

基于物联网的智能家居系统网络层设计

韩忠华^{1,2}, 吕哲¹, 王金涛³, 董晓婷⁴

(1. 沈阳建筑大学信息与工程学院, 辽宁 沈阳 110168; 2. 中国科学院沈阳自动化研究所数字工厂研究室, 辽宁 沈阳 110016; 3. 中国科学院沈阳自动化研究所工业控制网络与系统研究室, 辽宁 沈阳 110016; 4. 四川建筑职业技术学院电气工程系, 四川 德阳 618000)

摘要 目的 提出一种智能网关与服务器协调合作的网络层解决方案, 解决基于物联网的智能家居系统中各类感知设备、控制器与远程移动终端在网络层的互联互通问题. 方法 通过对系统网络层功能需求的分析, 设计基于 uC/OS-II 操作系统的多任务并发网关, 并结合 Linux 操作系统下的多进程技术与 IO 复用技术, 开发网络层服务器, 自主设计多层兼容可变长数据帧通信协议以保证系统各层次间数据交互的准确性与稳定性. 结果 系统网络层能够实现与感知层实时的数据交互, 解决了多移动终端并发请求、网络层 IO 阻塞、数据交互时序混乱的问题. 结论 系统网络层具备高度的并发与实时性, 可实现感知层中各类测控数据至应用层移动终端设备的双向数据交互, 具有一定的工程应用价值.

关键词 物联网; 网关; 多进程技术; IO 复用技术

中图分类号 TP388

文献标志码 A

The Network Layer Design of Intelligent Home System Based on IOT

HAN Zhonghua^{1,2}, Lü Zhe¹, WANG Jintao³, DONG Xiaoting⁴

(1. School of Information and Control Engineering, Shenyang Jianzhu University, Shenyang, China, 110168; 2. Digital Factory Department, Shenyang Institute of Automation, CAS, Shenyang, China, 110016; 3. Industrial Control Networks and Systems Department, Shenyang Institute of Automation, CAS, Shenyang, China, 110016; 4. Department of Electrical Engineering, Sichuan College of Architectural Technology, Deyang, China, 618000)

Abstract: In order to solve how the each kind of sensing devices and controllers, which belong to the intelligent home system based on the IOT, interconnect with remote mobile terminal on the network layer, a solution that intelligent gateway collaborates with server on the network layer is presented. Through the analysis about the functional requirement on network layer of the system, the intelligent gateway is designed based on the uC/OS-II operating system. The server in network layer is developed combined with the Multi-process technology and IO multiplexing technology based

收稿日期: 2016-05-26

基金项目: 国家自然科学基金青年基金项目(61502474)

作者简介: 韩忠华(1977—), 男, 副教授, 博士, 主要从事自动化系统集成技术、生产运作与管理、智能控制等方面研究.

on the Linux operating system. The accuracy and stability of the data interaction between each layers in the system are insured by the self-designed variable-length data frame multi-layer compatible communication protocol. The network layer can realize the real time data interaction between each layers in the system. The problems about multiple mobile terminals concurrent program, IO block and the temporal chaos in data interaction have been solved. Depending on the methods mentioned above, the network layer has the high degree of concurrency and real-time. The network layer has engineering application value with the function which can realize the data interaction from the TTC data in sensor layer to the mobile terminals in application layer.

Key words: IOT; gateway; multi-process technology; IO multiplexing technology

随着互联网技术的不断发展,移动互联网技术的逐步普及,物联网已成为现今信息技术领域的热门. 其广阔的发展前景受到各行业重视,基于物联网智能家居的研究与开发,已成为当前的热门研究方向,如何将家居中各类设备和环境的信息接入互联网是目前亟待解决并完善的重要问题之一^[1].

参照国内外对物联网结构体系的研究,基于物联网的智能家居系统采用分层体系结构,主要包括感知层、网络层、应用层^[2-4]. 智能家居系统中,网络层作为系统各层次间的核心层次,主要作用是实现由底层传感器至用户手机客户端的数据互通,保证数据交互的可靠性和实时性. 设计并实现造价合理、功能完善的网络层,是解决智能家居系统各层间数据交互问题的有效途径之一,也是国内外相关领域的研究热点.

目前,针对物联网智能家居方面的网关设计开发主要是设计嵌入式网络服务器作为网络层的唯一设备,参照叶高扬^[5]关于物联网网关的设计方案以及 ABDULHALIM M. F. 等^[6]人提出的采用无线方式将传感器采集数据接入互联网的构想,提出了基于 ARM11 或 ARM Cortex - A8 等高性能内核的嵌入式芯片的网关设计方案,这类方案能基本满足物联网系统数据流通、分析、处理的需求,但此类方案的造价成本及开发难度相对较高. 参考李晓丹^[7]设计的基于 STM32 的网关服务器,以及茅胜荣^[8]等人在设计的嵌入式 Web 服务器其处理器性能在大数据

量分析、多客户端请求的情形下,并不能完全满足功能需求.

基于此,针对目前网络层设计无法兼顾性能与造价的问题,笔者提出了一种智能网关与服务器协调合作的网络层解决方案. 在网络层网关实现了实时操作系统下的多任务并发功能. 在远端服务器设计中,结合了多进程、进程通讯、IO 复用等多种技术手段,实现了数据在整个系统中感知层、网络层、应用层的层间流通.

1 系统整体功能及结构

笔者设计的基于物联网的智能家居系统是具有感知层、网络层、应用层的三层系统. 如图1所示,网络层包含智能网关与服务器

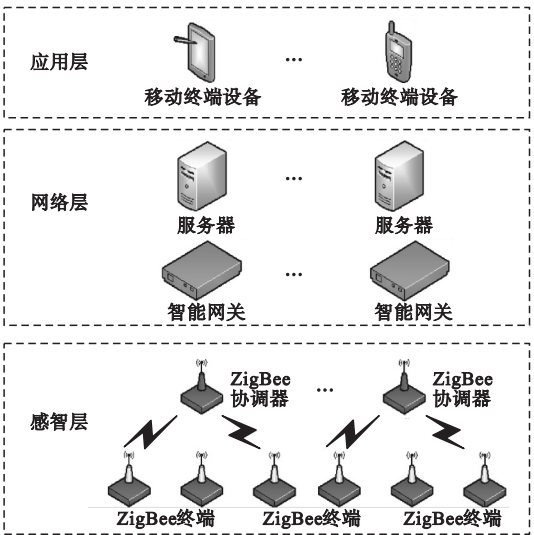


图 1 基于物联网的智能家居系统结构图
Fig. 1 The structure of the intelligent home system based on IOT

两大部分,主要作用是实现系统各层间数据实时快捷的交互,确保感知层信息有效接入互联网。

2 网络层功能及结构概述

网络层包括智能网关和远端服务器,综合考虑开发造价和系统性能需求这两方面因素,设计智能网关以 STM32F103VCT6 芯片作为核心处理器。STM32F103VCT6 是基于 Cortex - M3 内核,72 MHz 主频,256 kB flash,具有丰富外设资源的低功耗嵌入式微处理器。

智能网关的硬件结构图如图 2 所示,智能网关通过串行通信接口与感知层 ZigBee 网络的协调器相连,完成对感知层的数据交互,同时设计 ENC28J60 网络接口模块,通过有线网络的接入方式,实现与网络层服务器建立网络连接并通信的功能。

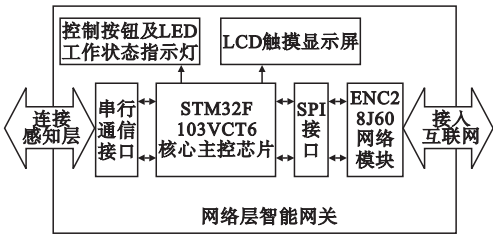


图 2 网关硬件结构图

Fig. 2 The structure of the hardware for gateway

本系统网关与现今国内设计的智能网关作为智能家居系统中的网络服务器这一思路略有差别,在笔者讨论的物联网智能家居系统中,智能网关被设计成具有连接感知层 ZigBee 网络协调器和网络层服务器功能的网络客户端,即网关主要工作是通过串口与感知层 ZigBee 网络协调器在约定的“感知层与网络层数据交互协议”下进行数据交互,并对不同类型的感知层数据进行判断分析。

感知层数据在网关接收到服务器发送的获取或者控制命令时,网关将由感知层获取的数据打包成网络层间的数据交互协议形式(网络层与应用层数据交互协议),发送至服

务器,交由其处理并继续发送至应用层。若通过串口从感知层接收到异常情况下的报警数据,网关直接将其发送给服务器,同时在约定的与感知层通信的协议下向感知层发送指令,及时处理异常信息。

基于上述内容,智能网关在网络层中作为网络层远端服务器的客户端的角色,并能够实现对感知层各传感器及设备状态数据的处理分析功能,同时可以将此类数据发送至网络层。与应用层多客户端通讯连接任务交由网络层服务器完成。

在网络层中,设计远端服务器。服务器在系统中的基本功能是沟通并协调网络层网关与应用层手机客户端间的数据交互,服务器中所完成数据交互功能也是建立在约定的网络层数据交互协议基础之上的。服务器在完成基本功能的同时,还具备对感知层上传数据存储和集中分析的功能。

考虑到服务器的功能扩展和服务器程序的可移植性,再设计过程中对服务器程序采用模块化设计思路,其中主要包括网络连接模块、进程通讯模块、协议解析模块、数据存储模块、数据分析模块。为了实现服务器程序在不同硬件平台上的兼容性和可移植性,在 Linux 操作系统下采用 C 语言编写服务器程序。并将其部署在一台具有 Intel Core i5 处理器、Ubuntu 操作系统平台的 PC 机上,移植到基于 ARM11 处理器,运行 ucLinux 操作系统的嵌入式硬件平台上进行测试。

3 网络层功能需求分析

网络层在整个系统中起到了联通整个系统的重要作用,实现各层的数据交互和协议转换,同时网络层一方面要求对感知层的上传数据具备较强的运算和处理能力,另一方面还需要向应用层提供相应的数据请求服务。针对网络层在物联网智能家居系统中的作用,笔者首先对网络层进行功能需求分析。

第一,设计网络层的过程中,应充分考虑

智能网关在处理感知层上传的数据和应用层下达的命令过程中的实时性要求。

第二,针对网络层需要处理大量网络连接请求的情况,设计的远端服务器应具有并发处理功能,具备处理多网关、多手机客户端并发连接请求的功能。

第三,在实现网络层并发功能的基础上,同时需要进一步研究网络层远端服务器进程之间的相互配合与协调的问题,应避免在服务器在多个进程并发过程中出现锁死、资源抢占、阻塞等待等现象的发生。

根据上述几点功能要求,在设计过程中采用如 I/O 复用、进程管道、信号量等多种技术手段,可以解决以上需求问题。

4 网关与服务器主要设计内容

4.1 基于 $\mu\text{C}/\text{OS}-\text{II}$ 嵌入式操作系统的多任务网络客户端设计

为满足网络层智能网关对实时性的要求,实现网关在工作过程中收发数据无阻塞,报警处理快速及时的功能。在设计过程中移植 $\mu\text{C}/\text{OS}-\text{II}$ 嵌入式操作系统至网关,并在其基础上对网关进行编程,满足各项功能需求。

4.1.1 操作系统与协议栈

$\mu\text{C}/\text{OS}-\text{II}$ 嵌入式操作系统是一个可移植、可固化的、可裁剪的、占先式多任务实时内核,它适用于多种微处理器、微控制器和数字处理芯片。其关于 CPU 硬件的相关代码通过汇编语言实现,其余部分代码由 C 语言编写。 $\mu\text{C}/\text{OS}-\text{II}$ 具有执行效率高、占用空间小、实时性能优良和可扩展性强等特点,最小内核可编译至 2 kB^[9-10]。

为了满足网关在网络层基本通讯功能,还需在网关之上移植 LWIP 轻量级 TCP/IP 协议栈,LWIP 协议栈在 $\mu\text{C}/\text{OS}-\text{II}$ 操作系统的支持下,可通过 RAW(原始套接字)、NETCONN 和 SOCKET 三种通讯编程接口进行网络应用程序的开发,本设计选择使

NETCONN 通讯编程接口,其与 RAW 和 SOCKET 二者相比,具有开发应用程序的开发难度低和执行效率高的特点^[11]。

综上所述,网络层智能网关是在 $\mu\text{C}/\text{OS}-\text{II}$ 嵌入式操作系统上开发的多任务网络连接器,并利用 LWIP 协议栈支持下的 NETCONN 接口进行网络通信功能设计,以实现系统各层间的数据的无阻塞收发及实时处理。

4.1.2 智能网关的多任务开发

智能网关在整个系统中的功能主要分为三大模块,即与远端服务器的网络通信模块,与感知层 ZigBee 的串口通信模块,处理感知层报警数据模块。

在网关设计过程中遵循该模块化思想,分别对网络通讯模块的数据接收和发送创建两个独立任务,避免收发数据过程中产生阻塞而干扰正常工作,确保数据流通的实时性。处理感知层数据报警,也创建一个独立任务,不受网络连接情况的影响,与其他处理任务并发执行,使报警情况能实时得到处理。与感知层 ZigBee 的串口数据通信,采用中断方式执行。

在网关中移植 $\mu\text{C}/\text{OS}-\text{II}$ 嵌入式操作系统,其支持多任务并发的功能主要是有系统提供的 OSTaskCreate() API 函数实现的。在网关完成个硬件资源初始化和系统初始化后,通过调用 OSTaskCreate() 可创建新的任务。该函数原为^[12]: $\text{INT8U OSTaskCreate}(\text{void}(*\text{task})(\text{void} * \text{p_arg}), \text{void} * \text{p_arg}, \text{OS_STK} * \text{ptos}, \text{INT8U prio})$ 。

其中,task 参数为指向任务代码的指针,p_arg 是传递给任务的参数,ptos 参数为指向任务栈顶参数,prio 是任务优先级。

以网关中的感知层报警数据模块为例,在创建处理报警数据任务时,先通过宏定义的方式确定任务堆栈大小、任务优先级,而后创建任务堆栈,最后声明任务函数,将上述参数填入 OSTaskCreate() 中,完成创建。网关

在实时操作系统支持下的多任务处理功能,可有效保证网关在处理感知层与网络层数据连通过程中的实时性与可靠性.

4.2 基于 Linux 操作系统的多进程技术

笔者描述智能家居系统网络层远端服务器,设计并运行在 Linux 操作系统之下,在智能家居系统中,服务器要实现与网络层智能网关客户端建立网络连接,同时也需要与应用层手机客户端进行网络连接以实现数据交互.要求服务器在对网关客户端、手机客户端进行数据收发的过程中,既要实现数据相互关联,又要避免网关客户端与手机客户端数据的相互影响与干扰.为此,笔者提出利用 Linux 多进程技术,设计服务器具有两个相互独立进程,其中一个进程用于与网关客户端建立连接,处理网关客户端的连接请求,另一个进程用于处理手机客户端连接请求,两个进程具有同样的 IP 地址,不同的端口号.

在 Linux 操作系统中,可在应用程序中调用 `fork()` 函数实现进程的创建,在一个应用进程中调用一次 `fork()`,产生一个新的进程,由 `fork()` 创建的进程被称为子进程,调用 `fork()` 函数的进程被称为父进程.

`fork()` 函数原形为 `pid_t fork(void)`,该函数被调用一次,返回两次.其中,子进程调用 `fork()` 函数,返回值为 0,父进程中 `fork()` 的返回值为新建子进程的进程 ID.

在服务器多进程设计过程中,服务器程序在定义好相关全局变量后,通过调用 `fork()` 函数创建子进程,子进程中运行处理网关连接请求的程序,父进程中运行处理手机客户端连接请求的程序.

4.3 Linux 操作系统下的进程间通信技术

在服务器完成双进程分别处理智能网关和手机客户端连接请求的基本功能设计之后.为了使感知层数据能及时反馈给应用层,应用层的命令数据能顺畅下达给感知层,服务器处理手机客户端以及网关的两个进程,应及时将各自从客户端收到的数据以进程通

讯的方式交给对方进程^[13],以便对方进程实时将所需数据发送各自连接的客户端.

远端服务器中两进程间通信的示意图如图 3 所示.

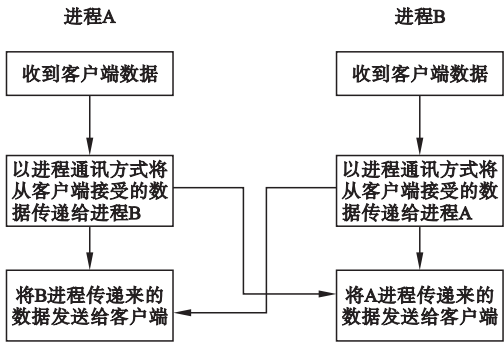


图 3 远端服务器中两进程间通信的示意图

Fig. 3 Diagram of the communicate between two process in server

分析图 3 所示的进程间通讯情形,远端服务器要满足进程间数据准确通讯,应充分考虑进程间通讯的方式及时序问题.笔者采用进程管道技术作为进程间通讯的方式,引入信号量同步思想,解决进程通讯时序配合问题.

4.3.1 进程管道

服务器进程间通讯采用最基本的 pipe 管道技术,实用而方便,可兼容各发行版本的 Linux 操作系统^[14]. pipe 管道技术是半双工(数据只能在一个方向上流通),pipe 管道只能在具有公共祖先的两个进程间使用.通常,一个管道由一个进程创建,在进程调用 `fork()` 后,创建的进程管道即可生效使用. pipe 管道通过调用 `pipe()` 函数实现,其函数原型为 `int pipe(int fd[2])`.

`pipe()` 函数经由参数 `fd` 返回两个文件描述符, `fd[0]` 为读取管道数据而打开, `fd[1]` 为向管道写数据而打开,管道的流向取决于我们对 `fd[]` 描述符的操作.对于父进程到子进程的管道,父进程关闭管道的读端(`fd[0]`),子进程关闭管道的写端(`fd[1]`),反之亦然.

对于文中描述的服务器程序,在父子进

程中分别创建一个 pipe 管道, pipe_par 和 pipe_chil, 分别对应父子进程的管道. 在子进程中关闭父进程管道 (pipe_par) 的写端, 父进程中关闭子进程管道的写端 (pipe_chil). 两进程中有接收到客户端数据后, 分别向自己进程的管道中写入接收数据, 这样, 另一个进程只需要在适当时刻读取对方进程管道, 获得数据, 这样就实现了进程间通讯.

这种在两个进程分别创建管道的方式, 有效弥补了 pipe 管道半双工通信的不足. 图 4 是服务器中进程管道数据流通示意图.

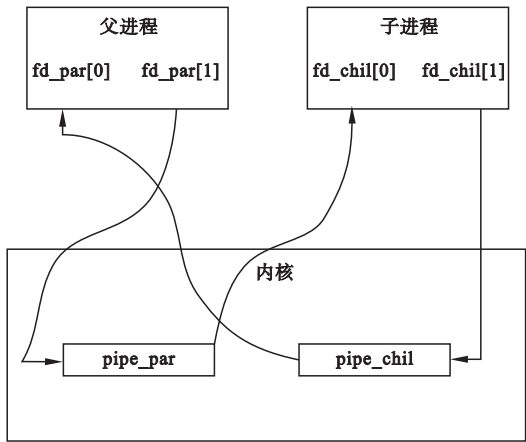


图 4 管道数据流通示意图

Fig. 4 Diagram of the data flow in pipe

4. 3. 2 信号量同步

解决了在 pipe 管道下两进程进行通讯的问题, 仍无法确保在进程通讯过程中数据传递的正确性与传递时序的合理性. 分析可知, 两进程在并发执行过程中, 若其中一个进程开始读取对方进程的 pipe 管道, 而对方进程并没有完成对 pipe 管道的写入, 这样就会造成进程未读到数据, 导致进程通讯错误.

在该服务器程序中, 可以将进程通讯流程抽象成为生产者与消费者模型. 服务器两个进程同时作为对方的生产者, 又作为彼此产品的消费者, 一个进程完成对自身进程通讯管道的数据写入后, 即完成了产品生产. 另一个进程此时可以消费产品, 即从完成写入的管道中读取数据^[15-16]. 在实际设计与编程过程中, 采用信号量同步方式, 对各进程的

pipe 管道进行资源保护, 当数据写入管道后, 该进程释放一个信号量 A, 另一个进程只有获取到信号量 A 时, 才能对对方进程完成写入的 pipe 管道进行数据读取.

图 5 表现了进程间使用信号量协调通讯的具体步骤, 各进程首先创建一个信号量, 在 Linux 系统中, 使用 semget() 函数创建一个信号量, 并通过 semctl() 函数设置信号量的初始值为 0, 当完成对进程所属的管道的数据写入后, 即可释放之前创建的信号量, 即对信号量的值进行 +1 操作, 当释放信号量后, 另一个进程既可以获取到该信号量.

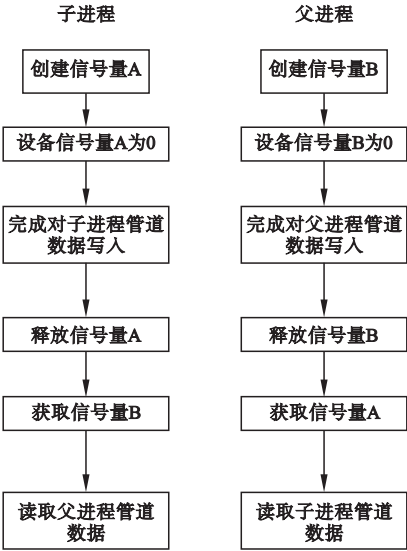


图 5 进程间使用信号量协调通讯过程

Fig. 5 Diagram of using the semaphore to coordinated the communication between two process

对信号量的释放及获取, 可通过操作结构体 structsem_buf 中的 sem_op 成员变量实现, 释放信号量时, sem_op = 1, 获取信号量时, sem_op = -1, 最后还应将结构体地址传递给函数 semop(). 在整个程序中, 当信号量的值为 0 时, 无法进行 sem_op = -1 (即信号量获取) 的操作, 依据上述流程, 就实现了服务器进程间通讯的信号量同步机制, 建立了合理完善的“生产者 - 消费者”模型, 确保当进程管道完成数据写入后才能被其他进程读

取数据。

4.4 通过 Epoll 技术实现服务器 I/O 复用

4.4.1 问题描述和解决方案的提出

在前文讨论的几种技术的综合运用下,远端服务器程序具备了基本的功能,即实现从网络层网关客户端至应用层手机客户端的数据流通,在实际过程中,服务器先从客户端收到数据后才能对自身的进程管道写入数据,即接收从客户端发来的数据存在阻塞,在远端服务器中,阻塞部分的示意如图 6 所示。

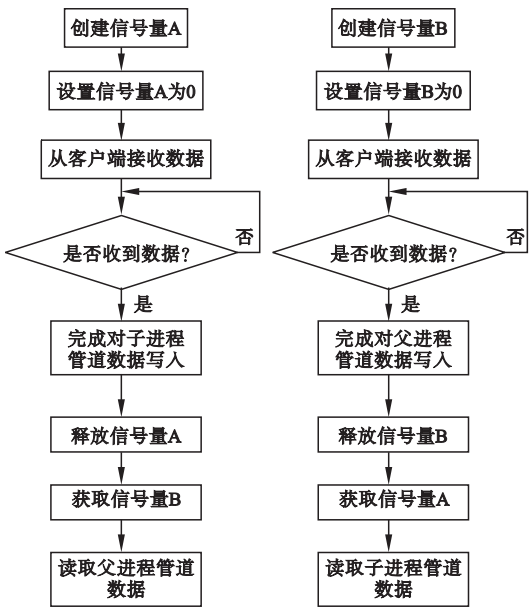


图 6 服务器在进程阻塞情况下的运行过程

Fig. 6 Diagram of the server operational process in case of the process blocking

在图 6 中可以分析出,当服务器程序与客户端建立连接后,服务器程序一直阻塞在接收数据的过程中,只有当一个接收到数据后,该进程才会执行后续操作.实际调试过程中,在这种阻塞状态下,只有当处理网关连接进程和处理手机客户端连接进程都接收到数据时,两进程间才会发生一次数据通信,并且这种工作情形并不支持多客户端连接,这并不符合网络层服务器的功能需求。

这种进程阻塞的状态在 Linux 环境下被称为 IO 阻塞,通常解决这种现象的办法有多进程处理阻塞、采用非阻塞 IO、使用异步 IO

方式、实现 IO 复用技术这几种方式。

综合考虑到处理效率、对处理器资源的消耗、实现的难度以及远端服务器的功能需求等因素,笔者提出一种基于 Epoll 实现 IO 复用(也称 IO 多路转接)的技术.解决服务器由于 IO 阻塞带来的问题的不便,并且能同时满足对多客户端连接进行高效处理的功能需求。

4.4.2 基于 Epoll 模型的 IO 复用

Epoll 模型是基于 Linux2.6 内核下为处理大批量句柄而作了改进的 poll 模型,这种模型能显著减少程序在大量并发连接中只有少量活跃的情况下的系统 CPU 利用率^[17].使用 Epoll 模型实现 IO 复用,用户将关心的网络套接字、读写文件描述符告知内核,当进程要进行 IO 操作时,向 Epoll 查询读写的状态,需要注意的是,Epoll 并不负责具体的 IO 操作,他所做的就是告诉进程当前是否可读或者可写,由用户控制读写操作,此时进程会有空闲处理其他操作^[17-18]。

在使用 Epoll 模型实现 IO 复用技术时,需要注意其工作方式分为 epoll_LT 和 epoll_ET. epoll_LT,水平触发模式,支持非阻塞和阻塞两种操作模式,此模式下,每个 epoll_wait()函数都将返回到事件提醒用户程序,只要关注的描述符可读;epoll_ET,边缘触发模式,只支持非阻塞的操作模式,它只会提示一次,直到下次再有数据流入之前都不会再提示了。

在网络层远端服务器进行 Epoll 模型编程时,主要用到三个函数。

Epoll 句柄创建函数. intepoll_creat(int size),此函数用来创建一个 epoll 句柄, size 标示监听内核的数量。

Epoll 事件注册函数, intepoll_ctl(intepfd, intop, intfd, struct epoll_event * event),这个函数的功能是完成事件注册, epfd 表示创建 epoll 句柄的返回值, op 表示下一步的行动。

fd 是需要监听的文件描述符,本服务器中将 socket 返回的套接字填入,实现网络连接过程中的 IO 复用. 第四个参数结构体是通知内核需要监听的事件.

等待事件函数, `intepoll_wait (intepfd, structepoll_event * events, intmaxevents, int timeout)`, 此函数用来等待事件触发, 其中, `maxevents` 表示事件集合的大小, `timeout` 是指超时时长. 这个参数的返回值是需要解决的事件的数量.

网络层服务器程序, 在创建了 `Epoll` 句柄并进入网络连接程序后, 每一次循环都调用一次 `epoll_wait()` 函数来查询所有网络接口的读写情况, 以实现网络数据收发、多客户端连接的非阻塞状态.

通过针对网络层的多进程通讯以及 IO 复用的设计与实现, 网络层远端服务器满足了在智能家居系统中的性能要求.

4.5 通讯协议

在实现高性能的网络层智能网关以及远端服务器功能设计后, 整个物联网智能家居系统在网络层的协调和处理下, 可实现由感知层到应用层的数据流通. 为了保证数据流通的正确性, 避免系统多层次数据交互过程的相互影响, 排除因系统在复杂环境下产生错误数据对整个系统产生干扰^[19-20], 在对网

络层设计过程中, 约定一套各层间数据交互的简单协议.

4.5.1 感知层与网络层数据交互协议

此部分协议是感知层 ZigBee 节点与网络层智能网关, 经由串口连接进行数据交互所遵循的协议. 协议内容包括帧头、长度、地址、类型、命令或数据、高低结束位.

帧头用于对数据合法性进行判断, 数据交互过程中首先进行对帧头的判断, 若不符合协议, 则交互双方直接判定为非法数据, 不做处理, 直接跳出, 准备下一次数据交互. 此部分协议中, 帧头都规定为 `0xa5`.

长度表示此次数据交互过程中的数据长度, 交互方应依据协议判断数据长度是否符合协议规定, 不符合协议规范长度的数据都判定为非法数据.

地址和类型, 分别明确表示了感知层的哪一个节点和对感知层的相应操作.

命令表示对感知层各节点下达的命令, 数据为感知层依据命令或产生报警后反馈至网络层的传感器采集数据或设备状态数据.

结束位包括高低两位, 分别为 `0x0d`, `0x0a`, 用于标记数据结束.

表 1 为网络层下达给感知层的命令的协议格式, 实际应用中可根据这种形式灵活扩充或裁剪.

表 1 网络层下达命令至感知层协议格式
Table 1 The data agreement from network layer to sensor layer

命令	帧头	长度	地址	类型	命令	结束位高	结束位低
打开 LED	A5	07	03	30	01	0D	0A
关闭 LED	A5	07	03	30	02	0D	0A
采集温度	A5	07	02	20	03	0D	0A
采集湿度	A5	07	02	21	03	0D	0A
采集光照	A5	0B	01	22	03	0D	0A

4.5.2 网络层与应用层数据交互协议

网络层与应用层数据交互协议内容包括帧头、长度、地址、类型、命令或数据(见表 2), 与感知层与网络层数据交互协议相比,

其内容上只是帧头变为 `0x5a`, 取消数据结尾标志, 数据流通方向上是在网络层与应用层之间. 该部分协议的最终目标也是经过网络层网关将此协议解析后再打包为与感知层交

互的协议,实现与感知层 ZigBee 节点通讯.

表 2 感知层反馈给网络层数据的协议格式

Table 2 The data agreement from sensor layer to network layer

数据名称	帧头	长度	地址	类型	数据	结束位高	结束位低
LED 点亮	A5	07	03	50	01	0D	0A
LED 熄灭	A5	07	03	50	02	0D	0A
当前温度	A5	07	02	01	DATA	0D	0A
当前湿度	A5	07	02	02	DATA	0D	0A
当前光照	A5	0B	01	03	DATA	0D	0A
温度高报警	A5	07	02	12	01	0D	0A
温度高解除	A5	07	02	12	02	0D	0A

5 结 语

笔者通过分析物联网智能家居系统的用户需求,提出了一种智能网关与远端服务器协调合作的解决方案.其中网络层智能网关低成本,功能简单,可实现入户安装,网络层服务器可实现大数据量分析运算,满足多客户端请求,实现对各不同用户网关的集中管理操作.在网络层支持多客户端连接的基础上,提出了结合进程通讯、IO 复用等多种技术的解决方案,使网络层具备高度的并发与实时性,能合理有效地处理并协调多层间的数据流通,大大提升了系统性能.同时预留了数据处理、在线分析、数据加密等功能接口,通过进一步的网络层功能完善,该网络层设计方案及相关技术在物联网和无线通信领域都具有广泛的应用前景.

参考文献

[1] 宁焕生,徐群玉.全球物联网发展及中国物联网建设若干思考[J].电子学报,2010,38(11):2590-2599.
(NING Huansheng, XU Qunyu. Research on global internet of things' developments and it's construction in China[J]. Actaelectronicasinica,2010,38(11):2590-2599.)

[2] 卢捍华,邵媛媛,孙洪波,等.物联网发展的标准问题[J].电信科学,2014,30(3):1-4.
(LU Hanhua, SHAO Yuanyuan, SUN Hongbo

et al. Study on standardization aspects of IOT [J]. Telecommunications science, 2014, 30 (3): 1 - 4.)

[3] 钱志鸿,王义君.物联网技术与应用研究[J].电子学报,2012,40(5):1023-1029.
(QIAN Zhihong, WANG Yijun. IoT technology and application [J]. Actaelectronicasinica, 2012, 40 (5): 1023 - 1029.)

[4] DATTA S K, BONNET C, NIKAEIN N. An IOT gateway centric architecture to provide novel M2M services [J]. IEEE world forum on internet of things, 2014, 00: 514 - 519.

[5] 叶高扬,毕冉.基于物联网的智能家居系统设计与实现[J].计算机应用,2014(增刊1):318-319.
(YE Gaoyang, BI Ran. Design and implementation of smart home system based on internet of things [J]. Journal of computer applications, 2014 (S1): 318 - 319.)

[6] ABDULHALIM M F, ATTEA B A. Multi-layer genetic algorithm for maximum disjoint reliable set covers problem in wireless sensor networks [J]. Wireless personal communications, 2014, 80 (1): 203 - 227.

[7] 李晓丹.基于STM32的物联网嵌入式网关的设计[J].计算机工程与应用,2015(4):61-65.
(LI Xiaodan. Design of IOT embedded gateway based on STM32 [J]. Computer engineering and applications, 2015 (4): 61 - 65.)

[8] 茅胜荣,肖家文,乔东海.用W5500构成的低成本嵌入式Web服务器[J].单片机与嵌入式系统应用,2016(4):49-51+59.

- (MAO Shengrong, XIAO Jiawen, QIAO Donghai. Low-cost embedded web server based on W5500[J]. Microcontrollers & embedded systems, 2016(4): 49 – 51 + 59.)
- [9] ZHOU P. Design of the cycle measurement system based on uC/OS-II_MCU[J]. Electronics world, 2013.
- [10] CHEN W S, WANG J P. Design of RFID card reading system based on LWIP and free RTOS [J]. Applied mechanics & materials, 2015, 734 (3): 916 – 920.
- [11] Lü F, CUI H M, WANG L, et al. Dynamic I/O-aware scheduling for batch-mode applications on chip multiprocessor systems of cluster platforms[J]. Journal of computer science & technology, 2014, 29(1): 21 – 37.
- [12] 沈婷婷, 金键, 毛伟. 多进程 DNS 服务器中通信方法的设计与实现[J]. 微电子学与计算机, 2011, 28(4): 76 – 80.
(SHEN Tingting, JIN Jian, MAO Wei. Design and implementation of communication method for multi-process DNS server[J]. Microelectronics & computer, 2011, 28(4): 76 – 80.)
- [13] 王崇文, 赵翼. 基于 Socket 的 IPC 消息平台设计[J]. 小型微型计算机系统, 2011, 32 (4): 588 – 593.
(WANG Chongwen, ZHAO Yi. Design of the socket-based IPC messaging platform[J]. Journal of Chinese computer systems, 2011, 32 (4): 588 – 593.)
- [14] 张南平, 徐静. 基于进程池的 Linux 并发服务器的研究[J]. 计算机与数字工程, 2009, 37 (1): 159 – 161.
(ZHANG Nanping, XU Jing. Research on concurrent server in Linux process-pool[J]. Computer and digital engineering, 2009, 37 (1): 159 – 161.)
- [15] 苟长庆, 杨乾明, 伍楠, 等. 共享存储可重构计算机软硬件通信的优化实现[J]. 计算机研究与发展, 2013, 50(8): 1637 – 1646.
(XUN Changqing, YANG Qianming, WU Nan, et al. Optimized software-hardware communications for shared memory reconfigurable computer[J]. Journal of computer research and development, 2013, 50(8): 1637 – 1646.)
- [16] 周海峰, 李俊, 刘嘉, 等. 基于 Linux 的外部 AD 驱动程序设计[J]. 仪表技术与传感器, 2015(5): 53 – 55.
(ZHOU Haifeng, LI Jun, LIU Jia, et al. Design of external AD driver based on Linux[J]. Instrument technique and sensor, 2015(5): 53 – 55.)
- [17] 张颖楠, 顾乃杰, 彭建章, 等. 一种内核级多进程负载均衡会话保持方法[J]. 计算机工程, 2014(3): 76 – 81.
(ZHANG Yingnan, GU Naijie, PENG Jianzhang, et al. A kernel level session-persistence method for multi-process load balancing [J]. Computer engineering, 2014 (3): 76 – 81.)
- [18] WU X, LONG X, WANG L. FlexPoll: adaptive event polling for network-intensive applications [J]. Frontiers of computer science. 2016, 10 (3): 532 – 542.
- [19] MA Xiaofu, CZAUSKI T, YANG T, et al. Wirelessly sensing medication administration [C]//Proceedings of the Wireless Health 2014 on National Institutes of Health. Bethesda, MD, USA: [s. n.], 2014: 1 – 8.
- [20] MA Xiaofu, FANG Yu, BAI Xingzhen. A balanced energy consumption clustering algorithm for heterogeneous energy wireless sensor networks[C]//IEEE International Conference on Wireless Communications, Networking and Information Security. Beijing: IEEE, 2010: 382 – 386.