文章编号:1671-4598(2018)06-0094-05 DOI:10.16526/j.cnki.11—4762/tp.2018.06.024 中图分类号:TP274 文献标识码:A

基于 STM32 的智慧路灯控制系统设计与实现

金山城,田 茂,段 沛,王雄兵,孙 军
(湖北大学 计算机与信息工程学院,武汉 430062)

摘要:随着物联网科技的不断发展,智慧路灯成为了智慧城市发展过程中不可或缺的重要组成部分;传统的城市照明路灯仅仅只能满足简单的照明需求,并且在控制局部照明上无法实现实时以及自由控制,只能按照季度的日出日落时间来固定设置路灯的开关灯,不仅浪费了人力物力,而且对于能源也是极大的浪费;设计了基于 STM32 的路灯集中控制器,该集中控制器通过 GPRS 与后台通信服务器连接,实现实时数据的回传、在线命令和策略的下发,最后对系统进行测试与分析;实验结果表明,该方案不仅解决了路灯的智能化控制,而且具有高度的可扩展性,极大地方便了城市照明,更实现了高并发通信,从而更安全可靠地达到对城市照明的目的。

关键词:智慧路灯;控制器;城市照明;GPRS

Design and Implementation of Intelligent Street Lamp Control System Based on STM32

Jin Shancheng, Tian Mao, Duan Pei, Wang Xiongbing, Sun Jun
(School of Computer Science and Information Engineering, College of Hubei University, Wuhan 430062, China)

Abstract: With the development of the Internet of Things, the intelligent street lamps have become an important part of the development of the Smart city. But the traditional urban street lamps can only meet the needs of simple lighting and can not achieve real-time and free control in the control of local lighting. The switches of street lamps which can only be set on the basis of the sunrise and sunset of the quarternot only waste manpower and material resources, but also greatly waste energy consumption. The integrated controllers of lambs which are designed based on STM32 can return data in real time, command online, delivery strategiesthrough the connection of GPRS and background communication server, finally, the system is tested and analyzed. Thus, The experimental results show thatit can not only solve the intelligent control of street lamps, but also have high scalability, more implementation of high concurrency communicationwhich greatly facilitate urban lighting. The system of street lambs can also store lighting strategies making urban lightning safer and more reliable.

Keywords: smart street lamp; controller; urban lighting; GPRS

0 引言

随着物联网科技的不断发展,智慧路灯成为了智慧城市中不可或缺的重要组成部分[1]。智慧路灯控制系统能够实现网络化的照明控制服务,并且通过与后台的通信,能实时的对路灯进行数据采集与回传,智能化的下发控制策略,实现对各个路灯的实时准确控制。通过无线 GPRS 通信技术,实现对路灯的远程集中控制与管理,具有远程照明控制、故障报警、远程抄表等功能,能够大大节省电力资源,提升城市照明管理水平,节省运维成本。

伴随着科技迅猛发展,智慧路灯的数量会越来越多,对于网络通信的要求也越来越高,无法满足现代路灯的数据通信需求。针对传统路灯系统所面临的通信问题,本系统采用基于 Netty 框架而设计的通信模块,不仅有效地实现了高并发通信,同时也更适应智慧城市发展的需求。

收稿日期:2018-03-30; **修回日期**:2018-04-23。

基金项目:湖北省教育厅科学技术研究计划青年人才项目(Q20161012)。

作者简介:金山城(1994-),男,湖北安陆人,硕士生,主要从事计算机应用系统方向的研究。

田 茂(1970-),男,湖北武汉人,副教授,硕士生导师,主要从事嵌入式系统设计,物联网技术方向的研究。

通讯作者:孙 军(1979-),女,湖北襄阳人,讲师,主要从事软件工程,网络安全方向的研究。

1 整体系统架构

智慧路灯控制系统主要由路灯集中控制器和后台通信系统两大部分组成。系统除具有自动早晚开关灯功能外,可利用策略控制,实现夜间其他时段自动通断功能,每个回路的采集器至终端控制平台的通讯为有线网络或无线传输方式,每个回路具有计量功能,通过 RS485 通讯、GPRS 等方式来实现远传电表的数据,路灯控制系统需将每个回路的用电量最终传输到后台进行统计分析,可自动生成用电量日报表、月报表和年报表,并与路灯管理云平台对接,并且集中控制器具有本地数据存储的功能,以保证系统的可靠性、稳定性和安全性。

从成本、灵活性等方面考虑单片机嵌入式系统可以满足需求,图 1 为智慧路灯控制系统组织结构图。本地集中控制器通过继电器模块控制现场路灯回路,集中控制器通过 GPRS 模块与服务器通信,实现后台通信系统指令下发和路灯集中控制器数据上传至服务器的功能,从而达到安全可靠的控制各个路灯的开关灯。

2 系统硬件设计

路灯集中控制器的硬件结构如图 2 所示。根据电路功能可以将硬件结构划分为核心控制模块、电源模块及各主要功能模块。

主要功能模块由 NorFlash 存储模块、三相数据采集模块、

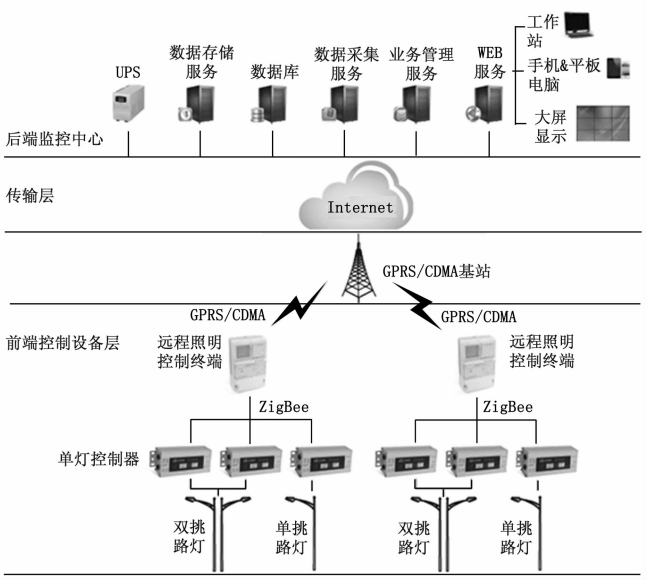

图 1　智慧路灯控制系统组织结构图

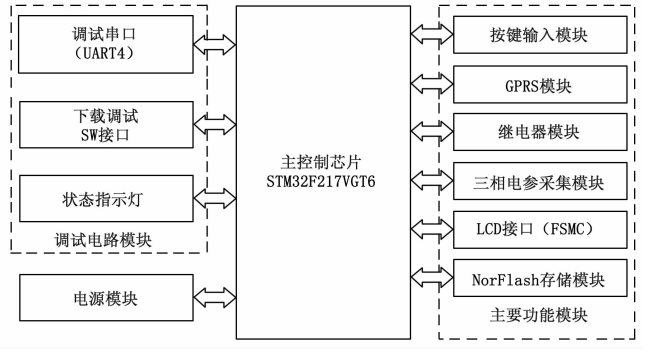

图 2　集中控制系统硬件结构图

GPRS 通信模块电路等组成。核心控制模块通过控制继电器输出模块来实现路灯回路控制；通过 GPRS 通信模块连接后台通信服务器，实现数据的上传、指令和策略的接收；NorFlash 存储模块对一些重要参数信息进行存储；三相数据采集模块采集路灯控制器各个回路的供电电压、电流和有功功率等电能数据[2]。

路灯集中控制器通过 GPRS 通信模块接收和发送所有路灯控制状态信号，并将数据记录在存储芯片中，路灯集中控制器处于后端监控中心和前端路灯设备之间，向上通过 GPRS 方式与后台通信系统联网通信，向下则是通过回路控制的方式，控制各个路灯开关灯；三相电参模块可以实现对电压、电流、有功功率等数据的采集，实时监控智慧路灯控制器的运行状况。

2.1　核心控制模块

核心控制模块综合考虑处理速度、低功耗、是否满足各个不同模块接口以及后期功能扩展等因素，系统选用基于 Cortex－M3[3]内核的 32 位微控制器 STM32F217VGT6 作为主控制器，主要因为它成本低、运行稳定可靠，并且它具有多路 SPI 通信接口，满足数据信息的采集的需求，多路串口也满足核心系统与子模块之间的数据通信要求。

2.2　NorFlash 存储模块

路灯集中控制器需要保存设备 ID、服务器 IP 及端口号、密钥等配置参数，并且需要存储路灯的控制策略。由于集中控制器需要存储的数据需求较大，在不影响整体系统的性能以及成本等条件下，选择具有 8 M 容量的 NorFlash 存储芯片 W25Q128，它具有功耗低，宽温度范围，并且高效的“连续读取”性和高安全性的特点，适合本系统设计的需求。

2.3　数据采集模块

数据采集模块的核心芯片是 ATT7022－EU，它是一片多功能高精度的三相电能专用计量芯片，通过它可以精准的采集路灯的电压、电流、有功功率，可以实时监测路灯的工作状态和耗电量。它通过 SPI 通信接口与核心控制器模块进行数据交互[4]，大大提高数据传输的效率，并且 SPI 具有全双工操作，操作简单。

2.4　GPRS 通信模块

路灯集中控制器通过 GPRS[5]通信模块与后台通信系统进行网络通信，并且保持实时在线，以便于实时控制路灯的开关灯和监测路灯控制器的工作状态。

GPRS 通信的资源利用率高，它引入了分组交换的传输模式，集中控制器只有在发送或接收数据期间才占用资源，这意味着多个集中控制器可高效率地共享同一无线信道，从而提高了资源的利用率；并且 GPRS 接入时间短，使得集中控制器与后台通信系统能够很快的进行连接通信；通信可靠性高，实现了 TCP/IP 协议的高效传输速率，在通信过程中，也能保证通信的安全性和稳定性。

3　系统软件设计

本路灯集中控制器软件系统是基于嵌入式操作系统 UCOS II 设计的[6]。UCOS II 是一款广泛应用于嵌入式控制器中的抢占式实时操作系统，具备多任务实时处理，任务灵活调度，多任务间信息传递，简单的时间管理和有效的内存管理等功能，并且 UCOSII[7]拥有良好的可扩展性和源码开放，用户可根据自己的具体需求来实现不同的功能。

路灯集中控制器软件系统的设计是基于软件功能采取多任务的方式实现的。主要分为 6 个任务：

1）按键任务（key _ task）：主要功能是获取按键的键值，并通过消息邮箱把键值传递给其他任务；

2）通信任务（communication _ task）：主要功能是控制器与后台服务器的通信连接，以及基于应用层通信协议的数据通信；

3）液晶显示任务（lcd _ task）：主要功能是显示控制器相关的状态和数据信息；

4）继电器任务（relay _ task）：主要功能是控制输出和电能参数的采集和处理；

5）电能采集任务（electricAcq _ task）：主要是负责电能参数的采集；

6）配置任务（config _ task）：主要能够通过上位机的配置助手实现对集中控制器的参数配置。

其中，通信任务是软件系统的核心部分，承担了 GPRS 通信模块的配置，路灯控制器与后台通信系统的连接，以及基于该连接的应用层通信协议的实现。

3.1　通信任务设计

通信任务的设计是路灯集中控制器与服务端通信模块之间联网通信的核心。通信任务主要包括系统初始化、GPRS 模块

初始化、建立 TCP/IP 连接、通信协议任务处理部分。

其中通信协议任务处理主要处理通讯状态转换，发送和接收数据，协议数据解析，以及数据处理等[8]。

软件系统的设计主要体现在使用 GPRS 模块建立 TCP/IP[9]连接，通过状态机的方式，实现不同运行状态之间的跳转。这些状态包括建立 TCP/IP 连接的状态流程，以及数据传输的状态流程，保证 TCP/IP 连接的稳定性以及数据传输的完整性和安全性[10]。

路灯集中控制器系统通信任务程序主流程如图 3 所示。

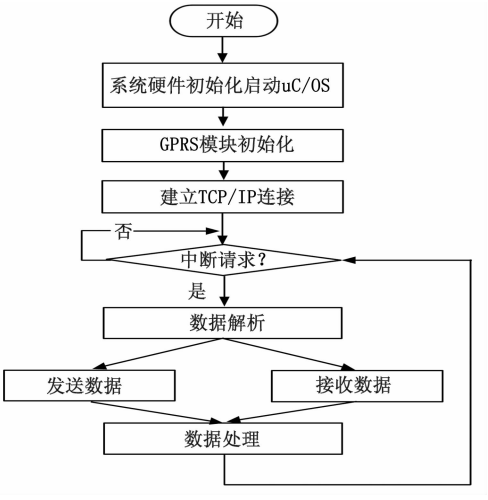

图 3 通信任务主流程图

3.2 应用层通信协议制定

根据智慧路灯控制系统的数据通信业务需求及对系统安全性方面的考虑，本系统设计了一套通信协议，路灯集中控制器依据此协议与服务端进行数据交互。

3.2.1 数据帧结构

通信数据帧结构用以规范路灯集中控制器与后台通信系统通信时的传输数据格式。数据帧结构由六部分组成，实际应用中顺序不能打乱，由左往右依次是包头、协议号、数据长度、数据类型、数据部分和 CRC 校验。数据帧结构如图 4 所示。

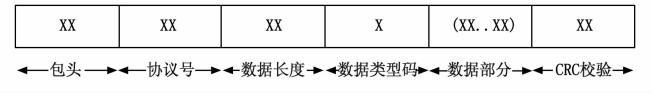

图 4 数据帧结构图

3.2.2 数据结构

集中控制器与服务端在进行数据通信时最终传递的是字节数组，双方在通信时数据结构必须一样，如表 1 数据结构所示，包头与协议号各占两个字节，分为高八位与低八位；数据长度的数据类型是整型，转换成字节占两个字节，长度 16 位，

表 1 数据结构

XX	2byte	包头
XX	2 byte	协议号
XX	2 byte	数据长度
X	1 byte	数据类型码
XX…XX	N byte	数据部分
XX	2 byte	CRC 校验码

数据长度包括数据类型和数据的长度之和；数据类型码占一个字节；数据部分则按实际确定字节长度，数据部分主要是联网认证、上传数据以及下发数据；最后是 CRC 校验码两个字节，对协议包头、协议号数据长度、数据类型码、数据等进行校验，保证数据的准确性与安全性。

3.2.3 策略定义

策略定义主要为策略定义固定时间段及是否使用经纬度开关灯等功能，作为抽象功能，可为一个或多个路灯集中控制器同时绑定关联定时策略。

本系统对于智慧路灯的控制主要采用了本地策略、经纬度策略、时间策略三种策略进行控制。

1）本地策略定义：本地策略是存储在路灯集中控制器本地的，主要用于断网情况下，依旧可以按照预先下发存储在本地的策略对路灯进行控制。

2）经纬度策略定义：经纬度策略是根据不同地理坐标点的日出日落时间以及日出日落偏移量来计算每一天的开关灯时间，如表 2 经纬度策略结构所示。

表 2 经纬度策略结构

名称	长度/byte	数据	描述
数据类型 ID	1	类别标识	下行
经纬度	2	两位小数	经度
	2	两位小数	纬度
日出偏移	1	有符号	分
日落偏移	1	有符号	分

3）时间策略定义：时间策略是用户自定义开关灯时间，根据不同路段的实际车流量以及所处位置等来自行设置开关灯时间，如表 3 所示。

表 3 时间策略结构

名称	长度/byte	数据	描述
数据类型 ID	1	类别标示	下行
策略编号	2	编号	编号
策略类型编号	1	0 非临时/1 临时	编号
回路	1	回路编号	数量
经纬度开关状态	1	0 关/1 开	状态
策略时间对	1	1 2 3	标记对数
策略时间 1	2	开灯时间	时分
	2	关灯时间	时分

4 高并发通信模块设计

为了解决高并发通信问题，本系统基于 Netty[11]框架设计了通信模块，在通信链路的应用层，为了保证集中控制器客户端与 Netty 通信系统服务端的数据传输的安全性、完整性和可扩展性，Netty 通信模块的业务流程按照制定的应用层协议进行设计。

4.1 Netty 逻辑架构

Netty 采用了三层架构进行设计与开发，它们由上往下分别是通信调度层 Reactor、职责链 ChannelPipeLine 和业务逻辑编排层 Service[12]，如图 5 所示。

第一层：Reactor 通信调度层，负责监听网络读写操作和连续操作，将网络层数据提取到内存缓冲区 ByteBuf 中，触发

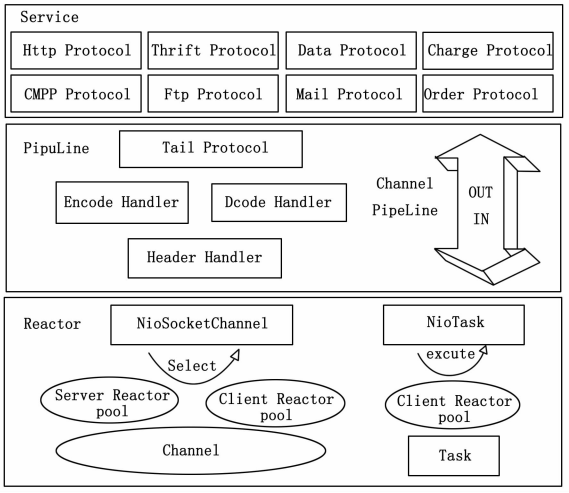

图 5　Netty 逻辑架构图

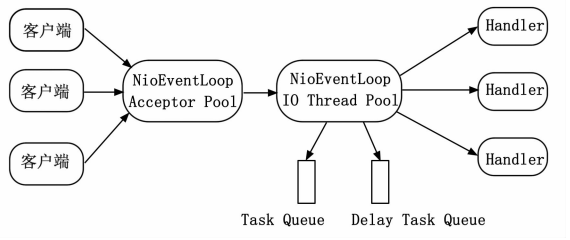

图 7　Netty 线程模型图

```
EventLoopGroup bossGroup = new NioEventLoopGroup(2);
EventLoopGroup workerGroup = new NioEventLoopGroup(8);
try {
    ServerBootstrap b = new ServerBootstrap();
    b.group(bossGroup, workerGroup);
            b.channel(NioServerSocketChannel.class);
            b.option(ChannelOption.SO_BACKLOG, 1024);
```

图 8　Netty 通信模块线程配置代码

各类网络事件。按照 Reactor 模式设计和实现的 Netty 架构，它在服务端的通信时序图，如图 6 所示。

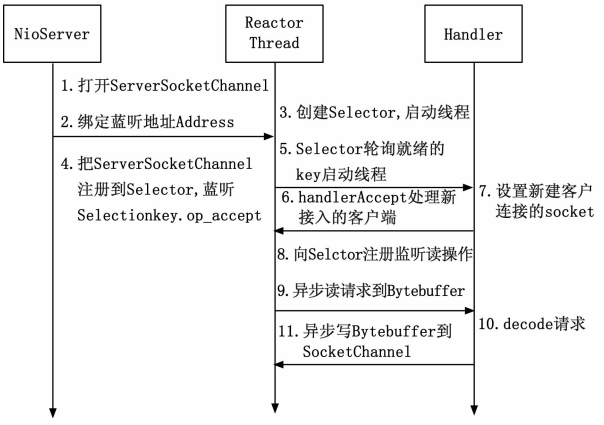

图 6　Netty 通信时序图

第二层：职责链 Pipline，它负责事件在职责链中有序传播，同时负责动态编排职责链。职责链可以选择监听和处理自我关联性较强的事件，它可以拦截处理和后向/前向传播事件。

第三层：最上层是业务逻辑编排层，业务逻辑编排层通常分为两种：一种是纯粹的业务逻辑编排，另一种是应用层协议插件，用于协议相关的编解码和链路管理。

4.2　Netty 线程模型

Netty 支持 Reactor 的单线程、多线程和主从多线程模型多种线程模型。线程模型可以通过设置不同的启动参数，调整线程池的线程个数、是否共享线程池等方式来切换，以便满足不同应用场景的需求[13]。Netty 线程模型如图 7 所示，对应的 Netty 通信模块线程配置代码如图 8 所示。

Netty 通信模块启动时，创建两个线程池 NioEventLoopGroup，实际上它们是两个独立的 Reactor 线程池。bossGroup 用来接收路灯集中控制器的 TCP 连接、初始化参数，将链路状态变更事件通知给 ChannelPipeline，这个线程池设置了两个线程。workerGroup 用来处理 I/O 读写事件、执行系统调用 Task、执行定时 Task 等，这个线程分配一个 Task，当这个 Task 完成时，线程就返回到线程池中，等待下一次分配调用。

创建系统 Task 的原因是，当 I/O 线程和用户线程同时操作一资源时，为了防止并发操作产生的锁竞争，会把用户线程封装为一个 Task 放入消息队列，由 I/O 线程负责执行，这样可以实现局部无锁化。而定时 Task 主要用于监控和检查等定时动作。尽管 Netty 支持多种线程模式，在实际应用中往往创建两个 NioEventLoopGroup，用于逻辑隔离 NIO Acceptor 和 NIO I/O 线程，尽量避免在 ChannelHandler 中自定义用户线程，一般的业务需求使用 NIO 线程组即可完成。每个线程池线程个数的设置没有统一标准，往往通过实际测试决定，一边测试一边调整。默认情况下会设置为 CPU 核数的两倍，常规的线程数量计算公式如下：

线程数量＝（线程总时间/瓶颈资源时间）＊瓶颈资源的线程并行数；

每秒查询率（QPS）＝1000/线程总时间＊线程数。

4.3　Netty 通信模块的实现

Netty 通信模块接收路灯集中控制器数据的流程如图 9 所示。

当 Netty 通信模块接收到路灯集中控制器发送的数据时，首先验证数据的包头和协议号，如果验证错误直接关闭当前连接，如果正确就判断当前路灯集中控制器是否已认证，遍历 map 全局变量，检查 map 中是否存在该集中控制器信息，如果没有就跳转到集中控制器认证流程，验证数据类型、数据类型 ID 是不是登录认证信息，验证通过就把路灯集中控制器信息存入到 map 全局变量中，此控制器以后在发送数据时就不再需要认证。如果之前检查的 map 中已经存在该控制器信息，则进行 CRC16 校验，校验通过就反馈数据包格式错误，通过就开始验证数据类型，检测数据类型是否为策略数据，如果不是则反馈数据包格式不正确。如果通信认证成功，则保持路灯集中控制器与 Netty 通信模块实时通信。

5　实现与分析

通过本地客户端配置软件，给路灯集中控制器配置设备密钥、设备 ID、IP 地址和端口号等基本信息，其中设备密钥用于通信协议的安全传输，设备 ID 用于确定设备身份，IP 地址和端口用于和后台通信系统网络连接，配置界面如图 10 所示。

配置完成后，基于 Web 的后台管理系统可以实时查看路

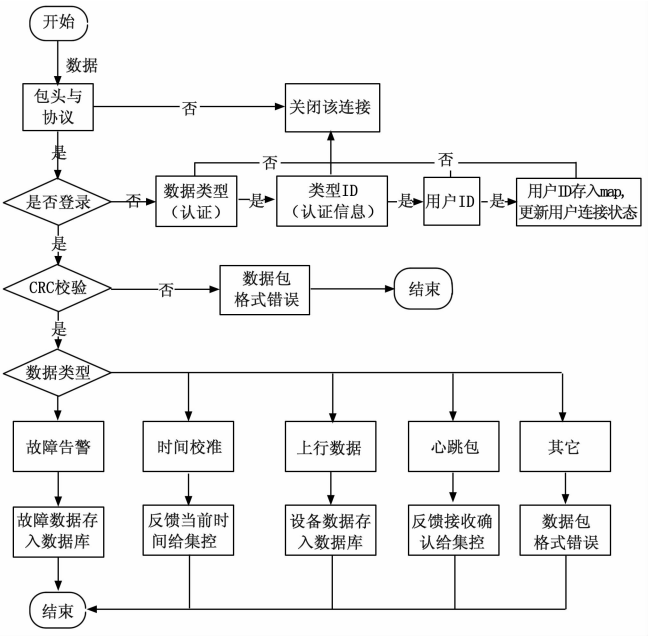

图 9 Netty 通信模块接收数据流程图

图 10 智慧路灯控制器信息配置界面

灯集中控制器的运行状态以及对路灯集中控制器下发时间策略、经纬度策略等控制策略指令，使路灯集中控制器能够通过不同的策略信息实现对道路上路灯断开和闭合的控制；为了保证路灯集中控制器的高可靠性和稳定性，还对路灯集中控制器做了本地策略信息的存储，以保证断网情况下，路灯集中控制器能够正常运行。

根据智慧路灯控制系统整体需求分析，本系统实现了路灯集中控制器与后台通信系统的联网通信，并且能够有效的下发路灯控制策略，而且能够实现策略本地存储。

表 4 系统配置

电脑型号	戴尔 PowerEdge R730 Rack Mount Chassis
处理器	英特尔 Xeon E5－2630 v4
内存容量	32.0GB(2133MHz)
硬盘	三星 EDLL(1999G)
主板	戴尔 0WCJNT
网卡	博通 NetXtreme BCM5720 Gigabit Ethernet PCIe/ 戴尔
操作系统	Windows 2016 Server Enterprise 64 位 SP1

本系统基本完成对于路灯的智能化控制，测试结果满足要求，然而基于 Netty 的通信模块还需要具备高并发的能力，要求能够同时处理多个路灯集中控制器的数据请求，采用专业的压力测试工具 Jmeter[14] 对 Netty 通信模块进行压力测试，通信服务端系统的配置如表 4 所示。测试方法是 Jmeter 分别模拟 0.5 到 5 万个路灯集中控制器同时向 Netty 服务端发送字节组数据，数组长度为 34，Netty 通信模块接收到后向路灯集中控制器反馈一条字节数组，数组长度为 8，Jmeter 测试生成聚合报告[15] 如表 5 所示。

表 5 聚合报告

S	*A*	*M*	90%_*line*	*Min*	*Max*	*E*%	*T*
0.5	49	42	69	39	93	0	2056
1	51	41	69	39	965	0	2704.1
2	54	41	68	39	1255	0	2434.1
3	50	41	68	39	1656	0	2383
4	49	41	69	39	2260	0	2400
5	52	41	68	39	1917	0	2398

表 5 中，*S*（Samples）是发送到服务器的通信请求事务数量（万）；*A*（Average）是平均完成一次响应消耗的时间，即平均响应时间（ms）；*M*（Median）是所有响应时间的中位数（ms）；90% _ *line* 是指 90%的用户请求的响应时间（ms）；*Min* 是服务器响应的最短时间（ms）；*Max* 是服务器响应的最长时间（ms）；*E*%（Error%）是请求的错误百分比；*T*（Throughput）是服务器每单位时间处理的请求数。

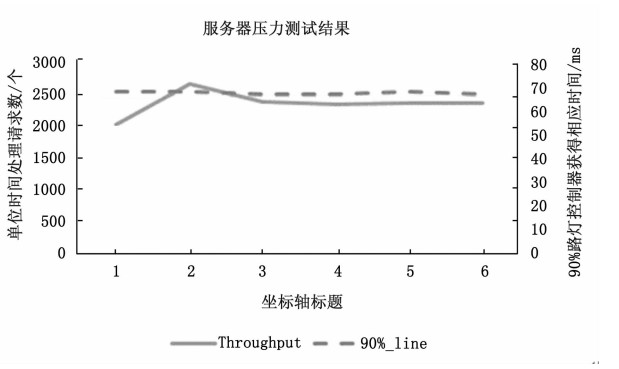

图 11 压力测试折线图

从聚合报告中数据可看出，并发量从 0.5 W 自增到 5 W 的过程中，Netty 通信模块错误百分比始终为零，平均响应时间基本稳定在 41 ms，单位时间处理的请求数（Throughput）集中在 2000 个以上，90%的客户请求响应时间在 69 ms 以内，对应的折线图如图所示。总体而言，在高并发环境下 Netty 通信模块响应及时、稳定、安全、可靠，它不仅可以满足智慧路灯控制系统的数据通信需求，而且可以为大规模城市部署智慧路灯集中控制器提供有力的技术支持和数据参考。

6 结论

随着信息时代的高速发展，智慧路灯在智慧城市的创建中起着不可或缺的作用。本文中设计的基于 STM32 的智慧路灯控制系统运行稳定可靠，有效地解决了传统路灯高耗能、高管理成本等一系列问题，并且本系统设计的通信协议具有高效的

（下转第 103 页）