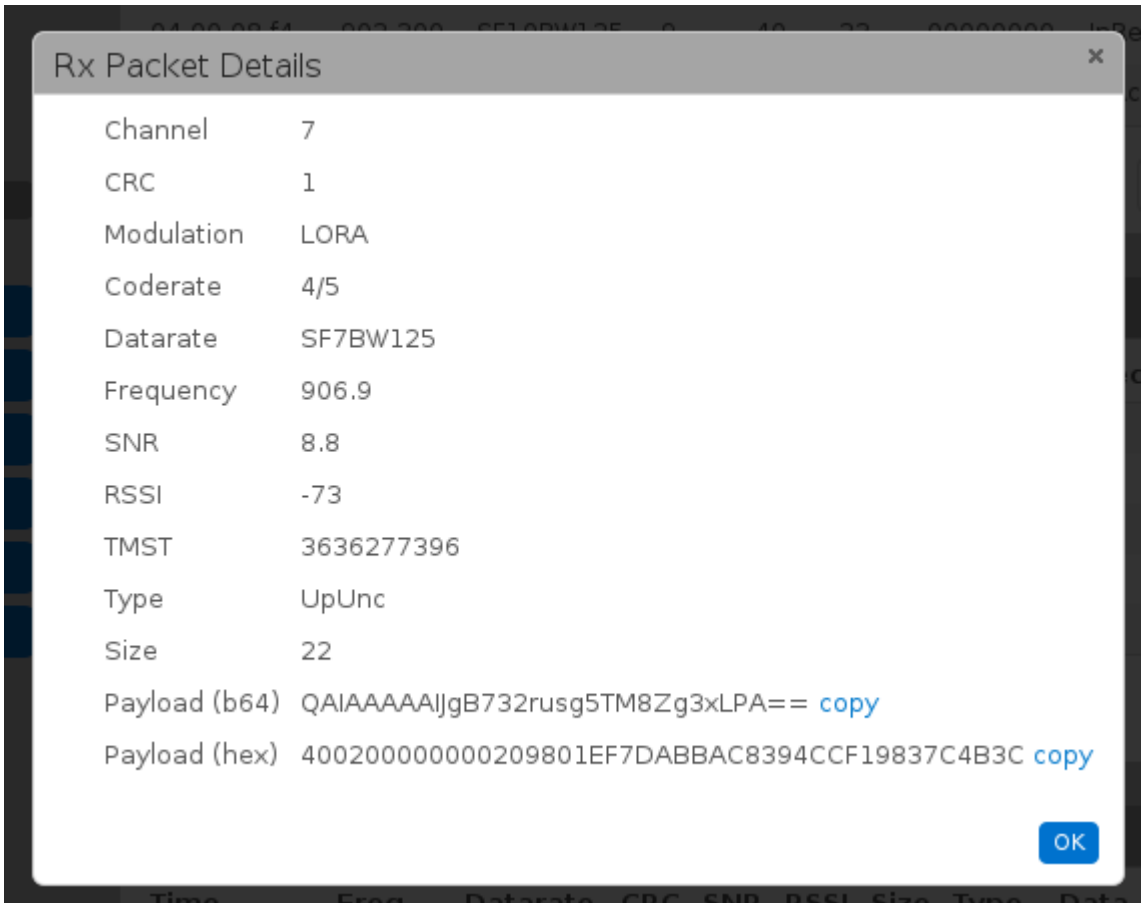


2.2.7.3.1 Columns

- **Time** – Internal clock timestamp from the gateway hardware
- **Freq** – Frequency the packet was received
- **Datarate** – Datarate the packet was received
- **CRC** – ERR or OK if packet passed the cyclic-redundancy-check, packets that fail the CRC filter may be caused by environmental noise. Because packets can be received below the noise floor some false positives with low SNR values may be randomly received.

- **SNR** – Signal to noise ratio of received packet -20 to 20 dB, a negative SNR notes packet was received below the noise floor. SNR is the best indicator of LoRa link quality.
- **RSSI** – Received signal strength during packet reception. Includes noise and signal strength. See SNR for the best indicator of LoRa link quality.
- **Size** – Size in bytes of the received packet
- **Type** – Type of packet received if discernible from first byte of packet.
- **Data** – Packet bytes received in Base64
- **Details** – View packet details

2.2.7.3.2 Details



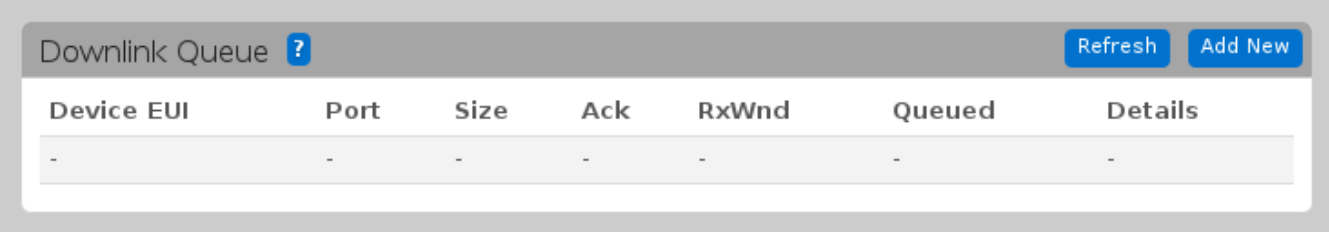
2.2.7.3.2.1 Fields

- **Channel** – Gateway channel the packet was received on, a single gateway can listen on 10 channels. 8 channels listen for 125 Khz packets using SF12-SF7 spreading factor. Two additional channels can be configured for FSK and a LoRa packets at a fixed bandwidth (125, 250 or 500 KHz) and fixed spreading factor.
- **CRC** – Value one or negative one to indicate passing the CRC filter

- Modulation – Packet modulation LORA or FSK
- Coderate – Forward error correction indicated in the packet, LoRaWAN uses only 4/5 FEC.
- Datarate – Datarate, spreading factor and bandwidth of the received packet
- Frequency – Frequency in MHz of the received packet
- **SNR** – Signal to noise ratio of received packet -20 to 20 dB, a negative SNR notes packet was received below the noise floor. SNR is the best indicator of LoRa link quality.
- **RSSI** – Received signal strength during packet reception. Includes noise and signal strength. See SNR for the best indicator of LoRa link quality.
- **TMST** – Clock timestamp for the gateway radio to synchronize the downlink transmission with the Rx windows opened on the end-device
- **Type** – Type of packet indicated by first byte of packet
- **Size** – Size in bytes of the received packet
- **Payload (b64)** – Full packet bytes in base64
- **Payload (hex)** – Full packet bytes in hexadecimal

2.2.8 Downlink Queue

Packets queued and waiting for downlink to end-devices can be viewed, added or removed.



2.2.8.1 Columns

- **Device EUI** – End-device identifier
- **Port** – App Port to send in LoRaWAN port field
- **Size** – Size in bytes of packet
- **Ack** – Number of retries for confirmed packet or 0 for unconfirmed packet
- **RxWnd** – Rx window the packet should be sent, 0, 1 or 2. 0 is used for first available window or Class C window.
- **Queued** – Time packet was queued
- **Details** – View details or remove the packet from the queue

2.2.8.2 Add Downlink Queue Item

2.2.8.2.1 Fields

- **Dev EUI** – Destination end-device identifier to receive the packet
- **App Port** – Port to use in LoRaWAN packet header, applications should use 1-223. 224 and above are reserved for special utilities. Port 0 is used for MAC commands in the payload.
  - The network server will send a Port 0 packet to the end-device, but it does NOT read the MAC commands in the payload and apply the changes to the local end-device state.
- **Data Format** – Choose Hex or Base64 data to input into Data field
- **Data** – Payload to add to the downlink, format must match the Data Format setting
- **Ack Attempts** – Number of times to retry packet downlink and receive ACK from end-device. After attempts are exhausted the downlink will be removed from the queue and a dropped event will be relayed to the application.
- **Rx Window** – Choose Rx window to transmit packet in 1 or 2 for Class A end-devices. Leave as 0 for first available window or for Class C end-devices.

3 mLinux

The update network server has an updated command interface and lora-query utility to request statistics and device information.

3.1 New lora-query commands

A few examples of new commands are given in the next subsections.

3.1.1 To view all commands

lora-query -x help

```
MTS Lora Server Command Help

Commands:
stats - list current stats
  reset - reset stats for network, gateways and end-devices
gateway - gateway commands
  list - list connected gateways
     format: gateway list [json]
  delete - remove a gateway from the list
     format: device gateway <GW-EUI>
device - end-device commands
  add - add a new end-device record
     format: device add <DEV-JSON>
     example: device add '{"deveui":"00-80-00-00-00-00-e1-9c","class":"C"}'
  stats - show end-device statistics
  update - update end-device configuration or session info
     format: device update <DEV-EUI> <FIELD> <VALUE>
     example: device update 00-80-00-00-00-00-e1-9c class C
     fields: class, nskey, dskey, ulc, dlc
     format: device update <DEV-JSON>
     example: device update '{"deveui":"00-80-00-00-00-00-e1-9c","class":"C"}'
     fields: class, name, serial_number, product_id, hardware_version, firmware_version, lorawan_version
  delete - delete an end-device configuration, session and packet records
     format: device delete <DEV-EUI>
  config - show configuration for a specific device
  reset - reset end-device session counters
     format: device reset <DEV-EUI>
  list - list end-devices configured in the network server
     format: device list [json | json file <path>]
     example: device list json
     example: device list json file /tmp/devices.json
  keygen - generate a unique end-device key using zero-touch settings
     format: device keygen <DEV-EUI> [APP-EUI]
session - session commands
  add - add a session for a device
     format: session add <DEV-JSON>
     example: session add '{"deveui":"00-80-00-00-00-00-e1-9c","dev_addr":"00112233","appeui":"00-88-88-88-00-00-e1-9c","joineui":"00-99-99-99-00-
00-e1-9c","net_id":"000017","app
_senc_key":"531bd9c5ec5d8ba5ef3b262cebfb3e66","fnwk_sint_key":"531bd9c5ec5d8ba5ef3b262cebfb3e66"}'
     fields: deveui, appeui, joineui, dev_addr, net_id, app_senc_key, fnwk_sint_key
  delete - remove a device session
     format: session delete <DEV-EUI>
  reset - reset session counters
     format: session reset <DEV-EUI>
  list - show current device sessions
     format: session list [json | json file <path> ]
     example: session list jsonvi /
     example: session list json file /tmp/sessions.json
packet - packet commands
  join - list all validated join packets
     format: packet join [json]
  up - list all validated uplink packets
     format: packet up [json]
  down - list all downlink packets
     format: packet down [json]
  list - list all packets: join, up and down
     format: packet list [json]
```

```
  queue - list downlink queue packets to be sent to end-device
      format: packet queue [json]
    add - add a packet to the downlink queue
      format: packet queue add <PACKET-JSON>
      fields: deveui, data, ack, ack_retries, rx_wnd
    delete - delete all downlinks for a specific device
      format: packet queue delete <DEV-EUI>
    delete - delete one downlink for a specific device
      format: packet queue delete <DEV-EUI> <ID>
database - database commands
  backup - backup database to flash memory
config - show network server configuration
debug - change debug level
ping - ping the network server command port
help - display this help
quit - command network server process to stop

- add 'json' modifier to request output in json
```

3.1.2 View end-devices list

lora-query -x devices

lora-query -x devices list json

3.1.3 View sessions list

lora-query -x sessions

lora-query -x sessions list json

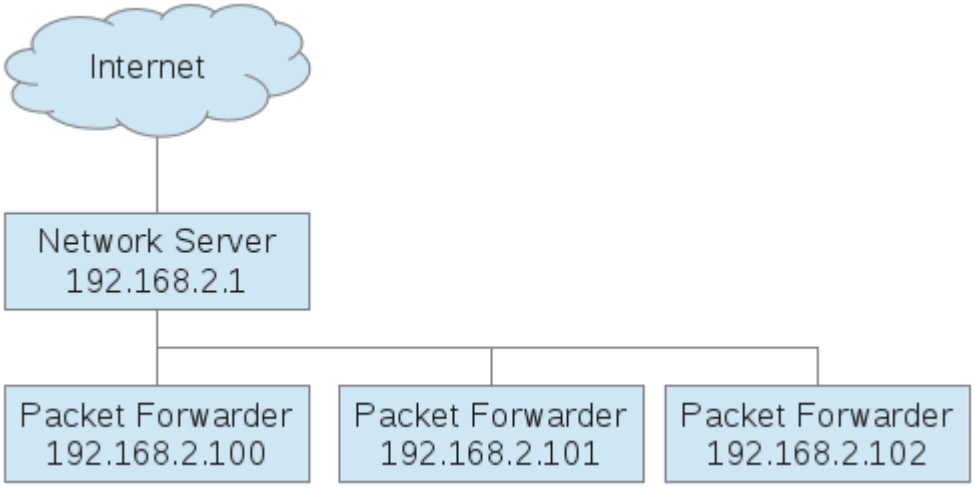
4 Multiple Gateway Deployments

A network of Conduits can be created with several setup as packet forwarder and one as a central network server.

This configuration can be used to increase the area of the network or the number of channels supported. The capacity of the network will still be limited to a single instance of the network server, approx 2000 end-devices in default configuration.

The /var/config directory is limited to 8 MB. Custom applications may also be installed in the /var/config directory reducing the space available for the database.

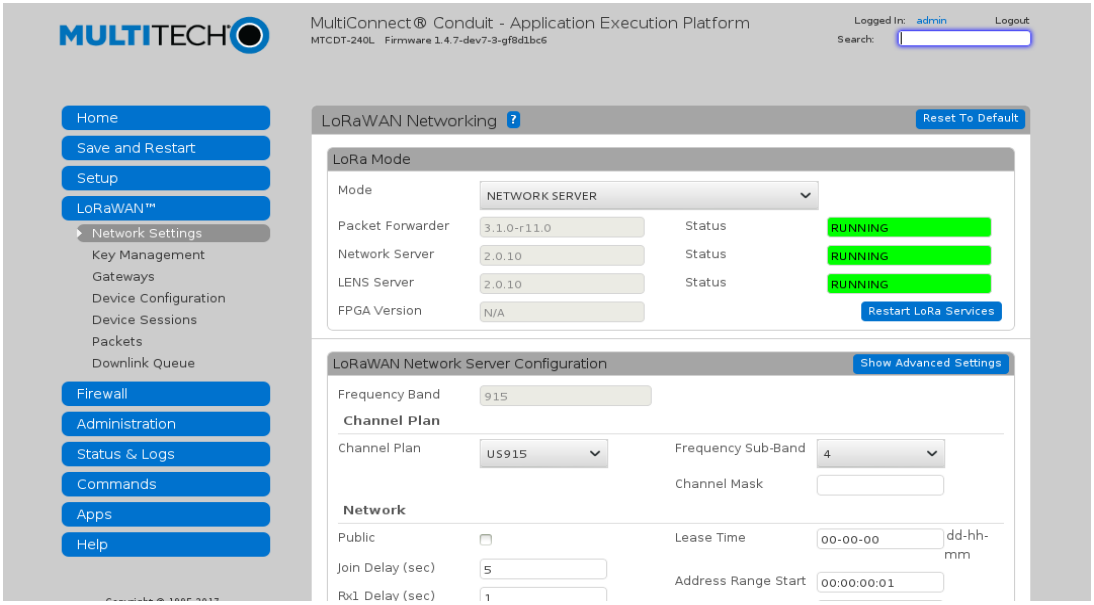
It is possible to increase the supported devices by installing an SD card and moving the custom application and network server database to it.



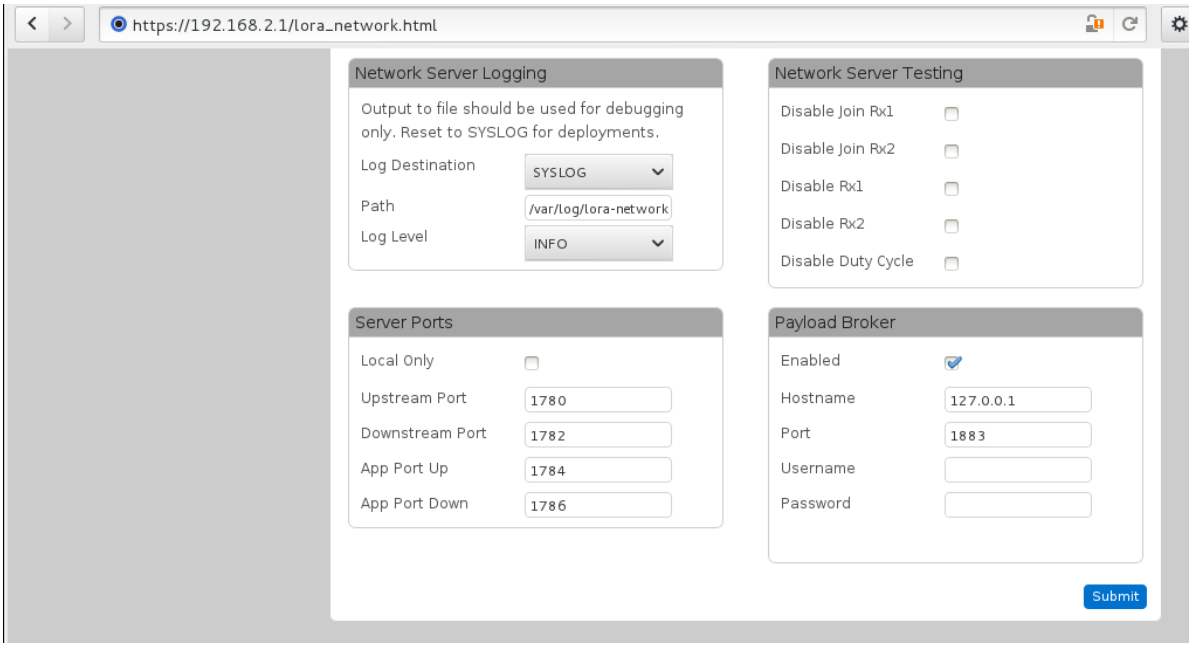
4.1 On Network Server Conduit

The central network server will handle all end-device session information, authenticate uplinks and author downlink packets.

4.1.1 Configure Network Server to accept connections from remote packet forwarders.

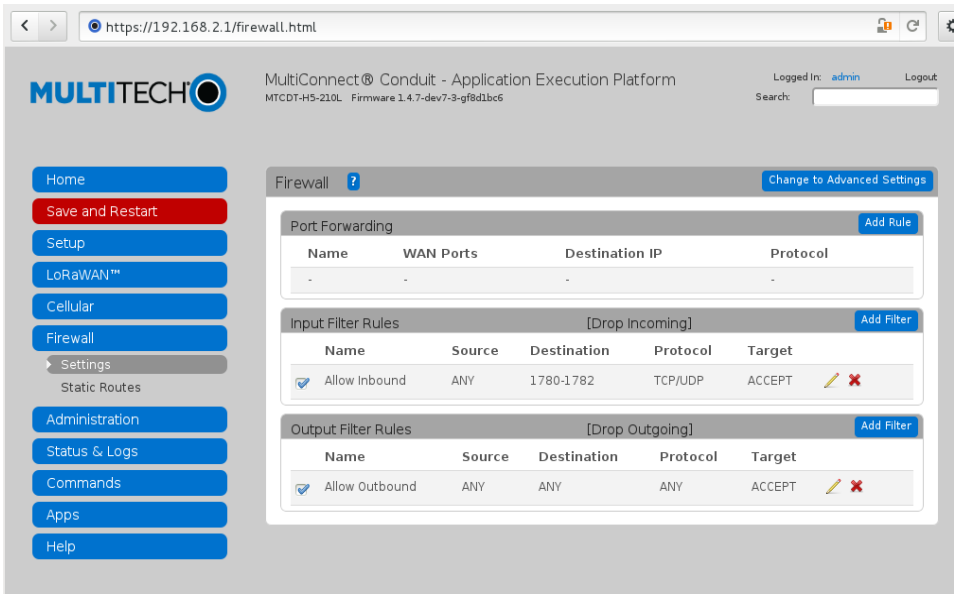


1. Go to **LoRaWAN > Network Settings** on Conduit
2. Click **Show Advanced Settings**
3. Under **Server Ports** verify **Local Only** is unchecked to allow incoming connections from the packet forwarder Conduits.

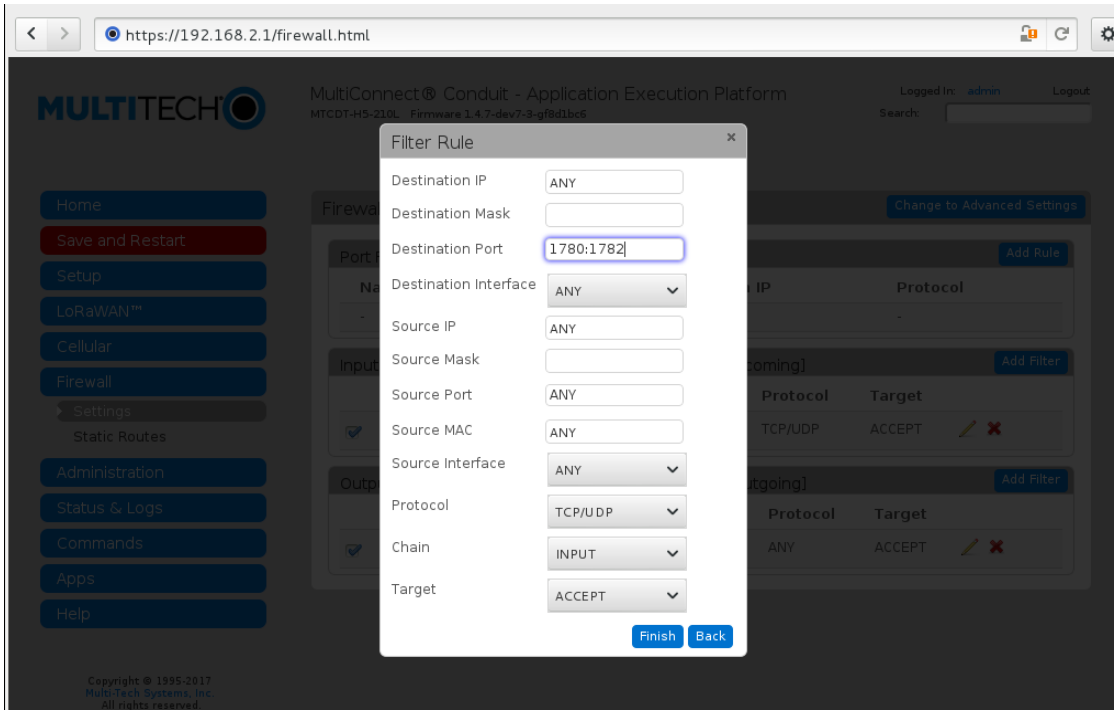


4. Click Submit

5. Go to **Firewall > Settings**



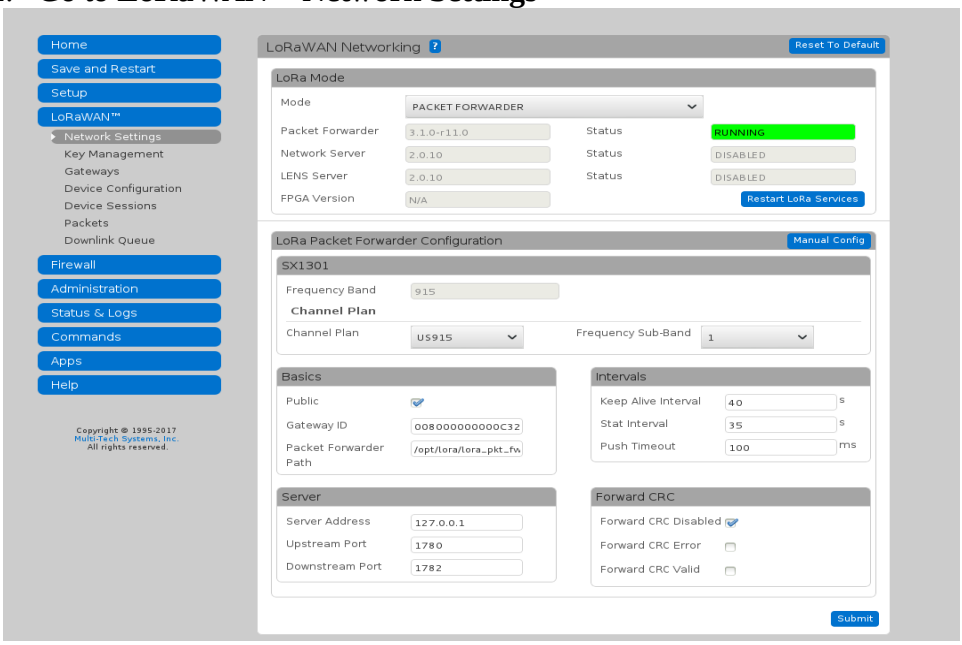
6. Enable Allow Inbound Input Filter Rule, change allowed ports to 1780 and 1782



7. Save and Restart Conduit

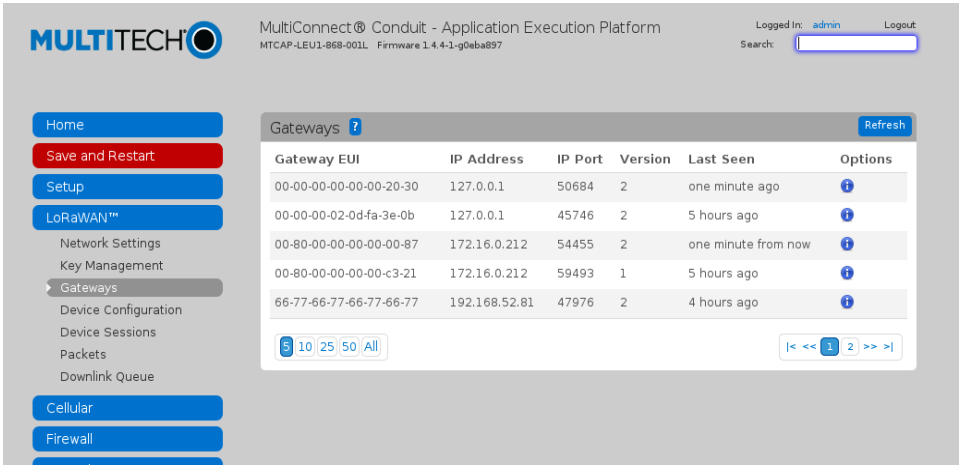
4.1.2 On a Forwarding Conduit

1. Go to **LoRaWAN** > **Network Settings**



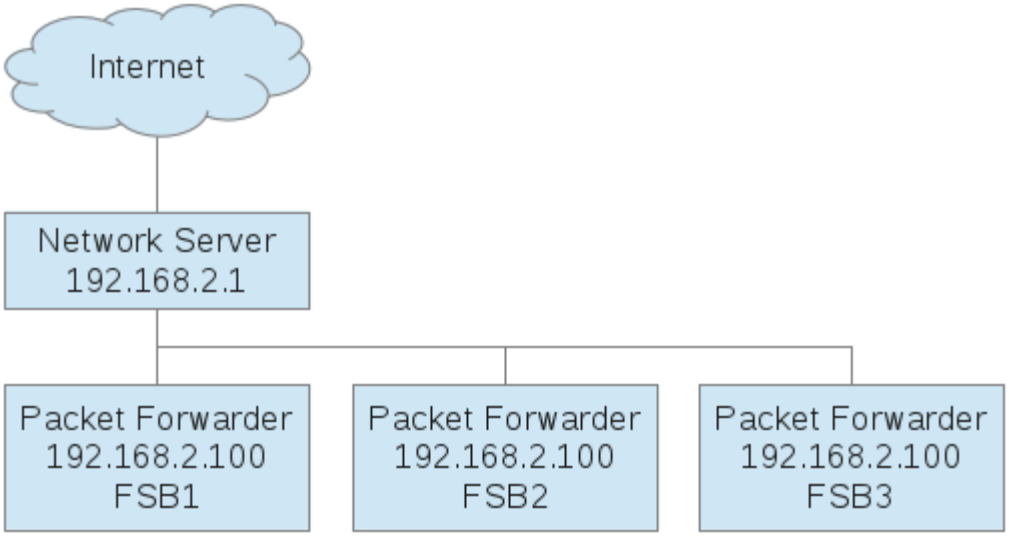
2. Enable Packet Forwarder mode
3. Set Public under Basics section
4. In Server section, *settings must match those of Master Conduit*
    1. Set Server Address to IP address
    2. Set Upstream Port to 1780
    3. Set Downstream Port to 1782
5. Click Submit

2. Back on First Conduit see that Gateway shows in list



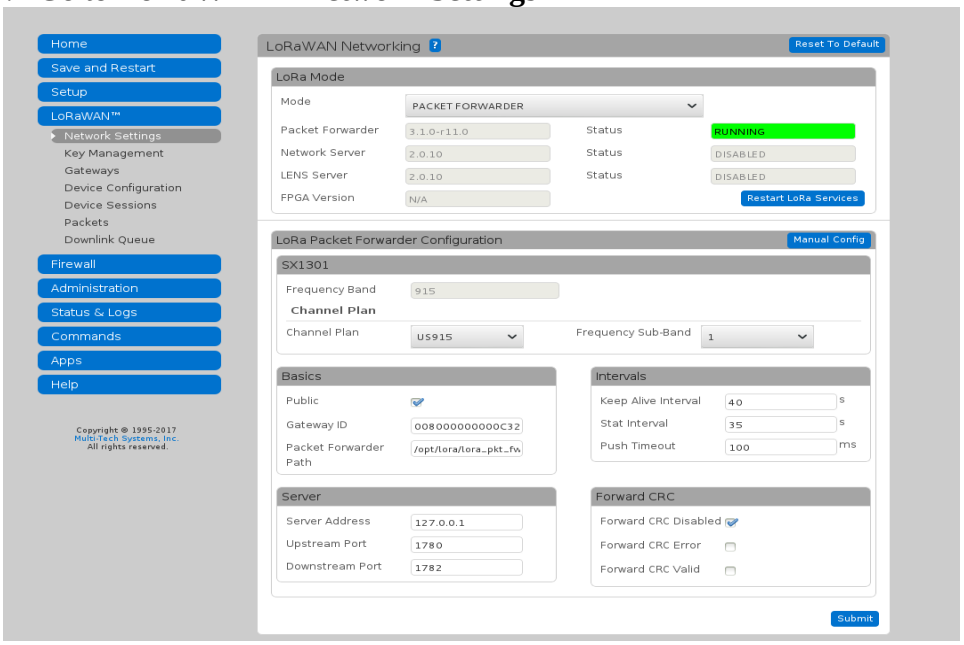
4.1.3 Extending Supported Channels

Additional channels can be supported by the network by enabled different Frequency Sub-Band settings on each forwarding Gateway. The supported channels must be relayed to the Network Server using the Channel Mask setting. This will allow the full set of channels to be enabled on the end-device following that OTAA join.



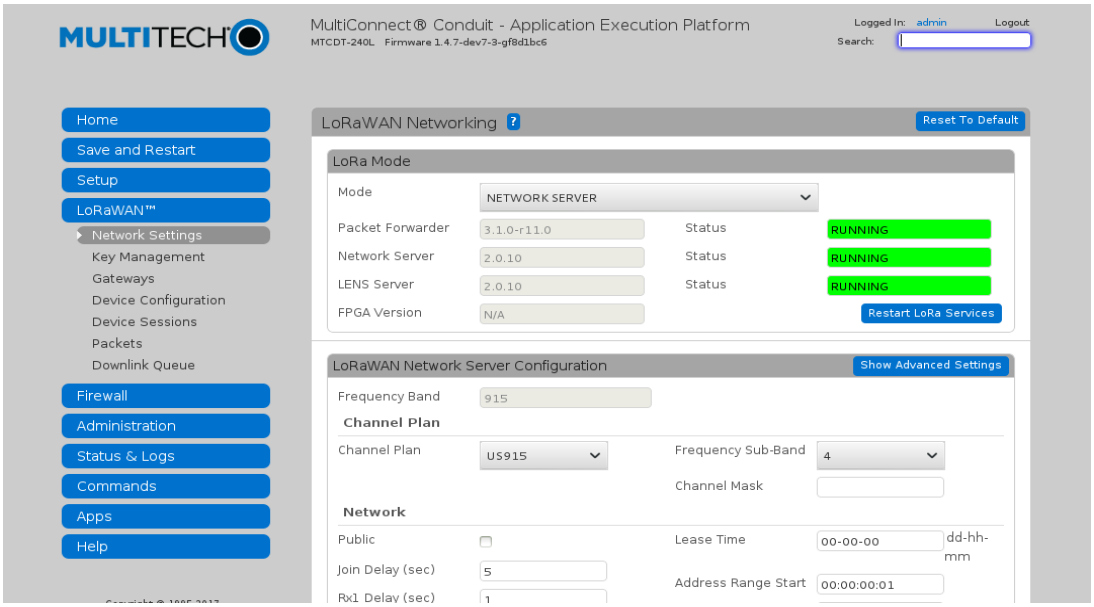
4.1.3.1 On a Forwarding Conduit

1. Go to **LoRaWAN** > **Network Settings**



2. Set the Frequency Sub-Band to the desired setting
3. Click Submit
4. Save and Restart the Conduit

4.1.4 Configure Network Server to support additional channels



1. Go to **LoRaWAN > Network Settings** on Conduit
2. Set channel mask to enable the additional channels
   - FSB 1, 2 and 3 – `00070000000000FFFFFF`
   - FSB 1, 2, 3 and 4 – `000F00000000FFFFFFFF`
   - FSB 5, 6, 7 and 8 – `00F0FFFFFFFF00000000`
   - FSB 1 and 8 – `0081FF000000000000FF`
3. Click Submit
4. Save and Restart the Conduit

5 AEP 1.4.11 Other Changes

5.1 Changes

- LoRa Network Server version 2.0.19
- Update node.js to version 0.10.48

- Update Node-RED to version 0.15.3
- LoRaWAN Menu and new pages
- Packet Forwarder Conduit GUI update
- FW Switch for Dual FW image radios
- Telit Firmware Upgrade
- Call Home Enable/Disable commands
- LTE Radio support for MTCAP
- Radio firmware upgrade restart and increase upload timeout
- Added global_conf.json setting for packet forwarder autoquit if not connected to network server for 10 minutes
- Added timeout for network server to restart lora services if there is no packet forwarder communication for 10 minutes

5.2 Bug Fixes

- Statistics - Fix memory buffer and cache values
- Debug Options – Fix available syslog level filters, add CONFIG and TRACE
- GPS – Display position data correctly
- Time – Time zone setting correction after reset to defaults
- Radio - Add support for LDC3 radio (MTCDT-LDC3-246A-JP)
- Custom Apps - Update user defined defaults to backup and restore custom apps installed in /var/config/app

5.3 Known Issues

- OpenVPN - The 2.1.3 version of OpenVPN is vulnerable to SWEET32 and needs to be upgraded or patched
- RS485 - Full Duplex not working with Node Red
- Web UI - Changing HTTP and HTTPS ports causes login issues for customers (AEP v1.4.1)
- PPP - /etc/ppp/ip-up and ip-down do not call run-parts
- Home page doesn't display "Idle timeout occured, waiting for demand" after cellular dial-on-demand idle timeout
- Firewall - Input/Output Filter rules rules failed to load when Protocol is set to "ANY"
- Firewall - "Source MAC" is ignored in the rules

- Firewall - Input/Output Filter rules: the description is always empty for new rules
- Radio - Signal strength bars displayed in Cellular (ppp0) do not match LED's on CDT
- Radio - LAT3 radio not supported
- Bluetooth - Statistics shows 0 bytes sent
- Bluetooth - Show Log displays nothing in WiFi/Bluetooth Statistics
- WiFi - Access Point and WiFi as WAN not working concurrently for 5G
- WiFi - Statistics counts packets, not bytes
- Custom Apps – current behavior after reset to factory defaults. Apps in /var/config directory are removed. Apps installed on an SD card will not be removed by the reset to defaults action.

6 Copyright

7 Trademarks