SEMTECH

WHITE PAPER

LoRa Basics™ Modem Relay: A Low-Cost Battery Powered Network Extender

Patrick van Eijk, *Senior Director IoT Solutions*

LoRa®

semtech.com

The LoRaWAN® Relay Specification is a new range extension feature created by the LoRa Alliance® and supported by Semtech in the LoRa Basics™ Modem v4 stack. This new relay feature enables the creation of low-cost battery-powered range extenders for extreme corner cases where even the impressive link-budget of LoRa® modulation is not sufficient to make a connection.

Introduction

There is a limit to the link-budget of every wireless technology, even for LoRa modulation which has an exceptional link-budget of up to 158dB with a transmit (TX) power level of 22dBm (160mW). Corner cases can arise where, for example, a few LoRaWAN connected water or gas meters simply won't connect to any gateway because the RF path is too lossy. Until recently, the only remedy has been to install an additional gateway closer to these stranded meters. Unfortunately, any gateway installation comes with both capex and opex expenses making this a prohibitive choice for some deployments; installing an additional gateway to connect a few stranded meters can be overkill. The alternative can now be found in a recently developed solution from the LoRa Alliance, called LoRaWAN relay as shown in *Figure 1*. This white paper will give an overview of the relay implementation and the interaction of the relay with both the end device and the network server. We will conclude with a few use case examples.

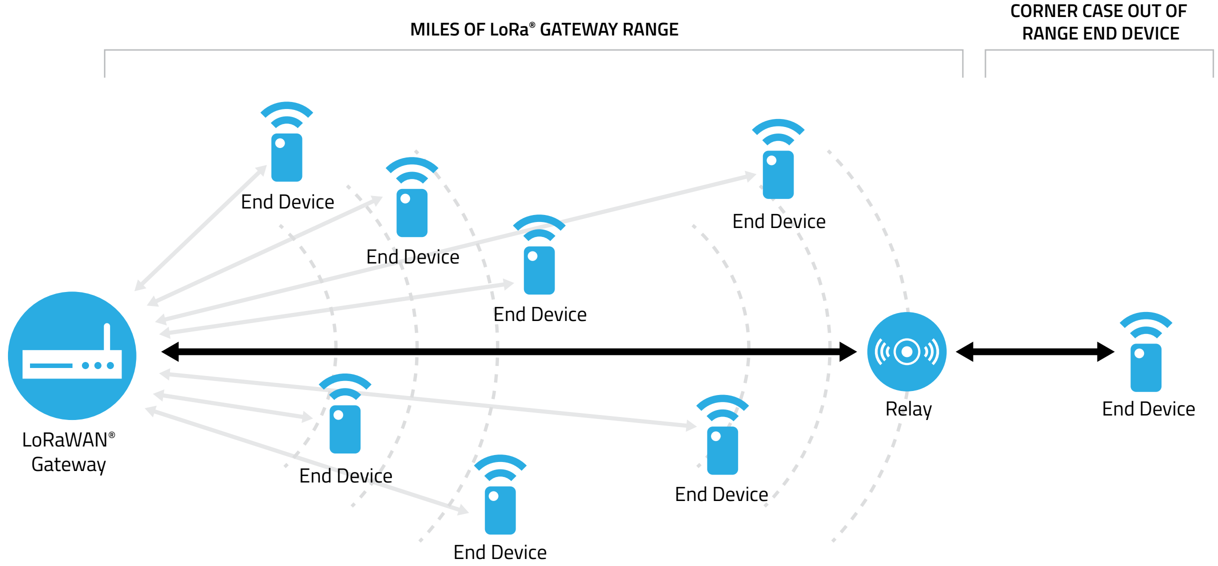


Figure 1: *Relay Device*

Relay Challenges

The LoRaWAN specification allows for three different device classes as shown in *Figure 2: class A, B and C*. The majority of LoRaWAN end devices in the field are class A devices that send uplinks infrequently, randomly, or event-based, and asynchronously. Herein lies the main challenge in the development of a battery-operated range extender. How do you create a range extending protocol that allows your relay device to capture the traffic from one or more LoRaWAN end devices without having to be powered on 100% of the time?

Most of the energy consumed by the relay will not come from the infrequent retransmissions of the LoRaWAN uplinks or downlinks it needs to relay. Instead, most of its battery power is used while listening for LoRa activity on its radio channel. Therefore, by only listening for a very short (~3-6 ms) time per second, we can extend the battery life of a relay. The second challenge is the synchronization between the end device and the relay. If the relay only listens for a short time for channel activity each second, the end device sending the data needs to synchronize with the relay or the uplinks from the end device will never reach the relay.

How Is a LoRaWAN Relay Implemented?

A battery-operated relay is technically also a Class A LoRaWAN end device with its own unique device identifier and root keys that allow it to connect to a specific LoRaWAN network. Its main purpose is to listen for other LoRaWAN end devices that want to transmit their LoRaWAN uplink (frames) to the same network server to which the relay is already connected. The relay uses the same single channel LoRa transceiver IC (the radio chip) as any other LoRaWAN end device. These LoRa transceiver ICs have a special ability to detect a LoRa preamble pattern (a unique repetitive pattern) while operating in a very low power mode called Channel Activity Detect (CAD). Semtech's application note AN1200.48 provides more details on CAD. While operating in this CAD mode the LoRa radio will perform a scan on a specific radio channel for a user-selectable number of LoRa symbols. If LoRa activity is detected on the channel the radio will wake up additional on-chip logic to process the expected LoRa packet. If no preamble activity is detected the radio will go back to sleep. A relay will be able to support up to 16 different end devices.

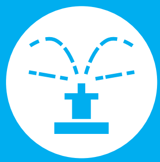
Note, it is possible to add relay functionality to any LoRaWAN end device, like a temperature sensor or a water-meter. In this case however, battery-life prediction could become challenging since additional battery power is needed to support the relay function, impacting the battery life of the end device.

Figure 2: *LoRaWAN Device Classes*

LoRaWAN®
DEVICE CLASSES

CLASS – A

- Battery Powered Devices
- Report status infrequently
- No low latency actuation requirements
- Extremely low energy
- Example: Temperature sensors



CLASS – B

- Battery Powered Devices
- Report status infrequently
- Low latency actuation requirements
- Low Energy
- Example: Irrigation valves

CLASS – C

- Main powered devices
- Constantly listens for Downlinks (except when transmitting)
- Low latency actuation
- Example: Streetlights

Wake-on-Radio (WOR) Frames

As part of the relay protocol a Wake-On-Radio (WOR) frame that is sent from the end device to the range-extending relay has been created. The goal for this WOR frame is to "wake-up" the relay and communicate specific information about the LoRaWAN frame that needs to be forwarded to the network server. *Figure 3* below shows a high-level diagram of a relay that connects an end device with a gateway.

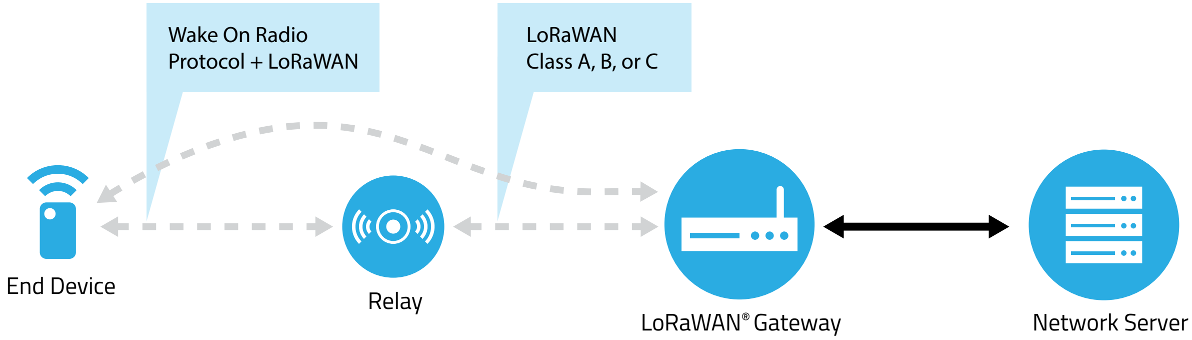


Figure 3: *Relay Mechanism Principle*

With a relay that only wakes up once a second, the LoRa WOR frame preamble needs to be at least 1 second long. After successful reception of the initial WOR frame the relay responds to the end device with a WOR Acknowledgement (WOR ACK) frame which contains certain information which allows the end device to synchronize itself with the relay and significantly shorten its WOR frame length. Next the end device will send its LoRaWAN uplink to the relay. Subsequently, the relay will forward this LoRaWAN frame to the network server using its own LoRaWAN connection. A new LoRa reception window (in addition to the existing RX1 and RX2 windows) called RXR has been created to a relayed end device to receive forwarded downlinks (frames coming back from the network server through the relay). *Figure 3* depicts the message flow between an application server and an end device connected through a relay. There are two different types of WOR frames that an end device can send to a relay. The first one is called a Relay Join-Request that an end device will send to join its LoRaWAN network. The second one is called a Relay Class A Uplink which the end device will send when it is ready to send an uplink to the associated application server.

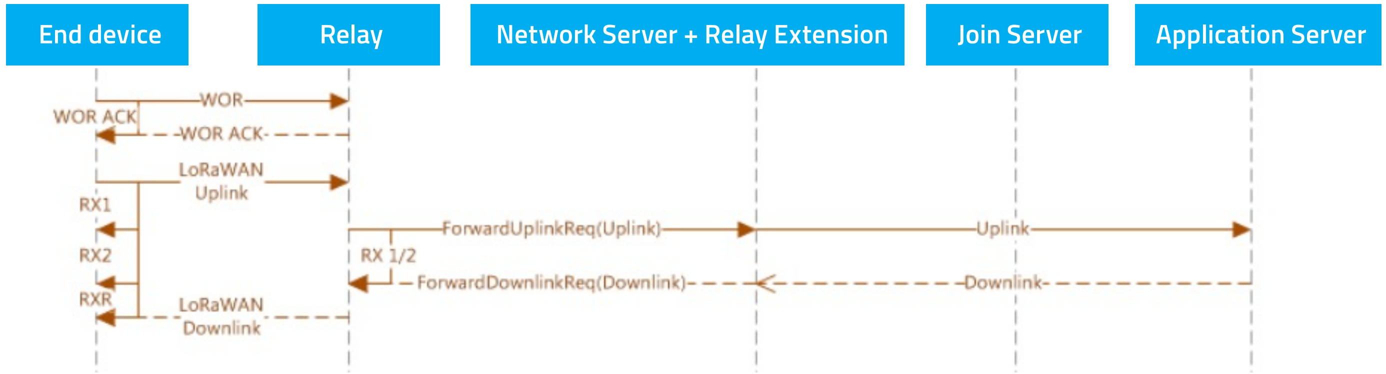


Figure 4: *End device messaging through a LoRaWAN Relay*

WOR Radio Channels and Data Rates

For the North American region (US915), the LoRa Alliance has designated a total of four radio channels dedicated to the relay. Two are for the WOR frames being sent from the end device and the other two are for the WOR ACK frames sent from the relay to the end device. The relay channel parameters for each region can be found in the latest Regional Parameters Specification 1.0.4. The channel parameters listed below are for North America. These four channels use a fixed channel bandwidth (BW) of 500kHz and a fixed spreading factor of SF10.

Table 1: US915 REGIONAL RELAY CHANNEL PARAMETERS

CHANNEL INDEX	0	1
Frequency WOR (MHz)	916.7	919.9
Frequency WOR ACK (MHz)	918.3	921.5
Spreading Factor	SF10	
Channel Bandwidth	500kHz	

Synchronization Considerations

A LoRaWAN end device that has not yet synchronized with a nearby relay is unaware of several key parameters:

- The CAD periodicity of the relay
- Delay between the CAD to RX mode of the relay
- The data rate (DR) used by the relay to connect to the network server (channel BW and SF)
- The relay crystal oscillator frequency accuracy in ppm

Therefore, the end device is unable to calculate the exact time when the relay will be listening to the default channel. Before the end device has received its first WOR ACK frame, it will assume that the CAD periodicity is 1s (1,000ms), a relay real time clock accuracy of 40ppm and that it takes 8 LoRa symbols to move from the CAD to the RX state. In the WOR ACK frame that the relay sends to the end device to acknowledge the WOR Uplink frame there is a 3-byte encrypted payload that provides information on the key-parameters, allowing the end device to calculate the expected CAD window and significantly shorten its WOR frame preamble symbol sequence. Please refer to Appendix 1 of the latest technical specification on the relay from the LoRa Alliance where a detailed example is provided on how to calculate the expected CAD window: https://resources.lora-alliance.org/technical-specifications/ts011-1-0-0-relay

WOR Frame Security

Both the relay and the end device connecting through the relay have their own individual device identifier and root keys which allow them to derive individual application and network session keys for secure communication with the LoRaWAN network server and end device related application server. However, these session keys cannot be used to encrypt the WOR and WOR ACK frames that are sent between the relay and the end device. Therefore, a new root key has been defined called the Root Relay Session Key. This key is used by both the end device and the relay to derive two new session keys that provide encryption and data integrity for the communication between the end device and the relay. Once an end device has received a Join Accept downlink it can derive the Root Relay Session Key from its Network Session Key. The relay will receive a copy of the actual Root Relay Session Key directly from the network server in a specific MAC command (more on MAC commands on page7 *“Relay MAC Commands”).*

Message Forwarding and Encapsulation

Once the relay has received a valid LoRaWAN uplink or Join-request frame from the end device on one of the two dedicated 500kHz channels *(as listed in Table 1)*, it will forward this frame to the network server. The end device will know to not exceed the payload size related to the data rate (DR) that the relay will use when forwarding the payload data from the end device to the network server since the end device is notified of this specific DR as part of the fields in the WOR ACK frame. There are 6 bytes of metadata overhead representing the below listed fields that the relay will forward to the network server in additional to the uplink or join-request frame:

- Receive Signal Strength Indicator (RSSI) of the received uplink or join-request frame
- Signal-to-Noise Ratio (SNR) of the received uplink or join-request frame
- RX frequency of the LoRaWAN uplink or join-request frame
- RX data rate the LoRaWAN uplink or join-request frame
- WOR channel used for the WOR frame

The original uplink frame from the end device will now be wrapped inside the payload of the LoRaWAN frame from the relay to the network server with 6 prepended bytes of metadata overhead. Let us look at a quick example.

If the original WOR uplink from the end device to the relay was a LoRaWAN frame with 11 bytes of payload, the payload of the relayed LoRaWAN frame needs to at least contain enough space for the original frame:

9 bytes of overhead (MHDR, FHDR +FPort Fields) + 11 bytes of payload + 4 bytes of MIC (Message Integrity Code) + 6 bytes of metadata (added by the relay) for a total of 30 bytes.

Therefore, the Data Rate (DR) of the relay connection to the network server can never be DR0, it needs to be DR1 or higher since the payload associated with DR0 is limited to 11 bytes max. Fortunately data rate 1 (DR1) supports a payload of 53 bytes, which provides adequate space for our example above. It is up to the network operator who manages the network server to ensure that the data rate between the relay and the network server provides enough payload capability for the end device traffic.

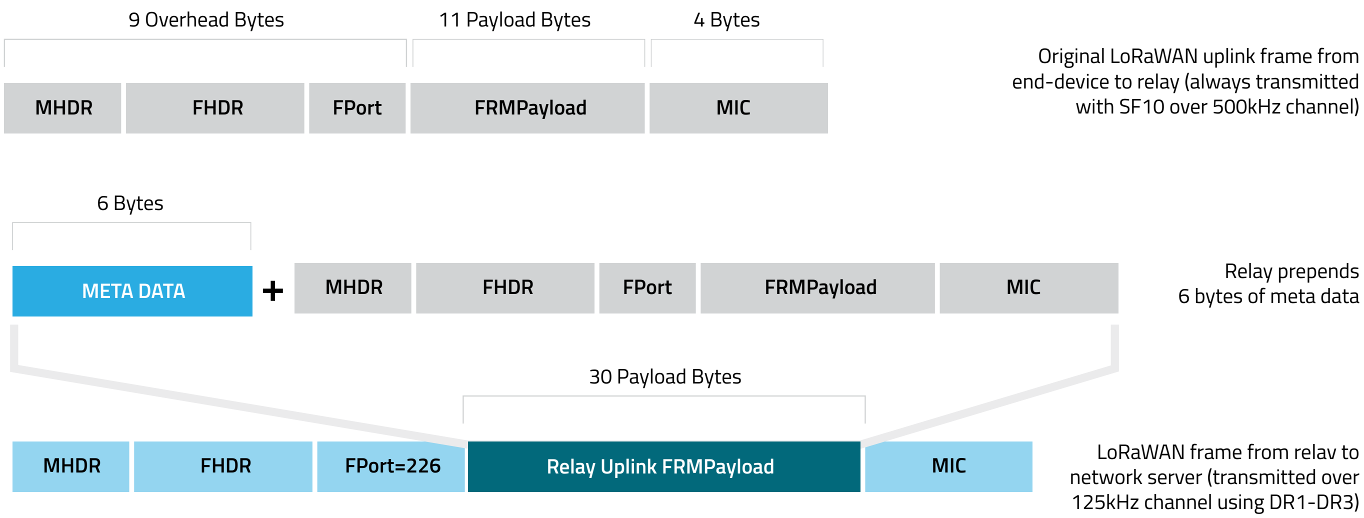


Figure 5: *Uplink Payload Example from End device to the Network Server via Relay*

The relay will send the uplink frame from the end device on a specific frame port (FPort = 226) to the network server. The relay will use its Network Session Key to encrypt the new frame so that the network server can strip off the encryption and metadata and forward the uplink frame that was encapsulated to the appropriate Application Server as can be seen in *Figure 4.*

In a similar fashion, a downlink message from either the Application Server or a Join Accept downlink message from the Join Server can be sent from the network server via the relay back to the end device.

Relay MAC Commands

The relay feature defines a set of MAC Commands that allows the network server to configure the relay-related parameters for both the end device and for the relay itself including the delivery of the Relay Root Session Key as described on page 5 *"WOR Frame Security"*. Through MAC commands, additional key features can be enabled such as a second relay frequency channel, Join Request Uplink filtering capability based on the JoinEUI and DevEUI fields, and token buckets limiting the number of Uplinks and Join Request uplinks to preserve relay battery life.

Please refer to the latest LoRa Alliance relay specification for more information.

Relay Hardware and Software Implementation Requirements

A relay device can be implemented with one of two different Semtech sub-GHz LoRa transceivers families, the SX126X or the LR11XX, or with any third-party module or SIP based on these two variants. The SX127X family of LoRa transceivers is not suitable for a relay design because of its limited CAD functionality. Unlike a regular LoRaWAN end device, a relay will require a temperature compensated oscillator (TCXO) to meet the strict real time clock accuracy requirements described on page 4 *"Synchronization Considerations"*. A relay device that will be mounted outside could benefit from a small solar cell to charge its battery.

For any end device to be able to communicate with a relay, it needs to have a LoRaWAN stack that supports the relay specification. Semtech has released a version of its LoRa Basics Modem that supports the LoRaWAN Relay Specification TS011-1.0.0. The latest LoRa Basics Modem code can be found on GitHub: https://github.com/Lora-net. There is a single version of this relay code which will need to be compiled for either an end device or for a relay.

In order to support the relay feature, a network server will require an update. ChirpStack has recently announced a test release that supports relay: https://forum.chirpstack.io/t/release-chirpstack-v4-4-test-releases/16786.

Please contact your network server provider for information on their planned support for the LoRaWAN Relay Specification.

In a city with **20,000** water meters there could be **600** to **1,000** stranded meters.

Adding **50** to **100** battery operated relays in strategic locations (strapped to a streetlight for example) will solve connection issues.

Use Case Examples and Summary

There is a myriad of use cases for a relay that can bridge the gap between one or more end devices and a gateway using LoRaWAN that is out of link-budget reach for a host of different reasons. For example, in the hilly wine region of Napa Valley California it is hard to reach every well monitoring system from gateways placed strategically on select hill-top infrastructure because of the terrain's rugged topography. A strategically placed battery-operated relay will prevent the need for an additional gateway installation. Another very strong application for a relay can be found in Advanced Metering Infrastructure (AMI) network installations, or so-called smart metering. There is always a small percentage (3-5%) of meters that simply cannot connect to the nearest gateway. Water meters are most often either located in the basement of a residence or in a concrete pit with a metal or concrete lid. The superior link-budget of LoRaWAN connected water meters allows almost all these water meters to connect to one or more nearby gateways, but there are always exceptions. For example, in a city with 20,000 water meters there could be 600 to 1,000 stranded meters, often found in small clusters that can't find a gateway to connect to. Adding 50 to 100 battery operated relays in strategic locations (strapped to a streetlight for example) will solve the connection issues of these stranded meters and represents a very small capex expenditure compared to adding dozens of additional gateways which could cost 10x more when factoring in overall equipment and installation costs.

On a smaller scale there are other use cases as shown in *Figure 6*, such as in the smart building market vertical that could also greatly benefit from the Relay functionality.

What all these use cases have in common is the fact that with relay, the LoRa network coverage can be greatly extended at a fraction of the cost. In conclusion, the newly developed relay feature allows the creation of unique low-cost battery-operated devices that provide a range extending solution for very hard to reach LoRaWAN end devices in various market verticals such as smart metering, smart agriculture, oil and gas, and smart building. We expect to see relay hardware from several OEMs expected to enter the market later this year.

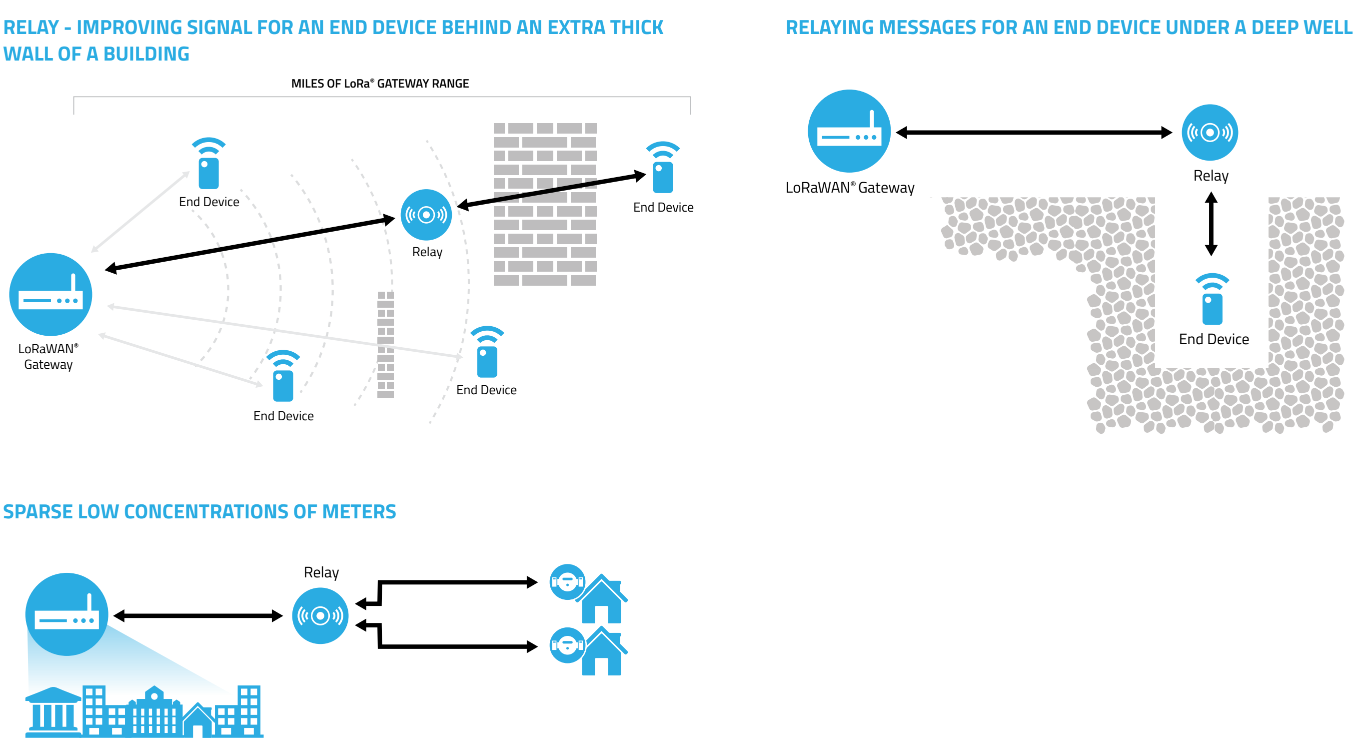


Figure 6: *Relay Use Case Examples*

FOR MORE INFORMATION ON THIS RELEASE, PLEASE REFER TO THE FOLLOWING LINKS:

https://github.com/Lora-net/SWL2001/tree/feature/relay

https://lora-alliance.org/

https://learn.semtech.com/

https://deviceroy.com/relay

https://lora-alliance.org/lora-alliance-press-release/lora-alliance-announces-new-relay-feature-that-extends-lorawan-coverage-for-metering-utilities-smart-cities-and-industrial-applications/

https://tektelic.com/expertise/relay-mode-of-operation-extending-lorawan-coverage/

200 Flynn Road, Camarillo, California 93012 | 805-498-2111 | semtech.com | Find us, like us, follow us