

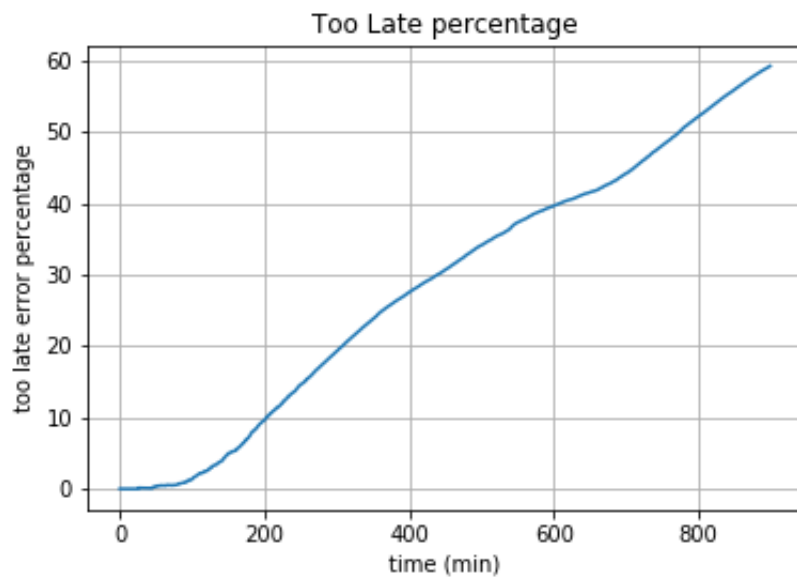
多终端大量上行测试的丢包分析

葛鑫 2019-02-28

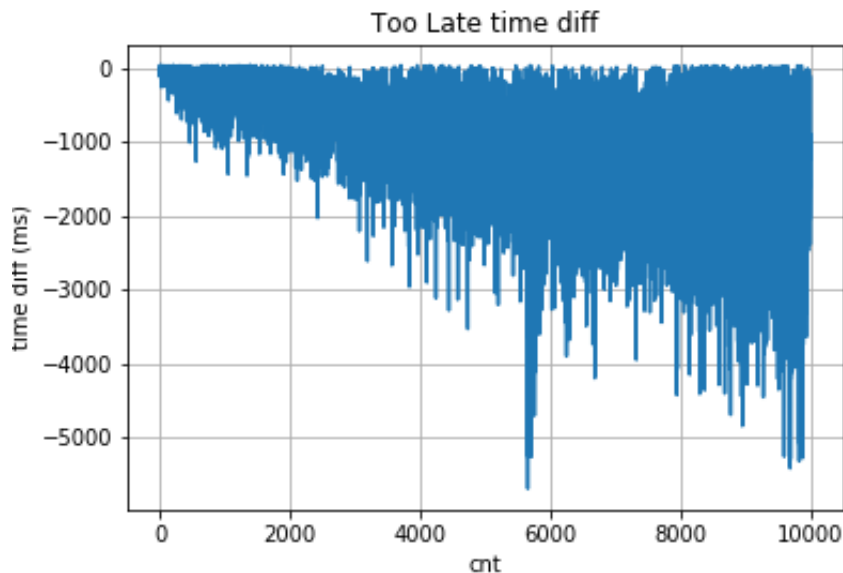
现象

使用多个终端并大量发送(间隔10s)上行包的情况下，丢包率随时间越来越大，几个小时后系统完全无法正常工作。具体表现为：

丢包都是网关too late下行拒绝



因此，对每一对相邻的上行 json up 和下行 json down，分析了包传给服务器的时间 t_1 和从服务器拿到下行的时间 t_2 ，计算差值，结果如下。纵坐标是 $t_2 - t_1$ 。



可以看到，时间差也随着时间越来越大，并且都是负值。

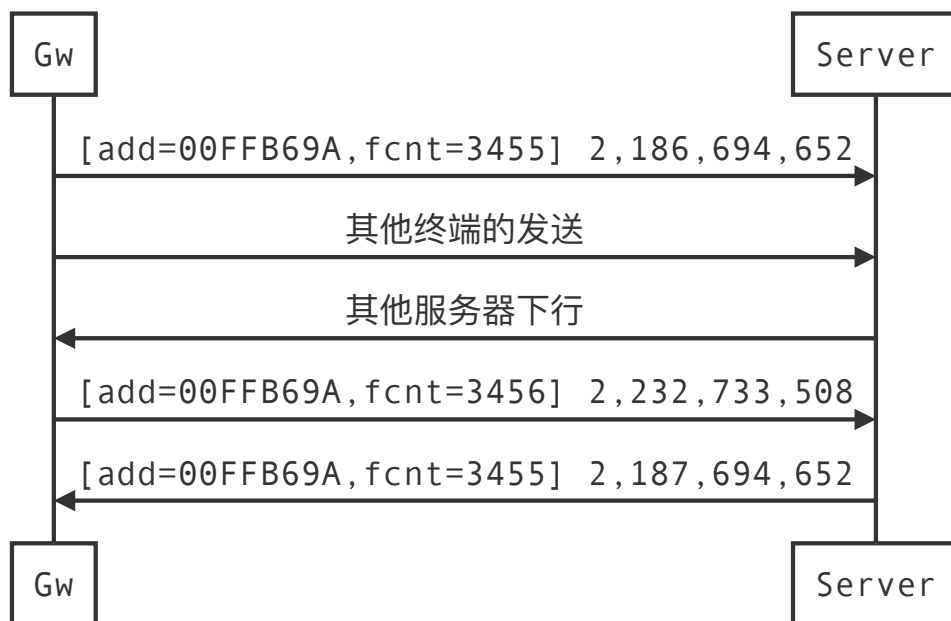
原因

为分析原因，找出了网关打印数据分析

```
JSON up: {"rxpk":{"tmst":2232733508,"time":"2019-01-24T13:04:39.782719000Z","tmms":1232341498782,"chan":5,"rfch":0,"freq":433.575000,"stat":1,"modu":"LORA","datr":"SF12BW125","codr":"4/5","lsnr":3.5,"rssi":-94,"size":17,"data":"gJq2/wCAGA0DM0AgvWzZrCg="}}
INFO: [down] PULL_ACK received in 0 ms
INFO: [down] PULL_RESP received - token[220:104] :)

JSON down: {"txpk":{"modu":"LORA","data":"oJq2/wCqfw0DzPqCqg+TW3s=","freq":433.175,"prea":8,"tmst":2.187694652E9,"rfch":0,"imme":false,"codr":"4/5","size":17,"nsrc":false,"datr":"SF12BW125","apol":false,"powe":27.5}}
##### CLASS A here #####
*****JSON down Payload*****
00->a0 01->9a 02->b6 03->ff 04->00 05->a0 06->7f 07->0d 08->03 09->cc 10->fa 11->82 12->a2 13->af 14->93 15->5b
src/jitqueue.c:233:jit_enqueue(): ERROR: Packet REJECTED, already too late to send it (current=2235202206, packet=2187694652, type=0)
ERROR: Packet REJECTED (jit error=1)
INFO: [down] PULL_RESP received - token[220:104] :)
```

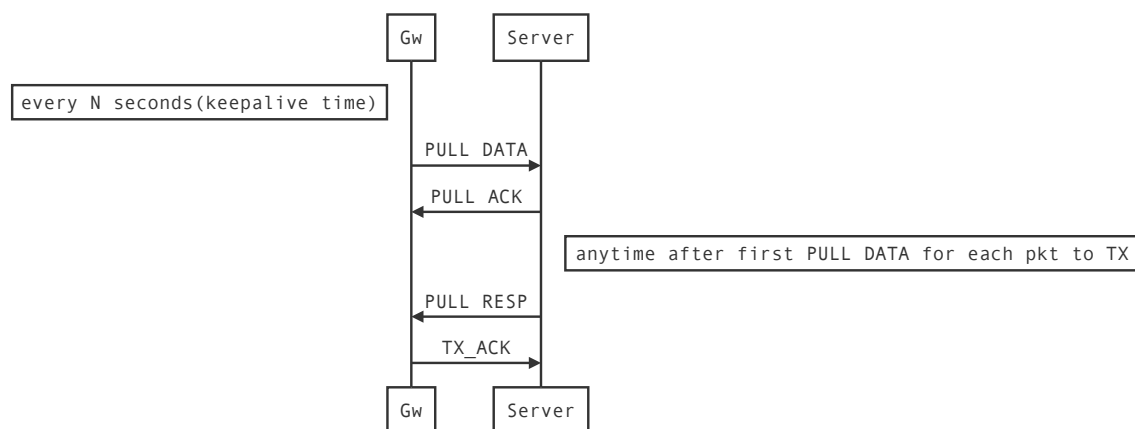
可以看到上行的上传时间是 2232733508，然后收到了服务器的下行，时间戳是 2187694652。按时序分析一下是。



最后一个服务器下行对应的不是 `add=00FFB69A, fcnt=3456`，而对应的 `fcnt=3455`，这个包在几十秒前就发送了，现在才收到对应下行。因此总是产生“too late”拒绝。

因此，真正的问题是：上行发送过多过快，服务器下行回复太慢

那么对于这个问题的原因，是因为keep alive机制导致的。根据网关-服务器下行通信协议



在网关-服务器上行协议中，网关只要有上行数据，就直接发给服务器。

但服务器想下行的话，一定要等一个 `PULL DATA`，而 `PULL DATA` 又每 `keepalive time` 发送一次，因此，每个 `keepalive time`，服务器只能发送一个 `JSON DOWN`。这是回复太慢的真正原因。

解决方案

基于以上原因，我认为以下解决方案。

- 更改 `keepalive` 机制：把 `PUSH DATA` 操作从上行线程中解耦出来，改成单独的线程，从而可以将 `keepalive time` 改得很小而尽量不影响系统性能。
 - 优点：是这个改动不难
 - 缺点是治标不治本，本质是改小 `keepalive time` 达到增大吞吐量，会加重网关的发送 `PULL DATA` 负担和服务器的回复 `PULL ACK` 负担
- 删除 `keepalive` 机制：这个机制存在的原因引用协议原文的说法就是

This data exchange is initialized by the gateway because it might be impossible for the server to send packets to the gateway if the gateway is behind a NAT.

所以如果可以保证网关到服务器之间没有NAT，那么keepalive机制没有存在的必要，因此可以直接删掉协议中 `PULL DATA` 和 `PULL ACK` 交互的部分

- 优点：本质上可以解决这个问题，并且协议上的改动不影响LoRaWAN
 - 缺点：如果网关和服务器之间存在NAT那么系统下行会瘫痪掉
- 更改服务器下行代码：更改服务器侧代码，一旦有机会发送下行，将所有积压的下行包全部封装在一个 `json down` 里（一个 `json down` 里有多个下行包）。更改网关侧代码：使其在解析 `json down` 时支持解析多个下行包。
 - 优点：本质解决问题，`keepalive time` 不用改小，有没有NAT都能用。
 - 缺点：代码工作量很大，尤其是服务器侧，需要判断时间，增加队列等。