

LoRa网关双通道进展报告

葛鑫 2017-4-21

本报告主要分为三个部分，第一部分是介绍截止到现在网关双通道SPI开发的进展，实现的功能，以及相比于单通道版本的代码区别。第二部分是为了双通道下行通信设计做了一个网关测试，主要关注点在同频干扰，这部分展示了测试过程和结果。第三部分是关于双通道下行通信方案的一些想法。

LoRa双路SPI开发进展

1. 开发进展及功能实现

- 网关能够打开两路SPI进行读写操作。
- 更改了 `HAL` 层代码(硬件抽象层)，增加了 `port` 参数，使对应函数能够操作指定的SPI。
- 增加了一个上行通信线程 `thread_up_1`，使每个线程对应一个SPI，功能上没有改变。
- 增加了一个配置文件 `global_conf_1.json`，使两路对应了不同的配置文件，从而实现分频接收功能(不同射频接收指定频率的包)，更改了 `lora_pkt_fwd.c` 中解析json配置文件的部分，使之能够解析和配置多份文件到不同的射频。目前 `433MHz` 频道使用的 `SPI0`，`434MHz` 频道使用的 `SPI1`。

2. 代码主要更改部分

- 配置文件 `global_conf.json` → `global_conf_0.json`，`global_conf_1.json`，两个文件对应两路SPI，通过更改文件里 `chan_multiSF_x` 中的字段 `enable` 实现对上行接收频率的选择。
- `lora_pkt_fwd.c` 中
 - 增加上行线程 `thread_up_1`
 - 增加解析并配置 `global_conf_1.json` 文件的代码
- `HAL` 层中
 - `lgw_start()`，`lgw_stop()`，`lgw_receive()`，`lgw_send()`，`lgw_status` 中都增加了 `port` 参数来指定对特定射频的操作
 - `lgw_board_setconf()`，`lgw_lbt_setconf()`，`lgw_rxrf_setconf()`，`lgw_rxif_setconf()`，`lgw_txgain_setconf()` 中屏蔽了对“网关是与否已配置参数”的检查，来重用这部分代码来进行多次配置文件。
 - `loragw_reg.c` 中把 `void* lgw_spi_target` 指针变量更换成指针数组 `void* lgw_spi_target[]`，对应不同的SPI。
 - `loragw_spi.native.c` 中所有函数添加了 `port` 参数。
 - 更改了测试代码，因为不改会报编译错误。
 - 所有代码更改可直接搜索“gexin”，每个更改处都留有注释。

LoRa网关同频干扰测试

1. 为什么要测试

目前还没有一个合适的下行通信设计方案，考虑到射频芯片在发送时不能接收，在接收时不能发送，当大量上行通信需求来临时，网关需要选择一路处于“空闲”的射频讲下行发送出去。那么此时有可能出现两种情况

- 同一时刻，使用相同频率的不同终端发来的包到达网关
- 某一时刻，由射频B发出的下行包的频率与射频A正在接收的上行包的频率一致

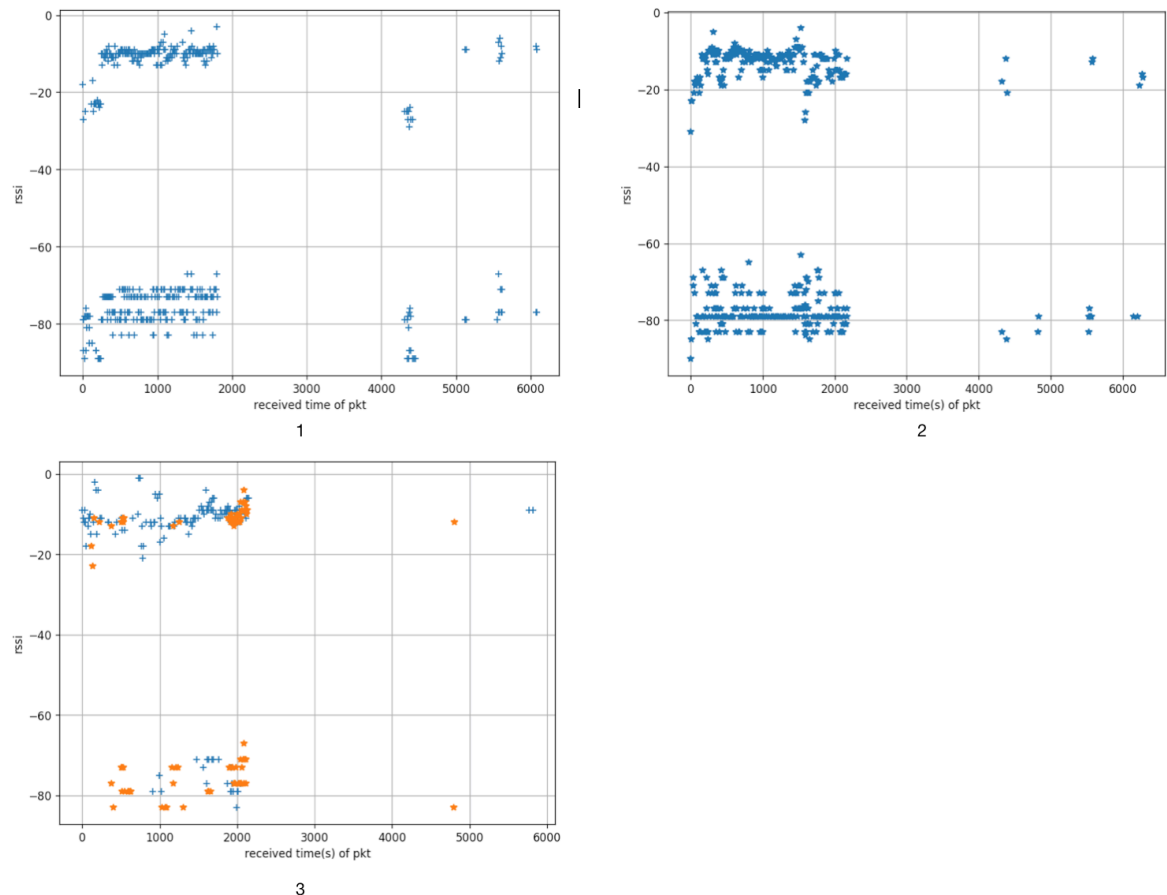
所以首先需要确定的是，在上述两种情况下，网关能否接收到上行的包。

2. 测试方案

使用两个终端，固定频率和速率，同时开启，网关接收后计算丢包率。为了确定有终端本身造成的影响，先分别对两个终端进行测试，再同时开启两个终端测试，总共三次。发送频率为 433.575MHz，速率为 SF12，每10秒发包一次，每次测试时长约30分钟。

测试环境：科B229，距离不超过1米，网关使用了天线。

3. 测试结果



图一图二分别为终端一，二分别测试时的情况，+ 为终端1发送的包，* 为终端2发送的包。图三是同时手动打开两个终端时网关的接收情况。

图中每一个点代表一个包，横轴是这个包的接收时间(与第一个包的相对时间，单位为秒)，纵轴是网关收到时的 rssi 值。

4. 测试分析

- 可以发现包的 rssi 值分化严重，rssi 值小于-60的包基本都是因为室内的多径效应明显产生的“重复包”，在室外测试时并没有如此严重的情况，并且它们也可以被很好的过滤掉。

- 因为每一次测试时间不超过40分钟，也就是 2400 秒，但是可以发现，每个图中都有时间大于 2400 的点存在，因此我找了网关的原始信息查找原因。以下图为例。

```
16: 0
17: 0
18: 0
19: 0
20: 0
21: 0
22: 0
loragw_gps.c lgw_cnt_utc: delta_sec:4294.846540
>>ns:879619499

JSON up: {"rxpk":[{"tmst":45923804,"time":"2018-04-18T15:54:05.846539999Z","tmms":
1208073264846,"chan":5,"rfch":0,"freq":433.575000,"stat":
1,"modu":"LORA","datr":"SF12BW125","codr":4/5,"lsnr":7.2,"rssi":-25,"size":
15,"data":"Q0Z85gGABAAADLd2Xr8xq"}]}
```

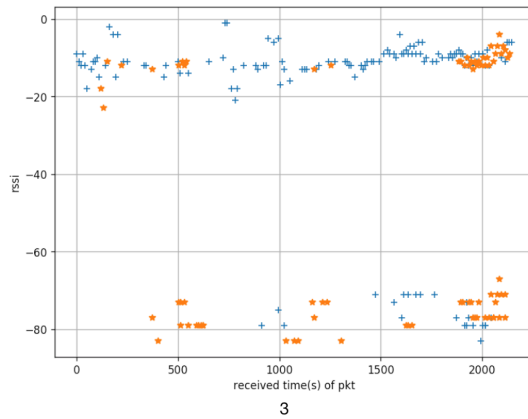
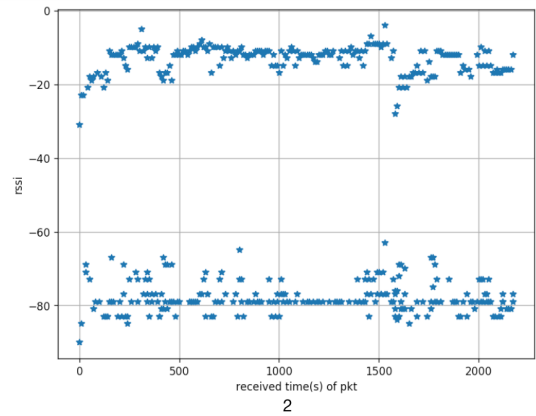
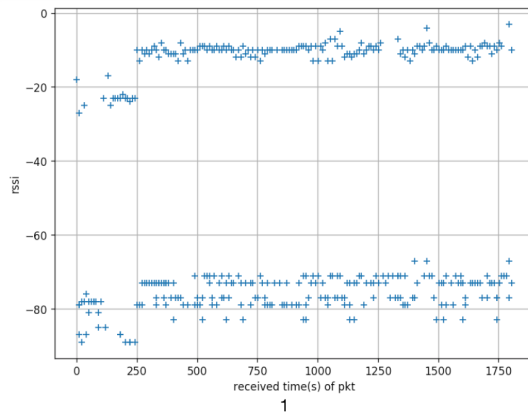
INFO: Received pkt from mote: 01E67CE6 (fcnt=5)
INFO: tx_start delay=1494 (1494.099976) - (1497, bw_delay=1.500000, notch_delay=1.400000)
lgw_send done: count_us=56933068
!!!!!!SEND TO NODE!!!!
*****Payload*****
0: 40
1: e6
2: 7c
3: e6
4: 1
5: 80
6: 5
7: 0
8: 3
9: c8
10: b9
11: 76
12: 82
13: f0
14: 6a
15: 0
16: 0
17: 0
18: 0
19: 0
20: 0
21: 0
22: 0
loragw_gps.c lgw_cnt_utc: delta_sec:0.888626
>>ns:846929930

```
JSON up: {"rxpk":[{"tmst":55933068,"time":"2018-04-18T14:42:40.888626000Z","tmms":
1208068979889,"chan":3,"rfch":0,"freq":433.175000,"stat":
1,"modu":"LORA","datr":"SF12BW125","codr":4/5,"lsnr":-3.8,"rssi":-81,"size":
15,"data":"Q0Z85gGABQADyLl2gvBq"}]}
```

两个包对应的接收时间， 14:42:40 秒接收到的包紧接在 15:54:05 秒接收到的包之后，很明显这是不可能的。而真实的接收时间时 14:42:40 的那个，但是测试时使用的固定频率是 433.575，也就是说，下面那个包的频率是假的，由 rssi 值也可以看出，可能是多径效应改变了频率？（我并不确定）。同样，由 rssi 值可以看到，第一个包是真的（频率也是对的），但是时间不对。

出现这样的现象的原因很可能是紫色框中的 delta_sec 计算错误引起的，本来这个变量是用来计算小数点后的时间，所以它的值应该只能小于1。验证了一下 15:54:05 减去 4294 秒后是 14:42:31，所以可以确定此bug是由这个变量计算错误引起的。

但是目前我还没有时间去详细检查为什么会出现这样的情况，只能先去掉这些“坏点”，得到下图。



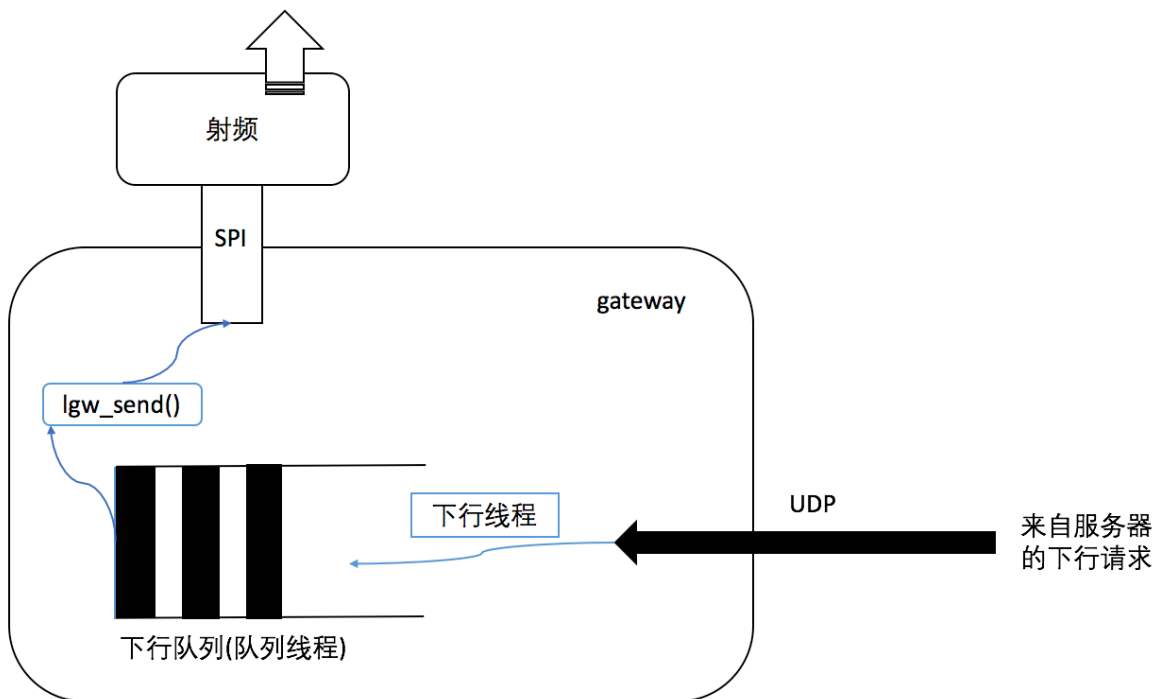
在这样的情况下的到的丢包率（不计算因多径而产生的重复包）：

- 终端一单独测试：10.2%
- 终端二单独测试：3.8%
- 终端一和终端二同时开启测试：总丢包率：65.5%，终端一单独的丢包率：47.7%，终端二单独丢包率83.2%。

可以发现，同频同速率的情况下终端间的相互干扰对网关接收来说有很大的影响。因此在设计下行通信SPI选择方案的时候需要考虑这个情况。

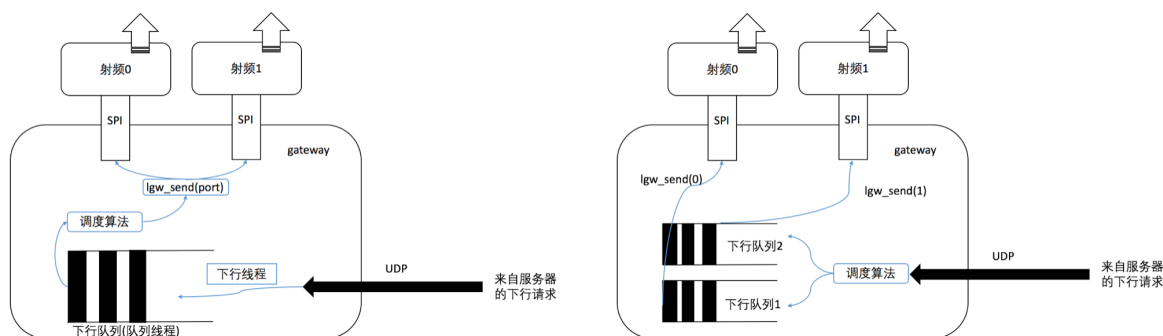
双通道下行通信设计方案

1. 单通道网关的下行通信模型



2. 双通道网关下行通信模型初步构想

目前大致有两种设计思路。分别如下。



我比较倾向于左图的设计，相比于右图可以少开一个队列线程，但对于调度算法的处理速度要求更高。因为队列线程是不停地扫描下一个需要发送出去的包，时间到达后出队直接发送，调度算法太慢的话发送时间就会比服务器指定的时间延后。左边实现起来也比较方便，更加灵活。

3. 调度算法

调度算法是为了扩大网关通信容量，并且下行走的哪个射频对于服务器来说是透明的。

在决定走哪一路的过程中有辅助函数可供调用和参考。

- `lgw_status(...)` 查询指定射频的状态，返回如 `TX_OFF` , `TX_SCHEDULED` , `TX_EMITTING` 等信息。

在单路网关中，如果查询失败直接放弃发送，如果查询成功但返回 `TX_EMITTING` 也放弃发送，返回 `TX_SCHEDULED` 的话就覆盖被scheduled的包，发送当前的这一个。发送若失败不进行任何处理。

现在能想到的是，如果是上面左图那样的网关，可以大致可以"强行"想到一个十分粗糙的调度算法。

假设：射频0只接收 433MHz 的上行，射频1只接收 434MHz 的上行，任何射频在接收上行的时候不能发送下行。

1. 网关维护一对变量 p_0, p_1 ，它们是基于历史上行数据统计后，通过某种算法计算出的在接下来一段时间，是否会有相应频率的包到达网关相应射频的概率。
2. 在有下行请求，并且出队后用 `lgw_status(...)` 分别对射频0和射频1进行查询。

我们视 返回值 `ERROR`，`TX_EMITTING`，`TX_OFF` 为不可发送状态。

视 `SUCCESS`，`TX_SCHEDULED`，`TX_FREE` 为可发送状态

- 两个射频都为不可发送状态：放弃此次下行通信。
- 某一个射频可发送，另一个不可发送：选择可发送的那一路。（若另一路的上行频率与此次需要发送的下行频率相同，是否会发生冲突）
- 两路射频都可以发送：对射频0来说，在这一段时间可能有上行包到达的概率为 p_0 ，对射频1，此值是 p_1 ，我们选择 $\min(p_0, p_1)$ 对应的那一路。（如果发送失败需不需要再尝试另一路？）

加粗的地方都是还需要深入研究的部分。

当然还有另一个实现更加简单的方案，由于每一个射频板上都有2片1255，两路就是4片1255，是否可以拿一片1255不接收上行，只发下行。（在有大于等于2个包需要在同一时刻发送这样的极端情况下这个方案行不通。）

总结一下：下行这部分说实话我并不知道要怎么做，因为面对的场景是大容量通信，由第二部分的测试可知在大容量通信的情况下丢包率应该会很感人，存在大量源源不断的上行的时候下行是发不出去的，此时需不需要使某片射频芯片只发下行，但是这样上行性能就更差了。同样大量下行的情况下上行也收不到，上下行的通信处于一种竞争状态，所以对于优化目标“通信容量”这个定义就比较难刻画，通信容量和质量(低丢包率)之间如何trade off？并且新增加的代码会给网关带来怎样的性能影响(目前双通道版本的网关启动时间就已经比单通道的慢了2~3秒)？希望老师能给予一些帮助和指导，谢谢！