

Toggle site



## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结

# LoRa Server Project 项目学习

📅 2018-02-26 | 🏠 开发

目前公司采用的LoRaWAN服务器是一个由开源开发者Orne Brocaar（自由职业软件工程师，专注go、Python，IoT，美国meetup网络社交聚会组“阿姆斯特丹Golang”的组织者）使用go语言编写的一个开源LoRaWAN服务器，项目名称为“LoRa Server Project”，开源于[GitHub](#)。LoRa Server Project是一个性能优越，功能完善，具备柔性框架的LoRa服务端，项目维护更新活跃。

本文描述的项目版本：LoRa Gateway Bridge 2.1.5\LoRa Server 0.20.1\LoRa App Server 0.12.0

## 🔗 项目简介

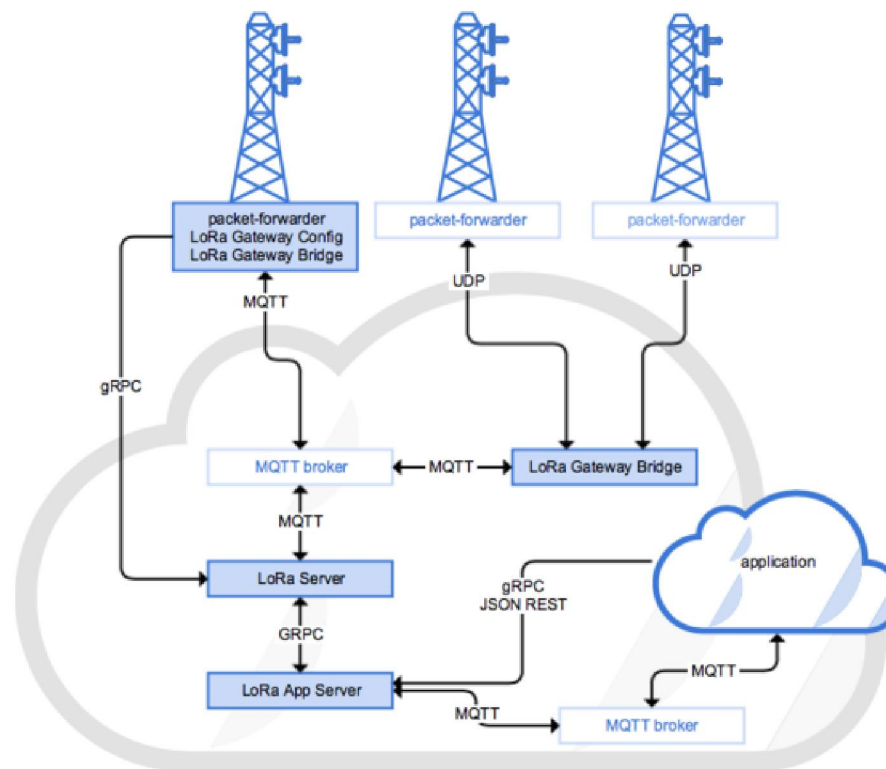
Lora Server project 是一套开源应用软件，实现从网关接收到节点数据一直到应用程序接收到数据这一段链路的处理。整个工程设计地非常灵活，这样可以用不同方式来使用它。例如 LoRa App Server 组件实现应用服务器组件，为用户提供一套Web UI来访问和修改他们的网关、应用程序和节点，还可以通过 gRPC and JSON REST APIs 编程接口来访问系统。而且，API设计地也很灵活，子系统可以用其他相同接口的软件来替代。



## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结



architecture

项目结构的主要部件可归纳如下：

- **packet-forwarder**: Semtech规定的一个用于网关与服务器传输的协议，网关需要运行packet-forwarder软件，通过 [packet-forwarder UDP protocol](#) 向服务端传输数据包,结构中相当于网关
- **MQTT broker**: MQTT代理服务，广播
- **LoRa Gateway Bridge**: 项目中的“网关桥”组件，是NC的一种变形
- **LoRa Server**: 项目中的“中心服务”组件，相当于NS
- **LoRa App Server**: 项目中的“应用服务”组件，相当于AS

Toggle site



## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结

- **MQTT**: 消息队列遥测传输，是一种即时通讯协议，通过一个代理（广播者）实现“订阅/发布”模式进行客户机之间的数据传输
- **UDP**: 用户数据报协议，在项目中指的是网关packetforwarder通过UDP传输层向固定的服务器IP发送mac负载数据包
- **gRPC**: 谷歌远程调用框架，是RPC框架中的一种实现，是一种可编程的应用接口，实现函数返回值能够在网络传输层中传输，应用通过RPC调用提供者的API可直接使用提供者程序内的函数。项目中用于AS，NS之间以及AS和应用之间的接口传输
- **JSON REST**: 指的是标准http RESTful风格的api，格式为JSON，rest是一种风格模式，名为“可重新表达的状态迁移”，是一种互联网发展趋势所形成的api格式标准。项目中用于AS和应用之间的接口传输

目前项目支持的LoRa功能：

- 设备类型：ClassA、ClassC
- 消息类型：(Un)confirmed uplink and downlink, Proprietary uplink and downlink
- 设备入网：ABP、OTAA
- 自适应速率ADR：支持

## 🔗 框架结构

### 🔗 LoRa Gateway Bridge

在LoRa Gateway Bridge（以下称作网关桥）之前，整个LoRaWAN流程中还存在两个概念

- 1 LoRa Node: 终端节点，遵循LoRaWAN协议规定的无线通讯模块或设备，通过LoRa网关向LoRa Network中传输数据
- 2 LoRa Gateway: 即网关，网关用于接收终端节点的数据，网关上必须运行 `packet-forwarder` 的一个实现，通过该软件与LoRaWAN服务端进行通讯

网关桥上下游结构图：

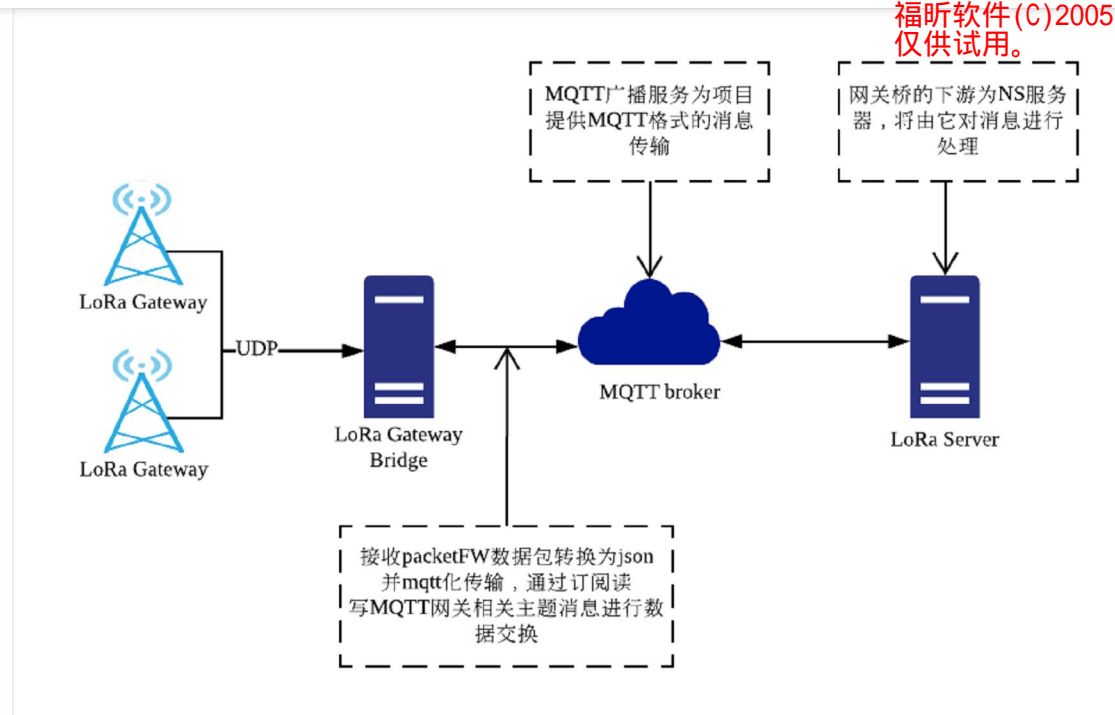


Toggle site

## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结



lora-gateway-bridge

网关桥是一个类似于LoRaWAN中规定的NC服务组件，LoRa Server Project中实现通讯的上游服务。作者将NC进行了变换，采用JSON化数据包后通过MQTT方式进行下游的通讯，比起直接使用UDP协议直接传输的优势和特点：

- 更容易进行debug，MQTT以及JSON数据更能够不接触组件内部进行debug
- 下发下行数据只需要获得相关网关的MQTT topic 地址，MQTT broker将自动路由至负责该网关的网关桥实例
- 提供了一个可安全传输的途径，使用MQTT over TLS 即MQTTS，可实现ssl加密传输，直接使用UDP不支持
- 在未来，不同版本的网关桥可以处理不同的网关协议，剩下的服务组件只需知道MQTT化的JSON消息格式

网关桥的进出入接口：

1 上游LoRa网关的通讯接口：UDP/IP 协议，端口为1700 （可配置）

2 下游通讯接口：MQTT 协议



### Catalog

You've read 100%

- 1. 项目简介
- 2. 框架结构
- 3. 总结

topic名称	订阅/发布	作用
gateway/{mac}/stats	发布	向NS发送网关数据统计，状态统计
gateway/{mac}/rx	发布	向NS发送终端设备的信息负载数据包
gateway/{mac}/tx	订阅	接收NS的下行消息发送

目前网关桥内具体程序运作流程尚待研究。

### LoRa Server

LoRa Server（以下称作网络服务器），是一个开源的LoRaWAN network-server 即NS服务器，它的任务是对从网关接收到的数据包进行“去重”和其他处理，处理LoRaWAN mac层以及对下行数据传输进行安排。

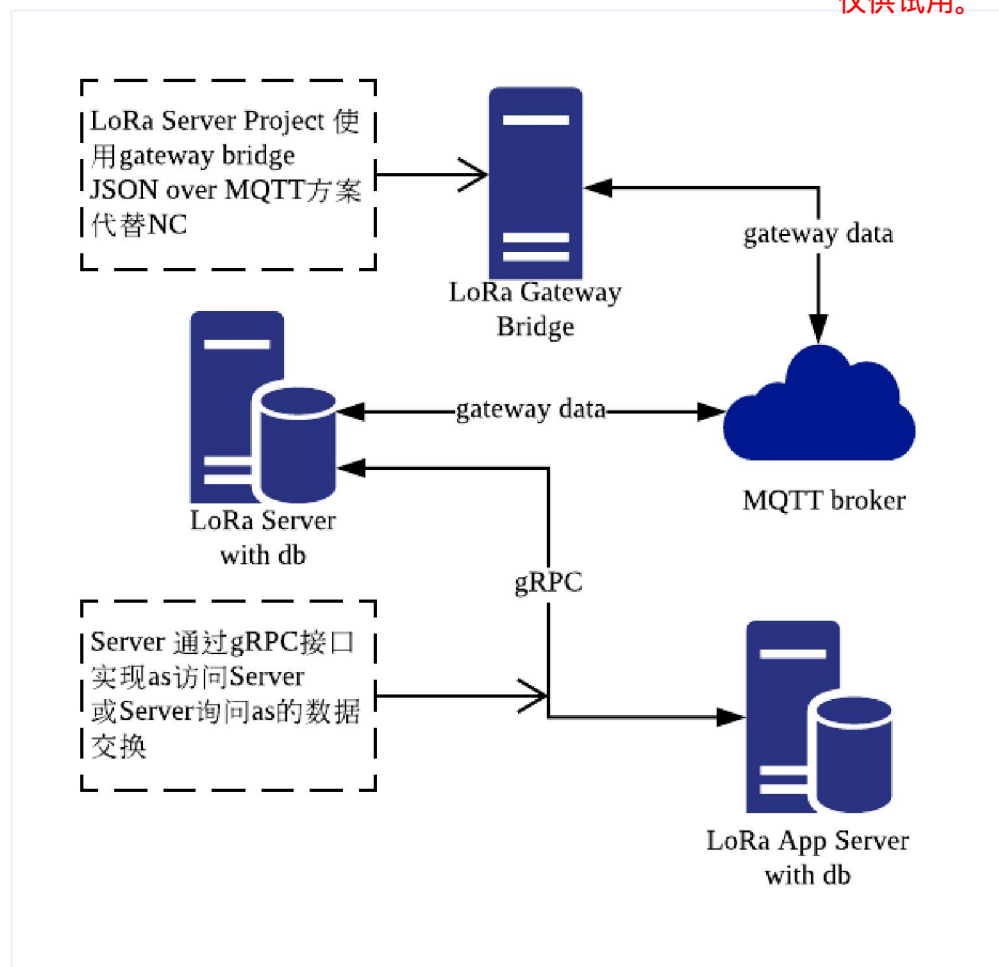
Toggle site

## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结

网络服务器的通讯结构图如下：



lora-gateway-bridge

网络服务器主要处理终端节点的数据网络：

- 1 为节点数据传输创建“会话”，当接请求入网时，网络服务器将向LoRa App Server 询问节点是否在AS上注册，判断是否接受入网
- 2 在一个激活的节点会话中，处理重复的数据（由不同网关上传），决定消息的来源网关

3 认证消息时间，确保消息不存在重复攻击。

4 向 LoRa App Server 推送数据，并询问是否返回消息。

Toggle site



## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结

网络服务器实现了一个基于gRPC的API，用户可根据此编写自己的App Server接入  
网络服务器的进出入接口：

- 1 与App Server 通讯的gRPC接口：gRPC调用，端口：8000（可配置）具体gRPC内容尚待研究
- 2 与网关桥通讯的MQTT接口：MQTT协议

topic名称	订阅/发布	作用
gateway/{mac}/stats	订阅	接收网关桥的网关数据统计和状态统计
gateway/{mac}/rx	订阅	接收网关桥的节点数据包上传
gateway/{mac}/tx	发布	向网关桥发送下行消息命令

网络服务器主要的数据库模型：

- 节点帧日志表：储存节点上下行消息的日志，其中的负载是加密的，用于debug
- 网关通道表：用于储存服务器可配置的网关通讯通道的模型（具体使用尚待研究）
- 设备消息队列：用于储存设备下行消息队列（针对不同类型节点，服务器的下行消息都先缓存与服务端，当适合发送时，服务端将数据下发，并从消息队列剔除）  
其他详细业务尚待研究

## LoRa App Server

LoRa App Server（以下简称AS）可以理解为一个终端节点身份的仓库，处理节点应用的上行负载消息和下行消息队列，储存节点的入网许可信息，负责对申请入网请求作出回应，消息的解密以及加密等。AS向外提供了节点信息，网关信息，服务端其他配置等操作的gRPC/http RESTful 接口，向MQTT broker暴露了各个节点的MQTT格式的实时消息接口，用户根据接口编写自己的应用或应用服务器。

AS的通讯结构图如下：

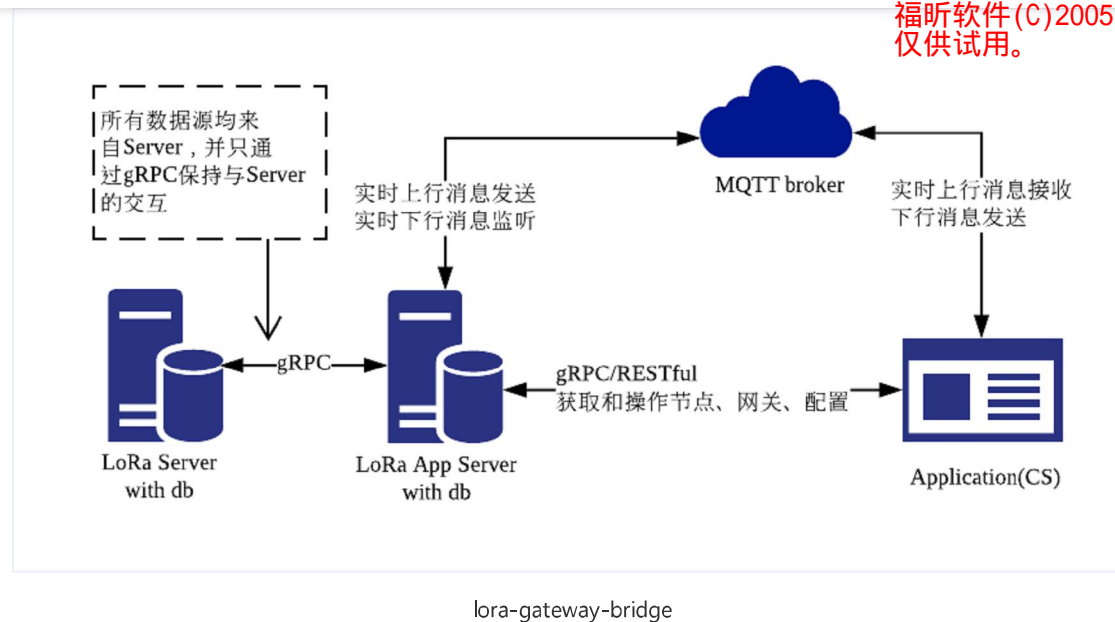


Toggle site

## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结



本页已使用福昕阅读器进行编辑。  
福昕软件(C)2005-2010，版权所有，  
仅供试用。

AS的功能模块主要如下：

- 1 登录功能：AS的所有API均使用JSON WEB TOKEN验证模式进行身份验证，用户必须进行登录后获取token，在后续接口调用中加入token后才能访问。同时，AS提供了一个web交互界面，访问端口为8080，使用https，用户可以直接使用此web交互界面进行节点、网关等操作
- 2 用户功能：AS把用户分为全局管理员以及普通用户区别，全局管员可操作任意模型的属性
- 3 组织功能：AS提供了组织权限，用户可隶属于某个组织，使用不同组织来划分使用组或公司
- 4 应用功能：一个应用隶属于一个组织，应用下可添加节点，根据应用隶属的组织安排用户的身份可实现用户对节点操作和可见性的权限控制
- 5 网关功能：一个网关隶属于一个组织，在AS上添加网关后，可以实时查看网关的状态信息（不添加也不影响传输）

AS的接口：

- 1 gRPC接口：使用grpc+grpc-gateway将http-RESTful与grpc合为一体的服务，同样为8080

2 HTTP 接口：访问 [hostname:8080/api](https://hostname:8080/api) 可查看 API 列表，调用同样使用 [hostname:8080/api](https://hostname:8080/api)







Catalog

You've read 100%

- 1. 项目简介
- 2. 框架结构
- 3. 总结

- 3 HTTP推送接口：在AS上注册了HTTP integrations后，节点的消息将直接由AS使用http推送至指定的服务终端
- 4 MQTT消息接口：

topic名称	订阅/发布	作用
application/{applD}/node/{devEUI}/rx	订阅	接收特定节点或应用的上行消息
application/{applD}/node/{devEUI}/join	订阅	接收特定节点或应用的设备入网事件
application/{applD}/node/{devEUI}/ack	订阅	接收特定节点或应用的响应（针对tx）事件
application/{applD}/node/{devEUI}/error	订阅	接收特定节点或应用的错误事件
application/{applD}/node/{devEUI}/tx	发布	通过MQTT向特定节点发送下行消息

AS中具体的业务实现尚待研究

其他组件

1.MQTT broker

在LoRa Server Project中起到重要作用的组件，实现标准的MQTT通讯协议。MQTT如同聊天室，客户机订阅主题接收消息，也可以在一个主题中发布消息。MQTT的引入使得整个工程组件之间的耦合程度大大降低，提高了组件之间的灵活度，降低了扩展和迭代的复杂度。使用JSON over MQTT代替NC是LoRaWAN服务端的创新

MQTT broker是可替换的组件，只要实现标准MQTT代理功能，均可以在工程中进行配置替换。工程默认使用mosquito mqtt broker

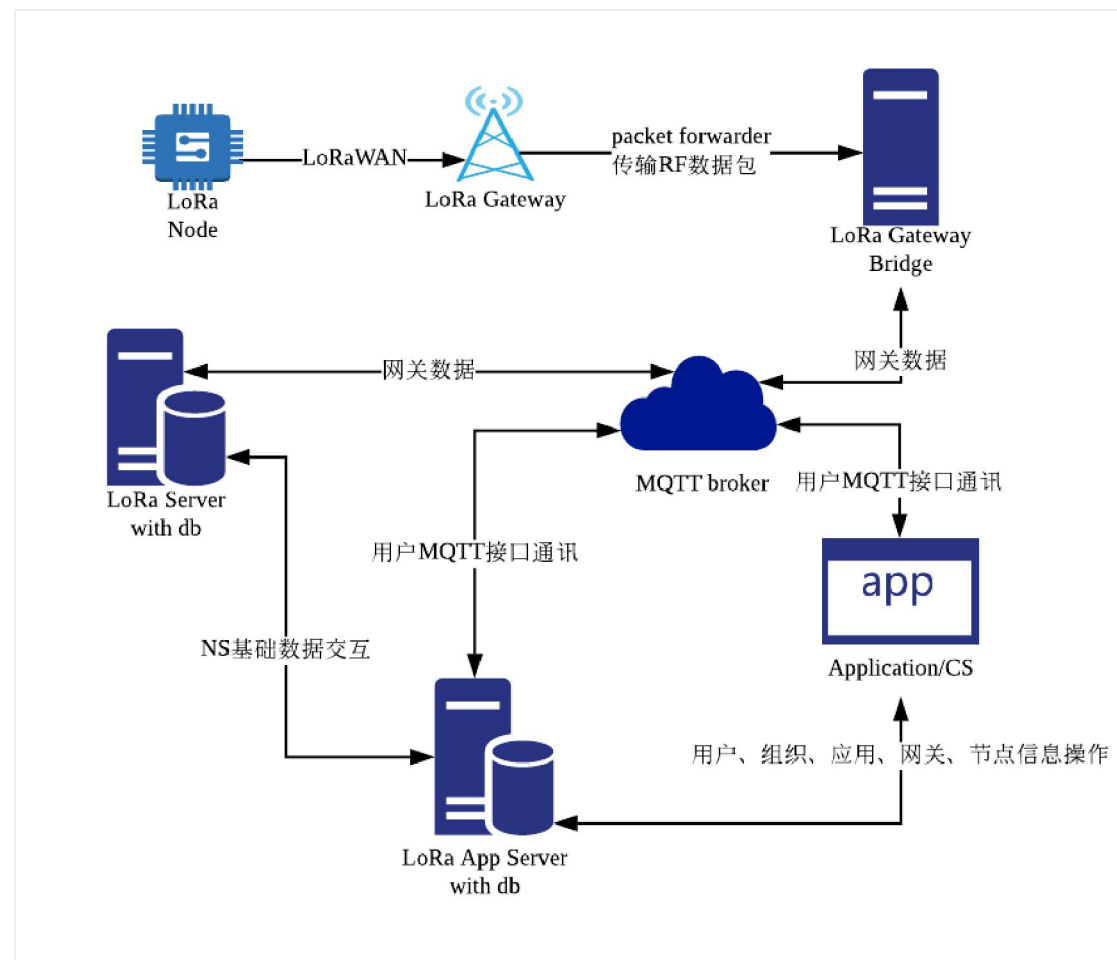
2.Redis

Redis作为一个运行于内存的数据库，提供极快速的储存响应，用于缓存数据和计算统计等，在LoRa Server Project中具体的业务尚待研究

3.PostgreSQL

PostgreSQL是新兴的开源结构型数据库，作为LoRa Server和LoRa App Server的数据储存工具

整个LoRa Server Project 在LoRaWAN实现的流程归整如下：



summary

一个完整的上行流程：

- 1 LoRa节点通过LoRaWAN协议向网关传输RF数据包；
- 1 LoRa网关上运行packet forwarder 软件，封装RF数据包并通过UDP传输至互联网上的LoRa Gateway Bridge；

3 LoRa Gateway Bridge 收到数据包并 JSON 化通过 MQTT 广播至 MQTT 代理服务。

Toggle site

## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结



## Catalog

You've read 100%

1. 项目简介
2. 框架结构
3. 总结

- 3 LoRa Server 通过MQTT订阅LoRa Gateway Bridge的消息，实时处理数据包并储存通知LoRa App Server；
- 4 LoRa App Server 通过gRPC读取LoRa Server 的信息或收到Server的推送，通过HTTP接口和MQTT广播向用户暴露；
- 5 用户/开发者调用LoRa App Server的HTTP接口（信息操作，下行发送）和订阅MQTT（消息监听，下行发送）完成消息获取。

### 一个完整的下行流程：

- 1 用户/开发者通过调用LoRa App Server 的HTTP接口（将一条负载加入下行消息队列）或向MQTT接口的下行topic发送消息；
- 1 一条负载会经LoRa App Server 加密后传输至LoRa Server进行缓存，缓存的空间称为 device queue 即下行消息队列，队列中每个消息都有固定的id，以及目前处于的状态，如pending等；
- 2 LoRa Server根据该下行消息对应的节点的类型，若为class C 该消息将直接由MQTT被直接发送至LoRa Gateway Bridge，为class A，Server将监听消息，计算下行时机再发送。发送后，消息从下行队列中永久删除，Server等待响应，若节点有响应，则Server通知 App Server 在MQTT的ack事件中作出消息触发；
- 3 LoRa Gateway Bridge 收到MQTT的tx信息，向LoRa Gateway发送数据包；
- 4 LoRa Gateway 向终端节点发送数据。

### MQTT的通讯作用：

- 1 用于LoRa Server 与 LoRa Gateway Bridge 的通讯
- 2 用于LoRa App Server 与 用户之间的通讯

到此，整个LoRa Server Project的框架解读完毕，文中可能存在理解和表述错误，其余更详细的逻辑仍需继续研究。

**Author:** [Hobai Riku](#)

**Link:** <https://blog.hobairiku.site/2018/02/26/LoRa-Server-Project/>

**Copyright Notice:** All articles in this blog are licensed under [CC BY-NC-SA 4.0](#) unless stating additionally.

