

高翔, 高强, 张华熊. 基于 Zigbee 长报文传控协议研究与实现[J]. 智能计算机与应用, 2024, 14(8): 23-28. DOI: 10.20169/j.issn.2095-2163.240804

基于 Zigbee 长报文传控协议研究与实现

高翔¹, 高强², 张华熊³

(1 浙江理工大学 信息科学与工程学院, 杭州 310018; 2 浙江传媒学院 媒体工程学院, 杭州 311231;

3 浙江理工大学 计算机科学与技术学院, 杭州 310018)

摘要: 越来越多的物联网设备采用 Zigbee 技术和网络通信协议实现互联互通, 但是 Zigbee 协议每片报文的最大长度仅为 128 字节, 严重限制了长报文在 Zigbee 协议上的传输。针对此问题, 本文提出了一种基于 Zigbee 的长报文传控协议, 设计了能够满足长报文分片和重组需求的协议头部字段, 开发了协议栈的程序, 并加入了超时机制和滑动窗口来保证传输质量和速率。实验结果表明该协议能够使 Zigbee 完成可靠和快速的长报文传输, 验证了其在嵌入式系统通信中的可行性、稳定性和可靠性。

关键词: 通信协议; 嵌入式; 物联网; Zigbee; 协议栈

中图分类号: TP393.1

文献标志码: A

文章编号: 2095-2163(2024)08-0023-06

Research and implementation of long message transmission control protocol based on Zigbee

GAO Xiang¹, GAO Qiang², ZHANG Huaxiong³

(1 School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China;

2 School of Media Engineering, Communication University of Zhejiang, Hangzhou 311231, China;

3 School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: More and more IoT devices are using Zigbee technology and network communication protocols to achieve interconnectivity. However, the maximum length of each message in the Zigbee protocol is only 128 bytes, which seriously limits the transmission of long messages on the Zigbee protocol. In response to this issue, this article proposes a Zigbee based long message transmission control protocol, designs protocol header fields that can meet the requirements of long message sharding and reassembly, develops protocol stack programs, and adds timeout mechanisms and sliding windows to ensure transmission quality and speed. The experimental results show that the protocol can enable Zigbee to complete reliable and fast long message transmission, verifying its feasibility, stability, and reliability in embedded system communication.

Key words: communication protocol; embedded; IoT; Zigbee; protocol stack

0 引言

物联网是利用二维码、RFID (Radio Frequency Identification)、各类传感器、敏感器件等使物体与互联网等各类网络相连, 获取无处不在的现实世界信息, 实现物与物、物与人之间的信息交互, 支持智能的信息化应用, 实现信息基础设施与物理基础设施的全面融合, 最终形成统一的智能基础设施^[1]。目前, 越来越多的终端设备基于 Wi-Fi、蓝牙或 Zigbee

这 3 种通信协议实现了互联接入。Zigbee 以低功耗、易部署、低成本、高可靠性和高安全性等诸多优势, 在工业互联网领域得到广泛应用^[2]。然而, 在实际应用过程中, 由于 Zigbee 协议最多只能支持每片 128 字节的报文长度, Zigbee 协议应用场景具有局限性^[3]。崔媛媛^[4]提出了一种基于 Zigbee 协议实现数据分片传输方法, 协调器接收数据后根据源地址定点反馈, 终端节点收到反馈后发送下一个分片, 数据重组时协调器接收并根据序号和标识信息

基金项目: 浙江省科技厅重点研发计划项目 (2021C03192)。

作者简介: 高翔 (1999-), 男, 硕士研究生, 主要研究方向: 嵌入式与物联网技术; 高强 (1986-), 男, 博士, 讲师, 主要研究方向: 软件定义网络, 网络安全, 嵌入式系统等; 张华熊 (1971-), 男, 博士, 教授, 主要研究方向: 智能信息处理。

收稿日期: 2023-04-24

重组数据分片,最后一个分片到达后开始重组完整数据包,提高了数据传输效率和完整性;Liu^[5]提出了一种基于FPGA(Field Programmable Gate Array)的多模MAC层控制器,使用该控制器帧封装、传输、自动确认传输和自动帧重传,但该方法需要额外的硬件设备,增加了设备部署成本。

本文针对上述问题,提出了一种基于Zigbee的长报文传控协议,根据长报文分片和重组需求,设计了协议头部字段、实现了协议栈的程序开发,同时加入了超时机制和滑动窗口来保证传输质量和速率;最后,实现该Zigbee传控协议,并对其进行了报文传输测试,验证了协议的有效性和可靠性。

1 常用无线传输网络技术分析

无线传输网络是指利用无线通信技术进行数据传输的网络。相比有线网络,无线传输网络具有部署方便、布局灵活、使用方便等优点,广泛应用于家庭、办公场所和公共场所等各个领域^[6]。随着无线通信技术的不断进步,无线传输网络的数据传输速率大大提高,应用场景也更加多样化。

针对物联网使用场景的不同,诞生了众多物联网无线协议。有针对近距离通信的RFID、低功耗蓝牙,也有中距离通信的Zigbee、Z-Wave、Wi-Fi技术,远距离的LoRa、NB-IoT技术,这些协议使得物联网设备能够应用于多种环境^[7]。在众多物联网无线协议中,Wi-Fi传输速度快、覆盖范围大,但能耗高、设备成本高^[8];蓝牙虽然功耗和成本低,但传输速率低,连接数量有限^[9];Zigbee具有能耗低、自组网、稳定等优点,但却存在设备兼容性低、数据帧短的问题^[10]。

1.1 Wi-Fi

Wi-Fi(全称Wireless Fidelity)是一种无线网络技术,实质是数字信号与无线电信号的转换,发送方与接收方分别实现了转换与还原数据内容的操作^[11]。Wi-Fi利用无线电波进行数据传输,使电子设备之间能够无线连接到互联网或局域网。Wi-Fi技术基于IEEE 802.11标准,使用2.4 GHz或5 GHz频段的无线电波进行通信^[12]。

Wi-Fi技术通过一个无线接入点(Access Point,简称AP)连接到有线网络,然后通过无线信号将数据传输到连接该无线接入点的设备上,实现无线网络连接^[13]。Wi-Fi技术具有高速数据传输、便捷的无线连接、易于部署和使用等优点,因此广泛应用于家庭、办公室、公共场所等各种场合。

1.2 蓝牙

蓝牙技术通常用于连接移动设备(如智能手机、平板电脑和笔记本电脑)与其他设备(如耳机、音箱、键盘和鼠标等),在新能源汽车领域和智能家居领域的应用广泛^[14]。蓝牙技术使用2.4 GHz频段的无线电波进行通信,最初的设计目的是为了消除通过有线连接进行数据传输时的复杂和不便。

在蓝牙技术的发展过程中,不断出现新的蓝牙版本,每个版本都有不同的功能和特性。最新版本的蓝牙技术是蓝牙5.2,具有更快的传输速度、更低的功耗、更强的安全性和更广泛的覆盖范围^[15]。

1.3 Zigbee

Zigbee是一种低功率、低数据传输速率的无线通信技术标准,主要用于智能家居、工业自动化、医疗设备、环境监测等领域。Zigbee技术是一种便宜、低功耗、低数据传输速率和短距离通信的解决方案,用于实现多设备之间的互连。

Zigbee使用IEEE 802.15.4标准,采用2.4 GHz或其他频段的无线电波进行通信^[16]。Zigbee最大传输速率为 $250 \text{ kbit} \cdot \text{s}^{-1}$,最远通信距离可以达到100 m左右。Zigbee网络可以支持多达65 535的设备。

上述3种协议对比见表1。

表1 Wi-Fi、蓝牙、Zigbee对比
Table 1 Compare with Wi-Fi, Bluetooth, Zigbee

协议	传输距离/m	传输速率	波段/GHz
Wi-Fi	100	11 Mbit · s ⁻¹ (IEEE 802.11b) 54 Mbit · s ⁻¹ (IEEE 802.11a) 9.6 Gbit · s ⁻¹ (IEEE 802.11ax)	2.4/5
蓝牙	10	1 Mbit · s ⁻¹	2.4-2.485
Zigbee	100	250 Kbit · s ⁻¹	2.4

2 Zigbee传输控制协议

2.1 设计思路

传输控制协议意在提供一个可靠的数据传输服务,从而确保数据的完整性和可靠性。设计一款Zigbee传控协议的重点和难点在于在嵌入式端以最小资源消耗来完成协议栈的运行。对于Zigbee协议来说,由于采用IEEE 802.15.4标准,每片数据长度最长只能达到128字节,因此在设计协议和协议栈时不能浪费任何一个字节。

1)长报文的传输问题

由于Zigbee协议本身最大支持每片128字节的数据长度,因此必须对长报文进行分片处理,涉及到分片,协议头中必须包含本次通信的标识号和当前

分片的序号,确保分片接收后的识别和匹配重组,并且要考虑判断分片结束的问题,需要添加分片结束标志位加以判断。为了解决分片传输会出现分片丢失问题,还需要在协议栈中加入超时重传的功能。

2) 长报文的传输速度控制问题

对于嵌入式设备来说,大部分情况系统内存不会很大,如果出现服务器端在相对短的时间内发送大量数据分片到达设备的情况,则可能会造成设备宕机,并且宕机的主要原因是因为报文长度过长而导致设备内存溢出,所以设计还需要考虑从协议角度控制发送速度和数量的问题。为了解决这个问题,需要加入滑动窗口机制。

3) 服务器端与设备之间的信息同步问题

无论是服务器端程序还是设备,都会存在重启、断电、宕机等问题,从而导致服务器端数据与设备现存数据不匹配问题。为了解决这个问题,需要添加

强制同步功能,以此来保证服务器端与设备总保持在同一报文的通信上。

4) 有效负载问题

上文所涉及的分片传输、超时机制、滑动窗口机制等等,无论设计怎样的协议,其最终目标是为了保证服务器端与设备之间传输数据的稳定性和准确性,因此与有效负载有关的数据长度、校验码等协议字段也是必不可少的要素。

2.2 技术实现

根据上述分析可知,协议头中必须包含本次通信的标识号、当前分片的序号、窗口大小、分片结束标志位、强制同步标志位、数据长度、校验码,协议栈需要包含超时重传和滑动窗口机制。为了协议栈通用性和易移植性,还需要增加片类型标志位,目的是为了无论是服务器端还是设备都可调用同一接口。具体协议头部结构及其解释见表 2。

表 2 协议头部结构及解释

Table 2 Protocol header structure and interpretation

名称	变量类型	解释
通信标识号	unsigned int	本次通信唯一确认号
当前分片号	unsigned char	当前分片的序号
可缓存分片数	unsigned char	最大可缓存分片数
保留位	unsigned char:5	保留
片类型	unsigned char:1	报文由服务器端到设备或相反
分片结束标志位	unsigned char:1	本片是否为结束分片
强制同步标志位	unsigned char:1	同步服务器端与设备的报文
数据长度	unsigned char	本片有效负载的长度
校验码	unsigned char	计算有效负载的 CRC8 的值

表 2 中,可缓存分片数的大小根据终端设备的改变而改变;片类型中报文由服务器端到设备端取值为 0,相反则取 1;当本片分片结束标志位为 1 时,表示本片为本次的最后一块,否则取 0。在协议头中需要注意保留位,片类型、分片结束标志位和强制同步标志位使用位域的方法,这 4 个标志位共享一个字节,按照表 2 中顺序,由高到低保存。

2.3 可缓存分片数机制

为了减少设备的计算量,保证小型嵌入式设备的正常运行,添加了可缓存分片数字段。因为大部分嵌入式设备并没有涉及操作系统,所以程序在编译好之后内存使用量已经确定,所以本文采用固定缓存分片数,如公式(1)所示:

$$\text{最大可缓存分片数} = \frac{(\text{内存总大小} - \text{内存使用量}) \times K}{\text{Zigbee 应用层数据最长长度}} \quad (1)$$

其中,内存总大小和内存使用量的单位均为字节, K 代表分片内存使用率,理论上取值范围为 $(0, 1]$,但为了避免内存溢出产生不可控的问题,建议 $K < 1$ 。Zigbee 应用层数据最长长度一般为 128 字节。

2.4 超时机制

通信过程中,接收方每收到一片数据,就会将此分片的识别号、当前分片号和接收成功标志返回,以此来告知发送方此分片接收成功。

计算超时时间的算法有很多,例如:在 Karn 算法使用重传计时器和忽略计时器测量方法来解决 TCP 中的计时器失序问题^[17];Jacobson 算法是一种改进的 RTT(Round-Trip Time)估算算法,用于计算和更新 TCP 连接中的 RTT 和超时时间^[18];NewReno 算法在处理丢失数据包时的快速重传和快速恢复机制上进行了优化^[19];TCP Vegas 算法基于拥塞控制

和 RTT 变化来调整超时时间,通过比较实际 RTT 和预期 RTT 来决定超时时间的增减^[20]。本文采用的是 Karn 算法。关于重传次数计算,根据实际情况而定,本文重传次数为 3 次。

2.5 滑动窗口机制

由于超时机制的存在,发送方在发送一片数据后必须停下来等待接收方的确认回复,收到之后才可以发送下一分片,导致发送效率低下。滑动窗口机制的改进在于其让发送方可以同时发送多个数据,然后等待接收方的确认,这样就大大提高了网络的利用率。窗口大小一般为最大可缓存分片数的 1/3。

2.6 分片与重组机制

分片与重组过程在服务器端较易实现和运行,然而设备端需要考虑可用内存的问题。因此在设备上电后,会主动向服务器端发送设备上数据,包中携带了包括设备识别码、最大可缓存分片数等与设备通信相关的信息,服务器将数据记录到本地,在此后的通信过程中,会遵循设备的最大可缓存分片的要求。设备也会根据最大可缓存分片数来申请一块内存,用于缓存与服务器之间的通信数据。

在通信过程中,服务器根据数据长度进行分片,并在协议头中使用相同的标识号和递增的分片号来实现分片,而终端设备也需要按照标识号和分片号进行重组。假设数据包分片数为 5,最大可缓存分片数为 10,窗口大小为 3,在超时机制和滑动窗口机制的协同作用下,分片与重组机制运行过程如图 1 所示。

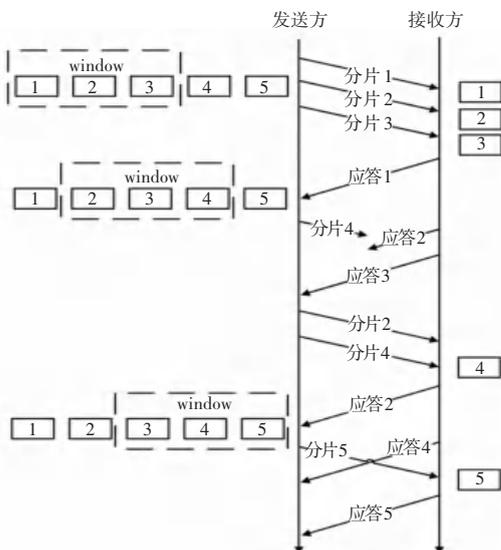


图 1 分片与重组机制运行过程

Fig. 1 Operation process of fragmentation and recombination mechanism

发送方现有一包完整数据经过分片后形成 5 片报文。根据协议规定,发送方根据滑动窗口机制同时发送 3 片报文到接收方,收到数据后,将前三片数据储存在缓存中,并且成功收到应答 1。发送方移动窗口到 2、3、4 分片位置,并发送了分片 4。但是由于不可控原因导致连接中断,分片 4 发送与应答 2 回复均失败,导致发送方进入等待应答状态。接收方成功回复了应答 3,但是由于发送方没有收到应答 2,所以继续等待并且窗口没有移动。应答 2 超时后发送方第二次发送分片 2,紧接着分片 4 超时,并第二次发送了分片 4,接收方收到分片 2 后发现已存在,则直接丢弃并回复应答 2,接收到分片 4 后直接保存并应答。发送方收到应答 2 后,移动窗口,发送分片 5,接收方收到分片 5 后应答,接收方收到应答 5,数据传输完成,本次通信结束。

3 测试

3.1 测试要求

本文对实际应用中的协议栈设计实现提出 4 种必须要考虑的情况:

1) 服务器发送报文,设备接收并应答,服务器接收应答成功

该情况属于双方正常通信情况,协议栈应该能够正常的收发报文,并且正常的进行超时判断和窗口滑动。

2) 服务器发送报文,设备没有收到

该情况属于网络状态问题,协议栈应该触发超时机制进行处理,超时机制处理结束之后还没有发送成功,则丢弃本次通信的所有分片并通知上层。

3) 服务器发送报文,设备接收,未通过数据校验

该情况属于数据传输问题,会出现数据长度验证错误或校验码校验错误,此时协议栈需要告诉上层应用并且需要在应答包中体现。

4) 服务器发送报文,设备接收,服务器未收到应答

该情况对应设备在收到上一条服务器分片后应答,但应答报文丢失,或应答时与服务器断连,会导致服务器在等待超时后以为上一个分片没发送成功,会再次发送一遍相同的分片,单片机再次收到一遍相同的分片,从而报告标识号重复或分片重复,服务器收到后可以知道设备上分片已经收到,只是返回时丢失,这时可继续发送下一个分片。

3.2 测试环境

测试主要在嵌入式设备上进行,主控芯片通过串口与 Zigbee 芯片进行通信,测试程序的嵌入式设备硬件环境信息见表 3。

表 3 设备硬件环境

Table 3 Device hardware environment

名称	详细信息
主控芯片	GD32 F303 RCT6
主频/MHz	108
Flash/kB	3 072
SRAM/kB	96
主控串口波特率/(bit · s ⁻¹)	115 200
Zigbee 芯片	Tuya ZT3L
主频/MHz	48
数据传输速率/(kbit · s ⁻¹)	250

后端程序部署在阿里云服务器上,服务器环境信息见表 4。

表 4 服务器环境

Table 4 Server environment

名称	详细信息
主控芯片	Intel(R) Xeon(R) Platinum 8269CY
主频/GHz	2.50
内存/GB	3
硬盘/GB	40
操作系统	CentOS 7
网卡带宽/Mbps	10
JDK 版本	17
Spring Boot	v2.5.6
TuyaCloud Java SDK	17.0.4.1

3.3 测试实例

本测试基于涂鸦云平台传输数据,由于平台限制,报文长度最长为 64 字节。

1) 分片功能测试

该测试共分为两种:单片数据传输测试和分片数据传输测试。单片数据传输中数据总长度为 31 字节,未达到 64 字节,所以报文无需分片,未分片报文传输运行日志如图 2 所示。

```

[DeviceService.java:70] 下发数据包的指令长度为 31
[ZigBeePacketUtil.java:33] 下发数据包的指令长度为 31
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 31
[ZigBeePacketUtil.java:85] 下发数据包的指令长度为 31
[MainService.java:181] 下发数据包的指令长度为 31
[DeviceService.java:70] 下发数据包的指令长度为 31
[DeviceService.java:141] 下发数据包的指令长度为 31

```

图 2 未分片报文传输

Fig. 2 Transport of unfragmented messages

分片数据传输,数据总长度 171 字节,超过 64 字节,执行分片传输功能。分片报文传输运行日志

如图 3 所示。

```

[DeviceService.java:70] 下发数据包的指令长度为 171
[ZigBeePacketUtil.java:33] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:85] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[DeviceService.java:181] 下发数据包的指令长度为 171
[DeviceService.java:181] 下发数据包的指令长度为 171
[DeviceService.java:141] 下发数据包的指令长度为 171

```

图 3 分片报文传输

Fig. 3 Transport of fragmented messages

2) 超时功能与窗口机制联合测试

无论窗口大小,无论分片多少,只要有一片数据没有接收成功,整包数据就会被丢弃。当分片数为 4,窗口为 1、超时次数为 3 时,该情况对应一收一答的报文传输机制,超时发送失败的情况如图 4 所示。

```

[DeviceService.java:70] 下发数据包的指令长度为 171
[ZigBeePacketUtil.java:33] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:85] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[DeviceService.java:181] 下发数据包的指令长度为 171
[DeviceService.java:181] 下发数据包的指令长度为 171
[DeviceService.java:141] 下发数据包的指令长度为 171

```

图 4 超时发送失败(窗口大小为 1)

Fig. 4 Timed out sending failed (the size of window is 1)

当分片数为 4、窗口为 3、超时次数为 3 时,属于对应报文分片数超过窗口大小,分片数据不可以一次性发出的情况,超时发送失败情况如图 5 所示。

```

[DeviceService.java:70] 下发数据包的指令长度为 171
[ZigBeePacketUtil.java:33] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:85] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[DeviceService.java:181] 下发数据包的指令长度为 171
[DeviceService.java:181] 下发数据包的指令长度为 171
[DeviceService.java:141] 下发数据包的指令长度为 171

```

图 5 超时发送失败(窗口大小为 3)

Fig. 5 Timed out sending failed (the size of window is 3)

当分片数为 4、窗口为 5、超时次数为 3 时,属于对应报文分片数小于窗口大小的情况,超时发送失败情况如图 6 所示。

```

[DeviceService.java:70] 下发数据包的指令长度为 171
[ZigBeePacketUtil.java:33] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:85] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[ZigBeePacketUtil.java:81] 下发数据包的指令长度为 43
[DeviceService.java:181] 下发数据包的指令长度为 171
[DeviceService.java:181] 下发数据包的指令长度为 171
[DeviceService.java:141] 下发数据包的指令长度为 171

```

图 6 超时发送失败(窗口大小为 5)

Fig. 6 Timed out sending failed (the size of window is 5)

4 结束语

本文从实际应用中的问题出发,设计了一种基于超时机制和窗口机制的 Zigbee 长报文传控协议,通过测试验证了此协议可以完成对长报文的分片传输功能,并且在遇到网络问题或者一些不可预测问题时,协议栈会按照设计的机制进行处理,该 Zigbee 长报文传控协议具有可行性、稳定性和可靠性。

参考文献

- [1] 李安民. 物联网与信息通信[J]. 电信科学, 2010,26(12):1-4.
- [2] 王立新, 郭凰, 杨佳宇, 等. 无线通信在结构健康监测系统的应用研究综述[J]. 科学技术与工程, 2023,23(6):2229-2241.
- [3] KAMAL S O, KHAN B M. Hardware implementation of IP-Enabled wireless sensor network using 6LoWPAN [C]//Proceedings of 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA). IEEE, 2021: 227-233.
- [4] 崔媛媛. 基于 ZigBee 和 QT 的农产品信息采集系统的设计与实现[D]. 长春: 吉林大学, 2019.
- [5] LIU W, CHEN Z, LUO G, et al. A MAC controller design for Zigbee and WiFi hybrid network based on FPGA [C]//Proceedings of 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). IEEE, 2018: 430-434.
- [6] 郑逢斌, 朱东伟, 臧文乾, 等. 边缘计算: 新型计算范式综述与应用研究[J]. 计算机科学与探索, 2020,14(4):541.
- [7] 张伟康, 曾凡平, 陶禹帆, 等. 物联网无线协议安全综述[J]. 信息安全学报, 2022,7(2):59-71.
- [8] SINGH S, RAY K K. Home automation system using internet of things [J]. International Journal of Computer Engineering and Applications, 2017,28(451): 2321-3469.
- [9] DAS S, GANGULY S, GHOSH S, et al. Bluetooth based sophisticated home automation system using smartphone [C]//Proceedings of IEEE. 2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI). New York: IEEE, 2016:236-240.
- [10] JAKOVLJEV S, SUBOTIĆ M, PAPP I. Realisation of a smart plug device based on Wi-Fi technology for use in home automation systems[C]//Proceedings of IEEE 2017 International Conference on Consumer Electronics (ICCE). New York: IEEE, 2017:327-328.
- [11] 包淳溢. 基于 WiFi 和蓝牙 5.0 的智能家居控制系统研究与设计[D]. 杭州: 浙江工业大学, 2020.
- [12] 刘新欣, 王晓东. 基于 IEEE 802.11 无线局域网的 VoIP 性能分析[J]. 计算机应用, 2010,30(S1):37-39.
- [13] 王劲阳. 一种基于无线专网的 CTC3.0 多岗位联动技术方案[J]. 铁路通信信号工程技术, 2024,21(2):26.
- [14] 朱昭华. 浅析蓝牙技术[J]. 电声技术, 2018,42(4):70-72.
- [15] WANG J, LIANG H. Bluetooth 5.2 technology and application [J]. International Journal of Advanced Network, Monitoring and Controls, 2021,6(2):32-36.
- [16] 崔文华. ZigBee 协议栈的研究与实现[D]. 上海: 华东师范大学, 2007.
- [17] 任克强, 刘昌时. 一种解决 TCP 协议重传多义性的有效算法[J]. 通信技术, 1993(1):25-29.
- [18] 张 钊, 肖宇峰, 张 华. 基于帧长的巨帧网络时延测量方法[J]. 郑州大学学报(理学版), 2013,45(1):56-60.
- [19] 陈 琳, 双雪芹. TCP 网络拥塞控制算法比较研究[J]. 长江大学学报(自然科学版)理工卷, 2010,7(1):60-62,79,383.
- [20] 蒋万春, 李昊阳, 陈晗瑜, 等. 网络拥塞控制方法综述[J/OL]. 软件学报, [2024-08-18]. [Http://doi.org/10.13328/j.cnki.jos.007045](http://doi.org/10.13328/j.cnki.jos.007045).