

# 杭州电子科技大学

## 硕士学位论文

题目：基于物联网技术的 SPD 实时监测系统设计

研究生 吴新虎

专业 电子与通信工程

指导教师 陈科明 副教授

完成日期 2021 年 3 月

杭州电子科技大学硕士学位论文

基于物联网技术的SPD实时监测系统设计

研 究 生： 吴新虎

指导教师： 陈科明 副教授

2021 年 3 月

**Dissertation Submitted to Hangzhou Dianzi University  
for the Degree of Master**

# **Design of SPD real time monitoring system based on Internet of things technology**

**Candidate: Wu Xin hu**

**Supervisor: Associate Prof. Chen Keming**

**March, 2021**

# 杭州电子科技大学

## 学位论文原创性声明和使用授权说明

### 原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名：吴新虎

日期：2021 年 3 月 12 日

### 学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署单位名称仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名：吴新虎

日期：2021 年 3 月 12 日

指导教师签名：陈明

日期：2021 年 3 月 12 日

## 摘 要

雷电是电子设备受损的主要原因之一，在需要保护设备的前端安装浪涌保护器（Surge Protection Device, SPD），能够在雷击环境下大幅降低被保护线路的瞬时过电压。当前众多的防雷项目中，SPD 使用数量大，安装位置复杂，导致人工检测维护 SPD 难度大、效率低、周期长、同时也会存在漏检、误检的风险，使得人们不能及时发现 SPD 的许多异常状态，存在风险。想要对相关异常进行及时处理，就需要准确、及时、稳定、全面的掌握 SPD 的实时运行状态，进行相应的修复工作，可见检测 SPD 的工作状态意义重大。

本文针对目前防雷技术的缺点，设计了一种基于物联网技术的 SPD 实时监测系统。通过自主设计的 LoRa 无线组网技术将 SPD 监测设备的信息实现无线采集，进行实时检测。通过 OpenWrt 路由器操作系统，将设备接入到互联网，向云平台上传 SPD 工作状态信息，实现对历史数据的存储。同时设计了云平台，后台和 WEB 客户端，实现对系统的远程控制和管理。将 SPD 监控设备的信息实时的显示在前端页面上。当设备产生异常时，进行自动报警推送。该监测系统具有实时性和记忆性的特点。

为了实现更大范围的 SPD 设备的监测，本文参考 IEEE802.15.4 协议和结合 LoRa 无线传感器的特性，提出一种基于时隙 CSMA/CA 算法的 LoRa 无线网络的设计。首先对退避算法的过程进行分析优化，再结合了 LoRa 的特性，对 MAC 协议进行设计，最后对无线组网的网络层和应用层进行设计。测试结果表明，该设计网络通信延迟小于 0.2S，楼宇内的水平通信范围为 80 米，垂直距离为 2 层楼高度，空旷户外测试距离 1300 米，户外有遮挡物的情况下可达 250 米。多节点定时上报周期为 4S 时，丢包率低于 0.5%，符合设计指标的要求。

为了提高系统采集效率和稳定性，本文对 SPD 监测系统数据进行链路分析，采用多层协作的方式。数据采集层利用了 modbus 的特点，提高了对 SPD 监控设备的查询效率。通过设备分组和制定相应的帧格式，降低发送的频率。数据传输层采用 OpenWrt 操作系统，网络情况不稳定时，设计断网续传的功能，同时实行定时差异化上报和汇聚上报，减少通信过程中冗余数据的开销。数据服务层，设计了设备替换和远程更新的机制。测试表明，SPD 数据变化时，2.5S 之内会将变化的数据推送到 WEB 客户端，对比透传方式下，数据流量节省了 70%。系统满负荷的情况下可以保存 10min 的离线数据。网关 http 要比 mqtt 的更新方式，效率提升 6 倍，符合设计指标的要求。

**关键词：**实时 SPD 监测、OpenWrt、LoRa 无线组网、CSMA/CA、分组上报。

## ABSTRACT

Thunder and lightning are one of the main reasons for the damage of electronic equipment. Installing Surge Protection Device (SPD) at the front end of the protection equipment can greatly reduce the instantaneous overvoltage of the protected line in the lightning environment. At present, in many lightning protection projects, SPD is used in a large number and its installation position is complex, which leads to the difficulty of manual detection and maintenance of SPD, low efficiency and long cycle. At the same time, there are also risks of missing detection and false detection, so that people can't find any abnormal conditions of SPD in time, which is risky. In order to deal with the relevant anomalies in time, it is necessary to grasp the real-time running condition of SPD accurately, timely, stably and comprehensively, and carry out the corresponding repair work. It is disputable that the detection of SPD working condition is of great significance.

According to the shortcomings of the current thunder and lightning protection technology, we have designed an SPD real-time monitoring system based on the Internet of Things technology in this article. The self-designed LoRa wireless networking technology is used to wirelessly collect the information of SPD monitoring equipment for real-time detection. The device is connected to the Internet, and the SPD working condition information is uploaded to the cloud platform to realize the storage of historical data by the means of the OpenWrt router operating system. At the same time, the cloud platform, background and web client are designed to realize the remote control and management of the system. The information of SPD monitoring equipment is displayed on the front page in real time. When the equipment is abnormal, it will send the alarm automatically. The monitoring system has the characteristics of real-time and memorable.

In order to achieve a wider range of SPD equipment monitoring, we propose a design of LoRa wireless network based on time slot CSMA / CA algorithm, referring to IEEE802.15.4 protocol and combining the characteristics of LoRa wireless sensor in this paper. First of all, the process of back-off algorithm is analyzed and optimized, and then the MAC protocol is designed based on the characteristics of LoRa. Finally, the network layer and application layer of wireless network are designed. The test results show that the network communication delay of the design is less than 0.2S, the horizontal communication range in the building is 80 meters, the vertical distance is the height of 2-story building, the distance in the open outdoor test is 1300 meters, and the outdoor test distance can reach 250 meters when there are obstacles. When the report period of multi node is 4S,

the packet loss rate is less than 0.5%, which meets the requirements of the design indicator.

In order to improve the collection efficiency and stability of the system, we have analyzed the data link of the SPD monitoring system by adopting multi-layer cooperation mode. The data acquisition layer uses the characteristics of modbus to improve the query efficiency of the SPD monitoring equipment. The frequency of transmission is reduced by grouping the devices and formulating the corresponding frame format. The Openwrt operating system is adopted in the data transmission layer. When the network condition is unstable, the function of continuous transmission after network interruption is designed. At the same time, timing differential reporting and aggregation reporting are implemented to reduce the cost of redundant data in the communication process. In the data service layer, the mechanism of device replacement and remote update is designed. The test shows that when the SPD data changes, the changed data will be pushed to the WEB client within 2.5s. Compared with the transparent transmission mode, the data flow is saved by 70%. Under the condition of full load, the system can save 10 minutes of offline data. The gateway http is six times more efficient than mqtt and meets the design criteria.

**Keywords:** Real time SPD monitoring, OpenWrt, LoRa wireless networking, CSMA/CA, packet reporting

## 目 录

|                                     |    |
|-------------------------------------|----|
| 摘 要 .....                           | I  |
| ABSTRACT .....                      | II |
| 第 1 章 绪论 .....                      | 1  |
| 1.1 研究背景和意义 .....                   | 1  |
| 1.2 SPD 监测相关技术现状.....               | 2  |
| 1.2.1 SPD 监测系统的基本原理 .....           | 2  |
| 1.2.2 SPD 监测相关技术研究现状.....           | 2  |
| 1.2.3 无线通信技术的发展现状.....              | 3  |
| 1.2.3.1 LoRa 通信的介绍.....             | 4  |
| 1.2.3.2 无线网络通信技术.....               | 4  |
| 1.2.3.3 无线网络拓扑结构.....               | 6  |
| 1.3 本文主要研究的内容及章节安排.....             | 6  |
| 第 2 章 基于物联网技术的 SPD 实时监测系统的详细设计..... | 8  |
| 2.1 系统总体设计 .....                    | 8  |
| 2.1.1 需求分析 .....                    | 8  |
| 2.1.2 总体框架.....                     | 8  |
| 2.2 设备总体结构设计 .....                  | 9  |
| 2.2.1 网关硬件设计.....                   | 9  |
| 2.2.2 网关软件设计.....                   | 11 |
| 2.2.3 LoRa 节点硬件设计 .....             | 13 |
| 2.2.4 LoRa 节点软件设计 .....             | 14 |
| 2.2.5 SPD 监测设备硬件设计 .....            | 15 |
| 2.2.6 SPD 监测设备软件设计 .....            | 17 |
| 2.3 应用层软件设计 .....                   | 18 |
| 2.3.1 云平台设计 .....                   | 18 |
| 2.3.1.1 设备云 .....                   | 18 |
| 2.3.1.2 业务云 .....                   | 19 |
| 2.3.2 前端设计 .....                    | 20 |
| 2.3.3 网关管理平台设计 .....                | 21 |

|                                              |    |
|----------------------------------------------|----|
| 2.3.4 系统展示 .....                             | 21 |
| 2.4 本章小结 .....                               | 26 |
| 第 3 章 基于时隙 CSMA/CA 算法的 LoRa 无线网络设计 .....     | 27 |
| 3.1 基于时隙 CSMA/CA 算法公平性的改进及实现 .....           | 27 |
| 3.1.1 传统时隙 CSMA/CA 算法的原理 .....               | 27 |
| 3.1.2 传统时隙 CSMA/CA 算法分析 .....                | 28 |
| 3.1.3 基于时隙 CSMA/CA 算法的公平性改进 .....            | 30 |
| 3.2 基于时隙 CSMA/CA 算法的 LoRa 无线网络 MAC 层设计 ..... | 32 |
| 3.2.1 基于时隙 CSMA/CA 算法 LoRa 的时隙选取 .....       | 33 |
| 3.2.2 LoRa 无线网络 MAC 层帧格式设计 .....             | 34 |
| 3.2.3 LoRa 无线网络的信道监测机制比较分析 .....             | 35 |
| 3.3 基于时隙 CSMA/CA 算法的 LoRa 无线网络的总体设计与实现 ..... | 36 |
| 3.3.1 LoRa 无线网络的拓扑结构 .....                   | 36 |
| 3.3.2 SWSN 无线通信协议设计 .....                    | 37 |
| 3.3.3 通信流程的设计 .....                          | 39 |
| 3.4 基于时隙 CSMA/CA 算法的无线网络性能测试 .....           | 41 |
| 3.4.1 算法的仿真与性能分析 .....                       | 41 |
| 3.4.2 多节点定时上报测试 .....                        | 43 |
| 3.4.3 系统检测机制测试和通信距离测试 .....                  | 45 |
| 3.4.3.1 信道检测机制测试 .....                       | 45 |
| 3.4.3.2 不同的 SF 和带宽网络通信总耗时 .....              | 46 |
| 3.4.3.3 不同楼层下的通信距离测试 .....                   | 47 |
| 3.4.3.4 测试户外广阔地面的通信距离 .....                  | 49 |
| 3.4.3.5 测试户外有遮挡物的通信距离 .....                  | 50 |
| 3.5 本章小结 .....                               | 51 |
| 第 4 章 多层式协作提高系统的稳定性和效率性 .....                | 52 |
| 4.1 数据采集层处理机制 .....                          | 53 |
| 4.1.1 数据连读策略 .....                           | 53 |
| 4.1.2 设备分组和信息混合策略 .....                      | 55 |
| 4.1.3 应用层中的帧格式设计 .....                       | 56 |
| 4.2 数据传输层的处理机制 .....                         | 57 |
| 4.2.1 数据主动上报策略 .....                         | 57 |
| 4.2.2 数据断网续传策略 .....                         | 58 |
| 4.2.2.1 时间同步机制 .....                         | 58 |

|                               |    |
|-------------------------------|----|
| 4.2.2.2 离线缓存队列 .....          | 58 |
| 4.2.3 网关离线检测和重连策略 .....       | 59 |
| 4.2.4 数据汇聚上报策略 .....          | 59 |
| 4.3 数据服务层的处理机制 .....          | 60 |
| 4.3.1 设备替换机制 .....            | 61 |
| 4.3.2 远程升级机制 .....            | 62 |
| 4.3.2.1 基于 MQTT 的远程更新机制 ..... | 62 |
| 4.3.2.2 基于 http 远程更新机制 .....  | 62 |
| 4.4 实验测试与结果分析 .....           | 63 |
| 4.4.1 多层协作上报测试 .....          | 63 |
| 4.4.2 数据连读功能测试 .....          | 65 |
| 4.4.3 数据汇聚上报测试 .....          | 66 |
| 4.4.4 数据离线缓存上报测试 .....        | 67 |
| 4.4.5 远程更新性能比较分析 .....        | 69 |
| 4.4.6 网关长时间在线监测 .....         | 70 |
| 4.4.7 系统整体测试 .....            | 71 |
| 4.4.7.1 数据推送实时性测试 .....       | 71 |
| 4.4.7.2 系统稳定性测试 .....         | 72 |
| 4.5 本章小结 .....                | 74 |
| 第 5 章 总结与展望 .....             | 75 |
| 5.1 总结 .....                  | 75 |
| 5.2 展望 .....                  | 76 |
| 致谢 .....                      | 77 |
| 参考文献 .....                    | 78 |
| 附 录 .....                     | 81 |

# 第1章 绪论

## 1.1 研究背景和意义

雷电天气下，为我们提供电力来源的电气设备受到了很大的威胁。一系列由雷击引发的电气设备故障问题，在对人们的人身安全造成威胁的同时，一般还会产生很大的经济损失。在雷击过程中会产生雷电过电压，其会对相应的电气设备造成严重的损害。我国境内时常发生雷击灾害，对建筑物及其内部的电气设备造成极大损害<sup>[1]</sup>。虽然大部分电气设备安装在建筑物内部，遭受直击雷的可能性比较小，但是雷击的电磁效应是造成电气设备故障的关键问题之一。当建筑物、建筑物附近及与建筑物相连的线路被雷击中时，会产生雷击电磁波脉冲，它会通过多种方式和途径入侵至建筑物内部，其中包括辐射、传导以及耦合等方式，造成有关设备的损坏，同时对人身财产安全产生极大威胁<sup>[2-4]</sup>。每年因雷电导致的经济损失呈上升的趋势<sup>[5-6]</sup>。以浙江省为例，在 2012 年至 2017 年中省内共发生 3391 起雷电灾害事件(28 起重大雷电灾害，268 起较大雷电灾害，3095 起一般雷电灾害)。其中，造成 129 起重大事故，死亡人数为 36 人，受伤 35 人。2539 件电子设备遭受损坏，24357 件公用设备遭受损坏，雷击导致建筑物损坏 268 次，累计造成的直接经济损失高达 16056.99 万元<sup>[7]</sup>。浙江省雷灾受损行业统计如下表 1.1 所示。

表 1.1 浙江省雷灾行业统计

| 受损行业       | 雷灾次数/次 | 经济损失/万元 |
|------------|--------|---------|
| 制造业        | 915    | 6578.42 |
| 通信         | 295    | 610.77  |
| 公共管理与社会组织  | 135    | 432.99  |
| 广电         | 119    | 734.40  |
| 城市住宅       | 79     | 120.17  |
| 旅游业、住宿和餐饮业 | 79     | 528.56  |
| 农居用户       | 845    | 2774.95 |
| 石油化工       | 71     | 584.59  |
| 其他         | 427    | 2292.35 |

由上表可以看出，雷灾影响着各行各业，给公民的人身安全和经济财产造成威胁，也给电气设备造成损害。下面列举两个近年来具体案例：

河南省某高速公路的收费站遭遇雷击，收费大厅中的 UPS、监控系统和电脑设备均被损坏，导致高速路严重堵塞，造成 60 多万的经济损失。

海南省某市的粮食仓库遭遇雷击，仓库内部的监控系统和电子设备均瘫痪，造成经济损失达 55 万余元。

这两起事故发生的原因就在于雷击发生时损坏了低压电气，因为雷击发生时，产生了电磁脉冲，而电磁脉冲会对电气设备造成损坏。SPD 能够对相关电气设备起到保护作用，使其免受电磁脉冲的侵害。中华人民共和国国家标准《建筑物电子信息系统防雷技术规范》<sup>[8]</sup>中规定：对建筑物防闪电侵入和高电位反击做出了具体的防护措施，必须严格按照条令执行对 SPD 进行相关设置。由此可见在防雷措施中浪涌保护器的重要性。

目前，我国浪涌保护器产业仍然处于“小、散、多”局面，多以小企业形象出现，没有主导企业<sup>[9]</sup>。浪涌保护器广泛应用于用电场合，和人们的生命财产安全息息相关，因此对浪涌保护器的实时监测就显得尤为重要，需要通过监测，实时了解浪涌保护器的工作状态和运行信息，从而对电气设备采取及时有效的防雷措施。SPD 是防雷系统中的关键器件，SPD 经过长时间的使用或者遭受雷击，会发生损坏、老化等现象，会直接影响到防雷的效果。SPD 监测系统的设计，可以有效的阻止可能出现的雷电危害途径，降低设备受损的风险。较多电气设备安装 SPD 时，会给检测人员带来繁重的工作量。当负责检测的人数很少或其他条件的限制下，相关单位检测时间可能是每半年或一年检测一次，很难及时发现 SPD 设备出现故障。只有下一次雷灾事件发生时，才反应过来，可见 SPD 监测系统的意义重大。

## 1.2 SPD 监测相关技术现状

### 1.2.1 SPD 监测系统的基本原理

对于 SPD 的检测，采集 SPD 的工作 SPD 失效、前端保护、温度、雷击次数、空开等关键参数。其工作原理为<sup>[10-11]</sup>：采集这些信息经过数模转换和信号处理交由相应的 MCU 进行分析，相应的 MCU 对采集到的数据进行处理，继而进行逻辑判断，最后进行显示，报警和通信等功能。结构框图如图 1.1 所示。

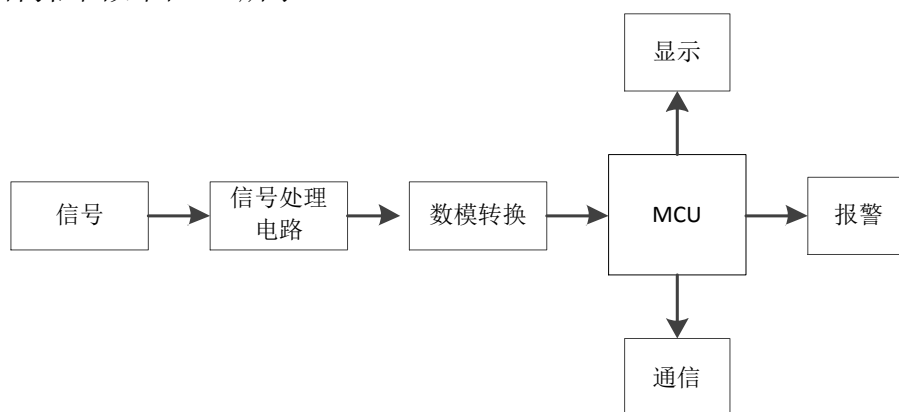


图 1.1 信号采集框图

### 1.2.2 SPD 监测相关技术研究现状

在人们的日常生活中，传统的 SPD 检测与定位仍然依赖人工判断，此种检测方法受人员

数量和经验影响比较大，会出现漏检、误检的状况。此种方法在检测过程中不仅浪费时间，而且实时性得不到保证。当发现故障和排除问题不及时的情况下，会造成经济损失甚至是安全事故。

文献[12]通过分析 SPD 中 MOV 器件的工作原理，制作了相应的硬件平台，编写了相应的程序，分析 SPD 脱扣点的温度和相关参数判断 SPD 性能的优劣。

文献[13]电涌保护器智能监测系统，这里主要是使用高速 AD 采样，数据缓存区电路是由 FIFO 寄存器和锁存器的组成，采集泄流数据的工作由单片机来完成，采集到的泄流数据点亮相对应的 LED 灯，对 SPD 的寿命进行显示。同时制作相应的 labview 上位机，具有显示采集数据和用户进行交互的功能，数据保存在本地。

文献[14]通过对中山市重要场所安装的 SPD 作为研究对象，采集 SPD 的各个参数，并对参数进行比较分析和相应参数的监测方法，系统稳定，且误差比较小，有报警和数据的存储的功能，但是监测范围比较局限。

文献[15]采用 zigbee 技术，实现了对高压避雷器的遥测，采集相应的数据，将数据进行上报相应的服务器。

文献[16-18]关于智能 SPD 的监控、数据采集、组网技术等方面提供一些新思路和相应的策略。

上述的 SPD 的监测技术，从理论到平台搭建，有本地的，有基于有线组网的，有基于无线 zigbee 的。考虑到对成本、用户体验、实现系统的准确性等需求，需要考虑相应的物联网技术，查询和监控更多的 SPD，使信息可以集中化，为接下来的防雷系统设计提供一定的数据分析。

### 1.2.3 无线通信技术的发展现状

当今时代，无线通信技术快速发展，物联网（IOT）技术在我们的工作和生活中发挥着越来越重要的作用。物联网是指设备上的各类传感器，比如温湿度、压力、气体等，将采集到的信息，通过二维码、条形码、射频识别、激光扫描等方式，与互联网技术相结合而形成的通信网络<sup>[19-20]</sup>。在无线技术的发展过程中，出现了移动、微波、卫星、短波通信等许多技术。当前无线组网通信，传统的无线组网技术已经无法满足人们对远距离、大容量、低功耗的要求。低功耗广域网 LPWAN（Low Power Wide Area Net Work）应运而生。LPWAN 解决了传统无线通信技术中的问题，使人们对远距离、大容量和低功耗的需求变为现实。最具有代表性的是 NB-IoT 和 LoRa。主要区别如下表 1.2 所示，可以得出，两种技术都能够实现远距离通信，具有较大的网络容量且功耗低。对比于前者，LoRa 具有更加成熟、成本低的等特点。

表 1.2 LoRa 与 NB-IoT 性能比较

|  | LoRa | NB-IoT |
|--|------|--------|
|--|------|--------|

续表 1.2

|          |                     |                 |
|----------|---------------------|-----------------|
| 频段 (MHZ) | 433、868、915         | 850、900, 授权频段   |
| 传输速率     | 250bps-21Kbps       | 160Kbps-250Kbps |
| 网络拓扑     | 自组网                 | 蜂窝网络            |
| 网络容量     | 10 万个               | 20 万个           |
| 距离       | 市区 1-4Km, 宽阔地带 16Km | >10Km           |
| 设备成本     | 较低, 易维护             | 适中              |
| 通信成本     | 低                   | 较高, 付费给运营商      |
| 功耗       | 低, 基于 ALOHA 协议      | 较高, 设备需要定期联网    |
| 发展现状     | 较成熟, 很多国家已经应用       | 2016 年公布, 待完善   |
| 应用       | 物联网农业、物联网抄表         | 物联网家居、共享业务      |

### 1.2.3.1 LoRa 通信的介绍

LoRa(Long Range), 是一种基于扩频技术的新兴无线通信方式。具有远距离、容量大、低功耗、可靠性高的特点。LoRa 联盟公布的数据表明, 现在全面构建 LoRa 网络的国家已有 9 个, 进行 LoRa 网络试点的国家有 56 个<sup>[21]</sup>。LoRa 采用的扩频调制技术, 有效信号低于信道底噪强度时, 也可以进行数据的传输。在相同的信道环境下, 相比于 FSK 的调制技术, 部署 LoRa 调制所提供的总信道容量增长了约 50%。在智能家居和智慧工厂, 可以实现快速部署, 广泛应用于数据量较小系统中<sup>[22]</sup>。LoRa 可调控带宽和频率, 使得其拥有多个频段, 可以配置多种发射速率, 使开发者可以根据实际情况进行灵活的使用, 降低开发难度<sup>[23]</sup>。LoRa 的网络架构如下图 1.2 所示。搭载相应的互联网的技术, 可以将传感器中的数据上报给服务器<sup>[24]</sup>。

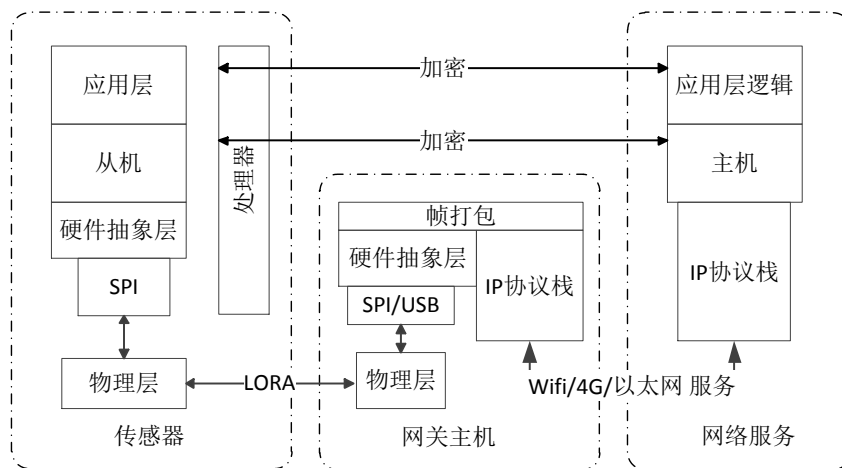


图 1.2 LoRa 网络架构

### 1.2.3.2 无线网络通信技术

在设计无线网络时, 要针对项目的背景, 进行无线网络的通信方式的选择, 是构成一个

可靠稳定网络的基石。通过不同的技术特性，可以将相应的无线通信技术按照频段、无线协议、传输速率、传输距离等标准进行划分。常见的无线通信技术如表 1.3 所示<sup>[25]</sup>。

表 1.3 常见的无线通信技术

|           | 主要频段/HZ        | 网络节点数 | 最大传输速率/bps | 无线协议           | 网络类型 | 调制方式    | 最大传输距离/室外 | 最大传输距离/室内 | 成本 | 应用   |
|-----------|----------------|-------|------------|----------------|------|---------|-----------|-----------|----|------|
| ZigBee    | 2.4G           | 65536 | 250k       | IEEE802        | mesh | GFSK    | 500m      | 20m       | 低  | 室内遥控 |
| WiFi      | 2.4G/5G        | 32    | 54M        | IEEE802        | 星型   | DSSS    | 50m       | 15m       | 较高 | 网络通信 |
| Bluetooth | 2.4G           | 7     | 720k       | IEEE802        | 星型   | GFSK    | 50m       | 5m        | 中等 | 文件传输 |
| Z-Wave    | 868M/915M      | 232   | 40k        | 私有协议           | mesh | GFSK    | 1km       | 30m       | 低  | 家居   |
| LoRaWAN   | 433M/470M/868M | 10000 | 21.8k      | LoRaWAN 国际联盟标准 | 星型   | LoRa 扩频 | 20km      | 100m      | 高  | 抄表   |

上表中所诉是目前常用的几种无线通信技术。针对不同的使用场景，选择不同的通信技术。其中 WiFi, Bluetooth, ZigBee 都应用于 2.4G 频段<sup>[26]</sup>，存在同频干扰，且都是基于 IEEE802 的协议。WiFi 多用于家庭和办公室以及公共场合，为手机、电脑提供联网服务，传输速率高，但是通信范围有限<sup>[27]</sup>，不适合室外的应用。Bluetooth：多用于移动设备、手机之间一对一或一对多的无线数据交换，传输距离较短，传输速率适中<sup>[28]</sup>。ZigBee：相比前两者，其具有低功耗的特点<sup>[29]</sup>，广泛使用在智能家居和仓储物流等领域，在可靠性，网络容量等有一定优势，但穿墙能力不如 LoRaWAN 工作频段。在室内室外的有通信要求的应用中，ZigBee 不占在优势，本系统中不推荐使用。Z-Wave 频段在国外使用的比较多<sup>[30]</sup>，但是在国内不能够直接使用，因为该频段未授权，所以不能应用在本系统中。LoRaWAN<sup>[31-32]</sup>国内的工作频段为 470M，LoRaWAN 协议能够实现远距离通信，具有较大的网络容量以及很强的抗干扰能力，在物理层中采用的是 LoRa 扩频调制方式，但是在 MAC 层采用异步 ALOHA 协议，信道利用率很低。研究表明<sup>[33]</sup>，在纯 ALOHA 协议中，其信道利用率最大仅有 18.4%，多个节点设备同时向网关发送数据时，会有很大的概率产生数据碰撞，为了数据的可靠传输，需要进行数据的重发，不仅影响了通信系统的稳定性，还浪费了信道的资源，导致了网络延时的增加。另外

LoRaWAN 部署成本较高，所以 LoRaWAN 也不适用于本系统。

### 1.2.3.3 无线网络拓扑结构

合适的网络拓扑结构可在保证信号传输稳定性的同时，拓展网络的传输距离。当前应用最广泛的几种网络拓扑结构包括：网状、树状、星型三种。如下图 1.3 所示。

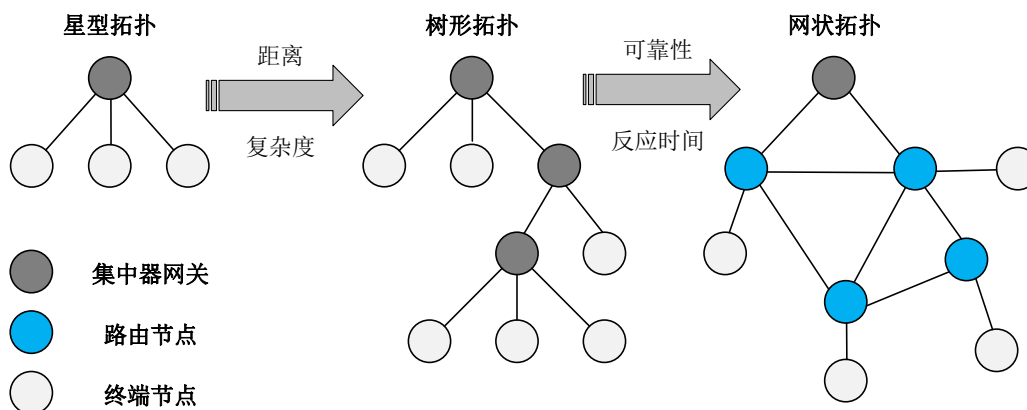


图 1.3 常用的网络拓扑结构

从上图可看出，星型网络拓扑结构中，各个节点与集中器网关相连接，也就决定了所有终端节点只能通过集中器网关进行通信，节点之间不能直接进行通信，优点是：结构简单、配置灵活，延时小；缺点是：网络共享能力较差，节点之间通信需要依靠网关，而且当网络中存在较多的节点设备时，会加重集中器网关处理的负担。树状网络拓扑结构本质上是对星型网络的拓展，当系统中存在多个星型网络时，集中器网关与总基站进行通信实现数据交互。想要扩大网络的通信范围，在网络中加入网关即可。网状网络拓扑结构也叫“多跳”网络，终端节点不仅能够做接入点（AP）也能够作为路由器，各个节点之间都可以直接进行通信，各节点感知网络状态的变化，可以自身进行维护，其稳定性较强。

## 1.3 本文主要研究的内容及章节安排

本文分析了现有 SPD 监测系统不足，设计了基于物联网技术的 SPD 实时监测系统，实现对楼宇、户外 SPD 的状态信息监测。通过 LoRa 无线组网的技术将设备进行互联和系统集成。通过智能网关将 SPD 的设备信息接入互联网。设计了云平台、前端客户端和网关管理客户端，实现了对设备运行状态的实时监测，满足用户对智能化生活的需求。本文基于时隙 CSMA/CA 算法进行研究，对 LoRa 无线网络进行设计。由于原有的时隙 CSMA/CA 算法，在通信的过程中，多个节点访问信道，首先访问成功的节点，再次参与竞争时，在接入信道的过程中具有较高的优先级，存在不公平性，针对该问题，对原有的退避算法进行分析，同时结合 LoRa 无线通信的技术，设计了 MAC 层、网络层和应用层协议。通过仿真和实物测试表明，对该算法的改进可以使 LoRa 无线网络中的节点接入信道的更为公平。针对本项目的特点，在小区，或则高层建筑物里，浪涌保护器的数量众多，采集的数据点的任务就会变得繁重。当设备较多时，网关采用时间片轮询和透传的方式，实时性就会打折扣。较多的数据需

要上报时，需要考虑流量资费和系统采集稳定性的问题。在数据的采集层，数据的传输层和数据的服层<sup>[34]</sup>进行优化，提高信息的实时性和系统的稳定性。

本章的内容安排如下：

第一章：绪论，对基于物联网技术的 SPD 实时监测系统设计的意义和研究背景进行详细介绍，分析 SPD 监测关键技术的研究现状和 SPD 监测的原理，最后对目前常用的无线通信技术进行分析讨论，分析了各个无线通信技术的优劣势，对 LoRa 技术进行介绍和相关的组网技术进行说明。

第二章：基于物联网的 SPD 实时监测系统的详细设计，首先进行系统总体设计，再对系统的各个部分进行介绍，包括智能网关、LoRa 节点、SPD 监测设备的硬件和软件的实现和应用层的设计，最后进行实物展示。

第三章：基于时隙 CSMA/CA 算法的 LoRa 无线网络设计。首先，对传统的时隙 CSMA/CA 算法进行分析，确定设计方向，结合了 LoRa 的特性，对时隙的 MAC 层的协议进行设计，再结合无线网络结构和本项目的特点对网络层和应用层进行设计，最后，测试无线网络的通信性能和通信距离。

第四章：多层式协作提高系统的稳定性和效率性。首先通过对本系统和传统的物联网网关存在的问题进行分析，结合本项目的特点，提出了多层协作的机制，数据服务层、数据采集层以及数据传输层进行优化，最后对相关优化和系统进行测试。

第五章：对全文的系统设计和关键技术研究加以总结，并对更深层次 SPD 实时监测系统提出建议和展望。

## 第2章 基于物联网技术的 SPD 实时监测系统的详细设计

前文分析了传统的 SPD 监测系统是如何设计与工作的，结合项目背景和现有技术，本系统中提出了相应的技术方案，下文展开描述。

### 2.1 系统总体设计

#### 2.1.1 需求分析

根据当前 SPD 监测系统的关键要素以及目前监测系统中存在的不足之处，对于需要设计的 SPD 监测系统，主要需求有以下部分：

(1) 改变传统人工检测方式，构建远距离小型无线网络通信，在楼层中要实现相应的 SPD 监测。在小区或者空旷的位置，实现户外 SPD 的监测，将 SPD 的设备的状态信息能准确上报到云平台。需要检测 3 层楼宇的 SPD 状态，户外的通信距离需要达到一千米，上报通信延时小于 3S。

(2) 相应的设备可以支持以太网、WiFi 和 4G 的联网方式，供安装人员进行选择。

(3) 开发配套的前端页面和相应的服务器，实时显示 SPD 状态信息，可查询相应的历史信息的数据，做数据报表分析，当产生报警信息时进行前端页面报警推送，可以短信告知相应的负责人，需要支持远程控制。

(4) 当需要更换设备时，可以通过前端页面操作，要实现人性化，便携化的原则。

#### 2.1.2 总体框架

基于以上的需求，对 SPD 监测系统进行设计，整体架构如图 2.1 所示，从下到上，可按功能进行区别，即数据采集层、传输层和服务层三个层次。

数据采集层是物理的最低层，主要是 LoRa 节点和 SPD 监测设备相连接，SPD 监测设备与 SPD 单元是硬件进行连接。可以采集 SPD 的状态信息，比如周围的环境温度、雷击次数、接地状态、前端保护状态、SPD 是否失效的状态等信息。

数据传输层是中间层，传输层主要由智能网关和 LoRa 节点，第一点：智能网关通过 wifi/以太网/4G 等联网方式接入云平台，将 SPD 的监测状态信息上报实时上报给云平台。第二点：智能网关对下挂载相应的 LoRa 板卡，与 LoRa 节点模块通过 SWSN 自主设计的组网的协议进行通信，相应的网络为星型结构网络。汇聚 SPD 的状态信息并进行上报，可以解决监测时布线困难的问题，实现 SPD 监控设备的统一管理。

数据服务层是最顶层，云服务器按功能可以分为设备云和业务云。设备云和智能网关建立连接，将上报的数据进行解析，并进行分类的储存。业务云具有具体的业务服务接口，外部可通过接入相应的 API 接口对其实现调用<sup>[35]</sup>。WEB 客户端和手机 APP 通过 HTTP 的请求，

调用数据云平台所提供的 API 接口来完成具体的应用<sup>[36]</sup>。同时可以访问网关上 Uhttpd 服务器来实现对网关的配置。

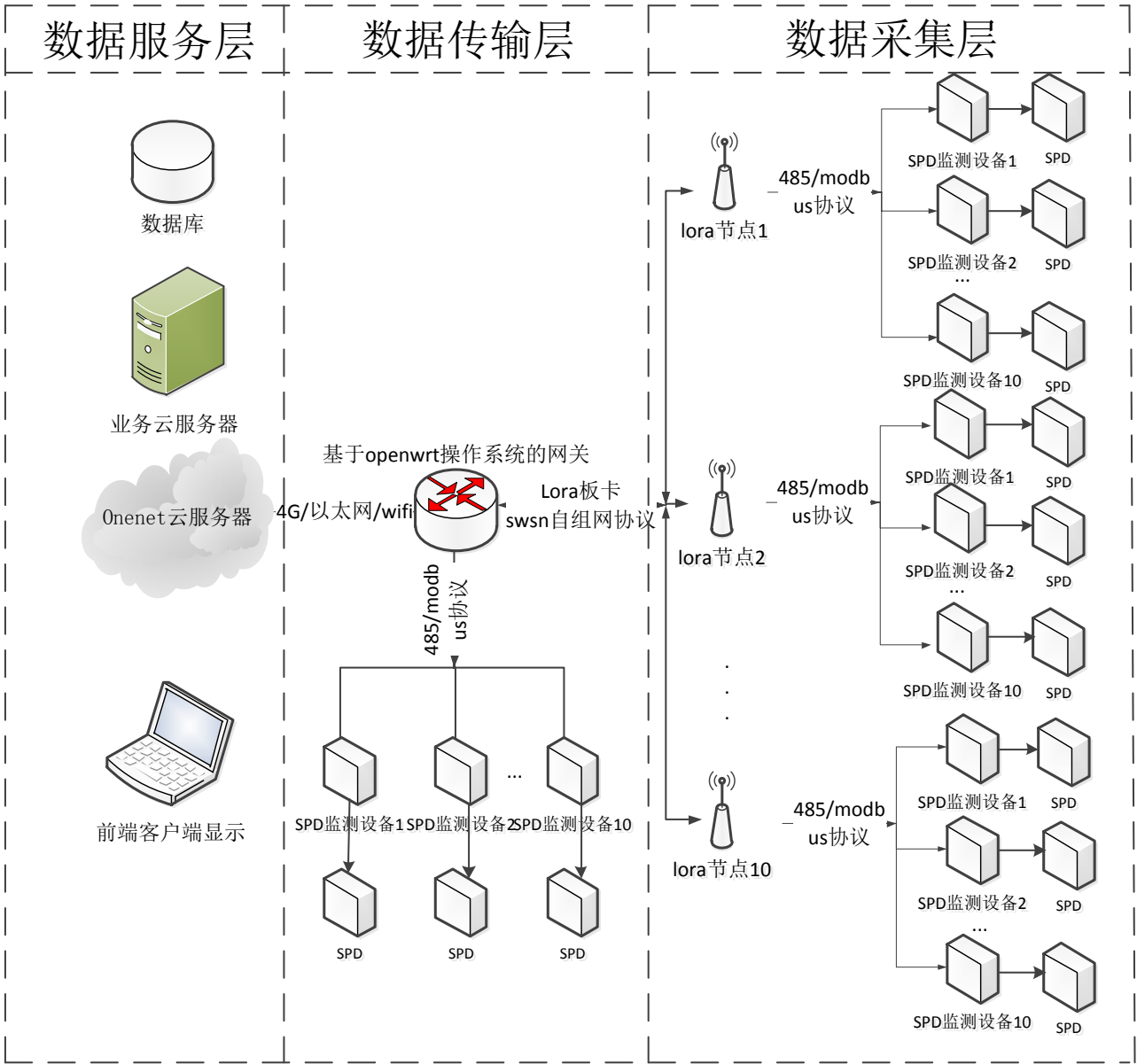


图 2.1 SPD 实时监测系统总体框图

## 2.2 设备总体结构设计

### 2.2.1 网关硬件设计

网关硬件功能总体框图如图 2.2 所示，网关硬件主要包括 AR9331、MCU、485/232、以太网和电源管理等模块。

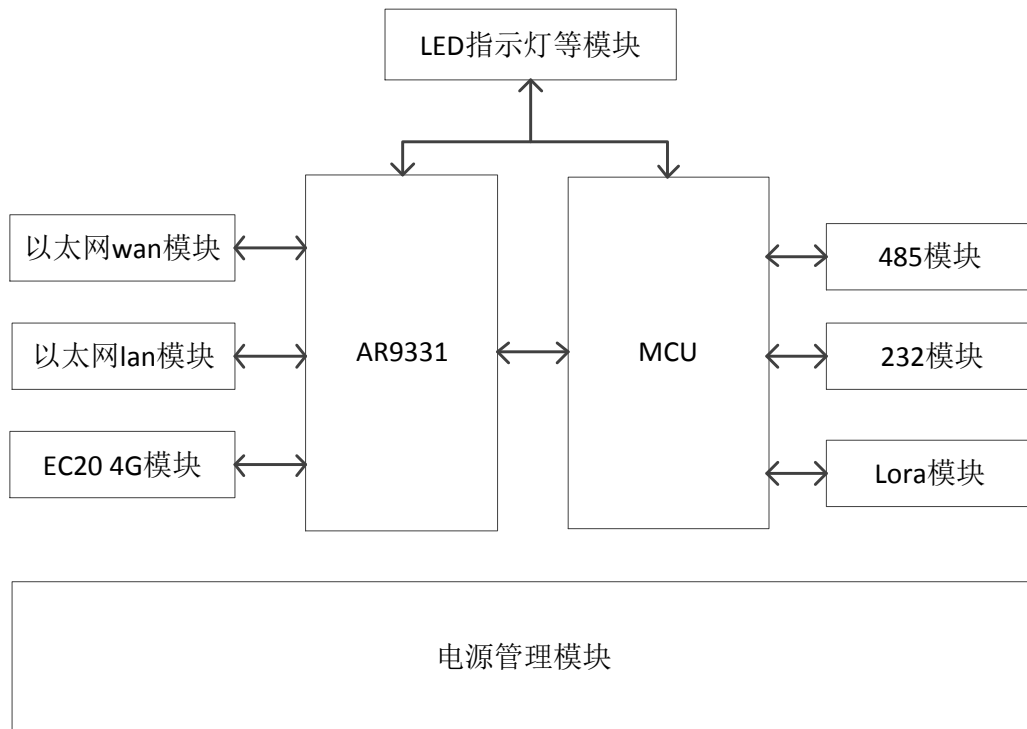


图 2.2 网关硬件总体框图

电源管理模块：24V 电源转 5V，给各种外设供电，比如隔离芯片、SP485、SP232 等。5V 转 3.3V，给 AR9331、MCU、4G 和 LoRa 等模块进行供电。

AR9331：是一款 2.4G 的 WiFi 芯片，具有 64MB 的 DDR2，16M 的 SPI FLASH。通过 USB 接口，挂载 EC20 的 4G 模块，插上物联网卡，实现 4G 拨号上网。连接两个 HR911102A 的水晶头接口，可以支持有线 lan 口和 wan 口的功能。具有一路 UART 通信，在该系统中，与 LoRa 板卡中的 MCU 通信，就是应用了该路串口。

MCU：这里应用 STM32F103RCT6 芯片，该芯片的片上资源丰富，Flash 容量较大，具有 SPI 通信、UART 通信和 DMA 控制器等。系统设计中用了 3 路串口，一路串口与 AR9331 进行交互，另外两个串口分别应用于 485 通信电路和 232 的通信电路中。LoRa 模块选择的是 SX1278 模组，MCU 采用 SPI 通信与 LoRa 模块进行交互。网关中的 LoRa 模块与节点上的 LoRa 模块实现无线通信。485 通信电路在本项目中可以通过有线连接相应的 SPD 监测设备，无节点的情况下可以进行单独检测。232 通信电路，是做拓展用的。

485/232 接口：这两个接口由于直接对接工业设备，所以分别在各自的电路当中加了对应的光耦隔离单元，增强抗干扰性。双向 TVS，保护芯片。

LED 指示灯等模块：共有 3 个指示灯，第一个是电源灯。电源上电，就会点亮。第二个是系统运行指示灯，当系统处于正常运行状态时会闪烁，其他情况系统中的主程序出现异常，需要排查问题。第三个是 4G 联网灯，快闪表示附着基站过程中，缓慢闪烁时表示附着成功，可以进行拨号上网。AR9331 与 MCU 之间不仅仅有串口，还有两路普通 IO 口相连接，用来

告知双方的工作状态。

LoRa 板卡上，有 2 个指示灯，第一个灯用来表示 LoRa 板卡的工作状态，闪烁表示系统正常运行。第二个灯在 LoRa 接收或发送数据时会进行亮灭。主要用来表示 LoRa 网络的状态。

### 2.2.2 网关软件设计

OpenWrt 是嵌入式 LINUX 发行版<sup>[37]</sup>，具有开源和应用广泛的特点。本系统所设计的网关，就是基于该操作系统进行编写，该操作系统常被用作路由器的开发，因此具有以太网和 WiFi 的联网功能，用户也可以自己裁剪，安装和编写相应的程序，实现对应的功能。在操作系统中进行了 4G 模块的挂载，实现了 4G 拨号上网的功能。自主编写了 Gateway\_SPD 的应用程序、网络选择、监测应用程序工作状态和错误收集等脚本。

Gateway\_SPD 应用层程序主要由以下部分构成，如下图 2.3 所示。主要由 4 个线程实现，OneNET 的数据收发线程、串口的数据收发线程、485 的数据收发线程和信息处理的线程组成，实现了图中的功能。

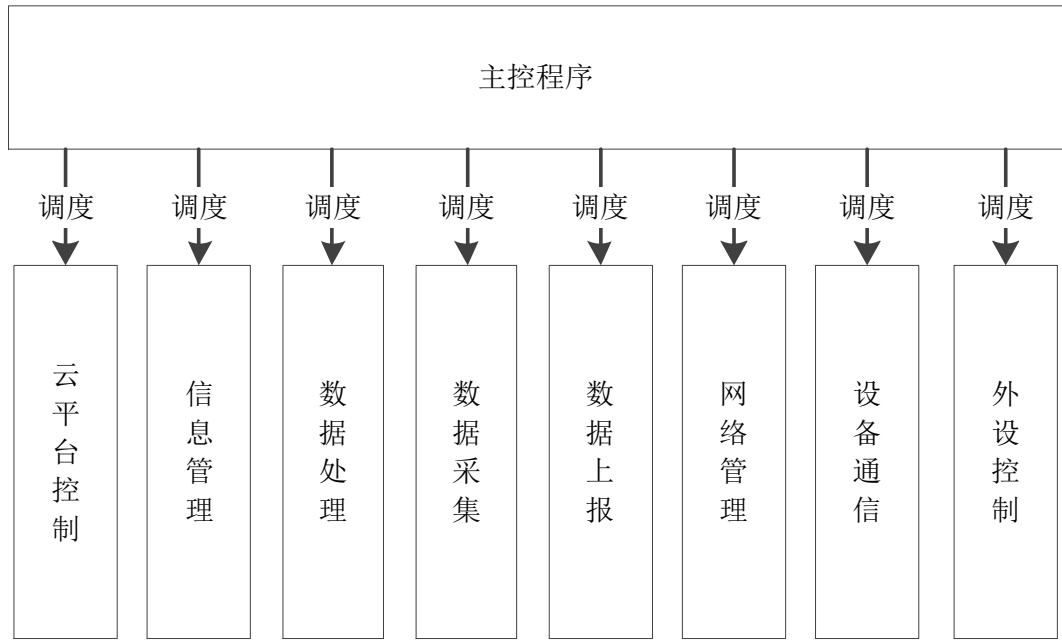


图 2.3 网关软件总体框图

网络选择处理的脚本，以及相应异常处理流程。判断网络是否连接成功，使用 ping 指令，去 ping 指定的域名服务器，中国移动、中国联通和中国电信通用的 DNS 是 114.114.114.114，也是本次设计中 ping 的 DNS。当需要对网络进行重启时，需要执行/etc/init.d/network restart 指令，进行重启网络的操作。具体如图 2.4 所示。

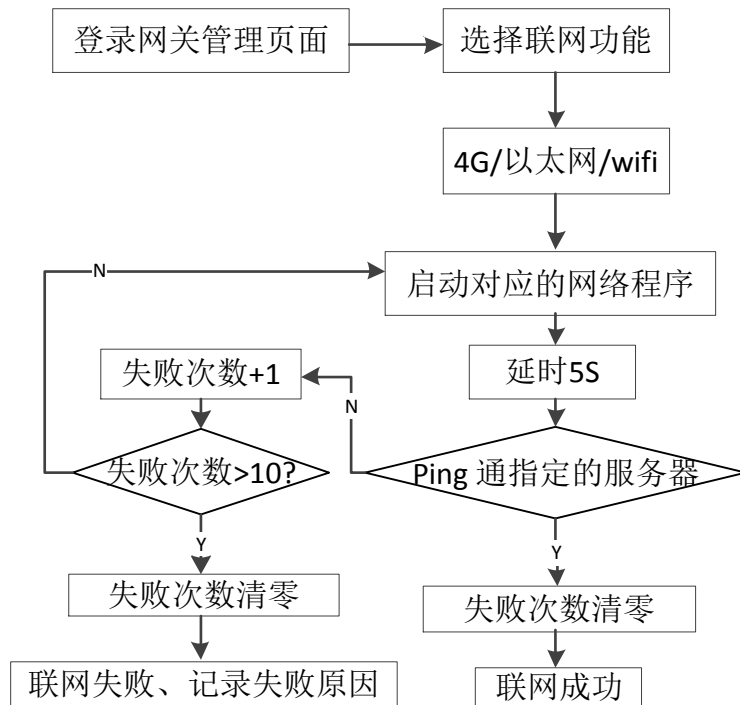


图 2.4 网关联网流程图

监测应用程序的脚本，执行 linux 中的 ps 命令，使用 awk 和 grep 查询字符串的指令去匹配当前运行的进程当中有没有 GatewaySPD 的字段。若有表示程序在正常运行中，没有需要重启 GatewaySPD 应用层软件。如图 2.5 所示。

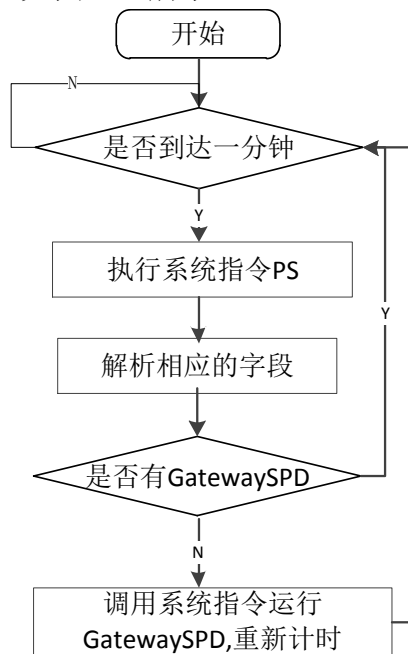


图 2.5 监测网关程序流程图

错误收集脚本，GatewaySPD 在运行过程中，会有相应的日志文件，当程序意外运行崩溃时，会产生 CORE 文件。当应用程序意外终止时，会触发相应的信号，错误脚本开始执行，

将日志文件和 CORE 文件打包成压缩包，使用文件传输协议上报给相应的后台。如图 2.6 所示。

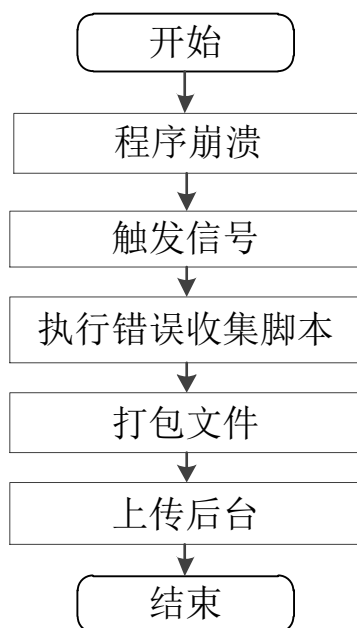


图 2.6 错误收集流程图

### 2.2.3 LoRa 节点硬件设计

节点硬件总框图如图 2.7 所示。主要包括电源管理模块、按键模块、485/232 接口等。

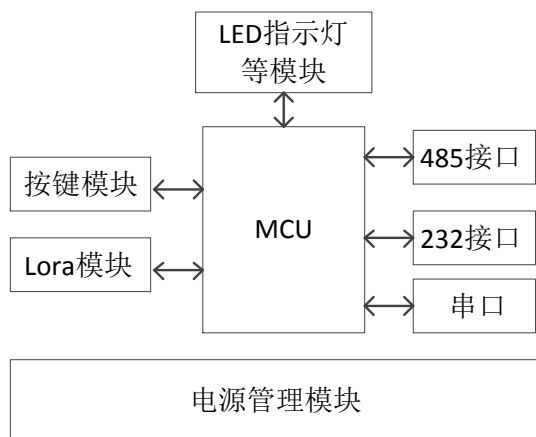


图 2.7 节点硬件总体框图

电源管理模块：24V 转 5V，5V 转 3.3V，给 MCU 和相关的外设进行供电。

MCU：这里选用的是 STM32F103C8T6，与网关中的 MCU 相比，外设资源、内存以及 FLASH 都大打折扣。性能没有区别，系统时钟扩频之后都可以到达 72M。满足本项目的需要，也节省了成本。

外设接口：MCU 的三路 UART 分别于 RS485 的通信电路，RS232 的通信电路以及 CH340 串口通信电路相连接。在本项目中 RS485 主要与 SPD 监测设备通信。CH340 的串口电路部分，主要用来配置节点设备，和调试作用。RS232 的通信电路可以用来做扩展。

**LoRa 模块：**与 MCU 的 SPI 引脚相连接，通过 SPI 时序对 LoRa 中的相关寄存器进行读取和配置。

**按键模块：**短按复位系统，长按恢复出厂设置。在测试的时候，可以修改按键功能为按下发送对应帧的功能。

**LED 指示灯等模块：**3 个指示灯，电源灯，上电状态指示。LoRa 组网连接灯，当联网成功后亮起。LoRa 数据收发灯，接收发送数据时会闪烁。

## 2.2.4 LoRa 节点软件设计

节点主要功能框图如下图 2.8 所示，包括了信息管理、信息采集、信息上报等功能。

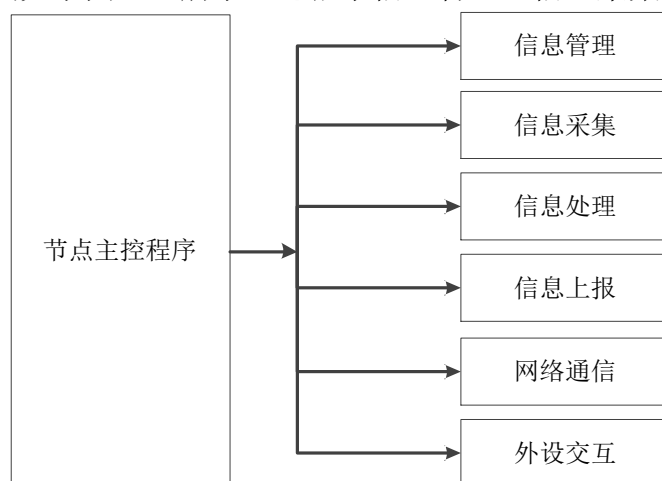


图 2.8 节点软件功能框图

网关执行添加设备和节点操作时。会将 SPD 监控模块栈号信息、所读寄存器的地址位和节点 MAC 地址进行下发。与之对应的 MAC 节点接收到相应的帧，进行帧解析，按照栈号，寄存器地址位和类型进行存储并开始查询对应的 SPD 监控模块。

节点信息链表的保存，链表中保存元素信息如下表 2.1 所示。信息有改动，就保存到 flash 中，节点设备上电自动加载。

表 2.1 节点保存的信息

|          |       |
|----------|-------|
| 设备栈号     | 1 个字节 |
| 设备的寄存器地址 | 2 个字节 |
| 设备的寄存器类型 | 1 个字节 |
| 当前寄存器的值  | 2 个字节 |
| 上一次寄存器的值 | 2 个字节 |

定时去查询 SPD 监控模块的信息，当信息发生变化的时候，将信息按照相应的帧格式上报给网关，或者当定时时间到的时候，实现定时数据上报。网关接受到帧后，会进行相应的回复。如果上报数据未得到回复，会进行数据的重发。具体流程图如图 2.9 所示。

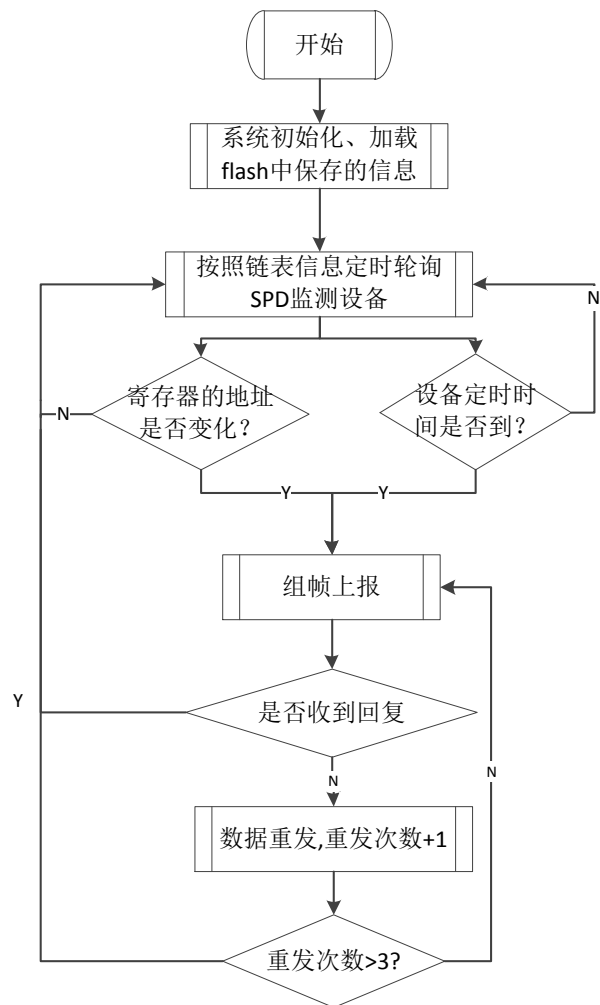


图 2.9 节点数据采集流程图

2.2.5 SPD 监测设备硬件设计

SPD 监测设备的硬件框图如图 2.10 所示，SPD 监测设备包括 MCU、数码管显示单元、LED/按键单元，接地监测单元、SPD 失效监测单元、485 接口单元、雷击检测单元、温度采集单元、前端保护器监测(SSD)单元<sup>[38]</sup>和电源管理模块。相关安装接线图，如图 2.11 所示。

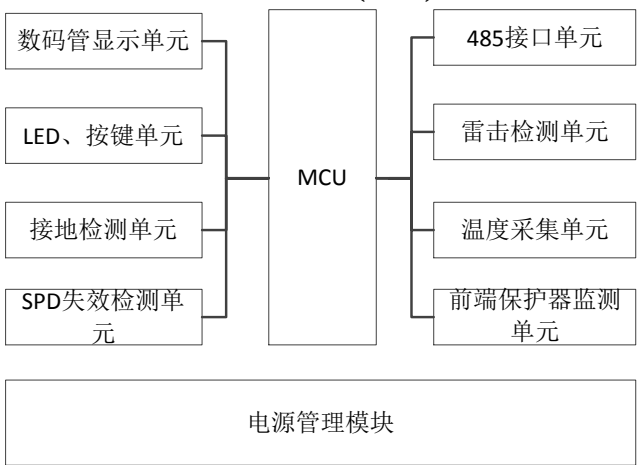


图 2.10 SPD 监测设备硬件框图

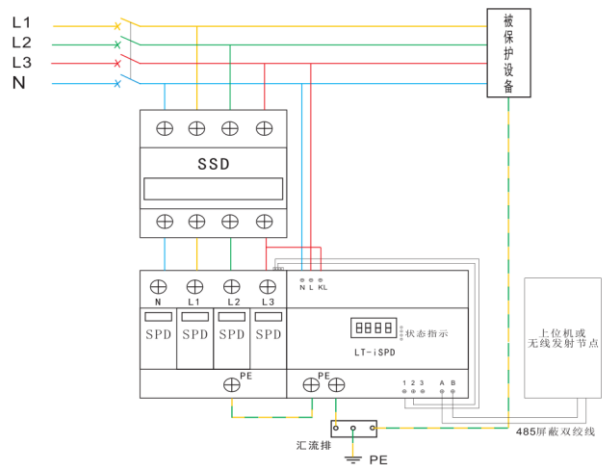


图 2.11 SPD 监测设备安装示意图

电源管理模块：是标准 220V 转 5V 的 AC-DC 电源。给 MCU 和其他外设进行供电。实际安装过程中，在设备的 N 和 L 端连接相应的 N 线和 L 线。

MCU：STC12C5A60S2，增强型 8051 单片机，8 路 ADC、10KRAM 和 20K 的 FALSH，满足项目的需求，也节省整个工程的成本。

数码管的显示单元和 LED\按键单元：是用来显示雷击次数和当前环境温度，和用户进行交互的部分，按下切换按键，切换显示雷击次数和温度。按下复位按键，清零雷击次数显示。

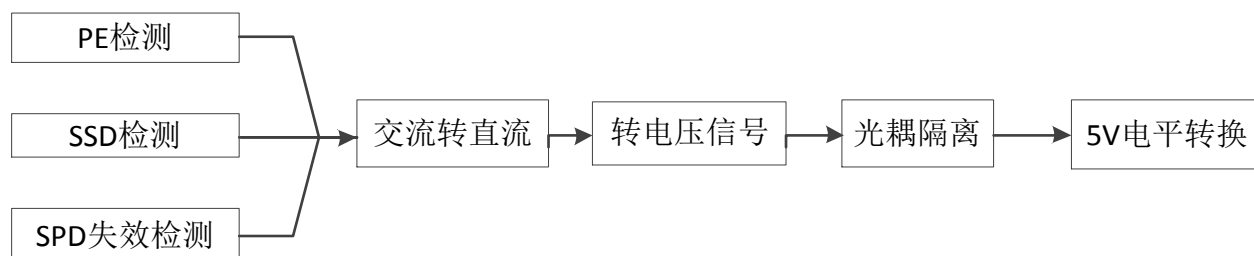


图 2.12 PE/SSD/SPD 失效检测框图

接地检测（PE）单元、SPD 失效检测单元、前端保护器（SSD）监测单元，监测原理相似，如图 2.12 所示。如安装图 2.11 所示，上述三个单元都接有用电端的一级。KL 连接 SSD 的一端，SPD 监测设备的 1、2 两个接口连接着 SPD。PE 检测单元与 SPD 的 PE 接口相连接。正常情况下，光耦隔离单元是导通的，MCU 的对应管脚检测为高电平，当出现 PE 连接异常、SPD 失效、前端保护器连接异常时，光耦截止，MCU 相应的检测管脚会检测到低电平，发出报警信号。

雷击检测单元：当产生雷击的时候 PE 接线部分，会有相应的电压信号，传输进入 SPD 监测设备当中，当监测到电压信号时，超过一定的伏值，光耦会导通，产生一个脉冲信号。MCU 进行捕获，雷击次数加一，具体如图 2.13 所示。

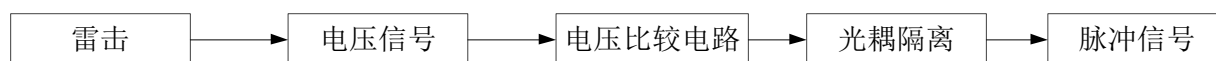


图 2.13 雷击检测单元框图

温度采集单元：如图 2.14 所示，本次设计中所选取的热敏电阻是负温度系数(NTC)的类型，用于检测周围环境的温度。温度越高，热敏电阻的阻值就越低。NTC 热敏电阻不但测温灵敏，价格也便宜，在生产生活中得到大量应用<sup>[39]</sup>。主要采用分压放大电路，使用电容滤波之后，送入单片机的 ADC 的单元进行数据采集，数据采集滤波之后，进行数码管的显示。



图 2.14 温度采集单元框图

## 2.2.6 SPD 监测设备软件设计

SPD 监测单元的软件框图如图 2.15 所示, 主要包括了 MODBUS 通信协议、接地检测、雷击计数、前端保护检测、温度检测、SPD 失效检测和 485 通信单元。

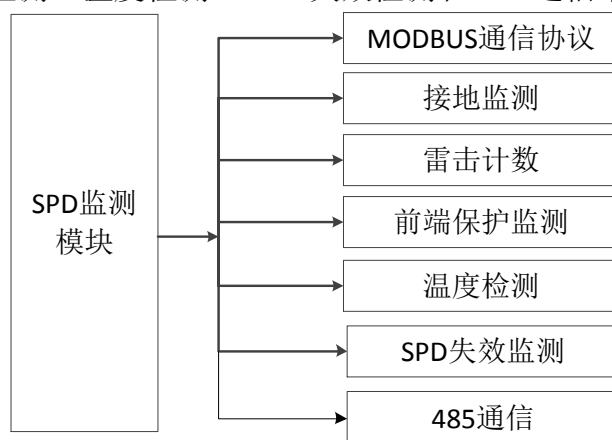


图 2.15 SPD 监测设备系统框图

经过硬件处理后, 这边可以通过 MCU 进行数据采样, 来分析处理数据。接地监测、前端保护检测、SPD 失效检测到达 MCU 时, 都会转变成 5V 的电平信号。MCU 判断 SPD 状态信息的方式是读取引脚上的高低电平。检测雷击次数, 经过硬件处理后, 传入 MCU 的是一个脉冲信号, MCU 通过中断检测机制, 去捕获相应的脉冲。检测到脉冲时, 实现雷击次数加一。具体的软件流程图如图 2.16。

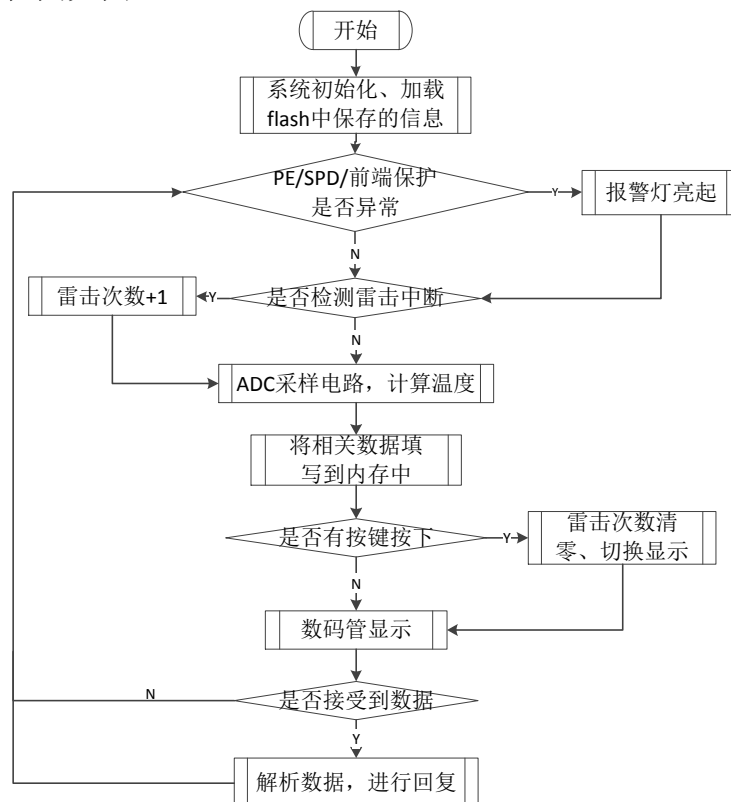


图 2.16 SPD 监测设备流程图

MODBUS 传输协议属于应用层报文传输协议，而应用层位于网络模型中的第七层。主要的机制为请求/应答机制，请求方为设备的主机，查询方为从机。从机不能主动发起申请，因此每一次的通信均需要主机主动发起通信请求。MODBUS 传输协议包括两种格式，即 RTU 和 ASCII 格式，两种格式都采用的大端模式。通信帧的结构如下图 2.17 所示，地址域即为从机设备的地址，通常称为设备栈号，占一个字节的空間。功能码用于提供特定的服务，在 1~255 的数值范围之内。比如读取线圈值或离散变量通常用功能码 01 和 02，03 功能码多用来读取保存在寄存器里的值。数据域中的内容因功能码的不同而差异。当查询寄存器的值时，需要有对应的寄存器起始地址、寄存器的个数。从机返回的内容为对应寄存器中的值。RTU 和 ASCII 采用不同的校验方式，前者采用 CRC 校验作为差错校验，而后者采用 LRC 校验方式。

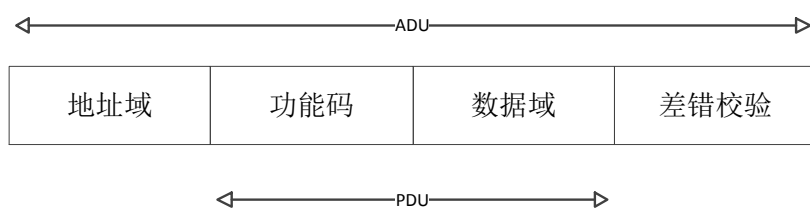


图 2.17 modbus 数据格式

将 SPD 的数据按照相应的地址存放，地址中的数据每个单元占两个字节。具体存放的数据单元如下表 2.2 所示：

表 2.2 SPD 数据存储地址和类型

| 功能       | 地址 | 字节数 |
|----------|----|-----|
| SPD 失效监测 | 21 | 2   |
| 接地检测     | 22 | 2   |
| 前端保护检测   | 27 | 2   |
| 雷击次数     | 30 | 2   |
| 温度       | 31 | 2   |

## 2.3 应用层软件设计

### 2.3.1 云平台设计

#### 2.3.1.1 设备云

本套系统中使用的云平台是中国移动的 OneNET 云平台，如图 2.18 和图 2.19 所示，设备将采集的信息按照一定的数据格式上报该平台。平台负责设备的存储和管理，以及和设备互联。网关通过 4G、WiFi、以太网的联网方式与云平台建立 TCP 连接，使用 MQTT<sup>[40]</sup>应用层协议，按照 JSON<sup>[41]</sup>的数据格式，将数据上报到 OneNET 设备云平台。同时业务云具有相应的 API 接口，供开发者使用，比如注册设备，注册产品。平台的管理页面如下图所示。开

发者可以按照实际的开发需要，进行监控设备管理、数据流查询、触发器设置等。

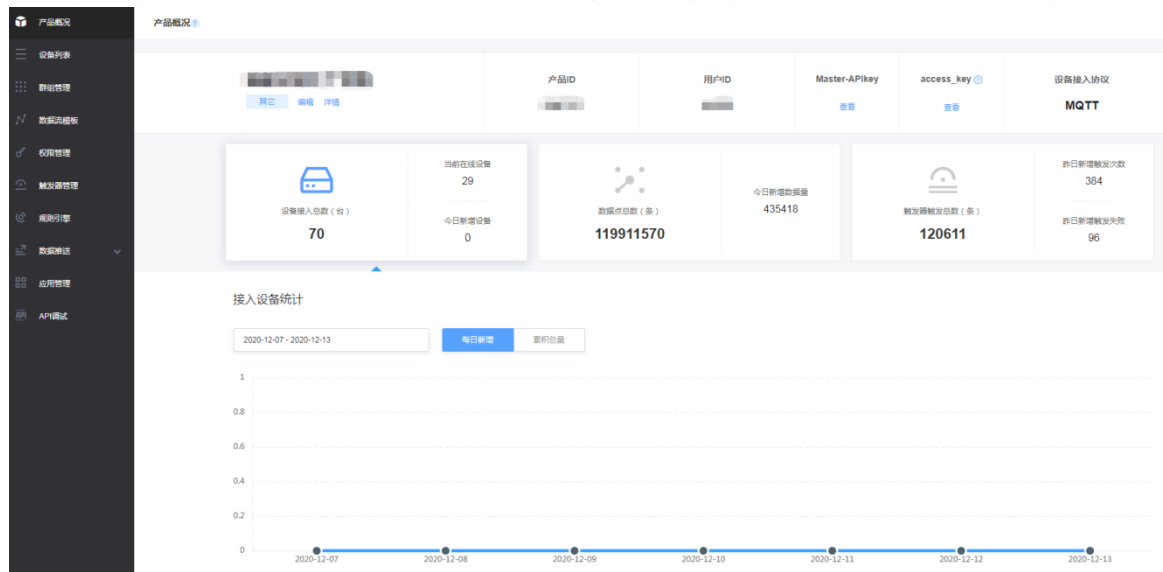


图 2.18 云平台展示图

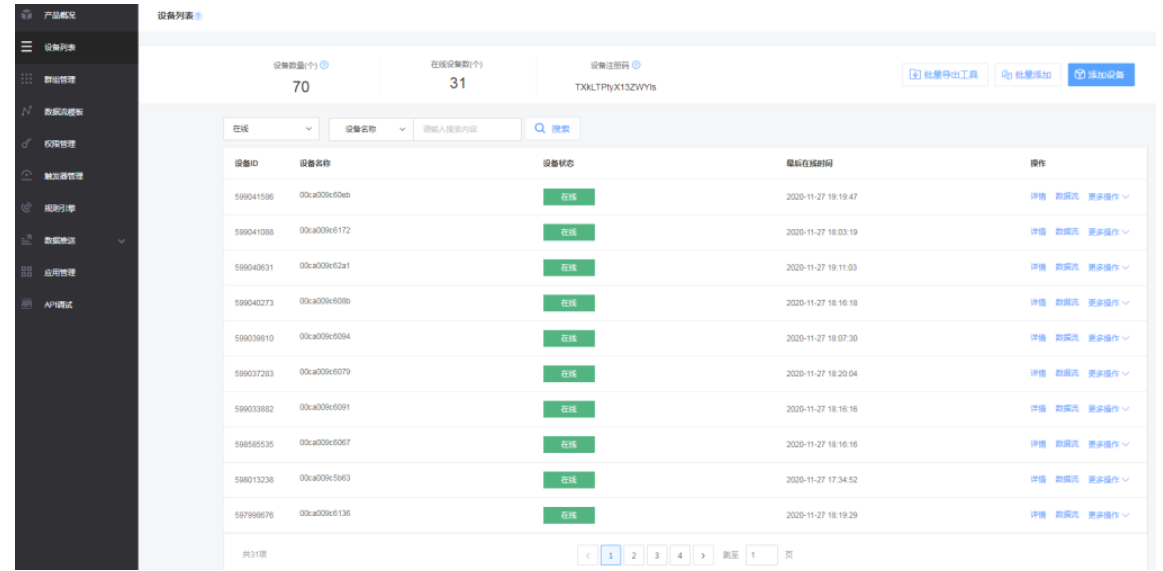


图 2.19 云平台设备在线展示

2.3.1.2 业务云

业务云的设计共三层，分为数据服务层、赋能层和存储层。同时支持客户端实现业务和存储支撑服务器集群。业务云的分层和职能如下图 2.20 所示。



图 2.20 业务云的设计逻辑

2.3.2 前端设计

前端客户端在整个系统中的作用是与用户进行交互、信息展示和消息推送的部分，用户通过前端页面进行配置，将 SPD 设备、节点、网关进行绑定。在设计的时候不仅仅考虑了用户角色，还涉及了管理员的角色。两种角色的操作权限是不同的，通过操作前端客户端对设备进行添加、删除、修改等操作。具体如下表 2.3 所示。

表 2.3 前端功能表

|                |                   | 拥有者   | 观察者 |
|----------------|-------------------|-------|-----|
| 条目             | 内容                | 超级管理员 | 用户  |
| 网关管理           | 添加设备              | ✓     |     |
|                | 删除设备              | ✓     |     |
|                | 功能设置              | ✓     |     |
| 节点和 SPD 监控设备管理 | 功能设置              | ✓     |     |
|                | 添加/删除/修改监控点       | ✓     |     |
|                | 添加/删除节点和 SPD 监控设备 | ✓     |     |
| 报警记录           | 报警查询              | ✓     | ✓   |
|                | 报警确认              | ✓     | ✓   |
|                | 权限管理              | ✓     |     |
|                | 功能设置              | ✓     |     |
| 历史数据           | 数据查询              | ✓     | ✓   |

续表 2.3

|      |      |   |   |
|------|------|---|---|
|      | 功能设置 | ✓ |   |
| 基本设置 | 信息查询 | ✓ | ✓ |
|      | 信息修改 | ✓ |   |

2.3.3 网关管理平台设计

网关管理平台在整个系统中功能包括配置和修改网关的信息、查看网关的内存、流量以及当前运行状态等。在使用的时候需要连接网关内部的 uhttpd 服务器。连接网关，在网页上输入相应的网关 ip，会向网关发送数据请求。OpenWrt 的 WEB 配置页面是通过 LUCI 实现的，如图 2.22 所示，LUCI 是基于 MVC 三层架构进行设计，其开发语言包括两种：HTML 语言和 Lua 脚本。本设计中编写了相应的前端网关管理平台的应用程序，网关管理平台具体实现如下图所示。

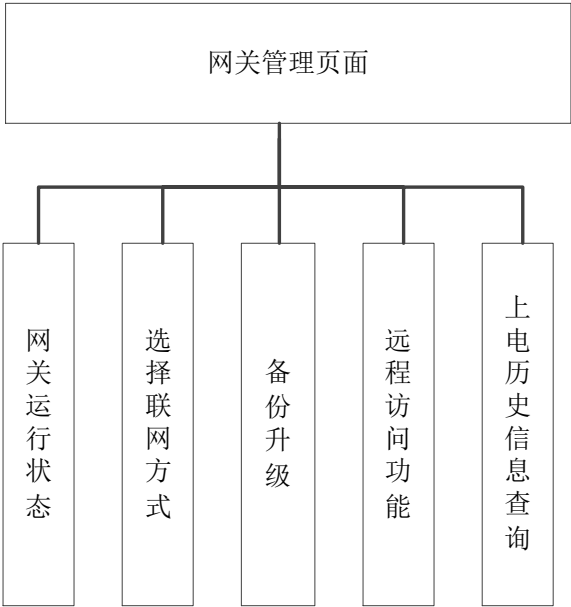


图 2.21 网关管理页面功能图

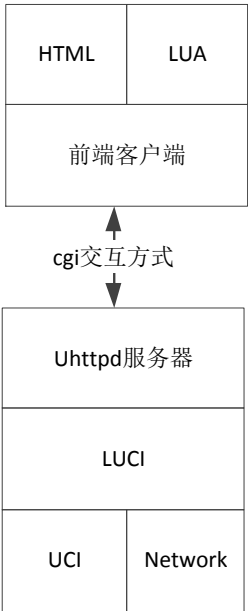


图 2.22 网关管理页面的实现方式

2.3.4 系统展示

智能网关的实物图如下图所示，主要包括顶板、底板和实物整体三个部分，底板主要包括了电源部分、AR9331、485/232 和以太网接口。顶板由电源、4G、MCU 和 LoRa 单元组成。



图 2.23 网关顶板



图 2.24 网关底板



图 2.25 网关实物 1



图 2.26 网关实物 2

节点实物图如下图所示，包含了节点正面、节点背面和节点的整体效果图。正面是由 MCU、电源、485/232/USART 等单元组成。背面只有一个 LoRa 模块。



图 2.27 节点正面



图 2.28 节点背面



图 2.29 节点实物

SPD 监测设备的实物图如下图所示，主要包括顶板、中层板和底板三个部分。底板主要是电源、PE 检测单元、雷击次数检测单元、SSD 检测单元和温度检测单元等。中层板是 MCU 和 485 的通信电路单元。顶板是数据交互单元。



图 2.30 设备底板



图 2.31 设备中层板



图 2.32 设备顶板



图 2.33 设备实物

SPD 实时监测系统的安装示意图如下：SSD、PE 和电源部分通过有线与 SPD 监测设备进行连接，SPD 安装在被保护设备的前端。LoRa 节点或者智能网关与 SPD 监测设备通过双绞线进行连接。

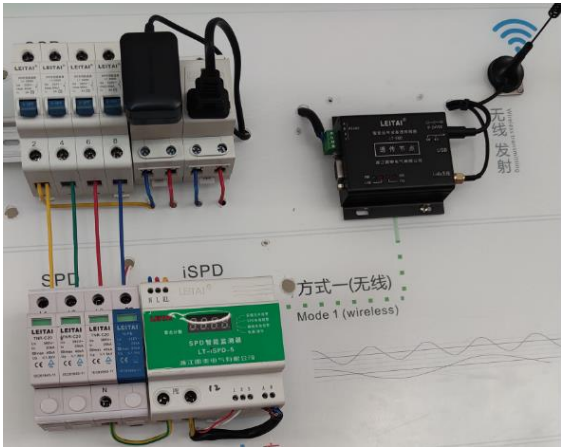


图 2.34 安装示意图 1

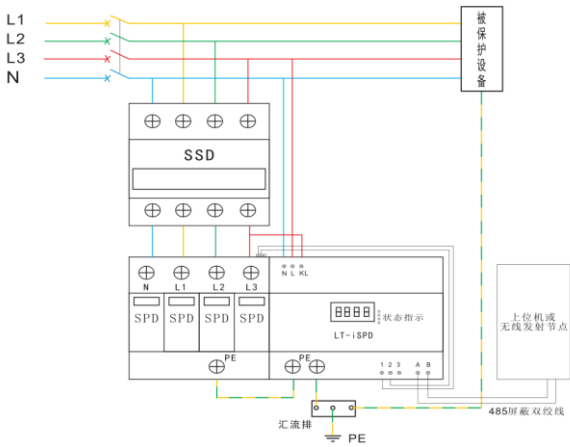


图 2.35 安装示意图 2

安装实物图如下：将 LoRa 节点和 SPD 监测设备与配电箱中相应的设备进行连接。



图 2.36 实物安装图示 1



图 2.37 实物安装图示 2

网关管理页面设计，如图 2.38 所示，可以通过网关的管理页面选择不同的联网方式。有以太网、WiFi 无线网络和 4G 移动网络拨号上网三个选项供用户选择。

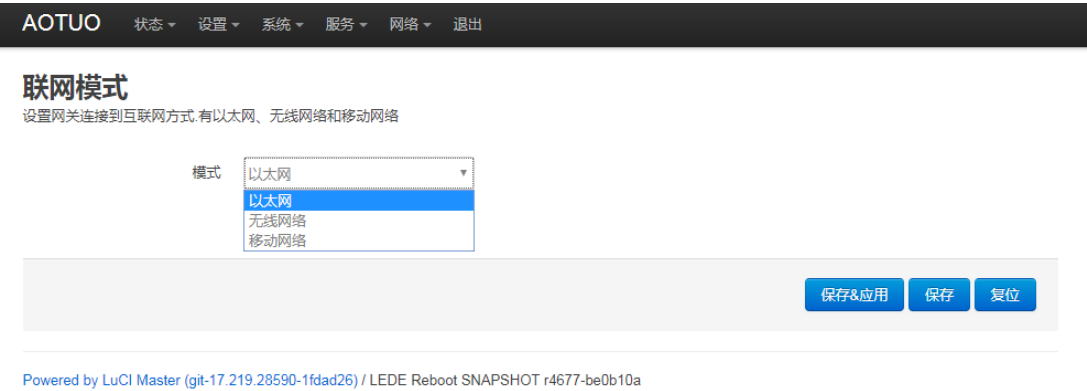


图 2.38 网关管理页面

WEB 客户端主页面设计，如下图 2.39 所示，登录以后，会有用户下管理的网关和 SPD 监测设备的信息，包含了网关安装的地理位置信息、SPD 监测设备的在线离线信息、以及相关报警信息。进行了图表数据的汇总和展示。



图 2.39 SPD 数据监控效果图

WEB 客户端添加页面设计，如图 2.40 所示，添加 SPD 监控设备和与之连接的 LoRa 节点的地址。包含了 SPD 监测设备的栈号、相应的寄存器地址和校验方式。

图 2.40 添加 SPD 监控设备

网关下的各个节点的 SPD 监测设备的信息，包含了 SPD 检测设备是否在线，在具体对应的 LoRa 节点的地址和设备名称等信息。其中状态栏中的圆点表示设备的在线和离线的状

态，离线（红色），在线（绿色）。具体如下图 2.41 所示：

| 序号 | 状态 | 名称      | 设备类型      | 安装时间                | 站号 | 节点       | 操作    |
|----|----|---------|-----------|---------------------|----|----------|-------|
| 1  | ●  | 5#      | LT_ISPD_5 | 2020-09-05 14:35:34 | 5  |          | 编辑 删除 |
| 2  | ●  | 二小总配电箱  | LT_ISPD_5 | 2020-07-27 11:12:28 | 9  |          | 编辑 删除 |
| 3  | ●  | 一小一修配电箱 | LT_CJ485  | 2019-11-19 08:47:02 | 60 |          | 编辑 删除 |
| 4  | ●  | 雷泰总配电箱  | LT_ISPD_5 | 2020-07-10 11:11:03 | 15 | 50700021 | 编辑 删除 |
| 5  | ●  | 雷泰一修配电箱 | LT_ISPD_5 | 2020-03-01 09:46:17 | 26 | 50700021 | 编辑 删除 |
| 6  | ●  | 测试10    | LT_ISPD_5 | 2019-11-19 14:29:19 | 10 | 50700022 | 编辑 删除 |
| 7  | ●  | 测试20    | LT_ISPD_5 | 2020-03-15 00:00:00 | 20 | 50700022 | 编辑 删除 |
| 8  | ●  | 避雷器1    | LT_CJ485  | 2020-07-30 10:59:49 | 1  | 50700023 | 编辑 删除 |
| 9  | ●  | 避雷器2    | LT_CJ485  | 2020-03-21 13:31:45 | 2  | 50700023 | 编辑 删除 |
| 10 | ●  | 避雷器3    | LT_CJ485  | 2020-03-08 00:00:00 | 53 | 50700023 | 编辑 删除 |

图 2.41 设备绑定信息图

报警推送，包含两种方式，如下图所示，一种是前端页面的推送，另一种是手机短信的推送。当出现报警信息时，会推送给与之对应的用户。

报警信息 关闭

网关名称：四板桥测试

| 设备名称  | 报警类型    | 报警次数 | 操作 |
|-------|---------|------|----|
| 51号泵房 | SPD失效报警 | 4    | 解除 |
| 避雷器2  | 空开报警    | 1308 | 解除 |
| 测试20  | SPD失效报警 | 828  | 解除 |

图 2.42 web 端页面报警推送



图 2.43 短信报警推送

当网关出现异常时，会将当时产生异常的环境进行打包，上传到对应的文件服务器。下图是各个网关上报的异常信息文件夹和文件夹下的压缩包。

| 名               | Size (KB) | 最后修改             | 所有者  |
|-----------------|-----------|------------------|------|
| 867394045863734 |           | 2020-05-19 01:29 | 1001 |
| 867394045685954 |           | 2020-05-19 01:23 | 1001 |
| 867394045632089 |           | 2020-07-16 23:26 | 1001 |
| 867394045550117 |           | 2020-05-19 01:02 | 1001 |
| 867160047934074 |           | 2020-08-05 03:45 | 1001 |
| 00ca02032641    |           | 2020-09-10 23:53 | 1001 |
| 00ca009c78a3    |           | 2020-11-11 21:39 | 1001 |
| 00ca009c78a0    |           | 2020-06-20 23:42 | 1001 |
| 00ca009c789d    |           | 2020-04-27 08:00 | 1001 |
| 00ca009c7894    |           | 2020-08-04 00:18 | 1001 |
| 00ca009c7891    |           | 2020-08-03 21:10 | 1001 |

图 2.44 异常文件收集图

| 名                   | Size (KB) | 最后修改             | 所有者  |
|---------------------|-----------|------------------|------|
| 20200804-194510.tar | 3         | 2020-08-05 03:45 | 1001 |
| 20200804-194447.tar | 3         | 2020-08-05 03:44 | 1001 |
| 20200804-194424.tar | 3         | 2020-08-05 03:44 | 1001 |
| 20200804-194403.tar | 3         | 2020-08-05 03:44 | 1001 |
| 20200804-194342.tar | 3         | 2020-08-05 03:43 | 1001 |
| 20200804-194323.tar | 3         | 2020-08-05 03:43 | 1001 |
| 20200804-194304.tar | 3         | 2020-08-05 03:43 | 1001 |
| 20200804-194245.tar | 3         | 2020-08-05 03:42 | 1001 |
| 20200804-194228.tar | 3         | 2020-08-05 03:42 | 1001 |
| 20200804-194212.tar | 3         | 2020-08-05 03:42 | 1001 |
| 20200804-194149.tar | 3         | 2020-08-05 03:41 | 1001 |

图 2.45 文件压缩包展示图

## 2.4 本章小结

本章首先对 SPD 实时监测系统的总体框架和工作流程进行介绍。然后对系统的各个部分进行设计，包括了硬件和软件。主要是设备层的物联网网关设计、LoRa 节点设计、SPD 监测设备设计和应用层设计。最后对各个部分的设备进行实物展示和平台展示。

### 第3章 基于时隙 CSMA/CA 算法的 LoRa 无线网络设计

在原有的时隙 CSMA/CA 算法当中，有多个节点同时想要发送数据时，需要使用传统的二进制退避算法去产生一个随机退避时间<sup>[41]</sup>，从而使节点在退避一段时间后接入信道以防止发生数据碰撞。这种方式比较简单，但是具有严重的不公平现象，会使得接入成功的节点，再次接入信道的优先级比较高<sup>[42]</sup>。本文对在原有的退避算法上进行了改进。进行了网络模型的仿真，证明了该算法相比传统二进制退避算法具有较好的公平性。同时分析了常用的网络结构和相应的协议，结合 LoRa 的特性对 SPD 监测系统的 LoRa 组网的网络架构、MAC 层的协议和网络协议和进行设计。最后通过实物进行测试分析。

#### 3.1 基于时隙 CSMA/CA 算法公平性的改进及实现

在 IEEE802.15.4 标准中，CSMA/CA 算法主要包括两种信道访问机制，分为时隙和非时隙两种<sup>[43]</sup>。两种算法各有优劣。非时隙 CSMA/CA 的过程中，不需要考虑时隙对齐，虽然有较好的实时性，但是当节点数目过多时，系统的吞吐量和丢包率性能不好，存在隐藏终端的问题和节点接入信道的不公平现象。而在时隙的 CSMA/CA 访问机制中，节点设备在发送数据之前需要发送一个短帧进行信道的预约，发送请求 RTS，接收方接收到该帧后需要回复一帧，允许发送 CTS，然后进行数据的收发，可以解决隐藏终端的问题。在网络结构中，节点数目较多时，系统的吞吐量和丢包率要比非时隙表现的好<sup>[44]</sup>。在本套 SPD 监测系统中，为了追求系统稳定和信息的准确性，从而选用了时隙的 CSMA/CA 退避算法接入信道机制。

##### 3.1.1 传统时隙 CSMA/CA 算法的原理

时隙 CSMA/CA 算法的原理<sup>[43]</sup>，利用空闲信道监测(Clear Channel Assessment, CCA)技术，节点设备在接入信道之前，需要判断信道是否被占用，以避免冲突。如图 3.1 所示，A 和 B 两个节点都可以在信道中发送数据，A 检测信道未被占用时，接入信道进行数据的发送，此时 B 会检测到信道是占用的状态，一直等到 ACK 信号完成后的一个时隙，节点 B 退避时间归零，重新接入信道，检测信道为空闲，开始发送。

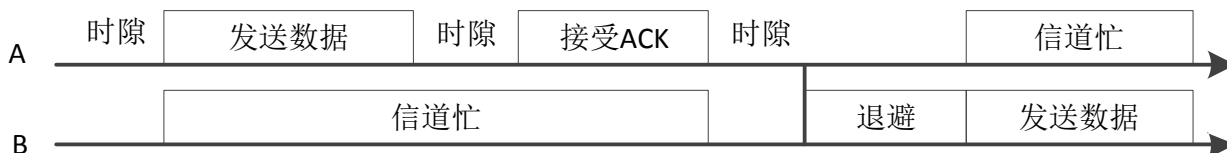


图 3.1 节点退避示意图

算法的工作方式如下：节点在进行数据的发送时，第一步需要到达下一个发送时隙，第

二步执行一次随机时隙的退避，等待退避时间递减为零，第三步进行信道 CCA 的检测，判断信道空闲与否，如果信道空闲，再进行检测，连续检测到两次空闲，可以抢占信道进行数据发送。反之则进入信号退避阶段，退避时长增加，失败次数自增。最后重新进行信道的检测，一直到数据发送成功为止，或者退避次数抵达最大值后放弃本次帧传输。具体流程图如图 3.2 所示。

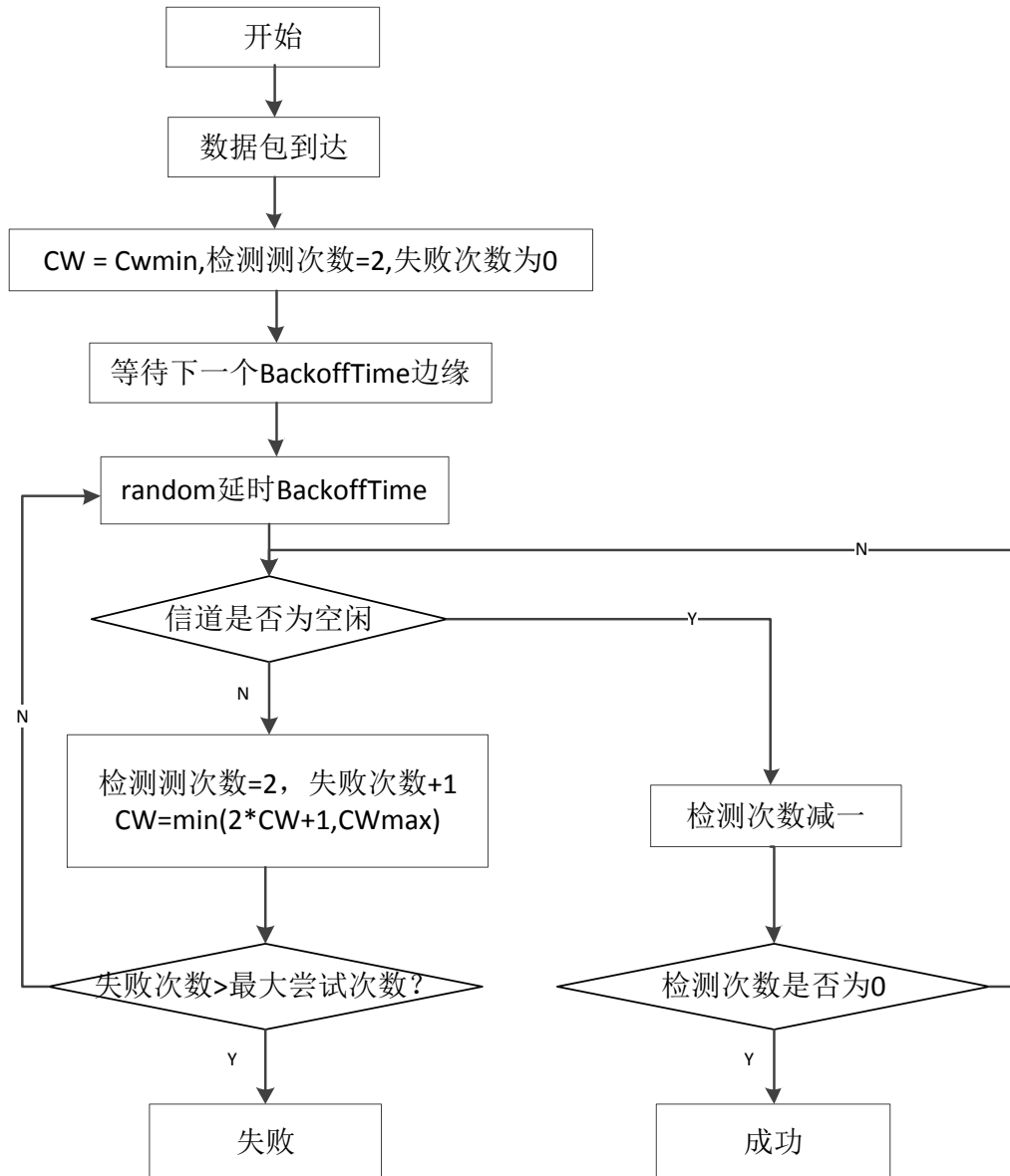


图 3.2 传统时隙 CSMA/CA 的流程图

### 3.1.2 传统时隙 CSMA/CA 算法分析

目前常用的退避算法是二进制退避算法。需要对竞争窗口(Contention Window, CW)的概念进行了解，其服从均匀分布。节点在进入退避阶段时，会从 $[0, CW]$ 的范围之间随机选取一个值，该随机值与时隙长度的乘积为退避计数器的初始值。当信道空闲时，退避计数器从初始值开始进行递减，递减为 0 时接入信道。当接入信道失败，竞争窗口 CW 的值会按指数

增加，一直到最大值  $CW_{max}$ ，当接入信道成功，数据传输完成才会被初始化为最小值  $CW_{min}$ 。

二进制退避算法中，退避时间可以按照公式(3.1)计算，其中  $random$  是从  $[0, CW]$  之间的随机值， $aSlotTime$  为时隙长度， $BackoffTime$  为退避时长。

$$BackoffTime = random \times aSlotTime \quad (3.1)$$

竞争窗口按照公式 (3.2) 进行计算。

$$CW = \begin{cases} \min(2 * CW + 1, CW_{max}) & \text{发生碰撞，数据重传} \\ CW_{min} & \text{发送成功} \end{cases} \quad (3.2)$$

传统的算法存在不公平现象，当有多个设备节点需要发送数据时，开始参与竞争信道环节。刚刚发送数据成功的节点，再次参与信道竞争时，该节点具有较高的优先级再次抢占到信道。如公式 3.2 所示，当节点发送成功数据的时候，竞争窗口  $CW$  将被初始化为最小值。而其他的节点访问信道失败时， $CW$  的值会随着重传的次数增加而增加。而当刚刚发送成功的节点与已经访问信道失败的节点产生竞争时，因为竞争窗口值  $CW$  的不同，产生的退避时间也是不同的。刚刚发送成功的节点的退避时间比较短，所以接入信道的概率就会更大。如图 3.3，描述了竞争窗口二进制指数增加的示意图。

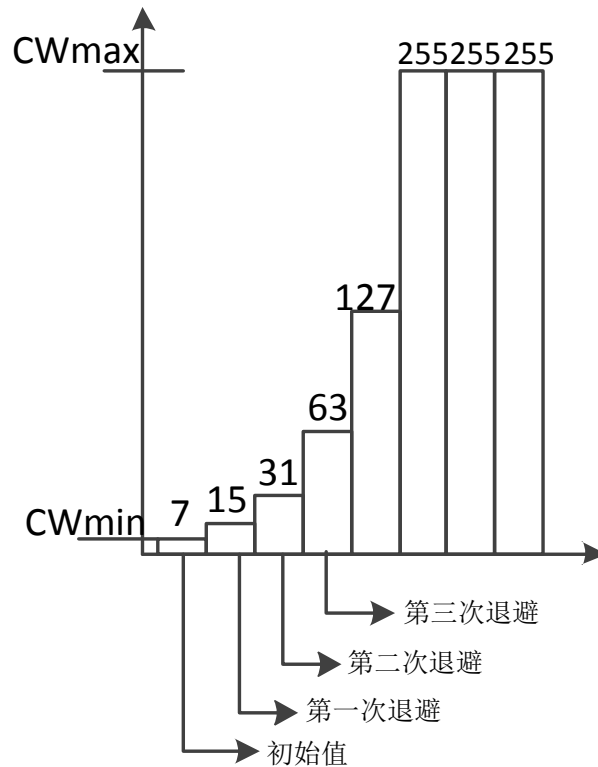


图 3.3 竞争窗口的示意图

$CW_{min}=7$ ,  $CW_{max}=255$ ,  $CW_{min} \leq CW \leq CW_{max}$ ，竞争窗口的初始值为  $CW_{min}$ ，当节点访问信道发生冲突时，竞争窗口的值将会按照二进制退避算法进行增加，一直达到最大的重传次

数，仍然接入信道失败的话，就放弃本次传输。当发送成功的时候，则将竞争窗口值初始化为  $CW_{\min}$ 。

### 3.1.3 基于时隙 CSMA/CA 算法的公平性改进

根据上文所述，原有的退避算法中公平性得不到保证，发送消息成功的节点可以优先获得接入信道的权限。退避算法的实质，还是调节竞争窗口  $CW$ 。也有很多论文此基础上提出了优化算法，像 MILD<sup>[46]</sup>和 EILD<sup>[47]</sup>算法。这两种算法同样存在不公平的问题。当网络状况在短时间内发生好转，竞争窗口倍数减小的思路会使得  $CW$  的值下降到合理范围区间需要很长一段时间，并且不一定能找到这个合理值<sup>[48]</sup>。本系统中采用的 LoRa 无线通信技术，传输数据的时延是在毫秒级的单位，如果采用传统的二进制退避算法，会导致节点在长时间竞争信道时，产生“饿死”现象。所以必须对相应的竞争窗口进行调节，使得节点接入信道具有公平性且适合本系统的特点。

结合实际情况和 LoRa 无线通信技术的问题，在原有的二进制退避算法的基础之上，本文提出了一种适用于 LoRa 的退避算法。算法是  $CW$  采用的策略是倍数递增、倍数和线性减小。需要在原有的二进制退避算法中，引入两个新的参数，竞争失败次数的门限  $n_{th}$  和竞争窗口中间值  $CW_{mid}$ 。实现方式如下。

竞争失败时

$$CW = \begin{cases} \min(2 * CW + 1, CW_{max}) & n < n_{th} \\ CW_{min} & n \geq n_{th}, \text{clear } n \end{cases} \quad (3.3)$$

竞争成功时

$$CW = \begin{cases} CW - d & CW \leq CW_{mid} \\ \min(CW / 4, CW_{mid}) & CW > CW_{mid} \end{cases} \quad (3.4)$$

当竞争失败时，未到达失败门限值时，竞争窗口  $CW$  成倍的增加，当超过竞争失败门限值时  $n_{th}$ ，恢复为最小的竞争窗口  $CW_{min}$  重新参与信道的竞争。当信道竞争成功时，竞争窗口的值是否超过竞争窗口中间值  $CW_{mid}$ ，当小于竞争窗口中间值  $CW_{mid}$  时，线性进行窗口值的减小。当大于竞争窗口中间值  $CW_{mid}$  时，竞争窗口  $CW$  成倍的减小。使得竞争窗口的区间维护在竞争窗口中间值  $CW_{mid}$  左右。从而保证了节点接入信道的机会相近，提高了公平性。

相关优化过后的流程图如图 3.4 所示：

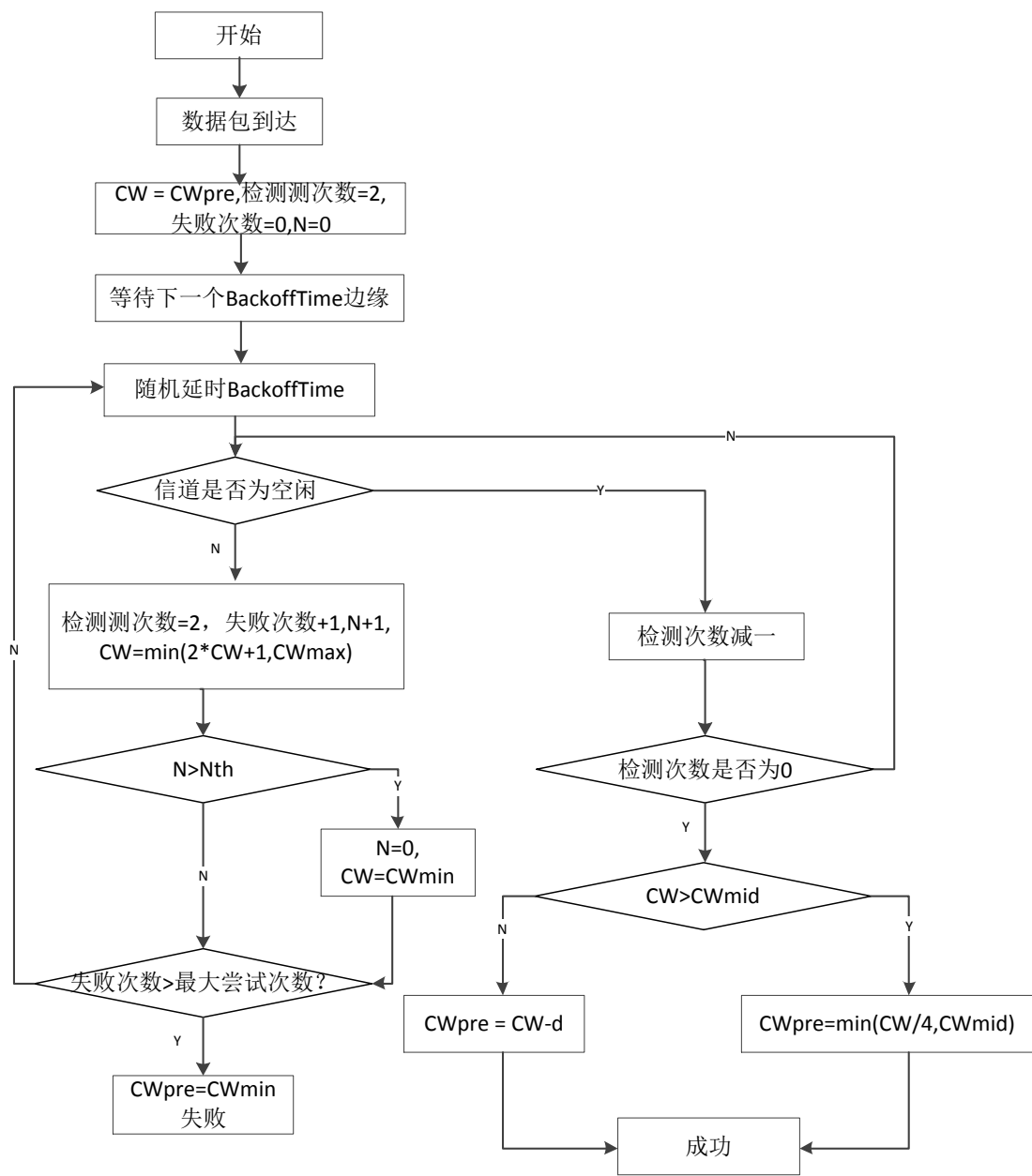


图 3.4 优化算法流程图

优化后的示意说明，当节点 A、B、C 都有数据需要发送时，且各个发送缓存中一直都有数据，分别采用传统退避算法和优化后的退避算法进行比对分析。传统退避如图 3.5 所示。



图 3.5 传统算法的退避示意图

优化过后的退避示意图如图 3.6 所示，在小于失败次数的门限  $n_{th}$  和竞争窗口中间值  $CW_{mid}$  下进行实现。



图 3.6 优化过后的退避示意图

优化过后的算法中节点 A、B、C 均可以发送数据，而传统算法中，节点 A 两次发送完数据，节点 B 和节点 C 没有抢占到信道，仍在执行退避过程。因为优化后的退避窗口值较大。

分析完成退避算法，需要移植到相应的硬件设施进行实现。下文结合本项目采用的 LoRa 技术，对 MAC 层进行设计，搭建相应的硬件平台进行测试。

### 3.2 基于时隙 CSMA/CA 算法的 LoRa 无线网络 MAC 层设计

IEEE802.11DCF 是基于 CSMA/CA 协议基础之上的标准协议，当网络负载较轻时，采用 DATA/ACK 的方式进行数据交互，当网络容量达到设定的阈值时，为了减小碰撞冲突的概率，会开启 RTS/CTS 的握手机制。牺牲较短的控制时延以换取整个网络性能的提升。如下

图 3.7 所示，是带有 RTS/CTS 接入信道方式的时隙图。源节点在接入信道之前，需要等待一个 DIFS 的时间，然后再进行信道的检测，如果当前信道没有占用，则发送一帧 RTS 帧，如果信道忙则启用退避机制。目标节点接收 RTS 帧以后，需要进行回复，表示可以接受数据，目标节点回复 CTS，CTS 的帧前后都有 SIFS 间隔。源节点收到 CTS，发送 DATA 数据，目标节点间隔 SIFS 回复 ACK。信道中的其他节点需要根据 RTS 和 CTS 帧中的字段不断进行更新 NAV 变量的更新操作，NAV 是传输消耗的总时间。

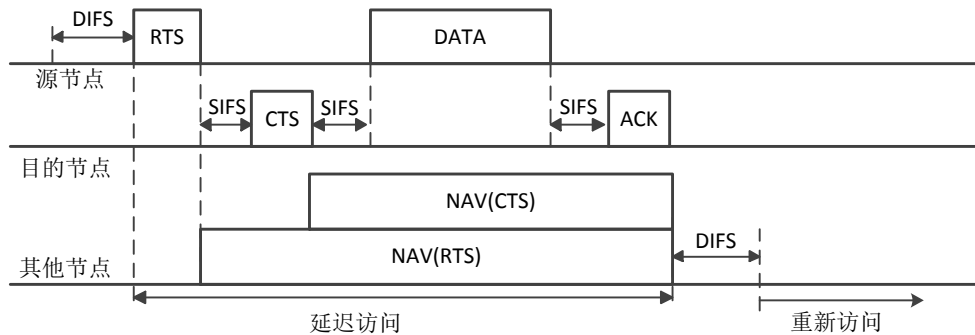


图 3.7 RTS/CTS 接入方式时隙图

下面对 DIFS 和 SIFS 进行讲解，如图 3.8 所示，其中 SIFS (Short Interframe Space) 是指最短帧间隔，在通信过程中主要作用是预约信道，保证数据帧的顺利传输。在进行数据交

换时，其他节点需要延时 DIFS 的时序，DIFS 的间隔时间时长大约 SIFS 的，所以 SIFS 的预约机制不会被打断。

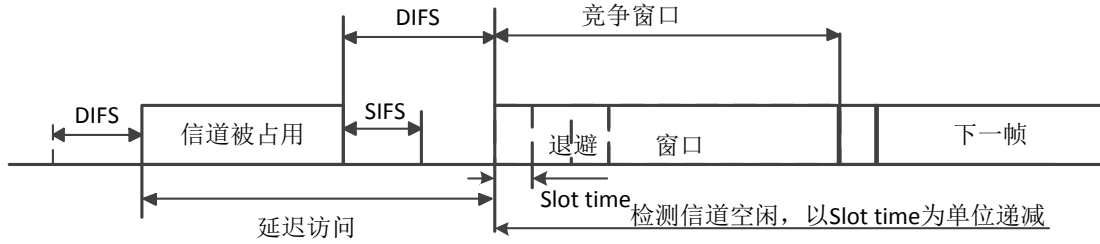


图 3.8 帧间间隔关系图

由图可知 SIFS 的使用是在 RTS、CTS 和 DATA 帧之后的间隔。SIFS 的大小可有公式(3.5)计算。

$$aSIFSTime = aRxRFDelay + aRxPLCPDelay + aMACPrcDelay + aRxTxTurnaroundTime \quad (3.5)$$

其中， $aRxRFDelay$  和  $aRxPLCPDelay$  是射频接收时延， $aMACPrcDelay$  是 MAC 协议处理时延， $aRxTxTurnaroundTime$  是射频的收发转换时间。

DIFS (Distributed coordination function Interframe Space) 是设备检测信道未被占用最少时间端，被作为每次交互之前的间隔，当设备连续检测了 DIFS 的时间间隔后，设备才能抢占信道。可按照公式 (3.6) 进行计算。

$$DIFS = aSIFSTime + 2 \times aSlotTime \quad (3.6)$$

### 3.2.1 基于时隙 CSMA/CA 算法 LoRa 的时隙选取

最小帧间间隔  $aSIFSTime$  分别由射频接收时延、MAC 协议的处理时延和射频转换时间所决定。对于 LoRa 扩频通信的技术，查询相应的官方数据手册可以得到，射频转换时间远大于接收和处理时延。由图 3.9 所示，发射器启动时间  $TS_{TR}$ 、接收器启动时间  $TS_{RE}$  和晶振振荡器  $TS_{OSC}$  共同决定了射频的转换时间。发射器和接收器的转换时间各为 1ms， $TS_{OSC}$  的特征值为 250us。那么最小  $aSIFSTime$  的取值取 3~5ms。

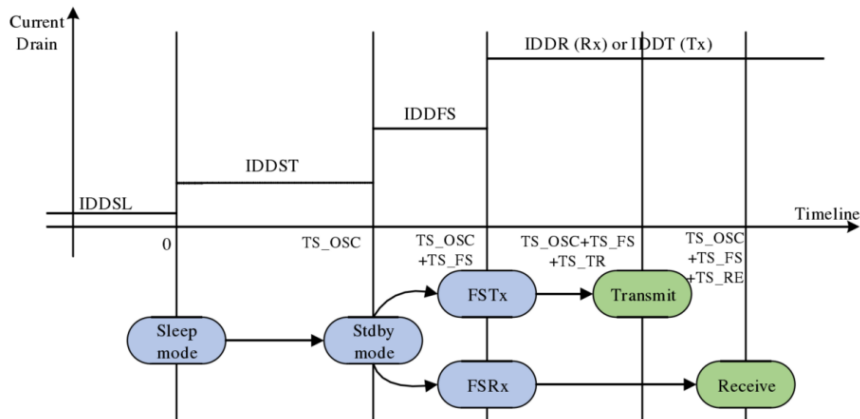


图 3.9 射频转换示意图

由上文可知 CAD 模式下可以用来检测信道是否被占用。时隙 aSlotTime 的时间分别由载波监听时间, 空中传播时间, 协议处理时间和射频转换时间所决定。载波监听的时长即为 CAD 检测的时常, CAD 检测的时常远远大于空中传播时长和程序中协议处理的时间。

CAD 的检测周期是由 LoRa 的扩频因子 (SF), 带宽 (BW) 共同决定的, 满足以下关系式。接收周期:  $T_{rec} = (2^{SF} + 32)/BW$ , 整个 CAD 周期:  $T_{cad} = 1.85 \times T_{rec}$ , 相应时间关系如下表。

表 3.1 不同 SF 对应的 CAD 周期

| BW(kHz) | SF | CAD 周期 (ms) |
|---------|----|-------------|
| 125     | 7  | 2.368       |
| 125     | 8  | 4.2624      |
| 125     | 9  | 8.0512      |
| 125     | 10 | 15.6288     |
| 125     | 11 | 30.784      |
| 125     | 12 | 61.0944     |

当射频模块选用相同的扩频因子和不同的带宽时, CAD 检测周期的时长很相近。当射频模块选用相同的带宽相同, 不同的扩频因子, CAD 的检测周期几乎成倍增长。其值小于 12 的时候, aSlotTime 时隙值定为 32ms, 为 12 的时候设置为 65ms。

确定了 aSIFSTime 和 aSlotTime 的数值, 时间间隔 DIFS 的数值和网络分配矢量的 NAV 的数值就可以按上文的方法进行计算。

### 3.2.2 LoRa 无线网络 MAC 层帧格式设计

RTS 的控制帧定义为 12 个字节, 如图 3.10 所示, 其帧格式为:

|     |     |      |     |       |       |
|-----|-----|------|-----|-------|-------|
| 帧长度 | 帧类型 | 目的地址 | 源地址 | 网络分矢量 | CRC校验 |
|-----|-----|------|-----|-------|-------|

图 3.10 网络帧格式 RTS

其中帧长度, 表示该帧的长度, 帧类型为 0x01, 节点和网关在程序中都会被定义一个唯一的 MAC 地址。地址均为 4 个字节。目的地址指的是, 在网络通信中该帧发往具体接受节点的 MAC 地址。源地址指的是, 此次通信的发起端的 MAC 地址。网络分配矢量 NAV\_RTS 在通信过程中是 RTS 帧的网络分配矢量, 计算方式如公式 3.7。CRC 表示校验位, 使用 CRC16 进行校验。

$$NAV\_RTS = 3 \times aSIFSTime + CTS\_Time + Data\_Time + ACK\_Time \quad (3.7)$$

CTS 的控制帧格式也为 12 个字节, 如图 3.11 所示。

|     |     |      |     |       |       |
|-----|-----|------|-----|-------|-------|
| 帧长度 | 帧类型 | 目的地址 | 源地址 | 网络分矢量 | CRC校验 |
|-----|-----|------|-----|-------|-------|

图 3.11 网络帧格式 CTS

帧类型为 0x02，网络分配矢量 NAV\_CTS 在通信过程中是 CTS 帧的网络分配矢量，计算公式 3.12。CRC 表示校验位。

$$\text{NAV\_CTS} = \text{NAV\_RTS} - \text{aSIFSTime} - \text{CTS\_Time} \quad (3.8)$$

DATA 的控制帧格式为 12 个字节和用户数据的长度组成，如图 3.12 所示。

|     |     |      |     |       |    |       |
|-----|-----|------|-----|-------|----|-------|
| 帧长度 | 帧类型 | 目的地址 | 源地址 | 网络分矢量 | 数据 | CRC校验 |
|-----|-----|------|-----|-------|----|-------|

图 3.12 网络帧格式 DATA

LoRa 的数据最高的发射速率，大约在 21Kbps 左右，本系统选用的速率从 6kbps 左右到 21kbps 左右，这里的数据长度为 60 个字节，加上 12 个字节的固定长度，可以保证一帧在 100ms 内传输完成。帧类型为 0x03，目的地址为要将信息发往网络中相应节点的物理地址，源地址为本机地址，即将要发送该帧的节点物理地址。数据为网络层交付的数据。CRC 为帧校验位。网络分矢量 NAV\_DATA 的计算方法如公式 3.9。

$$\text{NAV\_DATA} = \text{aSIFSTime} + \text{ACK\_Time} \quad (3.9)$$

ACK 的帧格式，如下图所示。

|     |     |      |     |       |
|-----|-----|------|-----|-------|
| 帧长度 | 帧类型 | 目的地址 | 源地址 | CRC校验 |
|-----|-----|------|-----|-------|

图 3.13 网络帧格式 ACK

帧类型为 0x04，目的地址为 DATA 帧中的源地址，而本帧的源地址为 DATA 帧中的目的地址。尾部加有相应的 CRC 校验位。

RES 的帧格式，如图 3.14 所示。

|     |     |      |     |       |
|-----|-----|------|-----|-------|
| 帧长度 | 帧类型 | 广播地址 | 源地址 | CRC校验 |
|-----|-----|------|-----|-------|

图 3.14 网络帧格式 RES

其中帧类型为 0x05，目的地址为广播地址 0xFFFFFFFF 这 4 个字节，而源地址为将要发送数据的节点地址，同理和 DATA 帧中一样，用户数据推荐最长为 60 字节。

### 3.2.3 LoRa 无线网络的信道监测机制比较分析

现在比较常用的信道检测技术是基于信号强度的做法，比如通过检测 RSSI 值。设备通过接收数据，判断接收信号的 RSSI 值是否超过所设定的阈值。这种方式存在以下的问题：

1. 信号在发送模式下检测结果不准确，必须在接收模式下才能进行，而 LoRa 无线技术属于半

双工的，无法实现数据的边收边发。2.在实际的楼宇和户外环境中，根据路径损耗模型可知<sup>[49]</sup>，无线信号的 RSSI 值会随机变化波动，导致检测结果不准确。对于 SPD 监测系统而言，采用的是 LoRa 的扩频调制技术，有效信息的信号强度比接收机的噪声信号强度还低，故不能用 RSSI 值来判断信道是否空闲。

对比于 RSSI 值检测信道的方式，采用 LoRa 无线通信技术中的活动信号检测技术，对信道是否忙碌进行检测，更具有优势。LoRa 的数据包组成，最前面一段部分为前导码，而 LoRa 的活动信道检测（CAD）部分可以通过检测无线信道是否存在 LoRa 前导码以判断信道是否被占用。当信号强度低于接收机底噪，通过 CAD 检测技术，仍然可以将 LoRa 的信号检测出来，前导码的检测与信号强度无关。对比于第一种 RSSI 值的检测技术，活动信道的检测技术更具有可靠性。

### 3.3 基于时隙 CSMA/CA 算法的 LoRa 无线网络的总体设计与实现

#### 3.3.1 LoRa 无线网络的拓扑结构

结合本项目的特点，需要选择合适的网络拓扑结构。不同的网络拓扑结构，具有不同的特点，可以使系统发挥出相应的优势，使系统更加的稳定。

基于 LoRa 的网络拓扑结构的性能比较如下表 3.2 所示：

表 3.2 网络特性比较

|     | 星型网络拓扑结构       | 网状网络拓扑结构          |
|-----|----------------|-------------------|
| 时延  | 小，实时性可控        | 大，信息最终接收时间不可控     |
| 能耗  | 低，节点可休眠        | 高，中继处于接受发送状态      |
| 容量  | 大，增加网关可以增加带宽   | 适中，节点越多，转发流量越多    |
| 跳数  | 单跳，网关与节点直接通信   | 多跳，需中继转发          |
| 可靠性 | 高，可及时发现丢包并重传数据 | 低，任意一跳失败则本次数据传输失败 |
| 复杂度 | 低，不需要路由转发      | 高，需处理路由转发         |
| 扩展性 | 强，增加网关即可       | 弱，需调整网络结构         |

从上表我们可以看出，在 LoRa 的无线通信网络中，星型网络通信比网状网络结构在距离、功耗、拓展性、可靠性、成本和实现方法上都具有一定的优势。结合本系统中的实际应用场景，SPD 监测设备和 SPD 都装在电箱或者防雷箱中，安装位置是固定的，节点无需移动。在本系统中，各个节点之间无需通信，都只需要与网关进行信息的传输，且本系统对数据的稳定性要求比较高。综合以上考虑因素，SPD 监测系统选用星型网络结构。相应的网络拓扑结构如下图 3.15 所示，以集中器网关为中心，各个节点与网关进行数据通信。

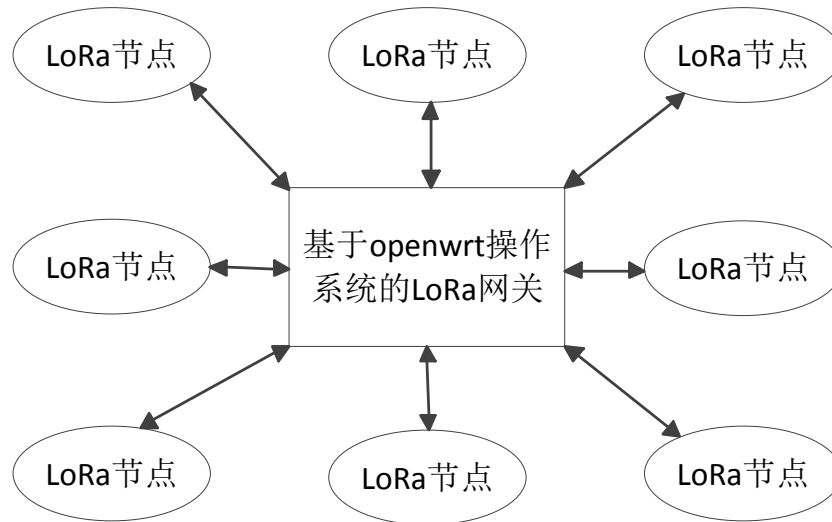


图 3.15 星型网络拓扑结构

针对无线通信进行设计时，对于网络通信的选择就显得非常重要了。在上文分析中，工程项目中经常采用到的协议：WiFi、蓝牙、ZigBee 和 LoRaWAN。对于室内的通信距离而言，除了 LoRaWAN，都不占优势，且前两者的网络容量比较小，不支持很多节点。LoRaWAN 的通信协议虽然具有 LoRa 的特性，但是底层的 MAC 协议的信道利用率很低，有关研究表明，其信道利用率最大只有 18.4%<sup>[33]</sup>。该协议在数据传输中容易产生数据的碰撞而导致重发，数据的重传在一定程度上不仅影响到了网络通信的可靠性，还会产生额外的功耗。

### 3.3.2 SWSN 无线通信协议设计

基于上述因素的考虑，并结合该 SPD 实时监测系统对室内穿透能力、室外远距离、稳定性和易维护等方面的需求，本系统自主设计了一种通信协议，命名为 SWSN 协议。该协议在物理层中采用了 LoRa 射频模块，具有 LoRaWAN 的特性。在 MAC 层以及 MAC 层之上和 LoRaWAN 完全不同，弥补了信道利用率低的缺点，保证了本系统的网络通信可以具有良好的实时性和稳定性。SWSN 协议参考 OSI 七层模型进行设计，这里对 SWSN 的物理层、MAC 层、网络层和应用层的功能点进行说明。相应的架构如下图 3.16 所示。

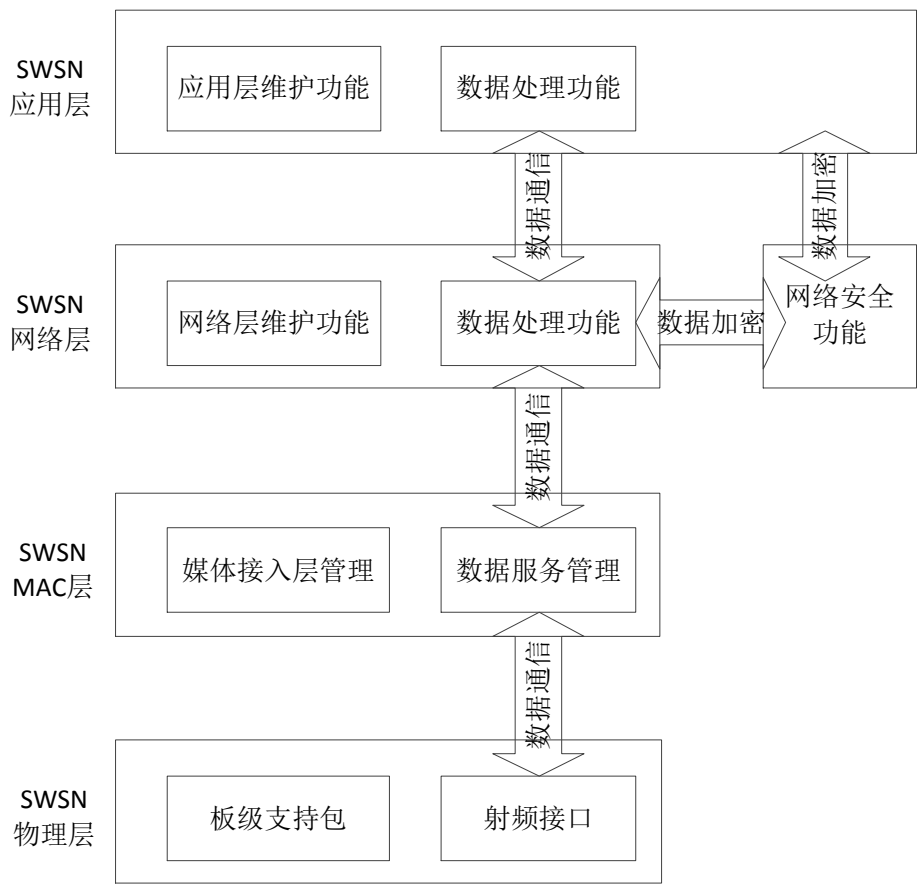


图 3.16 SWSN 网络协议架构

**SWSN 的物理层：**具有板级支持包和射频接口两部分，板级支持包指的是相应 MCU 的资源，比如 MCU 的时钟、SPI 和串口等外设的接口、系统的定时器、系统初始化和相应的中断处理。可以支持不同的硬件平台。而射频接口的主要功能则是实现射频模块数据的收发和信息读取，判断是否发送和接收完成数据。

**SWSN 的 MAC 层：**具有媒体接入控制层的管理和数据服务管理两部分组成，其中媒体接入控制层主要实现的是对信道资源的分配和信道检测机制，降低数据发生数据碰撞的概率。数据服务管理，则是将数据在物理层做一次过滤，对于不满足物理层帧格式的数据，不在进行网络层的处理。

**SWSN 的网络层：**主要职能是对上一层，也就是 MAC 层的数据进行解析处理，或者将数据传递给 MAC 层。对应用的数据进行网络层数据的打包，或者将网络层的数据解析出应用层的数据。在网络层中主要由网络层维护功能、数据处理功能和网络安全功能。

**SWSN 的应用层：**主要包括应用层维护功能和数据处理功能两个部分。与相应的上位机进行适配、配置网络中的相关参数和对网络通信进行评估。实现数据的采集和网络设备的管理等。

如图 3.17 所示，是本系统所设计的 LoRa 无线网络数据通信帧格式。

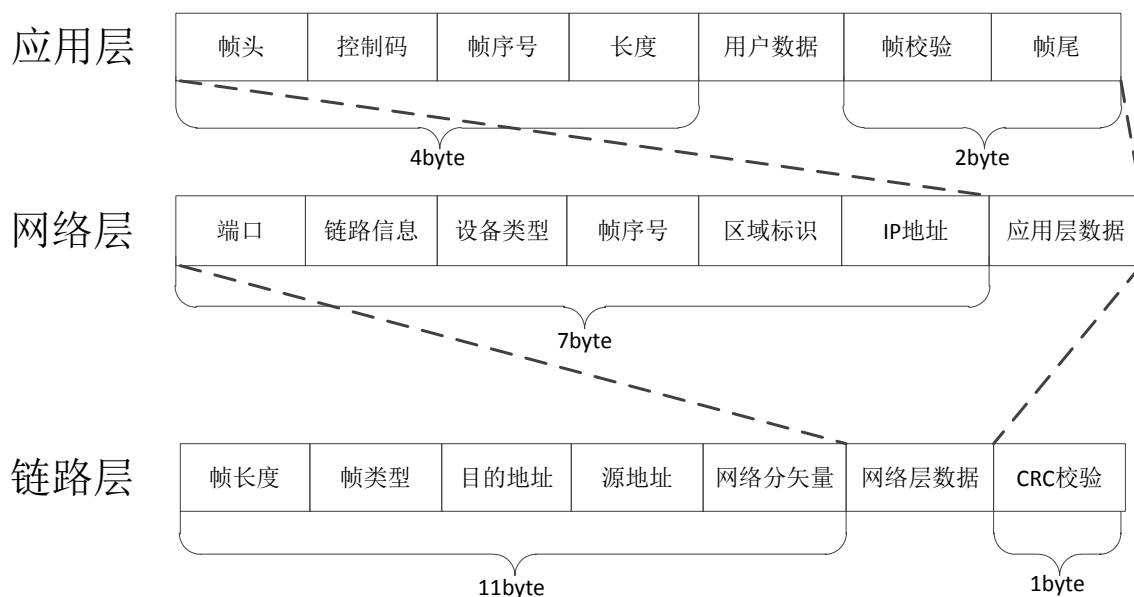


图 3.17 LoRa 组网内部数据帧格式

LoRa 无线网络通信协议中所涉及的数据帧格式如图所示。通过 LoRa 物理层接收到的数据,遵循相应的帧格式,包含了 MAC 层、网络层和应用层的数据。MAC 层和上文设计一致。在网络层中:是由端口、链路信息、设备类型、帧序号、区域标识和应用层数据组成。在应用层中,和 SPD 监测系统中的应用层一致。接收数据的时候,会一层一层进行解析,同理在发送信息时也需要将数据一层一层的打包,在进行数据的发送。

### 3.3.3 通信流程的设计

本系统中设计到节点入网和网关和节点数据通信两部分,具体流程图如下图 3.18 所示。节点入网由两个步骤组成,第一步:节点向网关发送 join 帧,join 帧中带有 Join 令牌标识。网关收到相应的 join 帧,检验令牌标识正确与否。如果正确回复一帧 Join Reply,这一帧中包含相应的 link 帧的令牌标识,否则不进行回复。第二步:节点将网关中的 Join Reply 帧中的 link 令牌标识取出,放入 link 帧中,发给相应的网关。网关收到进行解析,判断 link 令牌标识正确与否,若正确回复 link reply 帧,包含网络 IP 的信息,这样入网成功。数据通信部分采用了 RTS/CTS 的方案,实现网关和节点一应一答得机制。所有得通信流程都是经过 XTEA 的算法进行加密的。

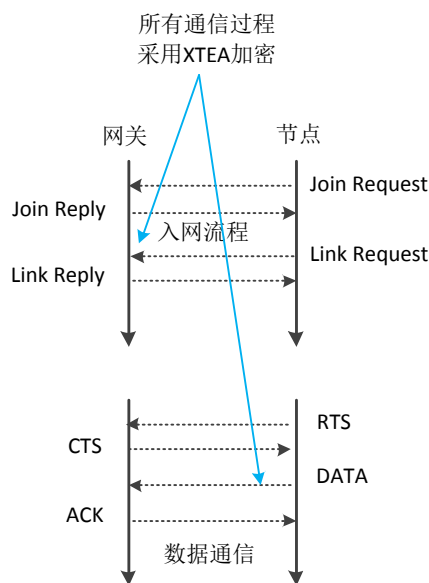


图 3.18 数据通信流程

除了上文所提到的 CSMA/CA 对信道监测的重发机制之外，还引入了应用层的数据重发机制。当信道检测失败的次数大于最大的信道检测次数时，认为该次数据通信失败。将该帧的数据写入失败发送缓存区，在一定的时间后再进行数据的重发。数据的重发和首次发送数据流程一样，如果再次达到最大的信道检测次数，则丢弃该帧。应用层数据失败重发机制的处理流程如下图所示。

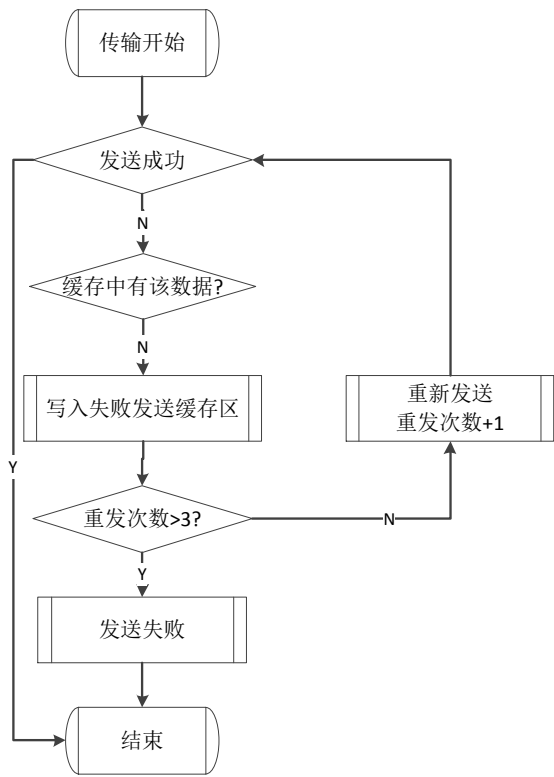


图 3.19 应用层失败重发

### 3.4 基于时隙 CSMA/CA 算法的无线网络性能测试

#### 3.4.1 算法的仿真与性能分析

NS2 (Network Simulator 2) 软件, 是一款可以用来仿真无线网络中的数据收发网络模拟软件<sup>[50]</sup>。可以通过修改相应的程序实现设置节点数目, 发包频率和网络拓扑等限制条件。根据 trace 文件分析观察数据的发送、接受、丢包等信息, 编写相应的 awk 脚本文件, 可以很方便的分析协议的通信时延、网络丢包率、网络吞吐量等性能。NS2 上集成了很多种 MAC 层协议, 比如常用的 CSMA、SMAC、以及 IEEE802.11 和 IEEE802.3 等协议。可以在 NS2 软件上模拟 LoRa 低速率的特性, 基于 IEEE802.11DCF 进行设计, 去修改相应的底层, 退避方案如下。

(1) 当节点访问信道, 参与信道竞争, 接入失败时, 竞争窗口的值以两倍的速度递增, 用来摆脱冲突。当竞争失败的次数超过一定的次数时, 丢弃该帧, 认为所处的通信范围内没有目标节点, 传输失败

(2) 当节点访问信道成功时, 竞争窗口按以下方案调整。

当目前的竞争窗口小于中间值  $CW_{mid}$  的时候, 可以理解为当前信道竞争较为缓和, 竞争窗口的值减去  $d$  值, 目的是为了使得竞争窗口波动缓和一些, 避免窗口递减过大。当目前的竞争窗口值比竞争窗口中间值  $CW_{mid}$  大时, 可以理解成当前信道竞争比较激烈, 竞争窗口按  $1/4$  的速率递减, 当小于等于  $CW_{mid}$  时, 回到递减  $d$  值的状态。

为了评估算法的优化效果, 需要与原二进制退避算法进行对比, 比对端到端的延时和公平性指数 FI(Fairness Index)。根据公式 (3-10)。公平性指数是指网络中的节点最大链路吞吐量 ( $Throughput_{max}$ ) 和最小链路吞吐量 ( $Throughput_{min}$ ) 的比值<sup>[51]</sup>, 用来反应当前网络里的链路吞吐量的变化情况。FI=1 时, 各个链路的吞吐量情况相当, 所有节点公平地利用信道资源。而 FI 越大时, 说明链路吞吐量情况差异比较大, 网络越发不公平。

$$FI = \frac{Throughput_{max}}{Throughput_{min}} \quad (3.10)$$

本节使用了 20 个节点, 在 500m\*500m 的仿真场景中各个节点处于静止的状态, 路由协议选用 AODV, 依次增加竞争节点的数目, 仿真 500s, 取 10 次的平均值。仿真的参数如下表所示。

表 3.3 仿真具体参数

|                    |    |                 |     |
|--------------------|----|-----------------|-----|
| 竞争窗口 $CW_{min}$    | 7  | 竞争窗口 $CW_{max}$ | 127 |
| 竞争窗口中间值 $CW_{mid}$ | 32 | RTS/Byte        | 12  |
| DIFS/ms            | 65 | 数据包 DATA/byte   | 64  |
| SIFS/ms            | 5  | CTS/Byte        | 12  |
| Slot 时隙/ms         | 30 | ACK/Byte        | 11  |

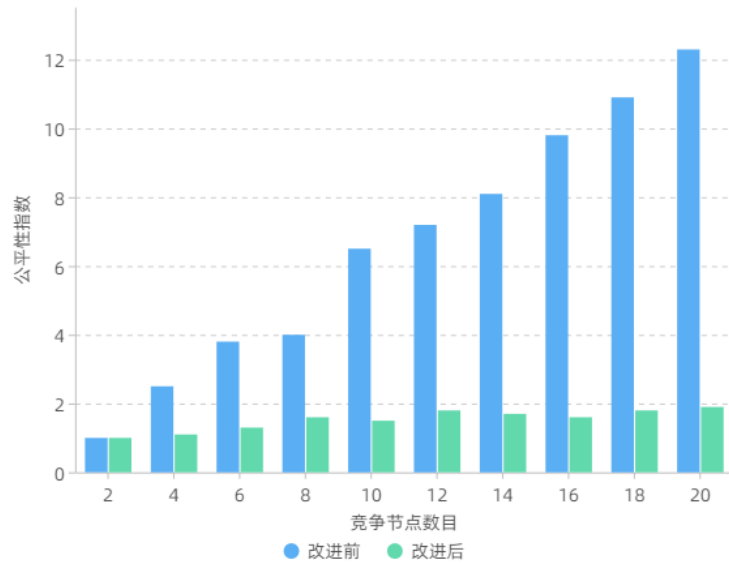


图 3.20 竞争节点和公平性指数

由图 3.20 可以得知，当节点的数量增加的时候，参与竞争的数量会越来越多，导致网络的负载越来越大，传统的二进制退避算法的公平性指数一直成递增的局势，增加幅值很大，而改进后的，公平性指数变化平缓。这是因为节点可以动态的调整竞争窗口的值，使竞争窗口值，在最优的范围内波动，不会出现某个节点的链路吞吐量过低或者过高。

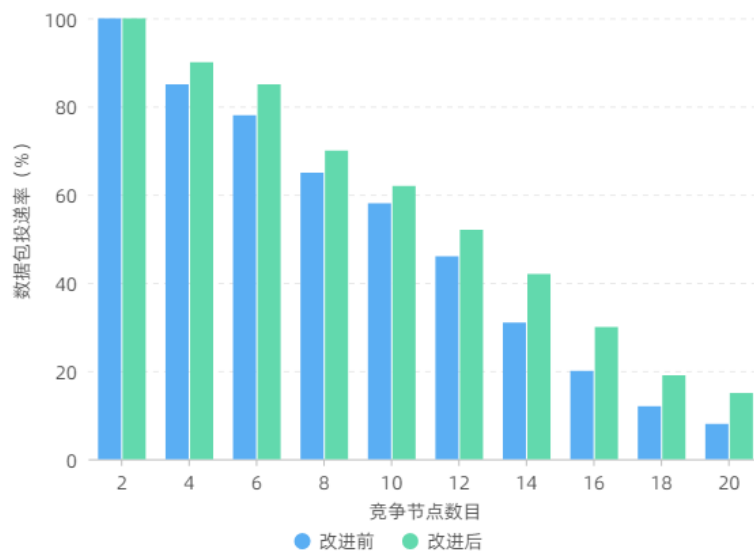


图 3.21 竞争节点和数据包投递率

由图 3.21 可以得知，当网络中参与竞争节点的数目越多，网络中的数据包投递率整体呈下降的趋势。因为设备数目的增加，各个设备抢占信道同时发送数据的概率增高，会导致了数据包冲突的可能性增加，丢包率也随之增加，则投递成功率就会降低。而改进后的算法比

原有的二进制退避算法，在同一个竞争级别时，投递率要有所提高。

3.4.2 多节点定时上报测试

测试场地：室内办公室。

测试方法：共有 10 个节点，节点定时上报信息给网关，网关通过上位机进行记录接收到的各个节点帧的计数。上报完 1000 帧，停止上报。记录各个节点得丢包率。具体图如下所示。

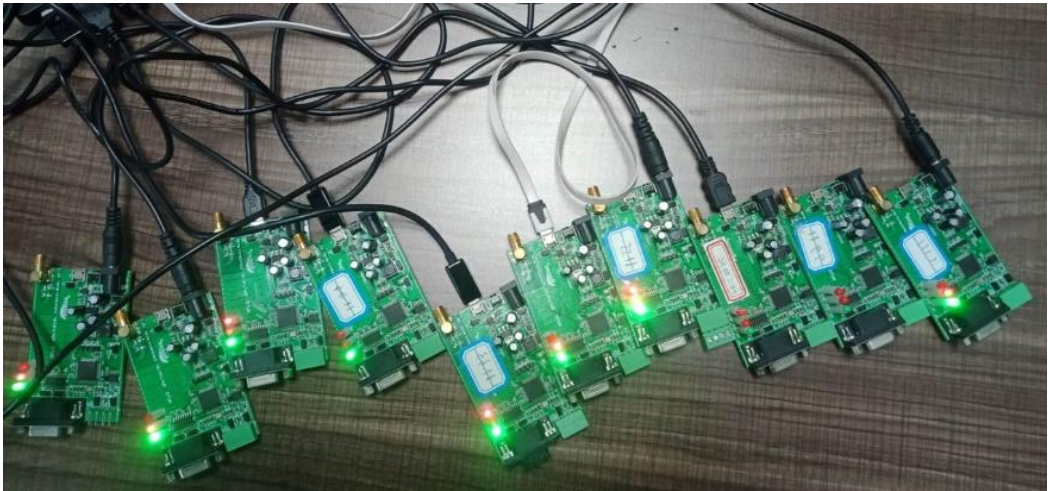


图 3.22 节点上报实物图



图 3.23 丢包率统计上位机

(1)第一组实验：记录传统的二进制退避算法，和优化过后的算法进行对比实验，采用定时 2S 周期进行测试，网关统计接收数据的帧数。

表 3.4 传统算法与优化后算法数据对比表

| 节点编号 | 传统算法 收/发 | 丢包率   | 优化算法 收/发 | 丢包率   |
|------|----------|-------|----------|-------|
| 1    | 853/1000 | 14.7% | 880/1000 | 12%   |
| 2    | 887/1000 | 11.3% | 878/1000 | 12.2% |

续表 3.4

|    |          |       |          |       |
|----|----------|-------|----------|-------|
| 3  | 886/1000 | 11.4% | 882/1000 | 11.8% |
| 4  | 867/1000 | 13.3% | 883/1000 | 11.7% |
| 5  | 888/1000 | 11.2% | 875/1000 | 12.5% |
| 6  | 893/1000 | 10.7% | 877/1000 | 12.3% |
| 7  | 865/1000 | 13.5% | 878/1000 | 12.2% |
| 8  | 868/1000 | 13.2% | 882/1000 | 11.8% |
| 9  | 873/1000 | 12.7% | 880/1000 | 12%   |
| 10 | 878/1000 | 12.2% | 876/1000 | 12.4% |

由上表可以得知，传统二进制退避算法中丢包率最高为 14.7%，最低为 10.7%。反应了节点抢占信道具有不公平现象，而优化过后的算法，丢包率均匀分布在 11.7%~12.4%之间，可以看出节点接入信道比传统算法公平。

(2)第二组实验：节点分别采用 3s、4s、5s 的定时时间上报。均采用优化过后的 CSMA/CA 算法。

表 3.5 定时上报测试

| 节点编号 | 定时 3S 收/发 | 丢包率  | 定时 4S 收/发 | 丢包率  | 定时 5S 收/发 | 丢包率  |
|------|-----------|------|-----------|------|-----------|------|
| 1    | 983/1000  | 1.7% | 1000/1000 | 0%   | 998/1000  | 0.2% |
| 2    | 982/1000  | 1.8% | 998/1000  | 0.2% | 1000/1000 | 0%   |
| 3    | 986/1000  | 1.4% | 1000/1000 | 0%   | 1000/1000 | 0%   |
| 4    | 984/1000  | 1.6% | 1000/1000 | 0%   | 1000/1000 | 0%   |
| 5    | 988/1000  | 1.2% | 998/1000  | 0.2% | 1000/1000 | 0%   |
| 6    | 993/1000  | 0.7% | 1000/1000 | 0%   | 999/1000  | 0.1% |
| 7    | 989/1000  | 1.1% | 1000/1000 | 0%   | 1000/1000 | 0%   |
| 8    | 988/1000  | 1.2% | 999/1000  | 0.1% | 1000/1000 | 0%   |
| 9    | 991/1000  | 0.9% | 997/1000  | 0.3% | 998/1000  | 0.2% |
| 10   | 988/1000  | 1.2% | 1000/1000 | 0%   | 1000/1000 | 0%   |

从上表可以看出，当定时周期较短时，网络服务是比较拥挤，信道访问失败的次数会增加。应用层的重发机制，也无法降低丢包率。定时周期为 4S 和 5S 的时候，相差不大，这是因为有足够的时间窗口去处理信息和竞争接入信道。本套系统的满负载的情况下，推荐定时上报周期为 4S 及 4S 以上。

### 3.4.3 系统检测机制测试和通信距离测试

#### 3.4.3.1 信道检测机制测试

设置 LoRa 的参数项如下表 3.6 所示。

表 3.6 LoRa 详细参数

| 性能性 | 扩频因子 SF | 带宽 BW  | 发送功率  | CR 编码率 |
|-----|---------|--------|-------|--------|
| 参数  | 7       | 125KHZ | 20dbm | 4/5    |

应用层 200 个字节，加上网络层头部和链路层的头部，总共为 225 个字节。一台节点负责发送这些数据，另一台节点专门负责检测信道。测试 20 次，测试信号占用信道的时间和 CAD 模式是否能检测到数据。记录 CAD 检测的次数，判断 CAD 除了能检测 LoRa 的前导码是否能检测有效信息。测试环境如下图：



图 3.24 实物测试场景

表 3.7 CAD 检测表

| LoRa 检测信道的次数 | 上下文时间 (ms) | CAD 检测的次数记录 | CAD 检测次数记录(次) |
|--------------|------------|-------------|---------------|
| 14 次         | 115ms      | 12          | 115 次         |
| 3 次          | 116ms      | 5           | 116 次         |
| 3 次          | 114ms      | 3           | 114 次         |

LoRa 检测上下文的平均时间为 115ms，CAD 检测的次数记录平均为 115 次。按上文所述，理论的 CAD 检测周期为 2.368ms，CAD 总周期时间为 272.32ms，信道占用时间为上下文的时间与 CAD 总周期时间之和。通过计算得到信道占用时间为 387.32ms。通过理论的计算器算得的时间为 351.49ms，如下图所示。

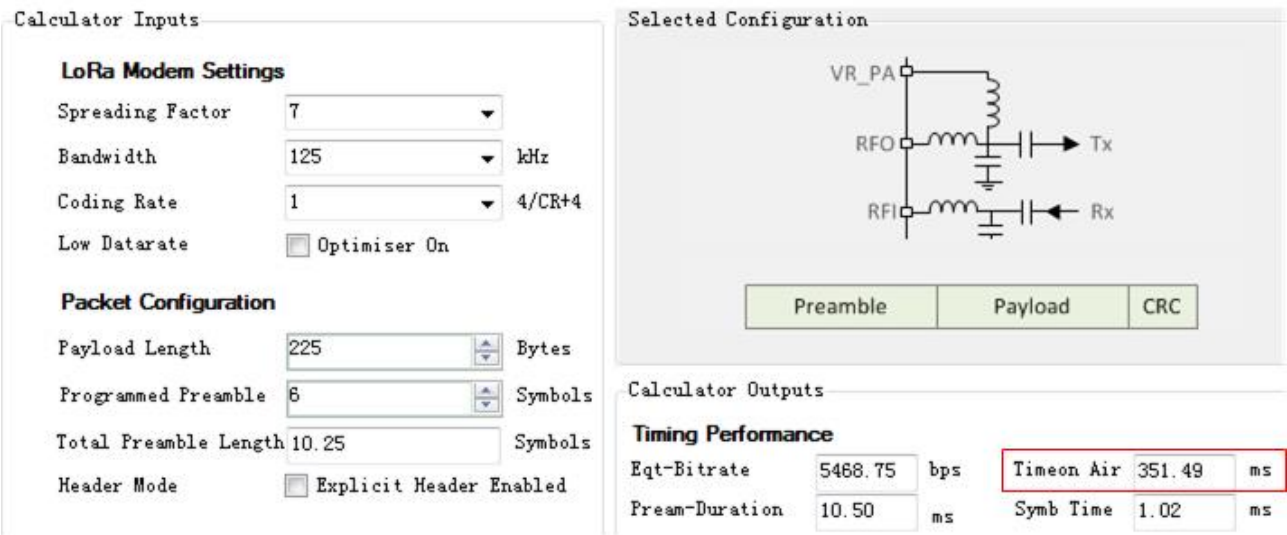


图 3.25 LoRa 参数计算

实际测试值约大于理论值，可以得到结论，CAD 不仅仅能检测前导码，也能够检测有效负载。

3.4.3.2 不同的 SF 和带宽网络通信总耗时

设备连接结构：上位机软件 A 和网关通过串口连接，节点和上位机软件 B 连接。上位机为自主设计的上位机，主要用来计算发送和接收之间的时延。

数据流走向：上位机软件 A→网关→节点→上位机软件 B，上位机软件 B→节点→网关→上位机软件 A。

网络往返耗时：无线网络中集中器和节点的通信时间，包括两次通信，集中器→节点，节点→集中器。

串口耗时：4 次串口通信时间。用户数据为 40 个字节，总数据为 65 个字节。

总耗时：整体完成通信的时间为网络往返耗时与串口耗时之和。



图 3.26 测试实物图

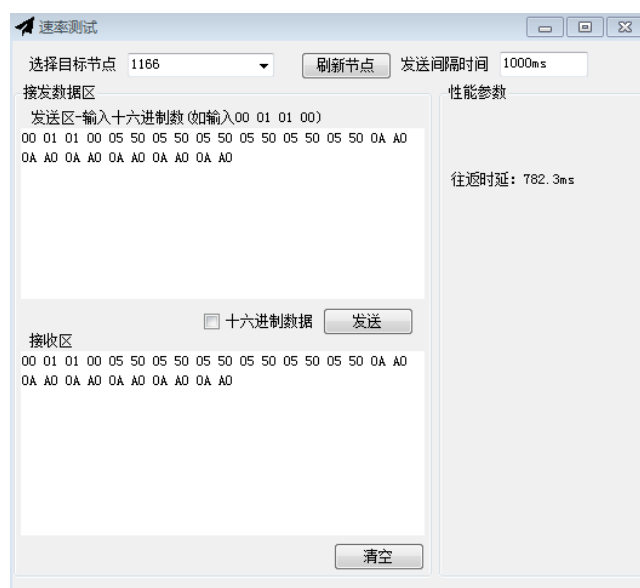


图 3.27 上位机透传页面

表 3.8 不同参数下的传输时间 (ms)

| SF \ BW (KHZ) | 7     | 8     | 9      | 10     | 11     | 12      |
|---------------|-------|-------|--------|--------|--------|---------|
| 31.2          | 542.4 | 997.5 | 1828.2 | 3330   | 6345.7 | 12675.3 |
| 41.7          | 407.9 | 749.2 | 1372.3 | 2498.5 | 5001.6 | 10321.1 |
| 62.5          | 273.3 | 500.8 | 916.2  | 1667.1 | 3335.8 | 5781.2  |
| 125           | 138.7 | 252.5 | 460.2  | 835.7  | 1670   | 3015.9  |
| 250           | 71.4  | 128.3 | 232.2  | 419.9  | 837.1  | 1510    |
| 500           | 37.8  | 66.3  | 118.2  | 212    | 420.7  | 757.1   |

测试数据基本和数据手册中说明一致，发现扩频因子和带宽影响通信的质量。在相同的 SF 下时，BW 增大会导致传输时间越长。在相同的 SF 条件下，BW 越大，传输时间越短。SF 为 7 和 8 时，BW 高于 62.5KHZ 时，通信时延小于 0.2S。

### 3.4.3.3 不同楼层下的通信距离测试

通过上述测试，挑选合适的参数进行不同楼层下的不同 SF 和带宽测试，本次测试在室内环境下测试了 LoRa 通信性能，测试地点为杭州下沙高科技孵化园区 B 幢 17~20 楼的通信情况。本次 LORA 在室内场地测试，网关的位于 D 区域的 5 号房间内，SF 分别为 7 和 8，带宽为 125KHz、250KHz 的通信情况，7 表示 125KHz，8 表示 250KHz，CR 均为 4/5，Tx\_Power 均为 20dBm，天线离地高度 1.5 米。节点在 18 19 20 楼进行固定点测试，比较不同参数的通信距离。上位机统计收发数据的页面如下图所示。



图 3.28 测试场景框图

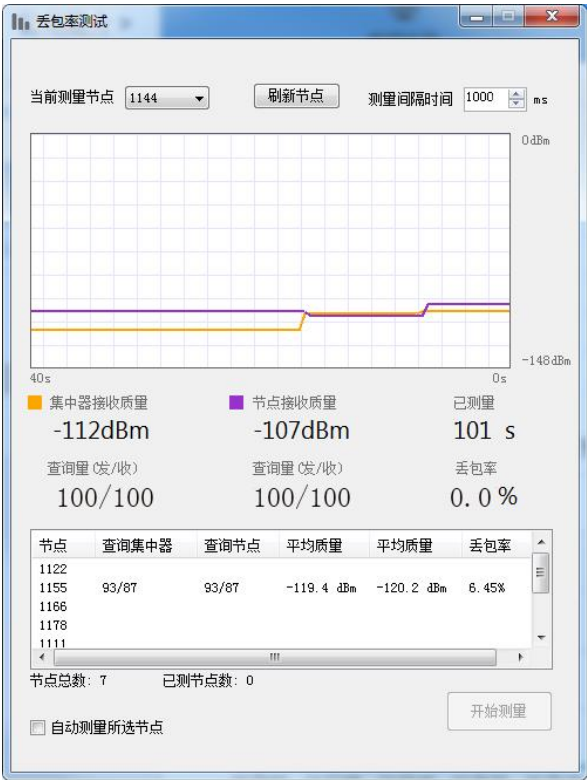


图 3.29 上位机统计页面

表 3.9 不同楼层下的通信距离测试

| 不同楼层下的通信距离测试 |         |        |         |      |     |
|--------------|---------|--------|---------|------|-----|
| 垂直距离（米）      | 水平距离(米) | SF /带宽 | 发/收     | 定时发送 | 丢包率 |
| 0（位于 17 楼）   | 80      | 7/7    | 100/100 | 1S   | 0%  |
|              | 80      | 7/8    | 100/100 | 1S   | 0%  |
|              | 80      | 8/8    | 100/100 | 1S   | 0%  |

续表 3.9

|        |    |     |         |    |     |
|--------|----|-----|---------|----|-----|
| 3 隔一层楼 | 50 | 7/7 | 100/100 | 1S | 0%  |
|        | 50 | 7/8 | 100/100 | 1S | 0%  |
|        | 50 | 8/8 | 100/100 | 1S | 0%  |
| 3 隔一层楼 | 60 | 7/7 | 100/100 | 1S | 0%  |
|        | 60 | 7/8 | 100/100 | 1S | 0%  |
|        | 60 | 8/8 | 100/100 | 1S | 0%  |
| 6 隔二层楼 | 30 | 7/7 | 100/100 | 1S | 0%  |
|        | 30 | 7/8 | 100/100 | 1S | 0%  |
|        | 30 | 8/8 | 100/100 | 1S | 0%  |
| 6 隔二层楼 | 40 | 7/7 | 100/100 | 1S | 0%  |
|        | 40 | 7/8 | 100/100 | 1S | 0%  |
|        | 40 | 8/8 | 100/100 | 1S | 0%  |
| 9 隔三层楼 | 15 | 7/7 | 100/15  | 1S | 85% |
|        | 15 | 7/8 | 100/17  | 1S | 83% |
|        | 15 | 8/8 | 100/13  | 1S | 87% |
| 9 隔三层楼 | 20 | 7/7 | 100/3   | 1S | 97% |
|        | 20 | 7/8 | 100/7   | 1S | 93% |
|        | 20 | 8/8 | 100/8   | 1S | 92% |

挑选出传输时间与传输距离，满足本项目需求点进行测试。这里对比了传输时间和传输距离，以及接收灵敏度和抗干扰能力的考虑，选用 SF 为 7，BW 为 125KHZ 的参数。在楼宇中水平距离测试，通信可以达到 80 米左右；在楼宇中的垂直距离测试，通信距离可隔两层楼。网关放在位于中间层的楼层，可以监测上下五层楼的 SPD 运行情况。

#### 3.4.3.4 测试户外广阔地面的通信距离

测试场地：空旷的场地。

测试方法：节点固定在一个地方，网关连接上位机，定时向节点发送数据，每次测试间隔 100 米。

射频的参数：SF 为 7，其他参数同上，天线离地高度 1.5 米。



图 3.30 实测场地

表 3.10 空旷户外测试数据

| 测试水平距离（米） | 发/收     | 定时发送 | 丢包率 |
|-----------|---------|------|-----|
| 300       | 100/100 | 1S   | 0%  |
| 400       | 100/100 | 1S   | 0%  |
| 500       | 100/100 | 1S   | 0%  |
| 700       | 100/100 | 1S   | 0%  |
| 800       | 100/100 | 1S   | 0%  |
| 900       | 100/100 | 1S   | 0%  |
| 1000      | 100/100 | 1S   | 0%  |
| 1100      | 100/100 | 1S   | 0%  |
| 1200      | 100/100 | 1S   | 0%  |
| 1300      | 100/100 | 1S   | 0%  |
| 1400      | 100/95  | 1S   | 5%  |
| 1500      | 100/73  | 1S   | 27% |
| 1600      | 100/52  | 1S   | 48% |
| 1700      | 100/0   | 1S   | --  |

由上表可知，空旷的环境当中，通信距离可以在 1300 米左右，根据本项目的需求，可以实现空旷环境下对户外 SPD 的监测。

3.4.3.5 测试户外有遮挡物的通信距离

测试场地：附近小区，如下图所示。

测试方法：节点固定在一个地方，网关连接上位机，定时向节点发送数据，测试完成，移动到下一个建筑物后方。

射频参数：同上，天线离地高度 1.5 米。



图 3.31 实测场地

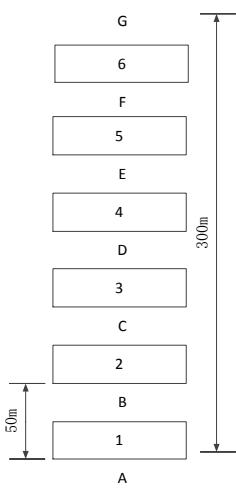


图 3.32 场地说明框图

测试数据如下表所示：

表 3.11 户外有遮挡物距离测试

| 网关位置 | 节点位置 | 相隔栋数 | 发/收     | 定时发送 | 丢包率 |
|------|------|------|---------|------|-----|
| A 点  | B 点  | 1    | 100/100 | 1S   | 0%  |
| A 点  | C 点  | 2    | 100/100 | 1S   | 0%  |
| A 点  | D 点  | 3    | 100/100 | 1S   | 0%  |
| A 点  | E 点  | 4    | 100/100 | 1S   | 0%  |
| A 点  | F 点  | 5    | 100/100 | 1S   | 0%  |
| A 点  | G 点  | 6    | 100/95  | 1S   | 5%  |

由上表可知，在户外有楼宇遮挡的情况下，在相隔 5 栋楼时，可以保证数据无丢包。根据本项目的需求，可以实现对小区中的 SPD 设备监测。

3.5 本章小结

本章基于时隙 CSMA/CA 的算法原理和 workflows 进行了介绍，首先分析了退避算法中的不公平性。针对原有的退避算法进行了改进，通过仿真进行了分析。结合本项目的特点，采用 LoRa 扩频调制的特性和相关网络拓扑结构，进行了相关 mac 协议和网络协议的设计。再经过实际测试，进行了性能的分析，选择合适的参数。对整个无线通信网络进行了户外，室内，多节点定时上报的测试。测试结果表明，通信时延小于 0.2S，节点和网关在同一楼层进行数据传输时的可靠通信距离最大水平距离约为 80 米，其信号可靠的穿透范围为两层楼，室外无遮挡物的通信距离可达 1.3km。有遮挡物时，和遮挡情况有关，本次测试有效通信距离可以达到 250 米。同时在可靠的通信范围内，原有的退避算法和改进算法进行实物测试对比，在 10 个节点定时上报的情况下，改进算法比原有的退避算法公平性表现的较好，多个节点定时 4S 向网关上报与数据时，丢包情况很低，表明该系统所设计的无线网络具有较高的公平性和可靠性。

第4章 多层式协作提高系统的稳定性和效率性

在设计本系统时和参考日常的物联网网关和无线组网的设计当中，存在以下几点问题：

- (1) 基本都是透传的模式，即将数据原封不动的进行下发，对于本套系统来说的话，就是物联网网关发帧到节点，节点通过 modbus 协议透传给相应的数据给 SPD 监测设备。SPD 监测设备将信息回复给节点，节点再将信息返还给网关的一个流程。SPD 监测设备的信息，通过一帧 modbus 帧，无法查询完全，需要连续发两到三帧，才能将 SPD 监测设备的信息读取完全。一个节点下挂多台 SPD 监测设备，一个网关下挂多台节点，这样的查询效率不高。本项目在小区或者楼层中，需要监控的设备比较多，需要考虑信息实时性的问题。
- (2) 在整个系统中，设备需要与云平台保持长连接<sup>[52]</sup>和实时数据上报，就会造成数据流量使用过多的问题，主要流量的使用还是数据上报的部分，频繁的上报数据，一些冗余数据，对于流量资源来说是一种浪费。系统未优化前，流量资费损耗严重。
- (3) 只要涉及到物联网的部分，网络情况的不稳定，是必须要考虑的因素，当网络条件不佳时，或者意外断网，一部分数据就可能丢失了。在本系统前期存在离线数据丢失的情况。
- (4) 物联网的设备，在已经安装使用在实际的应用环境中，有新的需求加入或者有需要优化的部分，需要对网关进行软件的更新，如果升级过程被打断，网关设备有程序崩溃的风险。本系统在早期程序更新时，时间花费比较长。

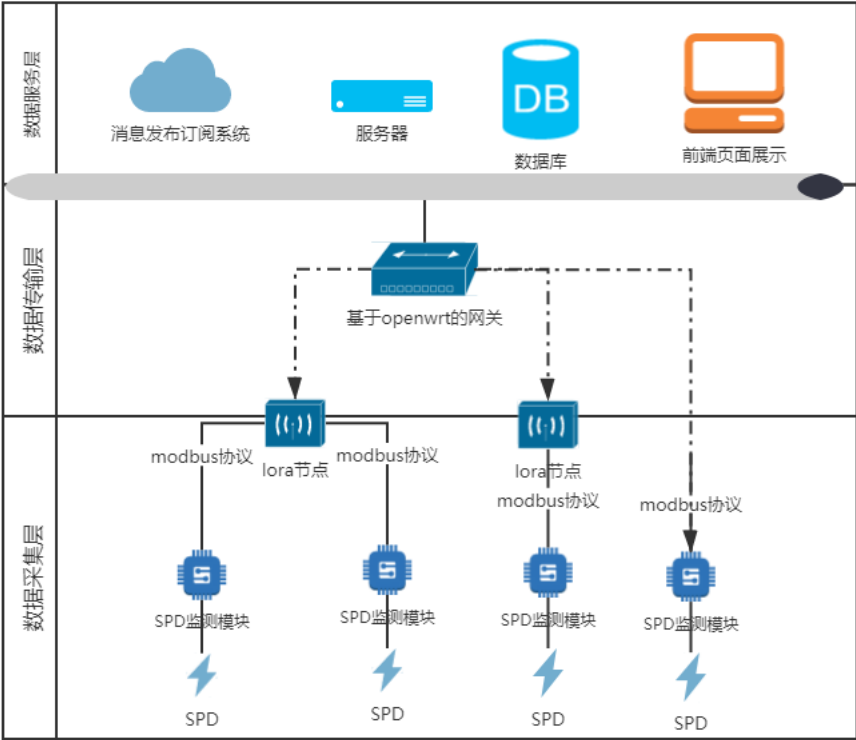


图 4.1 多层式协作框图

本文基于以上的问题，提出一种多层协作的方式，如上图 4.1 所示，数据采集层主要负责采集 SPD 监测设备的信息。数据传输层的功能，对上与数据服务层通信，对下收集数据采集层的信息。数据服务层则是进行数据汇总、分析和展示，和对网关进行远程的程序更新。

在数据采集层时：通过对节点的分组，让节点担任主动查询的角色，将信息在节点做一次数据缓存。通过调整相应的地址位，和数据长度位，进行 modbus 的数据连读操作，同时制定了相应的帧格式。在数据的传输层时：也做了一次缓存，将各个节点和 SPD 监测设备的信息进行缓存，当网络情况不稳定时，出现离线状况时，采集系统仍然在运行，将数据进行带有本地时间进行保存，当网络恢复正常时，实现数据的上报，同时提供数据汇聚的上报方式，减少了通信的开销。在数据服务层：可以通过选择定时上报时间和差异化上报方式，告知网关定时差异化上报，可在一定条件下减少流量的消耗。为了使系统更加稳定和操作的便利性，设备异常时进行替换，和对远程更新的方式进行了优化<sup>[53]</sup>。

## 4.1 数据采集层处理机制

### 4.1.1 数据连读策略

用户可以通过前端页面进行选择，勾选想要获取 SPD 监测设备的哪些寄存器中的数据，通过网关下发给节点，下发的指令中告知节点读取 SPD 监测设备的具体寄存器的地址和数据类型，节点存储相应的 modbus 设备的信息，将每个寄存器地址和数据单元设置为一个监控点。其特征就在于，利用 modbus 连读功能，重新将单个数据的读取，变为找到这些数据的连续规律，设置不同的间隔，读取相应的地址中的数据。将数据进行匹配，放入节点的缓存中。是一种通过排序，重新设置连续单元，来提升采集信息效率的方式。

如公式(4.1)所示  $address_{n-1}$  表示上一个监控点的地址， $address_n$  表示本次监控点地址，间隔阈值为 $\delta$ 。首先将地址按照从小到大进行排序， $address_n$  始终大于  $address_{n-1}$ 。当  $address_n$  与  $address_{n-1}$  的差值小于 $\delta$ 时，判定该地址与上一个监控点的地址是连续的。将  $address_{n-1}$  作为起始监控点， $address_n$  为终止监控点。

$$address_n - address_{n-1} \leq \delta \quad (4.1)$$

其中 $\delta$ 的阈值决定了数据连读的性能， $\delta$ 的值过小，监控点的数据过多时，仍然具有离散化的可能， $\delta$ 的值过大时，读取到的冗余数据增多，效率也会打折扣。

针对上述问题，设计了阈值计算的方案。通过逻辑代码寻找最优间隔阈值 $\delta$ ，找到轮询所有监控点所花费最短时间的 $\delta$ 值。如公式 4.2 所示，其中每次轮询完成所花费时间记为 $T_i$ ，所使用间隔阈值记为 $\delta_i$ ，将 $T_i$ 中的所花费时间最少的值所对应的 $\delta$ 值，记为最优值。 $\delta$ 的取值范围为 1~127。流程图如下图 4.2 所示。

$$\delta = \arg \min_{\delta_i} T_i(\delta_i), \quad \delta_i \in [1, 127] \quad (4.2)$$

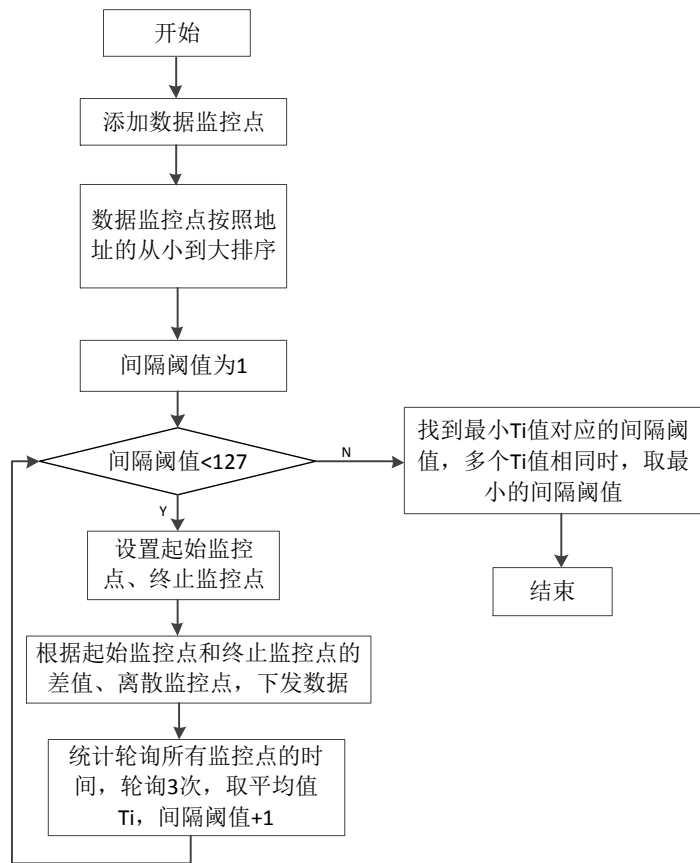


图 4.2 寻找最佳阈值

具体实现步骤如下：

第一步：节点中的数据格式如下图 4.3 所示，包含了寄存器的地址，数据类型，和数据单元。

|       |
|-------|
| 寄存器地址 |
| 数据类型  |
| 数据单元  |

图 4.3 数据格式单元

第二步：在前端页面进行监控点的选择，通过网关转发相应的信息的给节点，节点中添加的信息如下图 4.4 所示，地址是 4001、4006、4002、4005、4008、4012。

|      |      |      |      |      |      |
|------|------|------|------|------|------|
| 4001 | 4006 | 4002 | 4005 | 4008 | 4012 |
| 数据类型 | 数据类型 | 数据类型 | 数据类型 | 数据类型 | 数据类型 |
| 数据单元 | 数据单元 | 数据单元 | 数据单元 | 数据单元 | 数据单元 |

图 4.4 添加多个数据

第三步：节点将按照寄存器地址位数值的大小顺序进行排序。如下图 4.5 所示

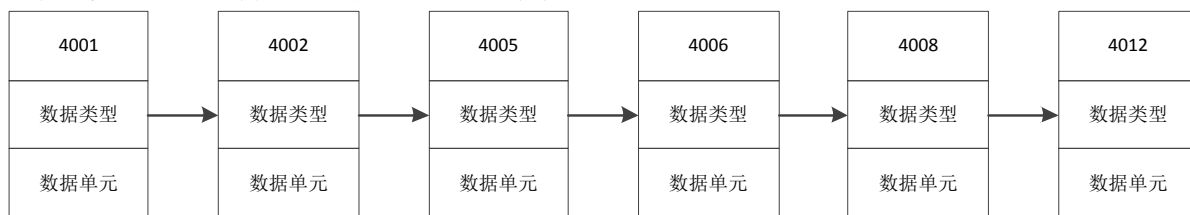


图 4.5 通过地址位排序

第四步：如下图 4.6 所示。设定 $\delta_i$ 值，根据 $\delta_i$ 值，找到了起始监控点和终止监控点。

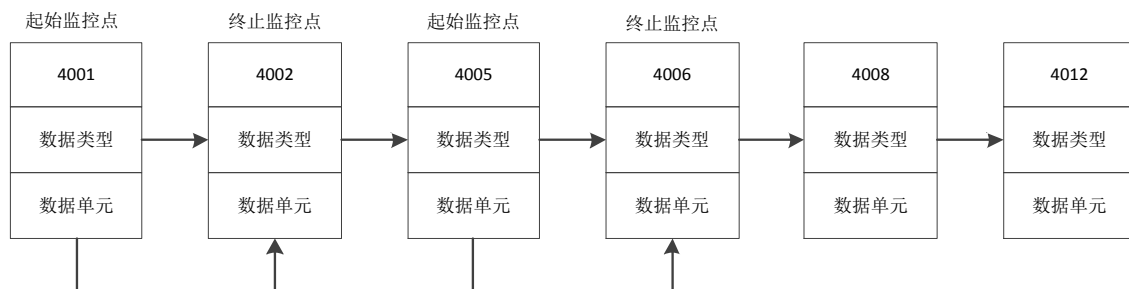


图 4.6 通过阈值排序

第五步：根据起始监控点和终止监控点的地址差值，判断要读多少个寄存器，重新设置读取的地址和读取数据的长度，组建相应的 modbus 数据查询帧，下发给相应的设备，返回的数据在逐个匹配到监控点中的数据单元中。数据单元在将这些寄存器的值放入缓存中。

第六步：计算轮询完成监控点一圈所花费的时间，轮询三次，计算平均值记为 $T_i$ 。重复第四步过程， $\delta_i$ 为最大值时终止。

第七步：统计 $T_i$ 中的最小值，对应的 $\delta_i$ 值，当 $T_i$ 中多个相等时，取 $\delta_i$ 的最小值。

如上述说明所示，当 $\delta$ 的值为 4 的时候，原来需要 4 次串口收发的过程，现在减少为 1 次。使得一帧查询到更多的信息，来提高查询的效率。

对于本系统来说，因为采集 SPD 监测设备中的信息是离散的，即地址 21 中保存着 SPD 失效监测的信息，地址 22 中保存着接地监测的信息，地址 27 中保存着前端保护检测的信息，地址 30 中保存着雷击次数的信息，地址 31 中保存着温度的信息。采用上述所提到的数据连读策略，原本需要 3 帧才能查询完 SPD 监测设备的信息，现在只需要发送 1 帧就完成了信息查询的功能。

#### 4.1.2 设备分组和信息混合策略

在无线网络中，网关下挂多台节点，一个节点下挂多台 SPD 监测设备。当网关下发指令给节点的时候，节点接收到指令信息进行解析，节点设备就能知道下挂了几台 SPD 监测设备，以及需要查询的每个 SPD 监测设备的栈号、寄存器地址和类型。节点就担任了透传网关的角色，按照保存的信息，依次轮询每一个 SPD 监测设备，将所得到的信息按照 SPD 监测设备进行分类，缓存各个 SPD 监测设备的信息和上一次查询 SPD 监测设备的信息。节点负责维

护 SPD 监测设备的信息，这样就分担了网关的查询负担。同时一个节点就相当于一个单元，汇聚了下面所有 SPD 监测设备的信息。网关只需向节点查询相应 SPD 监测设备的信息，省去了等待 SPD 监测设备查询回复的时间。

### 4.1.3 应用层中的帧格式设计

结合本项目的需求和 SPD 监测设备的特点，制定的数据格式如下图 4.7 所示。

| H  | C   | FRM | L  | N    | USER | CS  | T  |
|----|-----|-----|----|------|------|-----|----|
| 帧头 | 控制码 | 帧序号 | 长度 | 设备台数 | 用户数据 | 帧校验 | 帧尾 |

图 4.7 应用层帧细节展示

数据格式由帧头、控制码、帧序号、设备台数、用户数据、帧校验和帧尾所组成。在应用层传输的过程中都以字节为单位。这里为了描述方便，采用 16 进制进行说明。其中帧头标识为 0x68，帧尾标识为 0x16，各占了一个字节。

控制码标识该帧的属性，一个字节，字节中的每一位都带有特定的信息，从 D7 到 D0 表示从高位到低位。控制码的作用便于节点到网关之间的数据解析，具体如下图 4.8 所示。

| D7     | D6 | D5 | D4 | D3 | D2 | D1       | D0      |
|--------|----|----|----|----|----|----------|---------|
| 1: 告警帧 | 保留 |    |    |    |    | 0: 节点到网关 | 0: 数据读取 |
| 0: 普通帧 |    |    |    |    |    | 1: 网关到节点 | 1: 数据写入 |

图 4.8 功能码细节展示

帧序号占有一个字节，帧序号的作用，表示该帧在这一段的时间的唯一性。每一帧都有一个的帧序号，应用层可以依靠帧序号实现一应一答，或者用来调整接受帧的顺序，帧序号的重复，可以进行数据的过滤，防止重复处理。

长度位占一个字节，表示该帧除了帧头、控制码、帧序号后剩余的字节长度，数值范围 0~255，但是根据本系统的设计，后续的字节数据不会超过 60 个字节。通过对帧长度位的解析，可以计算出本帧的具体长度。

设备台数，使用了一个字节标识，在本系统中指的是，这一帧中包含的 SPD 监测设备的数量，比如设备台数为 2，则表示这一帧中有 2 台 SPD 监测设备的信息。

用户数据域，根据设备台数来调整和发起端是网关还是节点进行调整，这里以节点是数据发起端，进行描述。一台设备，需要一个用户数据域，两台时，则在第一台用户数据域后继续添加。具体如下图 4.9 所示，一个用户数据域中包含了 4 字节的 SPD 设备信息，栈号、报警、温度、雷击次数。报警单元包含了设备的在线离线信息、前端保护失效的报警、接地失效报警、SPD 失效报警。

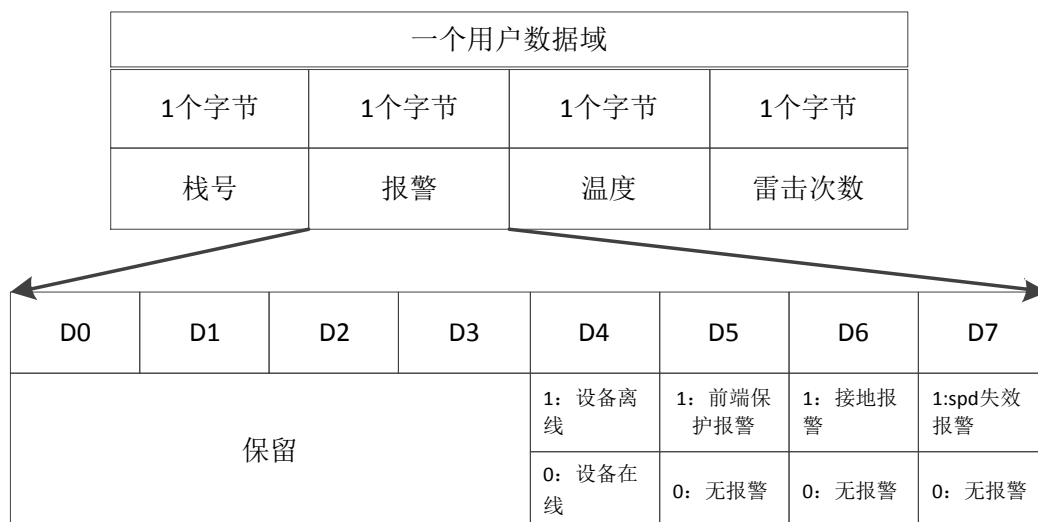


图 4.9 用户数据细节展示

帧校验采用的 checksum 的校验方式，判断此帧是否出现异常，占一个字节。

以上描述的是应用层帧协议，节点网关通过应用层的协议进行交互。需要通过帧的格式判断帧是否有效，当出现粘包和断包的情况下，按照帧头、帧长度位、帧校验位、帧尾，进行拆分。结合上面的分组策略，一台节点最多可以下挂 10 台 SPD 检测设备，可以查询到 10 个 SPD 监测设备的信息，一条应用层的帧最长为 47 个字节。如果直接采用 modbus 帧透传的方式，查询一台 SPD 监测设备，应用层一帧的开销就是 22 个字节。一个节点下挂 10 台 SPD 监测设备，一帧开销是 220 个字节。如果满负荷状态下，一个节点下挂 10 台监测 SPD，网络中存在 10 个节点。对于透传模式应用层帧的开销是 2200 个字节，基于本套系统，应用层帧开销 470 个字节。

## 4.2 数据传输层的处理机制

### 4.2.1 数据主动上报策略

基于透传模式下查询设备的状态信息，需要经过网关发帧到节点，节点将帧转发给 SPD 监测设备，SPD 监测设备收到信息再进行回复给节点，节点再将信息转发给网关。网关采用轮询的机制，按照定时的时间去轮询各个节点下的 SPD 监测设备，当网络中监测设备过多时，信息的实时性就打折扣。而基于数据采集层的策略，已经通过节点的分组，节点相当于网关，由节点去轮询下挂的 SPD 监测设备，节点中汇聚了下挂 SPD 监测设备的信息，为差异化上报和定时上报做准备。

(1) 差异化上报：比较各台数据是否出现变化，或者有无报警数据，如果数据出现上述两种情况，则需要立即执行上报数据给网关的操作，数据要遵循应用层的帧格式。

(2) 定时上报：数据无变化的一段时间，但是达到了定时周期的时间，节点会将下挂的 SPD 监测设备的信息，进行汇聚，组建相应的帧格式进行数据上报。

在整个网络中，因为有了节点分层的操作，所以一个网络中可以存在有相同的栈号的 SPD

监测设备的地址，前提是这些设备需要挂在不同的节点下面，增加了网络的容量。网关这边接收到节点汇聚上报的信息，也会开辟相应的缓存，进行数据缓存，也有差异化上报和定时上报的功能。

### 4.2.2 数据断网续传策略

当网络状况不好时，采集的数据就会丢失，存在丢失一部分重要数据的可能。这部分数据有可能就是分析定位 SPD 监测设备和 SPD 为何出现异常的关键数据所在。最开始采用的方案是由服务器进行时间维护，接受到数据，就认为是当前采集的数据。这样有一部分数据因为离线丢失，为了增强采集系统的稳定性，就需要保证数据是真实可靠的，那么采集到数据的时间就需要网关这边来维护，网关的时间需要与网络的时间进行同步，同时需要设计相应的缓存队列。

#### 4.2.2.1 时间同步机制

OpenWrt 中有 NTP（Network Time Protocol）时间网络协议的客户端，作用是网络中的各个计算机的时间进行同步。网关通过 4G、WiFi、以太网的联网方式连接到网络时，需要向网关中添加相应的 DNS 文件，保证能够进行域名解析。编写相应的脚本文件，将网关和网络时间进行同步，进行同步的流程图如下图 4.10 所示。

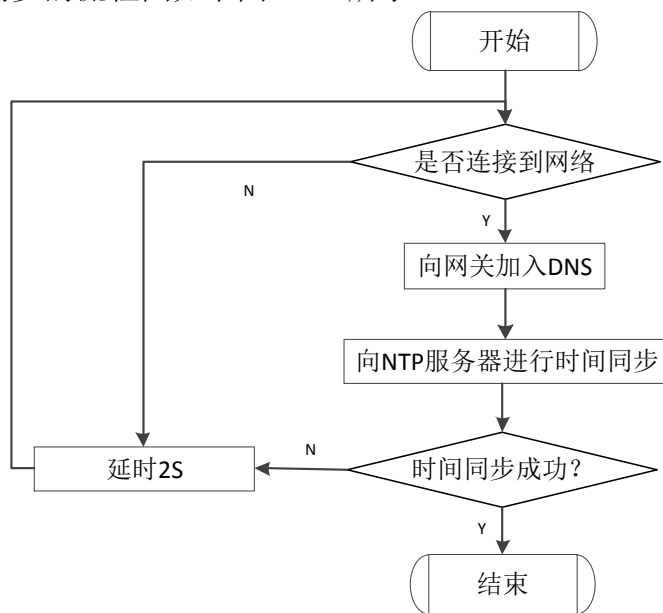


图 4.10 时间同步机制流程图

#### 4.2.2.2 离线缓存队列

当时间同步完成时，进行数据的采集，设计了离线缓存队列，其是一个常规存储的队列，不过当队列超过一定的数据量时，会删除头部数据，新来的数据依然插入到数据的尾部，不会出现内存溢出的问题，具体流程图如下图 4.11 所示。

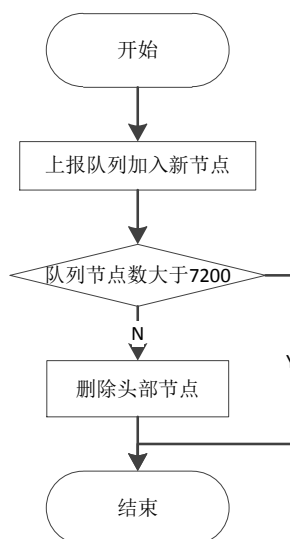


图 4.11 离线队列数据处理流程图

通过上述所提到的时间同步机制，保证当前时间和网络时间一致时。离线缓存中的数据单元的内容，内容的数据格式如下表 4.1 所示，表示 1000 这个监控点的在 2020 年 5 月 22 日 12 时 31 份 12 秒时的数据内容为 2234.5。

表 4.1 数据格式说明

|                                         |
|-----------------------------------------|
| {“1000”:{“2020-05-22 12:31:12”:2234.5}} |
|-----------------------------------------|

当网关因为网络问题，而产生离线时，节点汇聚 SPD 监测设备的信息，还在报往网关。网关会将这些离线的数据载入缓存中，等联网成功后继续上报给云平台。因为数据中含有相应的时间戳，依然可将当时采集的数据上报上去，云平台会按数据格式中的时间，进行数据的保存，并不会出现数据的丢失。

#### 4.2.3 网关离线检测和重连策略

网关与云平台建立连接，当没有数据上报时，会每 30S 定时向云平台上报呼吸帧，云平台收到上报的数据会给相应的回复。统计回复帧的条数，每一分钟检测回复帧的计数是否发生变化，如果有表示当前网络正常。如果没有，则表示当前网络异常，与云平台无法进行数据传输。同时云平台那边长时间没有收到网关的数据，则会断开连接。这时候如果是网络问题，会重新执行连接云平台的操作。离线检测机制可以在 1 分钟检测网络是否出现问题，重新连接上云平台的操作，只需要几秒钟的时间。如果是 4G、以太网、WiFi 出现问题，前文中也有相应的错误收集机制。

#### 4.2.4 数据汇聚上报策略

当数据进行上报的时候，单条进行上报，会造成很多额外的开销，如下图 4.12 所示。当有一条用户数据进行上报时，主要包括三个部分的首尾部：TCP/IP 首部，以太网的首部和尾部和 MQTT 应用层协议的首部。MQTT 的首部占了 2 个 byte，TCP/IP 固定首部占了 40 个 byte，不包含其他数据扩展的开销，而以太网的首部和尾部数据开销之和为 26 个 byte。这里面还没

有统计通信之前的 TCP 握手建立连接阶段的开销。也就是上报一条用户数据给云平台，至少会额外引入 68 个 byte 的数据开销。

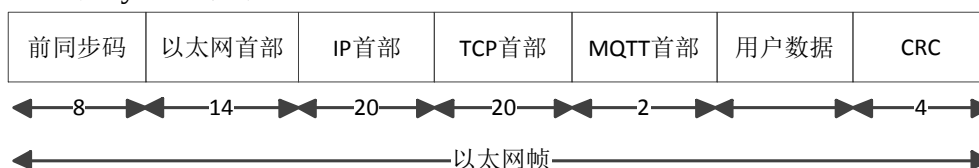


图 4.12 数据帧格式开销

为了减少额外的数据开销，同时使有效的数据也能够实时上报给云平台。设计了一个缓存空间，将要上报的数据放入缓存中，当缓存满时或者超过定时的时间时，则会将数据进行汇聚上报。具体流程图如下：

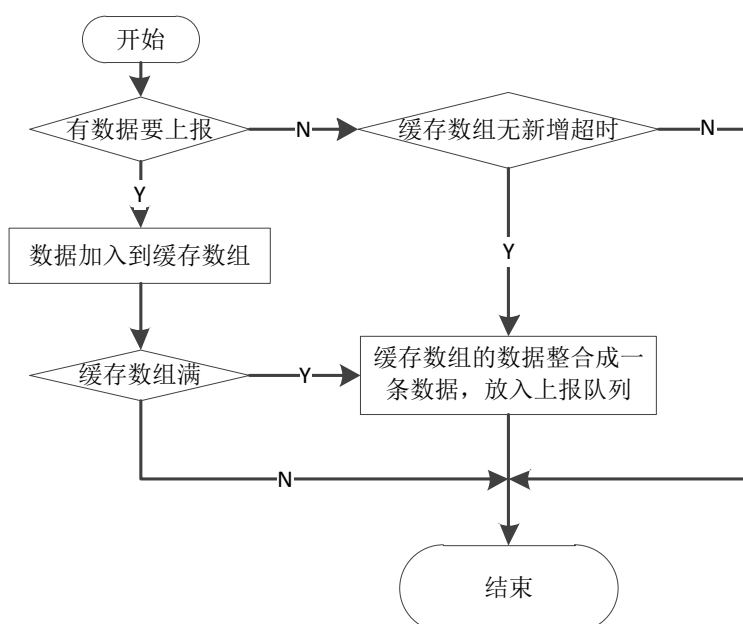


图 4.13 缓存数据处理流程图

将多条数据汇聚进行上报，相比一条一条进行上报，减少了传输中信息头部的数据开销，云平台也能进行数据解析，可以节省相应的流量资费。

### 4.3 数据服务层的处理机制

网关下挂多个节点设备，以及节点下挂多个 SPD 监测节点的设备，在实际使用过程中，在前端页面上逐一设备进行添加下发给网关，这时候网关中的信息保存了整体的网络结构的信息。当节点出现异常，只需替换一下节点，在前端页面中修改一下该替换节点的物理地址。而替换网关时，就需要数据服务层的协助。同时当网关的软件需要升级时，需要对设备进行远程更新，在升级过程中，不能出现意外情况。升级的过程越短，被打断的几率越小，升级成功的概率也会越大。

4.3.1 设备替换机制

前端页面中执行替换网关的操作，如下图所示，需要提交新网关的 SN 和密码，以表单的形式提交给后台。



图 4.14 web 端的替换网关页面



图 4.15 web 端填写替换网关信息

设备替换框图如下图 4.16 所示,后台将旧网关中的原数据表进行备份,生成备份数据表,包含了整个网关中的网络结构和设备查询信息。等待用户将新网关上电,连接到云平台。将备份数据表中的数据逐条下发,新网关接收到数据进行回复,同时将备份数据表已经添加成功的数据删除,添加成功的网关会有新数据表进行生成。添加完成的依据,就是备份数据表的数据清空,新数据表与原有的数据表中的数据相同。再将旧网关的数据从数据库中删除,旧网关数据清空。

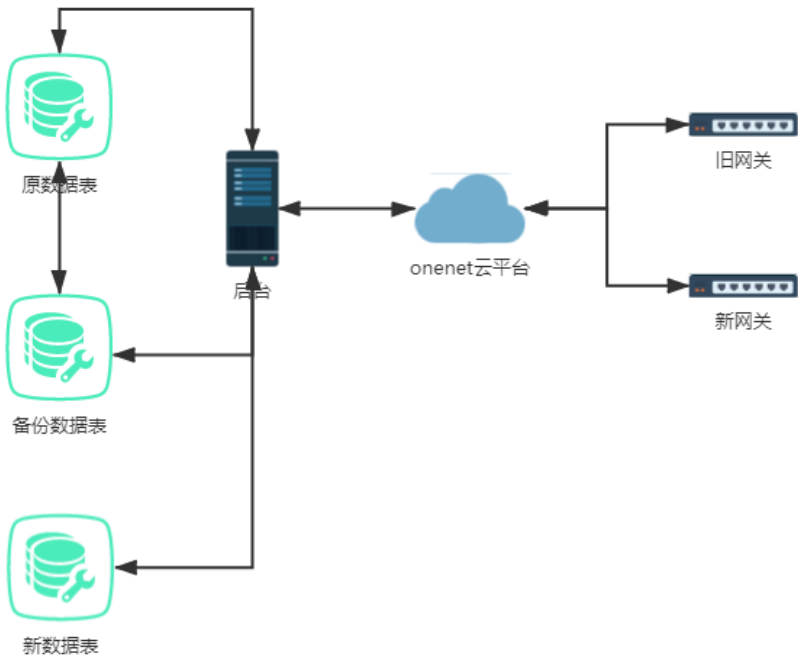


图 4.16 设备替换总框图

### 4.3.2 远程升级机制

远程更新的机制，是系统进行更新成败的关键。选择合适的更新方式，可以提高系统的稳定性。采用了两种远程更新的方式，具体的更新机制如下文所述。

#### 4.3.2.1 基于 MQTT 的远程更新机制

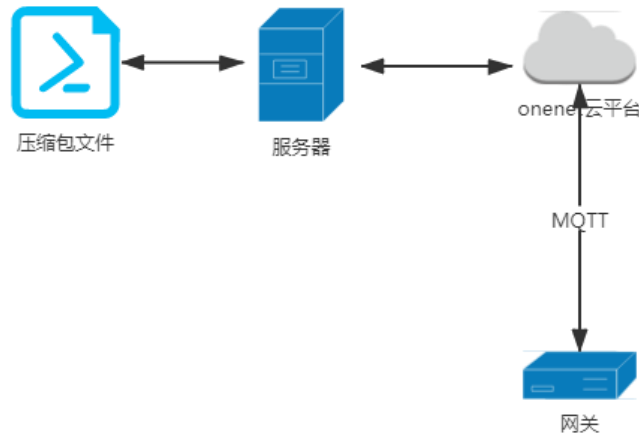


图 4.17 mqtt 更新图示

如图 4.17 所示，将最新版本号的固件打包成压缩包，将压缩包的文件，提交给后台服务器。后台服务器将压缩包转化为二进制的数，调用 OneNET 云平台提供的 API 接口，分包进行下发。在数据传输的期间需要遵循相应的帧格式，同样包括帧头、帧尾、控制码、帧计数器和 CRC 校验。分为起始帧、中间帧、和结束帧。网关接收到相应的数据，解析数据，判断帧协议，将有效内容取出，按压缩包的文件形式进行写入。每次通信根据帧序号进行同步，采用一一应答得机制，保证信息不出错。当文件发送完成时，发送更新结束帧，通知网关本次传输完成。网关执行脚本文件，关闭当前执行程序，进行压缩包得解压，再将解压得到得更新文件，拷贝到对应得文件夹，启动更新后得程序。

#### 4.3.2.2 基于 http 远程更新机制

对比于上述得 MQTT 更新方式，存在繁杂得数据通信过程。按照上述得流程，引入了新的机制。如图 4.18 所示，不再通过后台逐帧进行文件内容的发送，网关接收到数据进行校验回复的方式。而是通过阿里云的平台，将压缩包放置阿里云的服务器上。后台通过 OneNET 提供的 API 接口发送远程更新的指令给网关，告知网关需要进行应用程序的更新的版本号和对应的 url 地址，以及文件不同校验方式下的校验和。网关将会启动 http 的客户端与阿里云的服务器进行链接，将压缩包以文件下载的方式，下载到指定的目录。解压缩之后调用不同的校验方式程序，进行对接收文件的校验。与后台中所给的校验和是否一致，如果相同，上报 OneNET 云平台程序下载成功，执行新的程序安装。如果失败，上报失败信息给云平台，告知有错误，不执行新的程序安装。

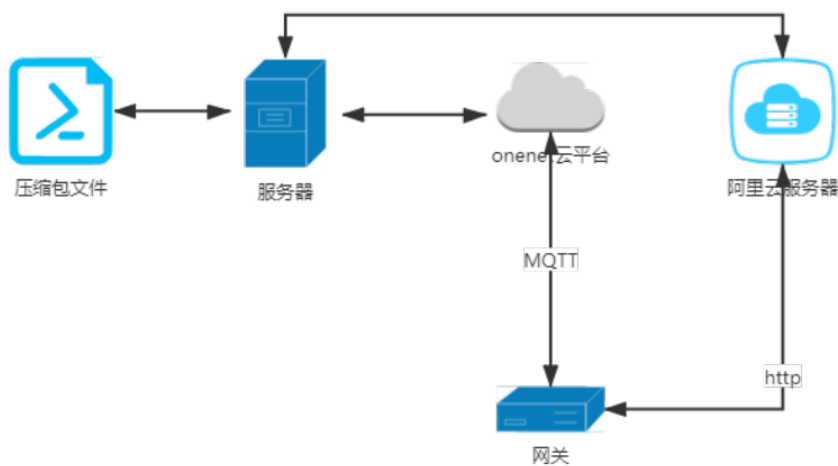


图 4.18 http 更新图示

4.4 实验测试与结果分析

4.4.1 多层协作上报测试

本次测试是主要对数据传输量的测试，从侧面可以反应流量的使用情况。测试该 SPD 实时监测系统分别采用透传策略和多层协作策略，对数据传输量进行对比测试，分析多层协作的处理策略在节省流量费用的上的优势。

测试方法：一台节点下挂 10 台 SPD 监控模块，启动网关，在网页的页面上配置相应的节点和 SPD 监测设备，测试时间为 1 小时，每十分钟记录一次，共记录 6 次。网关上报心跳帧，每 30 秒发送一次，每次 22byte，配置节点上报周期为 10 秒发送一次，每个监控点打包上报的数据长度为 43byte，系统中读取 5 个监控点。通过记录网关网卡上的数据，记录不同机制下的数据量传输的情况。下图是网页配置设备绑定图 4.19，和相关设备下的数据点展示图 4.20。

| 序号 | 状态 | 名称   | 设备类型     | 安装时间                | 站号 | 节点       | 操作          |
|----|----|------|----------|---------------------|----|----------|-------------|
| 1  | ✔  | 4851 | LT_CJ485 | 2020-11-01 22:01:00 | 1  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 2  | ✔  | 4852 | LT_CJ485 | 2020-10-27 00:00:00 | 2  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 3  | ✔  | 4853 | LT_CJ485 | 2020-11-12 14:41:16 | 3  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 4  | ✔  | 4854 | LT_CJ485 | 2020-11-12 14:41:31 | 4  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 5  | ✔  | 4855 | LT_CJ485 | 2020-11-12 14:42:17 | 5  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 6  | ✔  | 4856 | LT_CJ485 | 2020-11-12 14:42:33 | 6  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 7  | ✔  | 4857 | LT_CJ485 | 2020-11-12 14:43:01 | 7  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 8  | ✔  | 4858 | LT_CJ485 | 2020-11-12 14:43:52 | 8  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 9  | ✔  | 4859 | LT_CJ485 | 2020-11-12 14:44:13 | 9  | 50700001 | ⚙️ 编辑 🗑️ 删除 |
| 10 | ✔  | 485A | LT_CJ485 | 2020-11-12 14:46:21 | 10 | 50700001 | ⚙️ 编辑 🗑️ 删除 |

图 4.19 设备绑定

|       |                 |              |        |       |          |        |
|-------|-----------------|--------------|--------|-------|----------|--------|
| CJ485 | 一小一核配电柜<br>(60) | 雷击次数: 2      | SSD: 1 | 接地: 0 | SPD: 1   | 温度: 23 |
| ISPD  | 雷泰总配电柜(1<br>5)  | 雷击次数: 14     | SSD: 0 | PE: 1 | SPD失效: 0 | 温度: 23 |
| ISPD  | 测试10(10)        | 雷击次数:<br>215 | SSD: 0 | PE: 0 | SPD: 0   | 温度: 23 |
| ISPD  | 测试20(20)        | 雷击次数:<br>201 | 空开: 0  | 接地: 0 | SPD失效: 0 | 温度: 23 |
| CJ485 | 1#点(1)          | 雷击次数: 4      | SSD: 0 | 接地: 1 | 逐信: 0    | 温度: 31 |

图 4.20 部分 SPD 监测设备的数据点

表 4.2 数据量记录表

| 测试时间 (min) | 透传模式下数据量记录 (byte) | 多层协作式数据量记录(byte) |
|------------|-------------------|------------------|
| 10         | 49045             | 13318            |
| 20         | 48052             | 13578            |
| 30         | 48352             | 13657            |
| 40         | 49457             | 14376            |
| 50         | 48369             | 13521            |
| 60         | 48437             | 14292            |
| 平均值        | 48619             | 13790            |

从上表可以看出，多层协作式的数据量要远远小于透传的模式的数据量，数据量缩减了近 70%。在测试中通过 wireshark 抓包软件对网卡的数据进行抓包，如图 4.21 所示，发现一次通信过程中除了引入 198byte 额外开销，同时网卡上还有相应 DNS 域名解析的包和 ICMP 的包。通信过程中不仅仅有相应的数据传输，也有少量其他数据包开销。

|    |                 |                 |               |      |                                         |
|----|-----------------|-----------------|---------------|------|-----------------------------------------|
| 63 | 13:32:11.900506 | 114.114.114.114 | 10.99.15.29   | DNS  | Standard query response                 |
| 64 | 13:32:11.905522 | 114.114.114.114 | 10.99.15.29   | DNS  | Standard query response                 |
| 65 | 13:32:12.084518 | 8.8.8.8         | 10.99.15.29   | DNS  | Standard query response                 |
| 66 | 13:32:12.084793 | 10.99.15.29     | 8.8.8.8       | ICMP | Destination unreachable (Port unreachab |
| 67 | 13:32:20.030105 | 10.99.15.29     | 183.230.40.39 | TCP  | [TCP segment of a reassembled PDU]      |
| 68 | 13:32:20.123600 | 183.230.40.39   | 10.99.15.29   | TCP  | [TCP segment of a reassembled PDU]      |
| 69 | 13:32:20.123849 | 10.99.15.29     | 183.230.40.39 | TCP  | 47754 > 6002 [ACK] Seq=277 Ack=49 Win=3 |
| 70 | 13:32:31.020186 | 10.99.15.29     | 183.230.40.39 | TCP  | [TCP segment of a reassembled PDU]      |
| 71 | 13:32:31.149777 | 183.230.40.39   | 10.99.15.29   | TCP  | [TCP segment of a reassembled PDU]      |

图 4.21 wireshark 抓包图

虽然通信过程中有其他的流量包的引入，这是网络通信必须需要执行的步骤，测试环境下都

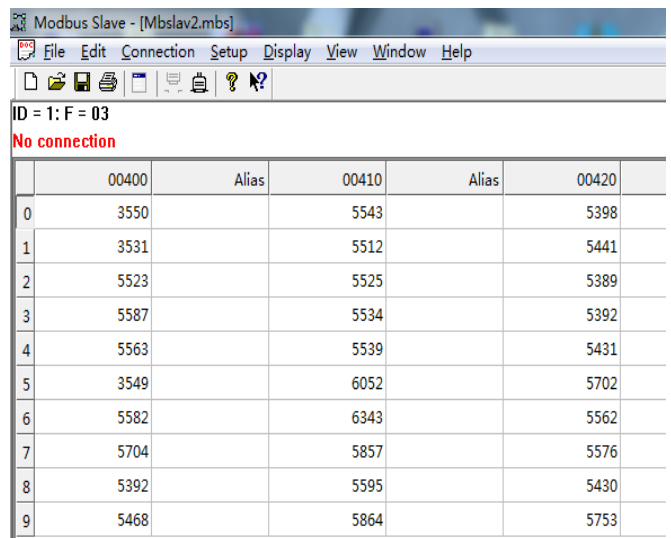
会存在, 相比于透传模式下, 多层协作的方式, 在节省数据流量使用的情况上更优, 数据量缩减了近 70%。

#### 4.4.2 数据连读功能测试

向 LoRa 节点添加相应的监控点, 监控点中的数据如下图 4.22 所示, LoRa 节点会按照下面添加监控点的地址 address, 进行数据的查询。Modbus 的节点采用的是电脑模拟的 modbusSlave 软件, 去模拟个个监控点的数据变化, 如图 4.23 所示。节点会按照地址的情况, 自动判断是否执行连读, 还是单独读取, 去和 modbusSlave 软件进行通信。实行节点轮询数据查询, 上报给网关。streamId 从 1021 一直到 1039。

```
//连读
{"streamId":"1021","regType":"03","dataNum":1,"dataSize":2,"address":400,
{"streamId":"1022","regType":"03","dataNum":1,"dataSize":2,"address":401,
{"streamId":"1023","regType":"03","dataNum":1,"dataSize":2,"address":402,
{"streamId":"1024","regType":"03","dataNum":1,"dataSize":2,"address":403,
//连读
{"streamId":"1025","regType":"03","dataNum":1,"dataSize":2,"address":405,
{"streamId":"1026","regType":"03","dataNum":1,"dataSize":2,"address":406,
{"streamId":"1027","regType":"03","dataNum":1,"dataSize":2,"address":407,
//连读
{"streamId":"1028","regType":"03","dataNum":1,"dataSize":2,"address":409,
{"streamId":"1029","regType":"03","dataNum":1,"dataSize":2,"address":410,
{"streamId":"102A","regType":"03","dataNum":1,"dataSize":2,"address":411,
{"streamId":"102B","regType":"03","dataNum":1,"dataSize":2,"address":412,
//连读
{"streamId":"102C","regType":"03","dataNum":1,"dataSize":2,"address":414,
{"streamId":"102D","regType":"03","dataNum":2,"dataSize":4,"address":415,
{"streamId":"102E","regType":"03","dataNum":2,"dataSize":4,"address":416,
{"streamId":"102F","regType":"03","dataNum":2,"dataSize":4,"address":417,
//单读
{"streamId":"1030","regType":"03","dataNum":1,"dataSize":2,"address":418,
//单读
{"streamId":"1031","regType":"03","dataNum":2,"dataSize":4,"address":419,
//连读
{"streamId":"1032","regType":"03","dataNum":2,"dataSize":4,"address":420,
{"streamId":"1033","regType":"03","dataNum":2,"dataSize":4,"address":421,
//连读
{"streamId":"1034","regType":"03","dataNum":2,"dataSize":4,"address":422,
{"streamId":"1035","regType":"03","dataNum":1,"dataSize":2,"address":423,
//连读
{"streamId":"1036","regType":"03","dataNum":2,"dataSize":4,"address":424,
{"streamId":"1037","regType":"03","dataNum":1,"dataSize":2,"address":425,
//连读
{"streamId":"1038","regType":"03","dataNum":1,"dataSize":2,"address":426,
{"streamId":"1039","regType":"03","dataNum":1,"dataSize":2,"address":427,
```

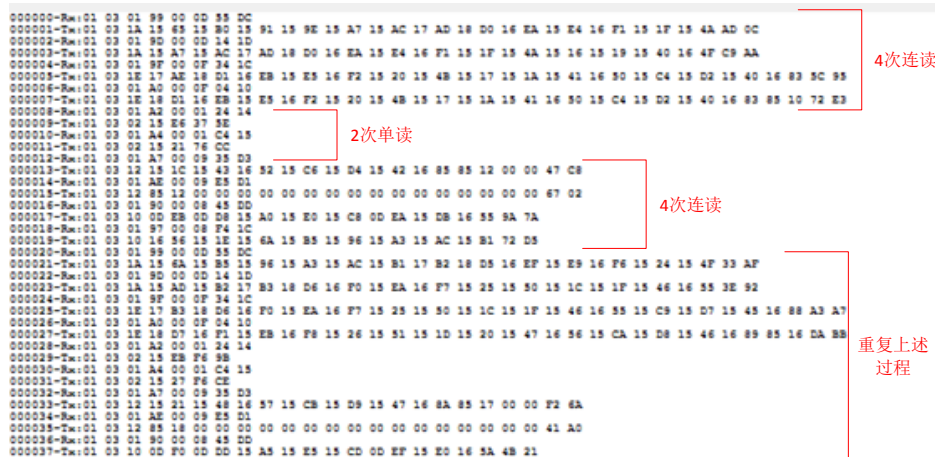
图 4.22 监控点数据



|   | 00400 | Alias | 00410 | Alias | 00420 |
|---|-------|-------|-------|-------|-------|
| 0 | 3550  |       | 5543  |       | 5398  |
| 1 | 3531  |       | 5512  |       | 5441  |
| 2 | 5523  |       | 5525  |       | 5389  |
| 3 | 5587  |       | 5534  |       | 5392  |
| 4 | 5563  |       | 5539  |       | 5431  |
| 5 | 3549  |       | 6052  |       | 5702  |
| 6 | 5582  |       | 6343  |       | 5562  |
| 7 | 5704  |       | 5857  |       | 5576  |
| 8 | 5392  |       | 5595  |       | 5430  |
| 9 | 5468  |       | 5864  |       | 5753  |

图 4.23 软件模拟寄存器变化

通过 modbus slave 软件监测数据的往来。如下图 4.24 所示, 可以看到前 4 次是连续读取, 中间两次单独读取, 最后又是 4 次连续读取。重复上述过程。



4次连读

2次单读

4次连读

重复上述过程

图 4.24 modbus 数据往来

节点将读取到的数据上报给网关, 网关将数据上报给云平台, 云平台中的数据展示如图 4.25 所示, 其中 streamId 为 1026 的数值是 4028, 以此类推。

|                                     |                                     |                                      |                                     |
|-------------------------------------|-------------------------------------|--------------------------------------|-------------------------------------|
| 1026<br>2020-06-23 15:06:57<br>4028 | 1031<br>2020-06-23 15:06:57<br>5876 | 1039<br>2020-06-23 15:06:57<br>34532 | 1038<br>2020-06-23 15:06:57<br>6231 |
| 102F<br>2020-06-23 15:06:57<br>6825 | 1035<br>2020-06-23 15:06:57<br>6040 | 1024<br>2020-06-23 15:06:57<br>6066  | 102C<br>2020-06-23 15:06:57<br>6005 |
| 1021<br>2020-06-23 15:06:57<br>4029 | 1030<br>2020-06-23 15:06:57<br>6078 | 1036<br>2020-06-23 15:06:57<br>6054  | 102A<br>2020-06-23 15:06:57<br>6023 |

图 4.25 云平台监控点数据

分别向两个节点中随机添加一百个离散的监控点，测试轮询 10 次，一个节点没有普通机制，另一个节点使用该机制，记录轮询完成所花费的时间。分别测试 5 次。

表 4.3 数据连读测试

| 次数         | 普通机制 (ms) | 连读机制(ms) |
|------------|-----------|----------|
| 1 次        | 14440     | 3200     |
| 2 次        | 14711     | 3270     |
| 3 次        | 14685     | 3203     |
| 4 次        | 14929     | 3291     |
| 5 次        | 14951     | 3308     |
| 平均值 (10 次) | 14743.2   | 3254.4   |
| 平均值 (1 次)  | 1474.32   | 325.44   |

由上述表格可以得出，普通机制轮询一遍监控点需要 1474.32ms，而加上数据连读机制后，只需要 325.44ms 就轮询了一遍监控点数据。轮询速率是普通机制的 4 倍，这是因为普通机制中串口的收发次数要远远大于连读机制的收发次数，离散的数据点越多，就越能体现出连读机制性能。LoRa 节点通过 modbus 中的地址位，判断是否执行连读功能，减少了 modbus 帧的查询次数，相比单次轮询读取，提高了本系统的查询效率。

#### 4.4.3 数据汇聚上报测试

通过数据的汇聚上报，可以减少流量的开销，这里测试汇聚上报的最大长度，测试环境同上，向网关中添加监控点，监控点的数量为 300 个。通过 modbus slave 软件模拟 modbus 从节点，网关通过串口转 485 与 modbus slave 软件相连接。通过监控点的文本提供的信息，去查询 modbus slave 软件中相应寄存器的地址，将数据读取完成之后，进行数据处理，放入到上报缓存，完成数据的上报。下图中提供的是上报汇聚的数据和总长度。

比如{"1020":{"2020-08-13 18:04:03":1234.5}}，这条数据表示，在 2020 年 8 月 13 号的 18 点 4 分 3 秒时，1020 这个监控点的数据为 1234.5。将 300 条这样的数据进行汇聚上报。减少了上报的开销。

```
OneNet_Publish msg len 4096
msg: {"2020":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,0,0,
69,56,78,123,23,0,0,0,00"},"2022":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,6
0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"2024":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,
03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"2026":{"2020-08-13 18:04:0
{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"2028":{"
0,0,0,0,00"},"2029":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,
,45,69,56,78,123,23,0,0,0,00"},"202B":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,
,0,0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"202D":{"2020-08-13 18:04:03":{"0,58,5,0,0,
04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"202F":{"2020-08-13 18:
30":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"203
3,23,0,0,0,0,00"},"2032":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123
6,56,45,69,56,78,123,23,0,0,0,00"},"2034":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856
,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"2036":{"2020-08-13 18:04:03":{"0,58,5,
3 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"2038":{"2020-08-13
,2039":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,123,23,0,0,0,00"},"
8 123,23,0,0,0,00"},"203B":{"2020-08-13 18:04:03":{"0,58,5,0,0,0,0,0,45,63,856,56,45,69,56,78,
```

图 4.26 数据打印信息 1

如图 4.26 所示，目前网关可以支持 4096 个字节数据的拼接上报，如果需要增加上报的长度的阈值，可以配置网关，增加数据的上报长度。相比于单条上报的形式，汇聚上报减少了信息头部的数据开销，节省了相关的资费。

#### 4.4.4 数据离线缓存上报测试

本次测试主要是针对网关离线存储功能的测试，因为采用了 OpenWrt 的操作系统，所以可以将相应的数据保存在系统之中，测试离线之后的数据是否能上报，主要目的验证系统的稳定性和采集数据的准确性。

如图 4.27 所示，在下午 5 点 42 分时断掉网线，在 5 点 45 分插上网线，观察这段期间，这段时间采集的数据是否能进行上报。如下图所示，第一个标识，表示 17 时 42 分 56 秒时，断掉网线，可以看到数据仍然在采集，每 10S 上报一次，一共采集了 27 条数据点，塞入环形队列的缓存中。Network unreachable，表示当前网络异常，无法连接到云平台。

```

2Add time Upload[2020-06-11 17:42:56]
2Add New Steam ID Upload[24]
2Add time Upload[2020-06-11 17:42:56]
2Add New Steam ID Upload[25]
2Add time Upload[2020-06-11 17:42:56]
2Add New Steam ID Upload[26]
2Add time Upload[2020-06-11 17:42:56]
(char*)onenet_cache_data_array[0].NowTime
(char*)onenet_cache_data_array[1].NowTime
(char*)onenet_cache_data_array[2].NowTime
(char*)onenet_cache_data_array[3].NowTime
(char*)onenet_cache_data_array[4].NowTime
(char*)onenet_cache_data_array[5].NowTime
(char*)onenet_cache_data_array[6].NowTime
(char*)onenet_cache_data_array[7].NowTime
(char*)onenet_cache_data_array[8].NowTime
(char*)onenet_cache_data_array[9].NowTime
(char*)onenet_cache_data_array[10].NowTime
(char*)onenet_cache_data_array[11].NowTime
(char*)onenet_cache_data_array[12].NowTime
(char*)onenet_cache_data_array[13].NowTime
(char*)onenet_cache_data_array[14].NowTime
(char*)onenet_cache_data_array[15].NowTime
(char*)onenet_cache_data_array[16].NowTime
(char*)onenet_cache_data_array[17].NowTime
(char*)onenet_cache_data_array[18].NowTime
(char*)onenet_cache_data_array[19].NowTime
(char*)onenet_cache_data_array[20].NowTime
(char*)onenet_cache_data_array[21].NowTime
(char*)onenet_cache_data_array[22].NowTime
(char*)onenet_cache_data_array[23].NowTime
(char*)onenet_cache_data_array[25].NowTime
(char*)onenet_cache_data_array[26].NowTime
TIME: 1591868578 ,FILE: onenet/onenet.c, LINE: 302
OneNetSubscribe topic: $dp

cur netstate : 0
pre netstate : 0
setNetConfig File /tmp/wifi_infor.txt Dont Exist;
TIME: 1591868578 ,FILE: onenet/onenet_socket.c, LINE:
socket id = 5

-----Clear Onenet Data Recv Queue-----
TIME: 1591868578 ,FILE: onenet/onenet_socket.c, LINE:
OneNet connect

connect: Network unreachable
TIME: 1591868578 ,FILE: onenet/onenet_socket.c, LINE:
Close socket id = 5

TIME: 1591868578 ,FILE: onenet/onenet_socket.c, LINE:

```

图 4.27 数据打印信息 2

在下午 5 点 45 分，接上网线，如下图 4.28 所示，在 17 时 45 分 5 秒时，数据仍然再采集，数据塞入了缓存当中。CONNECT SUCCEED 表示连接云平台成功。这时候会将缓存中的数据塞入上报队列中将数据进行汇聚上报。如图 4.29 所示 Send OneNET Data Over，表示将一条汇聚的数据上报。RECV ONENET PUBLISH ACK 表示 OneNET 云平台收到数据给予相应的回复。

```

2Add New Steam ID Upload[21]
2Add time Upload[2020-06-11 17:45:05]
2Add New Steam ID Upload[22]
2Add time Upload[2020-06-11 17:45:05]
2Add New Steam ID Upload[23]
2Add time Upload[2020-06-11 17:45:05]
2Add New Steam ID Upload[24]
2Add time Upload[2020-06-11 17:45:05]
2Add New Steam ID Upload[25]
2Add time Upload[2020-06-11 17:45:05]
2Add New Steam ID Upload[26]
2Add time Upload[2020-06-11 17:45:05]
(char*)onenet_cache_data_array[0].NowTime
(char*)onenet_cache_data_array[1].NowTime
(char*)onenet_cache_data_array[2].NowTime
(char*)onenet_cache_data_array[3].NowTime
(char*)onenet_cache_data_array[4].NowTime
(char*)onenet_cache_data_array[5].NowTime
(char*)onenet_cache_data_array[6].NowTime
(char*)onenet_cache_data_array[7].NowTime
(char*)onenet_cache_data_array[8].NowTime
(char*)onenet_cache_data_array[9].NowTime
(char*)onenet_cache_data_array[10].NowTime
(char*)onenet_cache_data_array[11].NowTime
(char*)onenet_cache_data_array[12].NowTime
(char*)onenet_cache_data_array[13].NowTime
(char*)onenet_cache_data_array[14].NowTime
(char*)onenet_cache_data_array[15].NowTime
(char*)onenet_cache_data_array[16].NowTime
(char*)onenet_cache_data_array[17].NowTime
(char*)onenet_cache_data_array[18].NowTime
(char*)onenet_cache_data_array[19].NowTime
(char*)onenet_cache_data_array[20].NowTime
(char*)onenet_cache_data_array[21].NowTime
(char*)onenet_cache_data_array[22].NowTime
(char*)onenet_cache_data_array[23].NowTime
(char*)onenet_cache_data_array[24].NowTime
(char*)onenet_cache_data_array[25].NowTime
(char*)onenet_cache_data_array[26].NowTime
TIME: 1591868705 ,FILE: onenet/onenet.c, LINE: 302
OneNet_Publish topic: sdp

TIME: 1591868705 ,FILE: onenet/onenet.c, LINE: 420
Tips: CONNECT SUCCEEDED

TIME: 1591868705 ,FILE: onenet/onenet.c, LINE: 628
streamid = 006
TIME: 1591868705 ,FILE: onenet/onenet.c, LINE: 641
value = 2
TIME: 1591868705 ,FILE: onenet/onenet.c, LINE: 302

```

图 4.28 数据打印信息 3

[illegible]

图 4.29 数据打印信息 4

观察云平台某一个数据监控点,17时42分到17时45分的数据是否有上报。如下图 4.30所示,17时42分到45分的数据有上报,没有丢失。

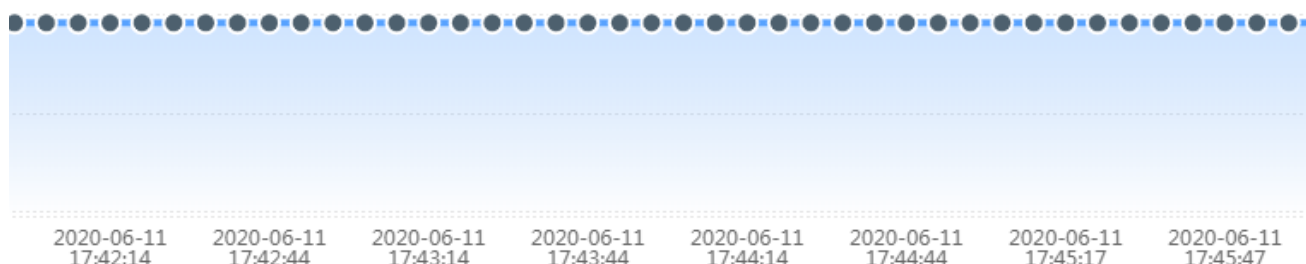


图 4.30 云平台数据值

对数据保存的时间进行测试，做 5 次实验，记录离线时保存的数据丢失的时间段。来反应离线数据保存的最长时间。

表 4.4 数据保存时长

| 次数 | 时间间隔 (min) | 信息丢失的时间(min) |
|----|------------|--------------|
| 1  | 5min       | 0min         |
| 2  | 10min      | 0min         |
| 3  | 15min      | 0min         |
| 4  | 20min      | 2min         |
| 5  | 25min      | 7min         |

由测试可以得出，当离线时间较长时，只能保存 18min 的信息。环形缓存队列的大小，可以根据需求进行设置，本套系统可以缓存 7200 条数据，系统满负荷的状态下，至少可以缓存 10 分钟的数据。当出现网络异常时，重新连接网络只需要几秒，保证了采集系统中的数据不丢失，提高了系统的信息准确性。

#### 4.4.5 远程更新性能比较分析

采用 mqtt 的一应一答的方式和 http 请求的方式进行远程更新，对比更新完成的速度，通过网关更新完成之后的上报云平台的版本号，判断当前是否更新成功。更新成功后，会在云平台上查询到相应的数据。如下图 4.31 所示，更新完成后的程序版本号为 01130916，之前的版本号为 01130915。



图 4.31 云平台网关上报版本号

在使用同样的网络，相同的测试环境下，对网关分别采用上述两种方式进行更新。新的固件程序为 230K，版本号高于网关本地版本号。记录相应的传输时间和成功率。

表 4.5 远程更新

| 测试次数 | mqtt 一应一答机制 (s) | 成功率  | http 请求更新机制(s) | 成功率  |
|------|-----------------|------|----------------|------|
| 1 次  | 183             | 100% | 25             | 100% |
| 2 次  | 196             | 100% | 25             | 100% |
| 3 次  | 185             | 100% | 28             | 100% |
| 4 次  | 192             | 100% | 26             | 100% |
| 5 次  | 187             | 100% | 27             | 100% |
| 平均   | 188.6           | 100% | 26.2           | 100% |

用上述两种方式对程序进行更新，在程序更新成功的机率上基本上是一致的，取决于当前网络条件的好坏。但是从程序更新成功的时长中，可以看出两种方式的优劣。mqtt 的一应一答得机制平均时长为 188.6S，而 http 的请求更新机制为 26.2S，效率提升 6 倍。这是因为后台服务器对固件进行分片，组装成相应的帧格式，调用 OneNET 得 API 接口，再通过 OneNET 平台进行下发。网关接受到需要进行帧序号比对，帧格式校验成功后进行回复，再将有效信息写入本地缓存中，后台收到回复才能继续进行数据的发送。这一部分的时间消耗要远远大于直接从阿里云服务器进行固件下载。

#### 4.4.6 网关长时间在线监测

对网关在线功能长时间测试，每个网关挂载相应的 4G 模块，网关插入配套的 4G 物联网卡，实现拨号上网，连接上 OneNET 的云平台。向云平台每分钟上报一条心跳帧，用来和云平台保持常连接，OneNET 云平台记录网关的在线情况。下图 4.32 是实物测试场景。

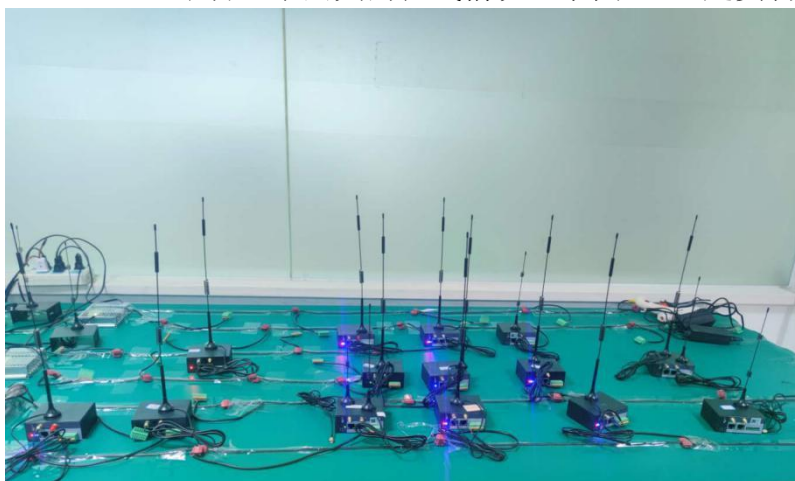


图 4.32 网关集体测试

表 4.6 网关在线监测

| 序号 | 设备上线时间           | 设备 SN    | 第一周 | 第二周 | 第三周 | 第四周 | 备注   |
|----|------------------|----------|-----|-----|-----|-----|------|
| 1  | 2020.5.18 18: 00 | 10101032 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 2  | 2020.5.18 18: 00 | 10101033 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 3  | 2020.5.18 18: 00 | 10101034 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 4  | 2020.5.18 18: 00 | 10101035 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 5  | 2020.5.18 18: 00 | 10101036 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 6  | 2020.5.18 18: 00 | 10101037 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 7  | 2020.5.18 18: 00 | 10101038 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 8  | 2020.5.18 18: 00 | 10101039 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 9  | 2020.5.18 18: 00 | 10101040 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |
| 10 | 2020.5.18 18: 00 | 10101041 | 无离线 | 无离线 | 无离线 | 无离线 | 已经寄出 |

监测了网关一个月，记录每星期是否有离线的数据，10 台设备均无离线，从侧面说明了系统的稳定性。

4.4.7 系统整体测试

4.4.7.1 数据推送实时性测试

(1) 测试方案与条件

数据推送实时性，即数据产生的时间到用户收到数据时间之间的间隔，本项目中主要测试报警的时间间隔。LoRa 节点和 SPD 监控设备进行连接，网关和各个设备上电。实物测试图如下。



图 4.33 实物测试图 1



图 4.34 实物测试图 2



图 4.35 实物测试图 3

(2) 测试步骤

- a.首先通过 web 客户端添加多组节点和 SPD 监控设备。
- b.设置多项报警条件，如下图 4.36 所示。

| 监控点名称 | 触发器名称   | 触发条件 | 报警内容  |
|-------|---------|------|-------|
| SPD   | yaoxin  | ==0  | 通信失效  |
| SSD   | ssd     | ==0  | SSD失效 |
| 温度    | wendu   | >30  | 超温报警  |
| SSD   | kongkai | ==1  | 空开脱扣  |
| PE    | jiedi   | ==1  | 接地失效  |
| 烟雾    | shuiwei | ==0  | 水位超限  |
| SPD失效 | shixiao | ==0  | 失效    |
| SSD   | ssd     | ==0  | SSD   |
| 温度    | wendu   | >30  | 超温    |

图 4.36 设置报警阈值

- c.硬件模拟触发报警，记录变化时间和接收到报警的时间。
- d.反复测试 10 次，记录时间。

表 4.7 报警时间记录

| 次数 | 报警时间 (ms) |
|----|-----------|
| 1  | 2231      |
| 2  | 2151      |
| 3  | 1983      |
| 4  | 2112      |
| 5  | 2001      |
| 6  | 1995      |
| 7  | 2281      |
| 8  | 2103      |
| 9  | 2202      |
| 10 | 2016      |

从表中的数据可以得知，报警推送时间基本稳定在 2.5S 之内。这期间主要包含了 SPD 监测设备的处理时间、LoRa 节点的处理和上报时间、网关处理和上报数据云平台的时间。数据云将数据推送给业务云平台，业务云平台查询数据库和相关用户的报警联系人信息，推送给相应的 WEB 客户端或者以短信的形式推送给用户的时间。

#### 4.4.7.2 系统稳定性测试

主要对整个 SPD 实时监测系统长时间提供高质量服务进行测试，需要考虑系统中各个部分是否稳定运行和各个部分能否实现自治并对外提供正常服务。

##### (1) 测试方案和环境

在阿里云的 ECS 中部署一整套的云服务平台，系统中添加了 5 台智能网关，每个智能网关下挂不同的数目的 LoRa 节点和不同数目的 SPD 监测设备，如图 4.37。设备安装在相应的配电柜中，如下图 4.39-4.42 所示。使用不同的账号在不同的客户端浏览器上登录，查看设备中的信息，操作网关下设备的 LoRa 节点和 SPD 监控设备。整个测试持续 15 天，需要记录浏览器的运行情况，SPD 监控设备，节点和网关是否出现掉线的情形，记录每一次数据通信是否出现异常，在整个测试当中，除了设备上电，不存在人工干预，均由系统自治。



图 4.37 web 端网关添加图



图 4.38 安装实物图 1



图 4.39 安装实物图 2



图 4.40 安装实物图 3

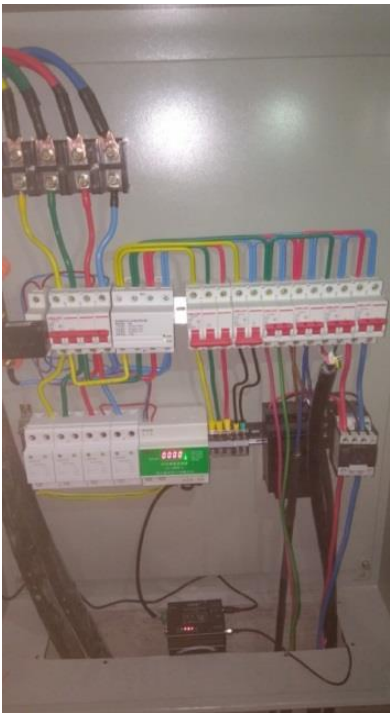


图 4.41 安装实物图 4

(2) 测试结果

测试结果如下表所示:

表 4.8 系统稳定性测试

| 时间    | 尝试次数 | 通信正常次数 | 通信失败次数 | 通过率  |
|-------|------|--------|--------|------|
| 第 1 天 | 100  | 100    | 0      | 100% |
| 第 2 天 | 100  | 100    | 0      | 100% |

续表 4.8

|        |     |     |   |      |
|--------|-----|-----|---|------|
| 第 3 天  | 100 | 100 | 0 | 100% |
| 第 4 天  | 100 | 100 | 0 | 100% |
| 第 5 天  | 100 | 100 | 0 | 100% |
| 第 6 天  | 100 | 100 | 0 | 100% |
| 第 7 天  | 100 | 100 | 0 | 100% |
| 第 8 天  | 100 | 99  | 1 | 99%  |
| 第 9 天  | 100 | 100 | 0 | 100% |
| 第 10 天 | 100 | 100 | 0 | 100% |
| 第 11 天 | 100 | 100 | 0 | 100% |
| 第 12 天 | 100 | 100 | 0 | 100% |
| 第 13 天 | 100 | 100 | 0 | 100% |
| 第 14 天 | 100 | 100 | 0 | 100% |
| 第 15 天 | 100 | 100 | 0 | 100% |

据表格所示，在 15 天当中，WEB 客户端每天执行相应的操作 100 次，通过浏览 WEB 端的日志，除了第 8 天请求失败了一次，其他时间段均正常。查看日志中失败的原因，数据重发无响应，导致操作失败，2 分钟后重新访问正常。经过排查，发现智能网关离线，因为测试环境中网络发生变化，联网功能受到干扰，导致 WEB 客户端操作失败。重新调整了测试环境的网络，恢复了通信链路，数据通信正常。通过测试，整个 SPD 实时监测系统的设计可以达到长时间稳定运行指标。

## 4.5 本章小结

本章主要是分析了如何提高 SPD 监测系统的稳定性和效率性，介绍了实时采集系统中常出现的问题。一个稳定的系统，是需要多方面进行设计和考虑的，本系统通过对采集、传输和服务三层协助的机制对系统进行优化。介绍了各个机制实现的原理，并通过实际的测试对多层次协作机制提高系统稳定性和效率性进行了验证。同时对于整个系统进行的测试与验证。测试表明 SPD 数据变化时，2.5S 之内会将变化的数据推送到 WEB 客户端，对比透传方式下，数据流量节省了 70%。系统满负荷的情况下可以保存 10min 的离线数据。网关 http 的更新方式要比 mqtt 的方式更新方式，效率提升 6 倍，符合设计指标的要求。

## 第5章 总结与展望

### 5.1 总结

本文针对实际项目---基于物联网的 SPD 实时监测系统的设计。雷电已成为电子设备受损主要的灾害之一，浪涌保护器的作用是保护电子设备免受雷击浪涌伤害。因此需要实时了解低压配电系统里的 SPD 的运行状态，可以及时处理相关的异常，进行相应的修复工作，可见检测 SPD 的工作状态意义重大。目前的防雷工程中，由于 SPD 使用数量大，安装位置复杂，人工检测维护 SPD 难度大，效率低、周期长、同时也会存在漏检、误检的风险。很多 SPD 出现问题不能及时发现，存在危险。本文基于物联网的 SPD 实时监测系统的设计正是在此背景下进行研究的，所做内容主要包括：

1.分析了当前 SPD 监测的技术，存在成本高、监测范围小以及人工检测难度大。在物联网技术快速发展的环境下，构建 SPD 实时监测系统的方案，为监测 SPD 的状态信息提供基础。

2.对 SPD 实时监测系统各个部分进行设计。由智能网关、LoRa 节点和 SPD 监控设备的硬件平台组成。SPD 监控设备实时采集 SPD 的状态信息，存储下来。智能网关和 LoRa 节点构建成星型组网，LoRa 节点采集 SPD 监控设备的信息，将各个 SPD 的信息上报给智能网关。智能网关通过网络通信技术，将 SPD 的实时数据上报给云平台。在 web 客户端可以实时查看 SPD 监控设备的信息。有报警信息可以进行推送，减轻了人工检测的负担。

3.针对物联网的使用场景，为了监控更多的 SPD 设备，参考 IEEE802.15.4 协议和利用 LoRa 的特性。对原有的时隙 CSMA/CA 算法存在多个节点访问信道时出现的不公平现象。对原有的算法进行分析优化，通过竞争窗口线性减小的策略，结合 LoRa 的特性，使得节点接入信道更为公平。同时对 LoRa 无线网络进行设计，包括了 MAC 层、网络层和应用层。通过仿真和实物验证，该算法提高了节点接入信道的公平性。同时针对实物在室内、户外、有遮挡物和定时上报的测试环境中进行测试，从测试数据得出该网络具有通信距离远，抗干扰能力强的特性，满足本项目的需求。

4.针对物联网多层协作进行设计，在具体的使用场景中，对设备的流量资费、系统稳定性、采集效率性和数据的真实性进行考虑。数据采集层的数据连读策略、数据组合上报和定时差异化上报，在一定情况下节省了数据的开销，提高了采集效率。数据传输层的网关同步网络时间、数据汇聚上报、数据离线缓存和断网续传的策略，节省了流量资费的开销和增加了系统信息的真实性。在数据服务层主要针对设备替换和远程更新机制进行设计，增强了系统的稳定性。最后对系统进行了长时间的测试，该系统运行稳定，达到设计要求。

## 5.2 展望

本文设计的基于物联网的 SPD 实时监测系统的设计，解决了人工检测维护 SPD 难度大，效率低、周期长、同时也会存在漏检、误检的风险的问题。个人在 SPD 设备端的经验不足，系统还有较大的改进空间。针对大数据、边缘计算和物联网的快速发展背景下，值得继续研究。

1.构建物联网的 SPD 实时监测系统，不仅仅可以检测 SPD 的设备状态信息，同时外接其他传感器，对当前的环境进行检测。以及可以检测防雷系统中的电压、电流、接地阻值、设备的功耗、雷击峰值等信息。

2.大数据的背景之下，可以进行 SPD 设备的参数分析，预测 SPD 的使用寿命，更加准确的检测出相关设备异常信息。同时分地理环境和使用场景去分析，为设计 SPD 提供有效数据支持。

3.边缘计算的技术背景之下，设备较多时，可以进一步扩大监测范围，提高信息采集的速率，给云平台减压，实现本地化的服务。

## 致谢

时光飞逝，转眼三年就过去了，研究生的生活即将画上句号。在杭州电子科技大学电子信息学院，我真切感受到了学院浓厚的学术氛围和高效的科研能力，也意识到自己在学术方面的不足。这三年求学生涯，是我人生的宝贵财富，感谢帮助过我的每一个人，感谢在杭电学习生活所经历的一切。

首先，感谢我的导师陈科明副教授，老师严谨的求学态度、渊博的学识积累和勤勉的工作精神深深的影响了我。为我提高专业知识和技能打下敦实的基础。此次论文写作，无论是选题、立新还是定稿，陈老师都严格要求。老师的细心指导，让我收获巨大。三年的师生情，陈老师让我看到一名专业人士应该具备的责任和态度。在今后的日子里，我也会牢记陈老师的教诲，做一名优秀的杭电人。

感谢我的同门、师弟师妹和实习时遇到的小伙伴们。在学习、工作和思想上给予我莫大的帮助。能够和他们成为同学和朋友让我感到非常荣幸。

感谢我的父母及家人，感谢这三年来你们对我的支持和鼓励，在我面临人生选择的迷茫之际，为我排忧解难，使我顺利完成学业。

感谢我的对象胡晓双在写论文期间默默支持我、鼓励我和帮助我一起修改论文。正是有你的支持，我才能顺利完成硕士论文。

感谢论文评阅和答辩委员会的老师，感谢你们的耐心，我一定会悉心听取你们宝贵的意见并积极改正。

最后，衷心地祝愿所有人，身体健康，工作顺利，让我们一起迎接更美好的明天！

## 参考文献

- [1] 姜畔.浅谈建筑外部防雷与接地[J].林业科技情报,2014,46(03):66-67+71.
- [2] 潘亮.民用建筑防雷设计中电涌保护器的选择[J].电气工程应用,2020(03):7-9.
- [3] 仲杰,殷华.现代气象防雷设备与技术及其应用探讨[J].科技经济导刊,2020,28(25):95+58.
- [4] 石光其. 智能建筑信息系统雷击电磁脉冲 LEMP 研究与防护设计[D].湖南大学,2004.
- [5] 田德宝,冯瑜骅,张雪慧,陈政豫.2012-2017 年全国雷电灾害事故统计分析[J].科技通报,2020,36(05):42-47.
- [6] 刘佼,肖稳安,陈红兵. 全国雷电灾害分析及雷灾经济损失预测[A]. 中国气象学会.第 28 届中国气象学会年会——S13 雷电物理、监测预警和防护[C].中国气象学会:中国气象学会,2011:10.
- [7] 张祎,李浩,边学文.2007-2013 年浙江省雷电灾害特征统计分析[J].气象与环境科学,2018,41(02):139-143.
- [8] GB 50343-2012, 建筑物电子信息系统防雷技术规范[S].
- [9] 储晶.浪涌 SPD 在乡村防雷系统中的应用[J].农村经济与科技,2017,28(20):4+6.
- [10]张毅刚.单片机原理及应用[M].高等教育出版社, 2004:159-187.
- [11]查丽斌, 王宛苹, 刘建岚. 电路与模拟电子技术基础[M]. 电子工业出版社, 2008: 37-52
- [12]何耀斌. SPD 在线监测系统设计[D].电子科技大学,2012.
- [13]仇慧宝. 电涌保护器的智能监测系统[D].河北工业大学,2015.
- [14]罗志勇. 中山市重点场所电源 SPD 在线安全监测系统[D].电子科技大学,2017.
- [15]李思南,刘黎,刘岩,汪卫国,赵勇.基于 ZigBee 无线网络技术的避雷器在线监测传感器设计[J].工业控制计算机,2011,24(04):69-70.]
- [16]车现法,陈波,王光博,饶辉新.防雷设施智能监测在云湛高速公路中的应用[J].广东公路交通,2020,46(03):53-58.]
- [17]黄俞成.基于物联网智能防雷监测系统对广电发射台改造的探讨[J].中国传媒科技,2020(05):120-125.
- [18]崔灿. 雷电在线监测系统的应用研究[A]. 中国气象学会、深圳市防雷协会.智能防雷网赢未来——2016 中国雷电防护高峰论坛论文集[C].中国气象学会、深圳市防雷协会:四川省电子学会,2016:11.
- [19]Amardeo C, Sarma, J G. Identities in the Future Internet of Things[J]. Wireless Personal Communications,2009,49(3):353-363.

- [20] Gustavo R G, Mario M O, Carlos D K. Early infrastructure of an Internet of Things in Spaces for Learning[C]. Eighth IEEE International Conference on Advanced Learning Technologies, 2008: 381-383.
- [21] 刘云浩. 物联网导论[M]. 北京: 科学出版社, 2010: 12-4.
- [22] 李金瑶. LoRa 物联网技术及应用[J]. 电子技术与软件工程, 2019(19): 5-6.
- [23] 赵静, 苏光添. LoRa 无线网络技术分析[J]. 移动通信, 2016, 40(21): 50-57.
- [24] 李泽宇. 基于 LoRa 的智能节水灌溉系统[D]. 呼和浩特: 内蒙古大学, 2019: 2-18.
- [25] 许杰. 物联网无线通信技术应用探讨[J]. 无线互联科技, 2018, 15(14): 19-20.
- [26] 短距离无线通信技术对比[J]. 电子技术应用, 2011, (03): 16-17.
- [27] Li Fei, Liu Min, Zhang Yue, Shen Weiming. A Two-Level WiFi Fingerprint-Based Indoor Localization Method for Dangerous Area Monitoring[J]. Sensors (Basel, Switzerland), 2019, 19(19).
- [28] 夏晶. Bluetooth 技术综述[J]. 科技信息(科学教研), 2007(26): 77.
- [29] 何燕翔. ZigBee 技术在智能建筑中的应用研究[J]. 通讯世界, 2019, 26(09): 5-6.
- [30] Jonathan D. Fuller, Benjamin W. Ramsey, Mason J. Rice, John M. Pecarina. Misuse-based detection of Z-Wave network attacks[J]. Computers & Security, 2017, 64.
- [31] Sanchez-Iborra Ramon, Sanchez-Gomez Jesus, Ballesta-Viñas Juan, Cano Maria-Dolores, Skarmeta Antonio F. Performance Evaluation of LoRa Considering Scenario Conditions[J]. Sensors (Basel, Switzerland), 2018, 18(3): 324-329.
- [32] 曾佳. 小型无线传感器网络的可靠性研究[D]. 杭州电子科技大学, 2015.
- [33] 赵琰琰, 蒋遂平, 车春立. 基于 LoRaWAN 的 MAC 层协议的研究与改进[J]. 计算机工程与设计, 2019, 40(05): 1276-1281+1293.
- [34] 白双星. 基于物联网的工业现场数据采集网络系统的设计与实现[D]. 北京交通大学, 2019.
- [35] Bist, M., Wariya, M., Agarwal, A.. Comparing delta, open stack and Xen Cloud Platforms: A survey on open source IaaS[P]. , 2013.
- [36] Carrozza, G., Battaglia, L., Manetti, V., Marotta, A., Canonico, R., Avallone, S.. On the Evaluation of VM Provisioning Time in Cloud Platforms for Mission-Critical Infrastructures[P]. Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on, 2014.
- [37] Fainelli F.. The OpenWrt embedded development framework[C]. Proceedings of the Free and Open Source Software Developers European Meeting, 2008.
- [38] 于尧. 浅析建筑物电气防雷系统优化设计[J]. 中国新技术新产品, 2019(05): 146-148.
- [39] 肖学华. NTC 热敏电阻温度传感器——高精度负温度系数[J]. 世界电子元器件, 1997(12): 53-55.

- [40] 龚永罡,付俊英,汪昕宇,王蕴琪,高爽.MQTT 协议在物联网中的应用研究[J].电脑与电信,2017(11):89-91+94.
- [41] 高静,段会川.JSON 数据传输效率研究[J].计算机工程与设计,2011,32(07):2267-2270.
- [42] 黎帮毅.基于时隙 CSMA\_CA 算法的 ZigBee 信道接入机制的研究[D].长春:吉林大学,2013:4-48.
- [43] 黄颖. IEEE802.15.4 时隙 CSMA/CA 算法性能研究[D].西安电子科技大学,2010.
- [44] 林静. IEEE 802.15.4 有时隙 CSMA/CA 在无线传感网络中的协议性能研究[D].电子科技大学,2007.
- [45] 尚克诚. IEEE 802.15.4 隐藏节点问题研究[D].吉林大学,2011.
- [46] Song N O, Kwak B J, Song J, et al. Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm[C] Vehicular Technology Conference, 2003. Vtc 2003-Spring. the, IEEE Semi. IEEE, 2003:2775-2778 vol.4.
- [47] Wu H, Cheng S, Peng Y, et al. IEEE 802.11 distributed coordination function (DCF): analysis and enhancement[C]IEEE International Conference on Communications. IEEE Xplore, 2002:605-609.
- [48] 李运鹏, 徐昌彪, 刘琳. 无线传感器网络中一种竞争窗口自适应 MAC 协议[J]. 传感器与微系统, 2010, 29(1):49-51.
- [49] 唐周益丹,姜宁康.基于 LoRa 的长距离室内定位的研究[J].计算机应用与软件,2018,35(04):148-154.
- [50] 柳永波. 基于 LoRa 的无线自组网 MAC 协议研究[D].西安电子科技大学,2017
- [51] 赵莹. Ad hoc 网络 MAC 层公平性研究[D].大连理工大学,2014.
- [52] 张琴. 基于 HTTP 长连接的 WEB 实时通信技术的研究[D].电子科技大学,2014.
- [53] 魏民,王艺.物联网云平台终端远程更新技术研究与应用[J].电信科学,2018,34(10):137-142.

## 附 录

### 作者在读期间发表的学术论文及参加的科研项目

#### 参加的科研项目：

- [1] 电工产品智能化技术服务，横向课题
- [2] 基于 LORA/NB-IOT 产品及 LORA 无线系统开发，横向课题
- [3] 暖通物联网平台开发，横向课题

#### 专利：

一种提高网关采集 modbus 设备效率的方法    CN111884915A    2020.11.03 ( 发明公开 )

一种基于 lora 组网的 spd 智能监测系统    CN212259016U    2020.12.29 ( 实用新型 )

一种基于改进灰狼优化算法的 5G 通信系统资源分配方法    CN111787543A    2020.10.16 ( 发明公开 )