**2017/2018学年北京理工大学计算机学院**

**《物联网创新项目训练》实验报告**

1. **基本信息**

实验题目：物联网创新项目训练

组长（学号）：刘天祺1320151097

组员（学号）：张新宇1320151090

王 维1120151781

罗士尧1320151094

朱笑含1320151091

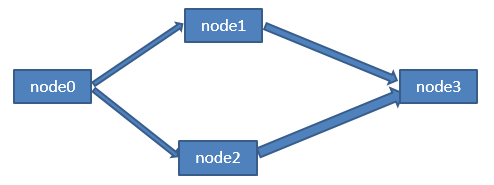
牛昕仑1320151093

王佳轩1120151780

1. **实验目的**
2. 能够在TinyOS平台下进行编程开发。掌握无线传感器节点之间通信的方式，能够自定义数据通信格式，进一步理解网络中的拥塞、丢包的情况。
3. 能够在TinyOS平台下进行编程开发。掌握无线传感器节点之间通信的方式，能够自定义数据通信格式，进一步理解网络中的拥塞、丢包的情况。了解传感器节点采集数据的方式。
4. **实验内容**

**实验一：**

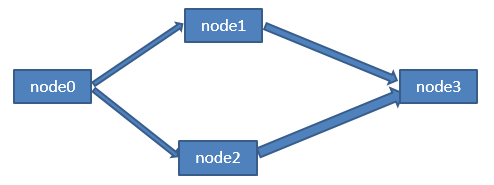
现在有4个无线传感器组成的网络，拓扑结构如下所示：



节点0是数据源节点，节点1，2收集节点0发送的数据。节点1，2将数据转发给节点3。我们组编写3份程序，一份烧录到节点0（发送），另外一份烧录节点1和2（奇偶数的转发），最后一份烧录到节点3（接收）。

实验二：

现在有4个无线传感器组成的网络，拓扑结构如下所示：



节点0为数据源节点，产生两种类型的数据，将生成的数据广播出去，节点1，2分别接收一种类型的数据，接着转发给节点3。我们组编写四份程序，第一份烧录到节点0（发送）第二份烧录到节点1(温度数据的转发)，第三份烧录节点2（随机数的转发），最后一份烧录到节点3（接收）。

1. **核心模块功能及实现描述**

实验一：0、每隔一秒发送含有自增序列的数据包

1/2、接受序列为奇/偶数的数据包并转发

3、接收数据包并打印数据及对应节点号

实现方法：1、在节点0设置计数器并使其周期性自增。

2、节点1、2接/发数据包长度的不同，

因此节点3不接受节点0广播的数据包。

实验二：0、前六十秒每隔一秒采集温度并广播发送，

后六十秒每隔一秒产生随机数并广播发送。

1、接收温度数据包并转发。

2、接受随机数数据包并转发。

3、接收数据包并打印数据及对应节点号，

　 根据本节点收包个数统计丢包情况。

实现方法：1、配置中引用了Read和RandomC组件并在程序中调用其接口。

2、定义的温度数据包与随机数数据包的结构体长度不同，

节点1、2因此能区分两种数据包。

1. **成员分工（介绍每位成员的分工、任务完成情况、工作量比重）**

刘天祺：代码的编写及调试，环境的搭建（完成）30%

张新宇：硬件调试，查阅资料（完成）10%

王　维：代码的编写及调试（完成）20%

罗士尧：代码的编写及调试（完成）20%

朱笑含：硬件调试，资料整理（完成）10%

牛昕仑：文档撰写，代码调试（完成）10%

1. **实验总结**

一、实验方面

实验验证出的一部分影响无线信号传输主要因素：

1、 物理的障碍物 ：当节点1,2与节点3需要穿墙传输信号时身处在墙壁后面的节点3只能接到很微弱的信号，或没有收到信号造成会大量丢包。

2、 如果在无线环境中存在多台无线设备还有可能存在频道冲突，无线信号串扰的问题，将1,2节点放在正在高速传输数据的路由器旁，数据包的传输会受到严重影响丢包50%以上。

此外，我们更加深入的了解了TinyOS的温度检测和随机数产生接口，温度传感器能准确测量出温度的数值，但是对于温度的变化测量存在延迟。

二、项目协作方面

本次实验历时3天，采用git进行版本控制与小组成员合作开发，摒弃了“编写代码->通过QQ\微信传输文件->人工整和代码”这样的原始开发方式模式。

实验第一天，组员聚在一起，搭配完TinyOS开发所用的Linux环境，并安装Git。我们以组长在Github上创建远程repo，组员fork的形式，进行协作开发。

组员修改代码后向组长pull request，组长同意后将修改的代码合并。组长更新代码后，组员通过fetch、merge的方式同步代码。

此外，由于项目开发需要用到许多命令，为了方便大家使用，我们将编译命令、查看端口数据命令、同步仓库代码等等命令整合到bash脚本中，并且可以由安装脚本一键配置，方便组内成员使用。

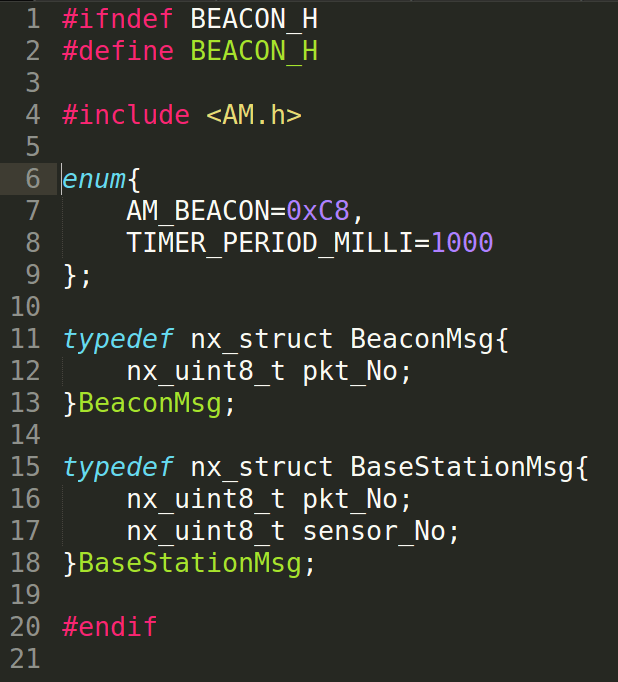
仓库地址：<https://github.com/TaQini/BIT_TOS>

使用Git进行实验，管理代码十分便捷，各个成员之间的代码能保持很高的同步率，此外各个成员可以同时进行代码开发（比如实验2中的随机数产生和温度测量就是通过2人同时开发然后整合的方式完成的），极大的加快了项目的开发速度，节约了宝贵的时间。

**附录：核心模块程序代码**

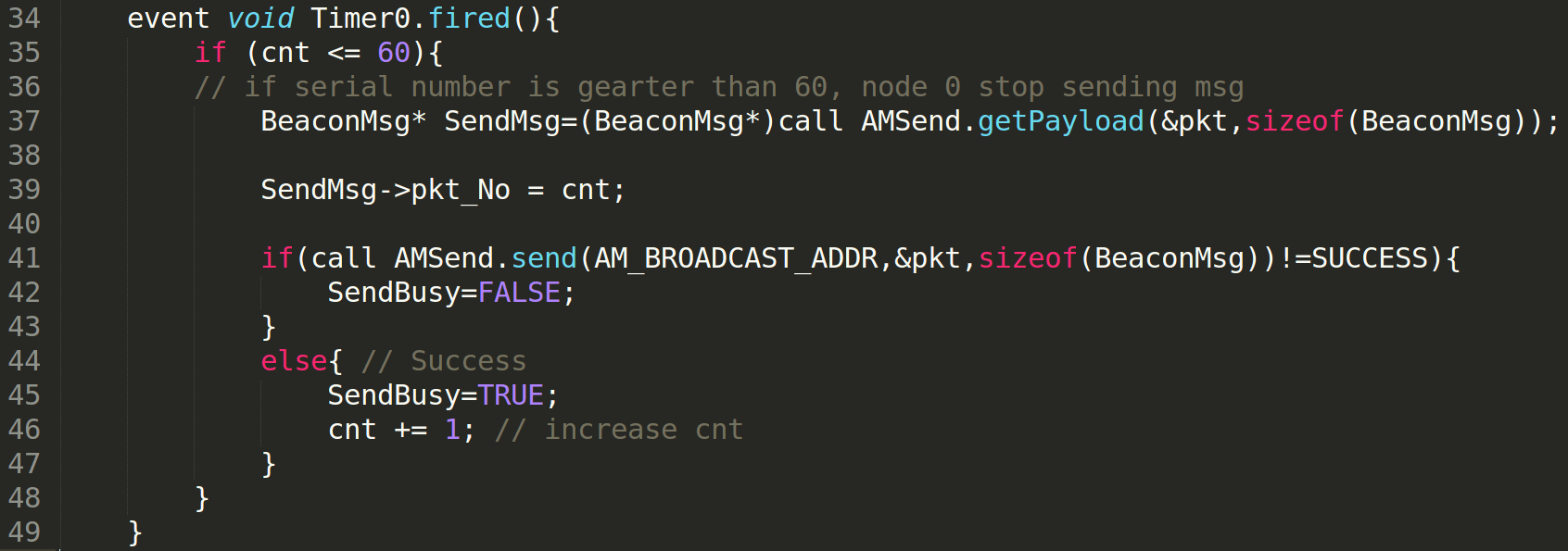
**实验1：**

**结构体：**

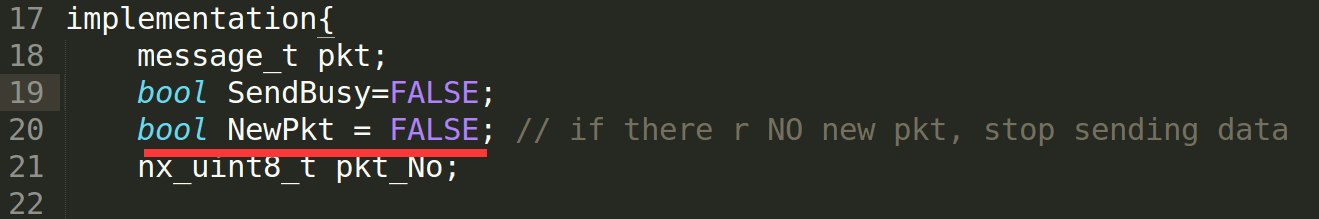


BeaconMsg用于节点0发送数据， BaseStationMsg 用于节点3接受数据

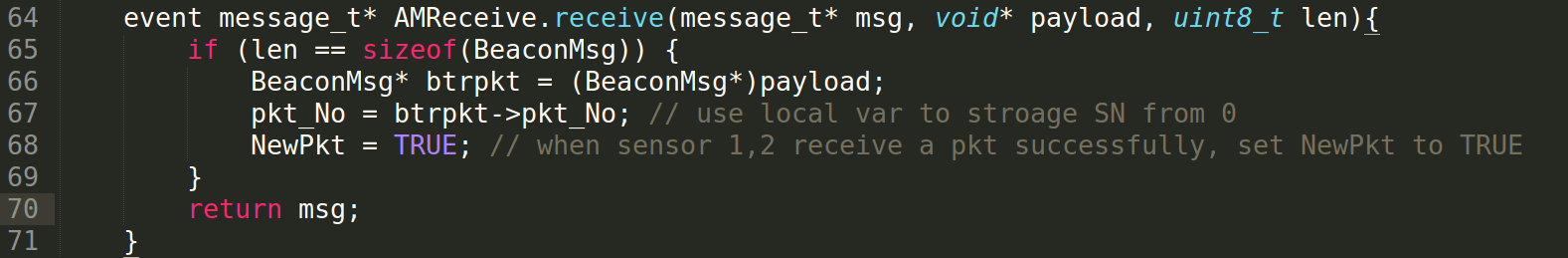
节点0 关键代码：发送递增的数据包。



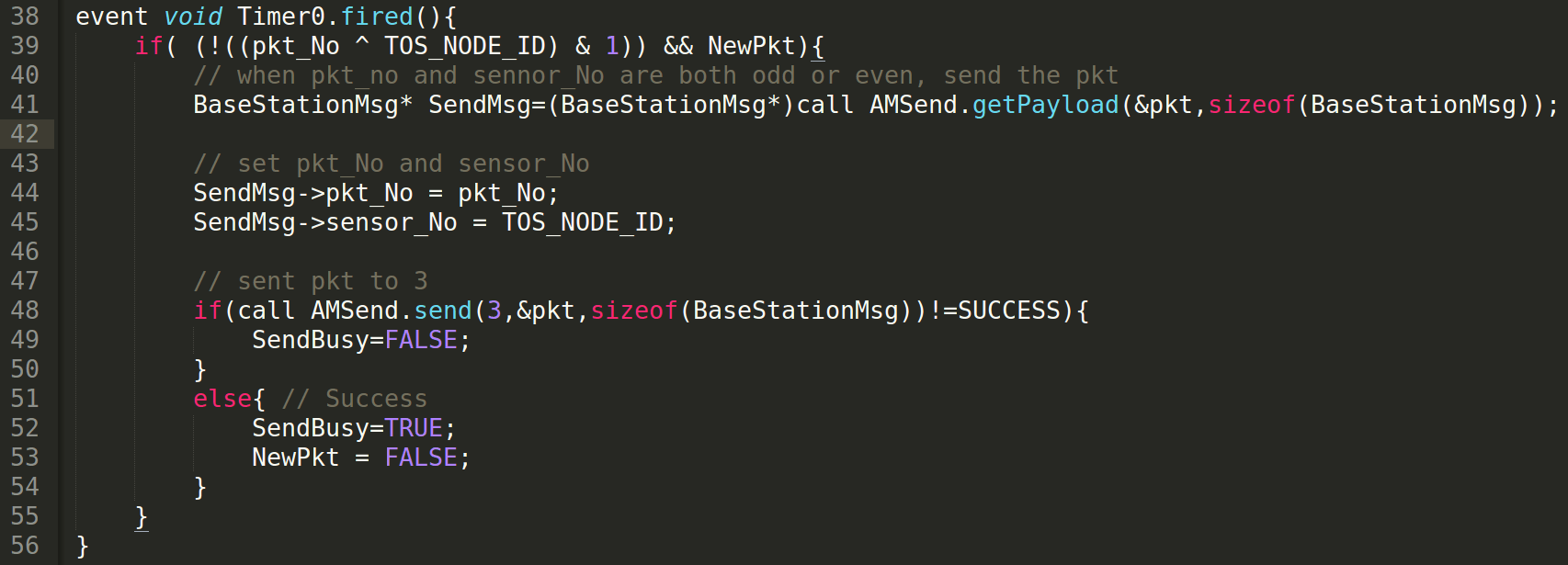
节点1,2 关键代码：设置NewPkt标志。



当节点收到数据包时，将节点0发送的序号存储，并将NewPkt标志设为True。



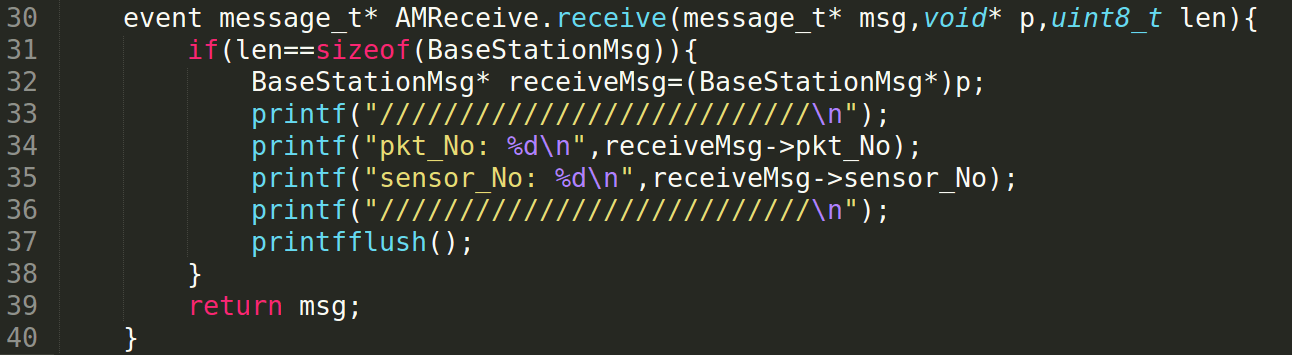
通过节点号与递增序号的关系区分节点1,2。(1号节点只收发奇数序号，2号只收发偶数序号)：



节点1,2将存储的序号以及自己的节点号打包为BaseStationMsg结构，并发送至节点3。

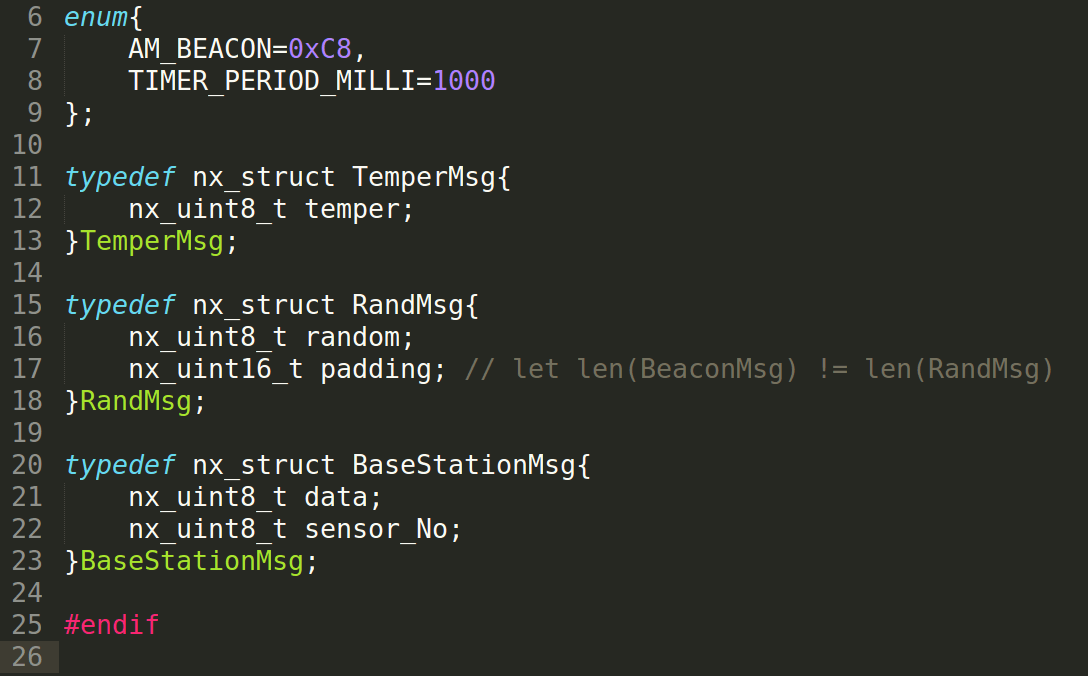
发送成功后，将NewPkt标志更改。设置该标志可以保证节点1,2在**收不到**数据包时，**不再继续**向节点3发送数据包。

节点3关键代码：打印接受到的序号数据以及节点号。



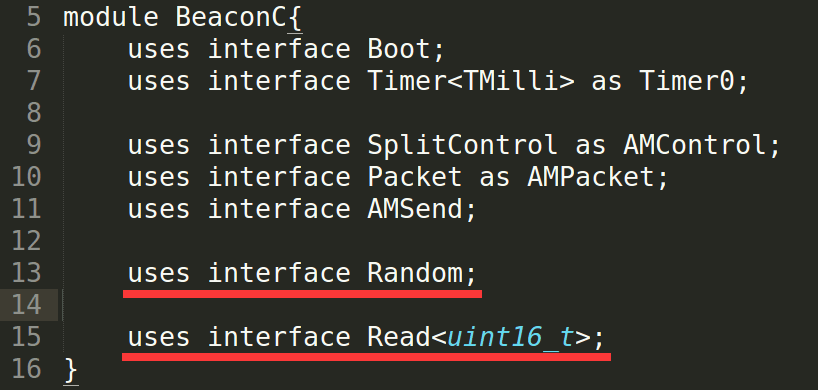
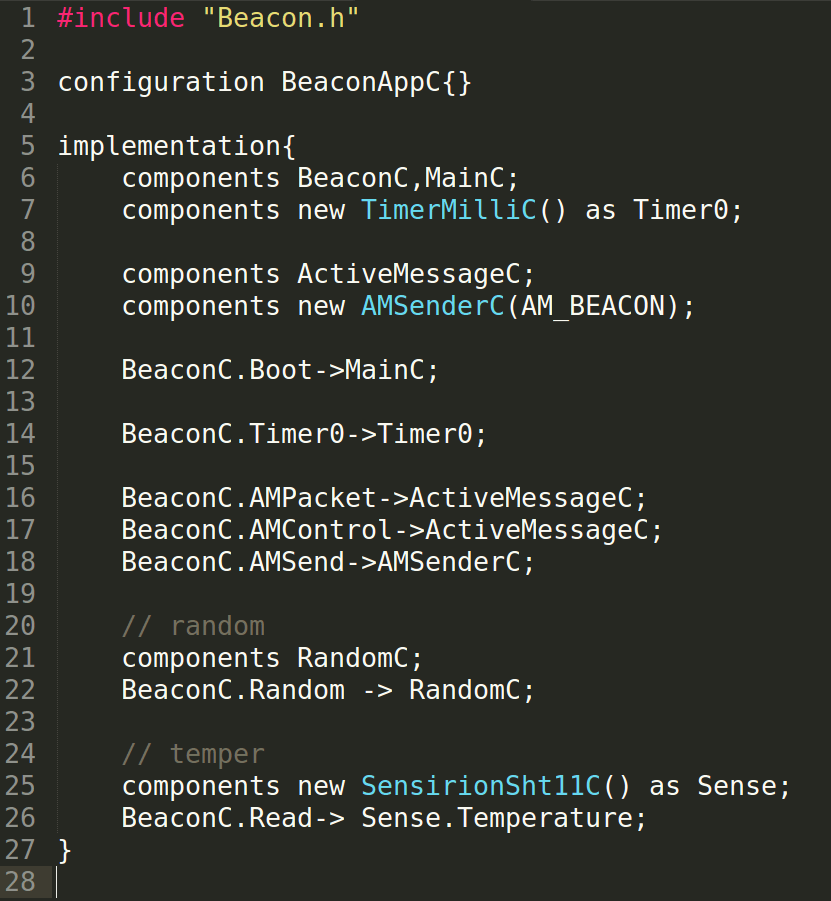
**实验2：**

**结构体：**



定义了3个结构体，结构体长度分别不同。如此设计是为了方便用数据包大小区分不同类型的数据（温度、随机数、传感器节点号）。

节点0模块配置：

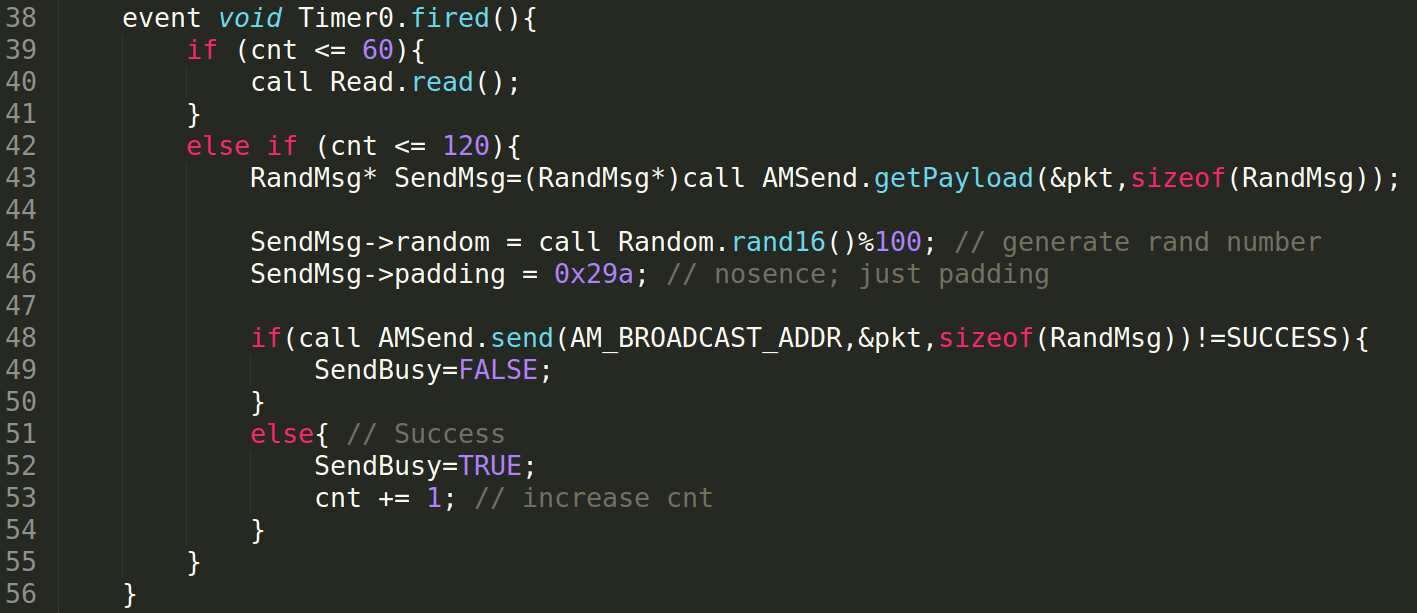


节点0关键代码：

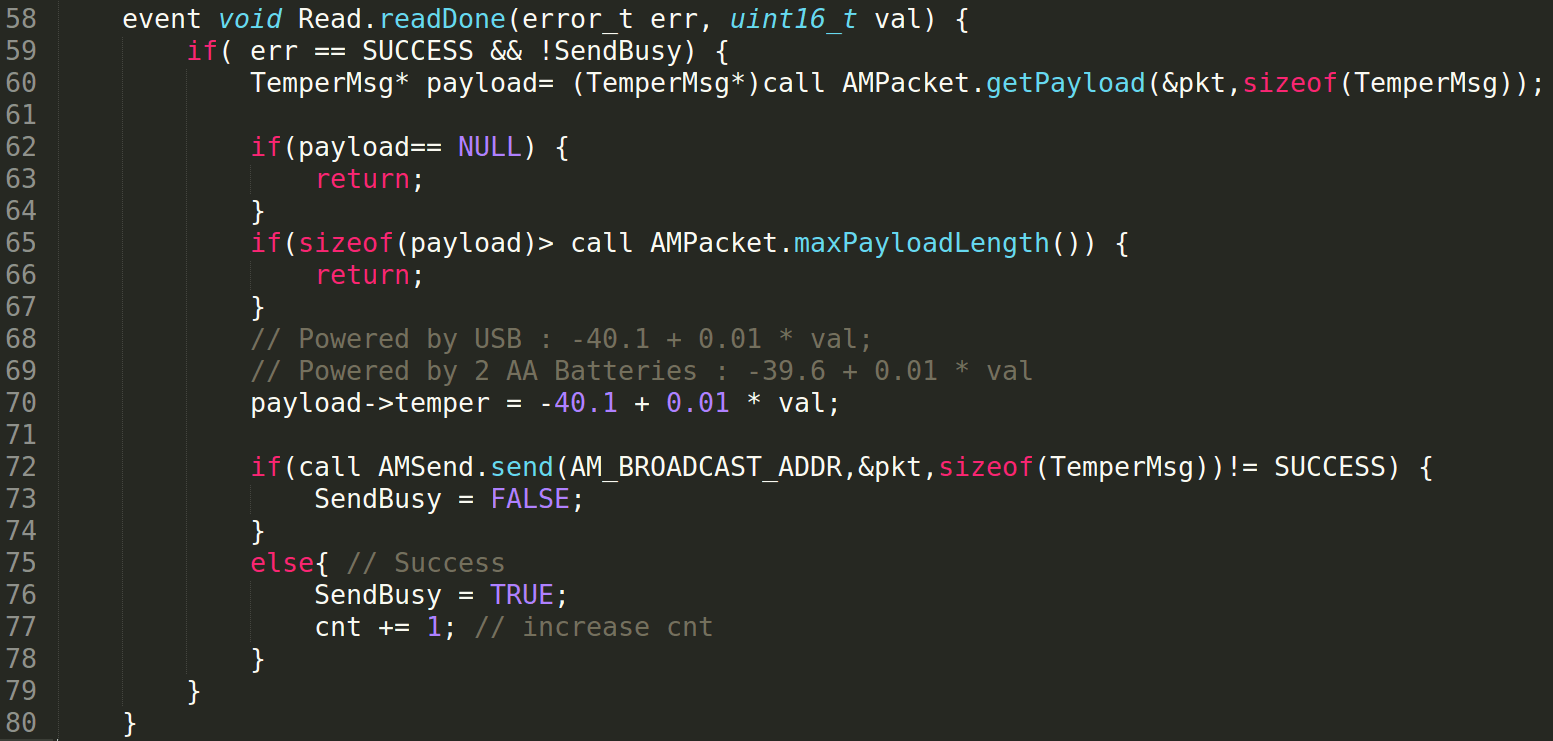
计数器cnt记录节点0发送的数据包个数。

收到前60个数据包时，读取采集温度。（发送代码在后面）

收到后60个数据包时，产生随机数与填充值打包(RandMsg)并广播发送。



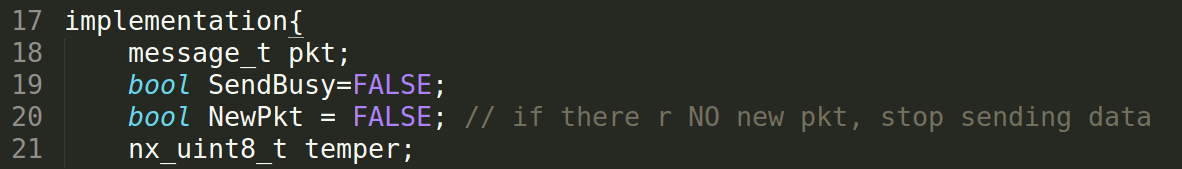
采集温度完成后，发送含有温度数据的数据包(TemperMsg)：



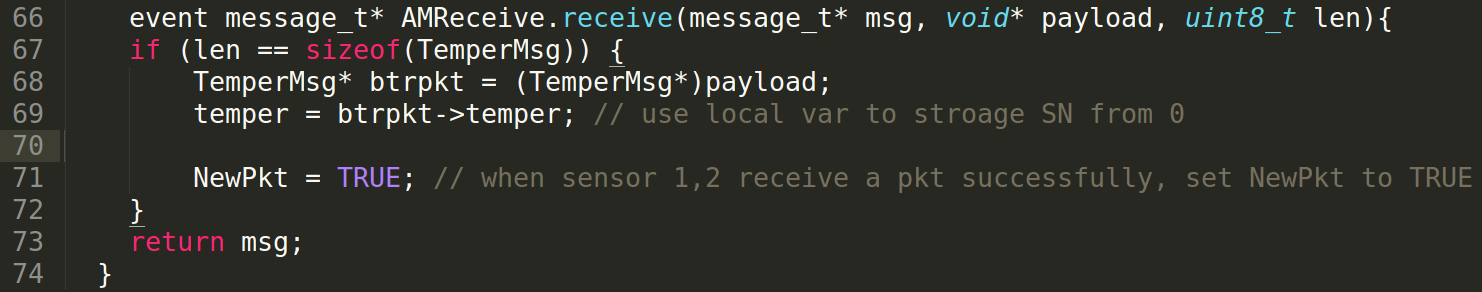
发送成功后计数器自增。

节点1关键代码：

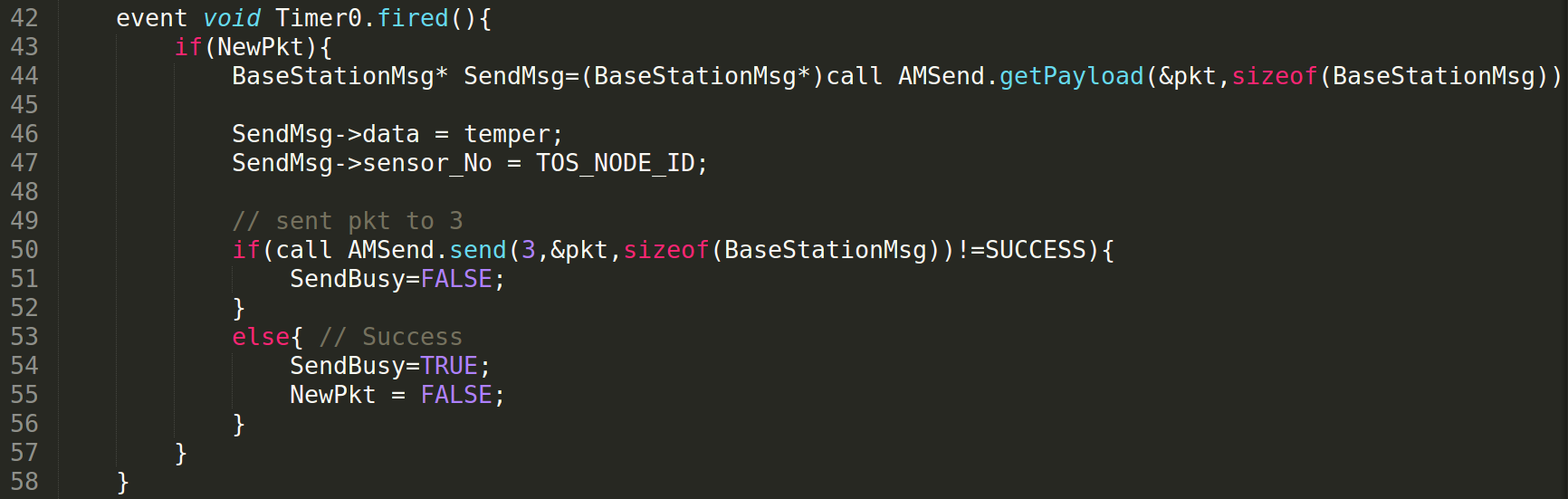
设置NewPkt标记，同实验1，不再赘述。



if语句用于筛选接受温度数据包（各种类型数据包长度不同）：

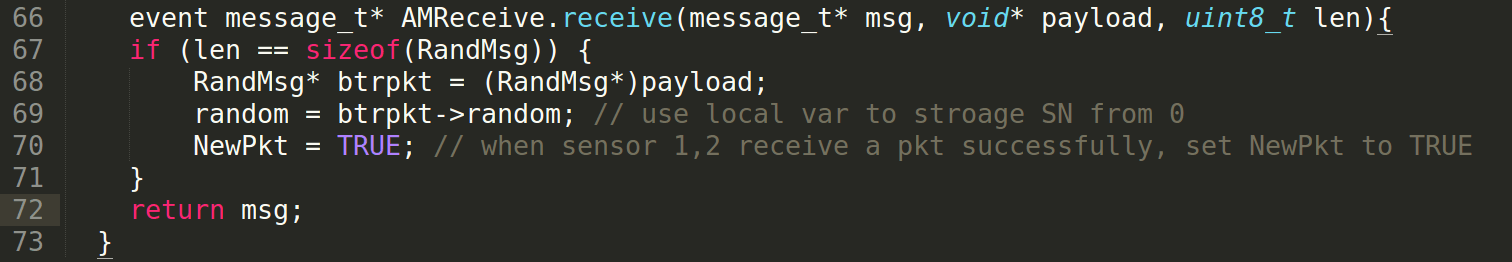


将温度数据与节点号打包(BaseStationMsg)，并发送至节点3。

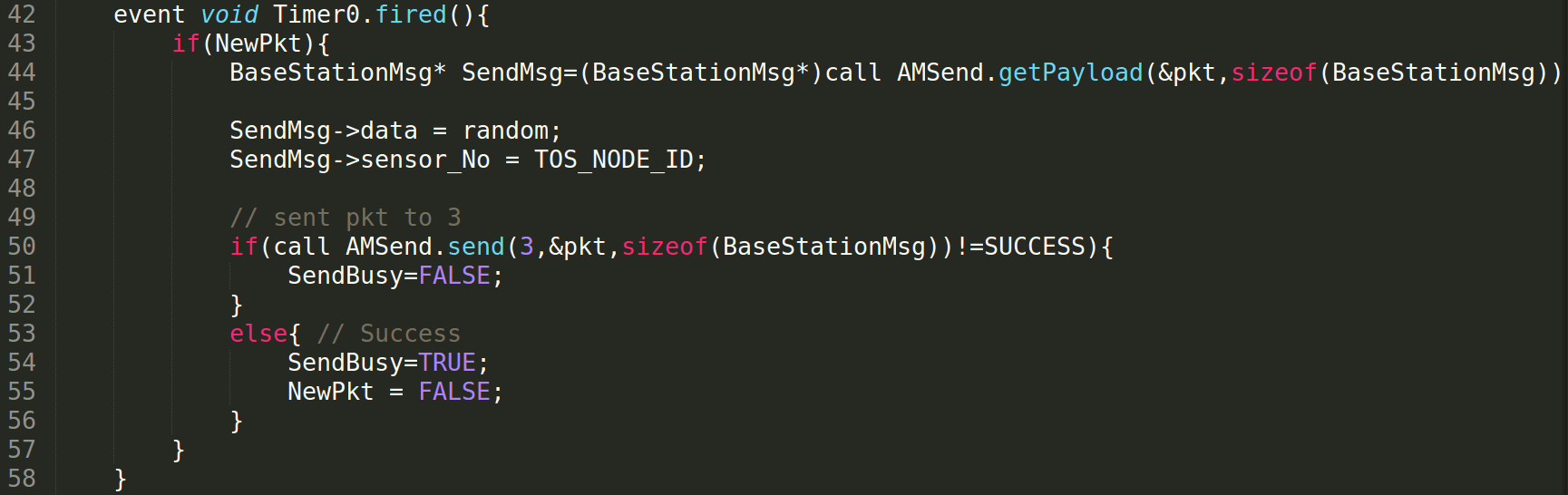


节点2关键代码：

与节点1代码功能类似，只是处理的数据不同（节点2处理RandMsg，节点1处理TemperMsg）。出于节约内存角度考虑，将节点1,2代码分离。

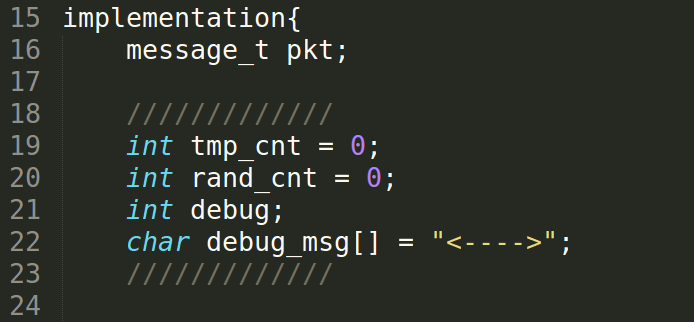


将温度数据与节点号打包(BaseStationMsg)，并发送至节点3。

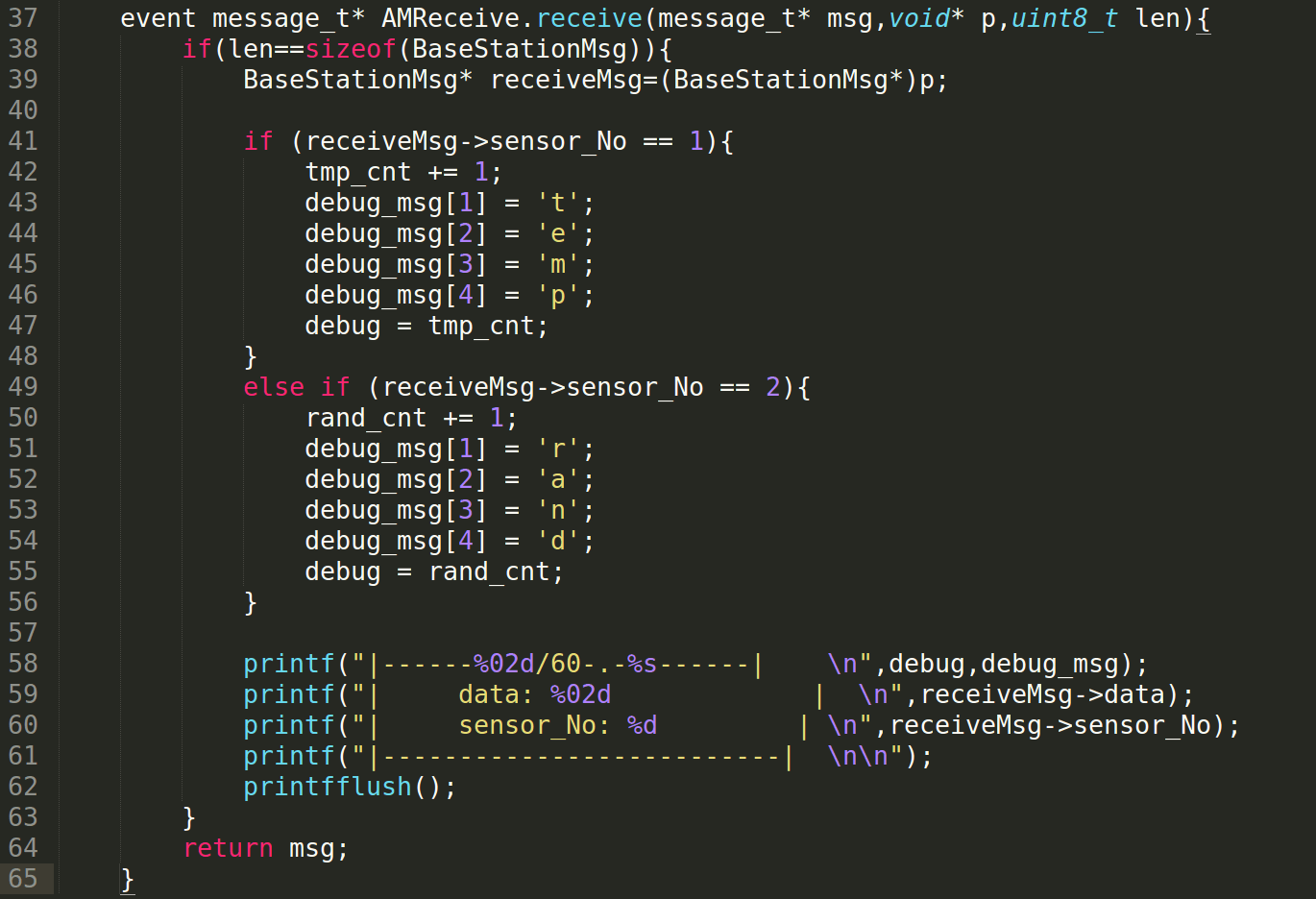


节点3关键代码：

设置两个用于记录收到数据包个数的计数器，用于统计收包情况。



打印接受到的数据时，根据节点号区分数据类型（温度\随机数）。



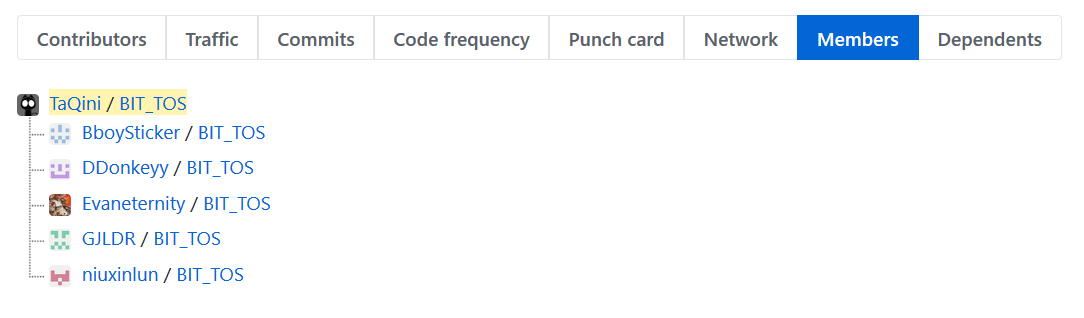
打印的信息包括收到的数据包数量/数据包总数，此处可直观的看出丢包情况。

**附录2：组内成员使用Git完成实验的感受**

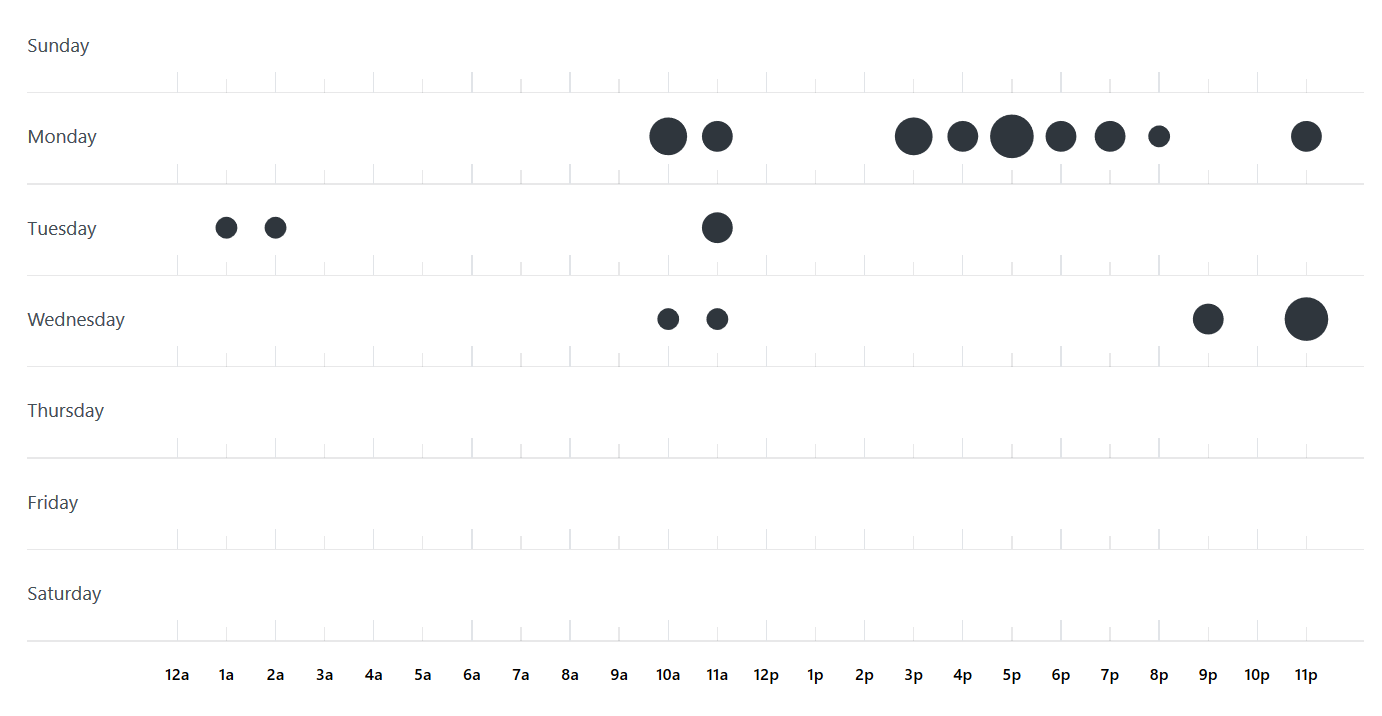
　　关于这回传感器实验，我收获的不仅仅是在TinyOS平台下进行编程开发及掌握无线传感器节点之间通信的方式，理解网络中的拥塞、丢包的情况，还有的是在组长的介绍下我体会到了git的实用性，在虚拟机上直接将代码传上git只需短短几行代码，而且在git上我们可以清晰地看到上传代码的区别，这样大大的方便了我去查看代码的改动。在上面我们小组成员可以就自己的问题发表观点，并在一起讨论，大大地提高了我们小组的交流程度，虽然我们的分工不同，但是在git上我们每个人都可以了解彼此的工作，可以从上面看出每个人对于项目的贡献程度，在某种程度上来说，这大大激发了我们对于项目开发的积极性，在提供方便快捷的使用体验的同时，更是加强了组员之间的合作能力，这种实用方便的工作平台是我们此次实验之外的又一大收获！

**附录３：Git使用截图**

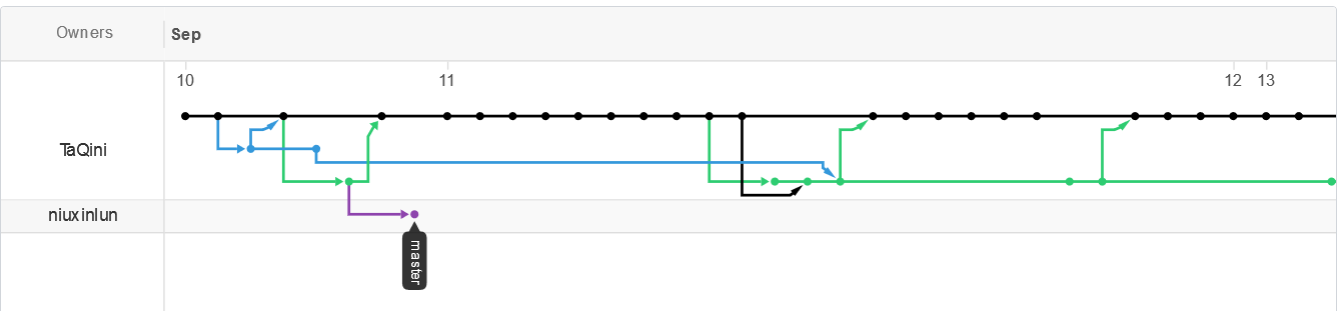
组内成员：

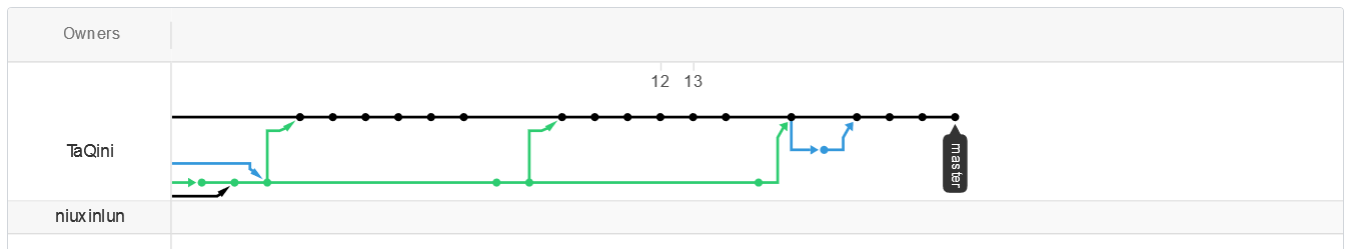


Punch Card：



Network：





**Contributors:**

