Cloud Server Software Design of Street Light Management System

Junxiao Yang [a], Yulin Qi [b], Na Li [c], Yang Li [d]

North China Electric Power University (Baoding), Baoding 071000, China.

[a]1348509718@qq.com, [b]qiyulin589@163.com, [c]2278131173@qq.com, [d]1654626486@qq.com

Keywords: street light, management system, cloud server.

Abstract. By studying the management status and power consumption status of street light system, put forward a management way which can save energy again based on energy-saving lights, and have designed cloud server software that can apply to the street light management system. Cell phone management terminal can command street light terminal. It can realize the remote control of a street light in regulating power, monitor running status of lights in real-time by having fixed IP address and port number of the cloud. This system can realize intelligent control of street light and also can save a lot of energy, so human resources and financial resources would be greatly reduced.

1. Introduction

According to the illumination status of most cities in our country, the main problems in management and control way are: first, The management is not smart enough to find fault and troubleshooting, which is extremely inefficient for the large-scale distribution of street lights; second, control mode is too single, which caused serious waste of electricity.

The cloud server software design of the streetlight management system can solve these problems well. The system can call parameters such as the voltage, current and power of each light directly through phone terminal , according to which we can know the operational status of the street lights in real time. The system also can regulate the brightness of lights according to the actual situation in the evening automatically, which broke the traditional way with the same brightness brightly through all the night, and let the waste of electricity greatly reduced [1].

2. System Design

This system consists of the cell phone terminal, cloud server, centralized controller, and street light terminal. The overall frame diagram of the system is shown in figure.1.

Cell phone management terminal: The phone control is an APP that was installed on the phone, through it’s interactive interface, we can send commands and receive information that about lights running status.

Cloud server: The cloud server mainly implements the forwarding of data and the monitoring of street lights. The server establishes the connection with the cell phone management terminal and the centralized controller through the socket communication, and then transfers the data through the data stream. Through the intermediate bridge of the server, the mobile phone and centralized controller can communicate with each other. Server software was written by the development environment named Eclipse based on the Java language.

Centralized controller: Centralized controllers communicate with the server through the GPRS network, which cannot only receive street light information from a server, but also can transfer the data that was measured by sensor to cell phone.

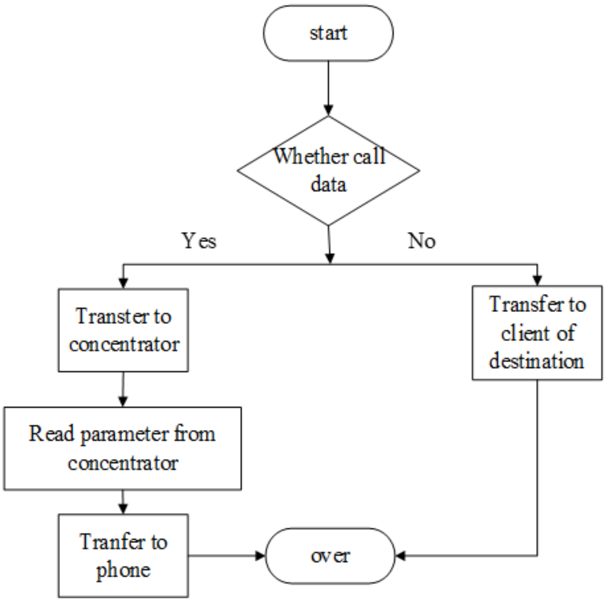

Figure 1. The framework of streetlight parsing protocol

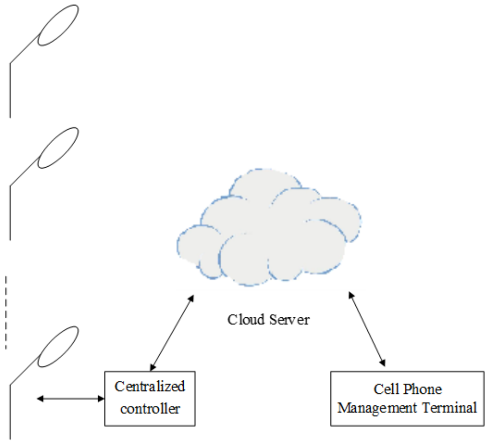

Figure 2. The flowchart of cloud server management system

Street light terminal: The street light terminal is the hardware that controls the street light directly.

3. Cloud Server Software Design and Implementation

3.1 Communication.

As a bridge, the cloud service should first receive the connection from cell phone manager and the centralized controller, then, it can transmit data independently and accurately. Both the cloud server and the two are connected by a socket. Socket is a set of socket specifications built on TCP/UDP[2].

First we need to establish ServerSocket object in cloud server, in this object, we set fixed server IP and port number, and then the server would have been monitoring the server port if there have clients want to connect. At this point, as long as we create the corresponding socket object on the phone side, in which we will write the fixed IP address and port number of the server. Server then would listen to the client and make a connection. When the connection is established, the data can be transmitted through the input and output stream [3].

3.2 The Forwarding Protocol of Cloud Server.

Whether the cloud server can be accurately forward the data to the destination client, the forwarding protocol plays a significant role. In the process of forwarding, it relies on the forwarding protocol to determine the final destination of a frame of data.

(1) The Number of Connections and the Connection ID

In order to distinguish between the cell phone terminal and the centralized controller, we specify that the client of the ID number 0-1023 is the control side of the phone; the client of the ID number 1024-65535 is the centralized controller.

(2) Upward Data

We call the data from the centralized controller to the cloud server and then go to the cell phone at last upward data. The upward data have 99 bytes per frame, and we illustrate dc-data as an example.

The frame data begin with 7E bytes. Bit 1-6 are the single light address. Bit 7-10 are specific dc reference byte, and the cell phone side can calculate the value of the parameters according to these bits.

(3) Downward Data

The downward data is the data that from the phone client to the street light. We set the length of the downward data is 21 bytes, consists of the ID of destination client, length of data and function bytes, the ID of destination client and length of data both are two bytes, function bytes have 17 bytes. Cell phones can realize the different control ways of street lights, including: drive reset, stop triggering, start, low operation, ultra-low light running, remove the dimming, summon an alternating current and direct current. Each function corresponds to a different command bytes, such as stop triggering corresponds to 0EDD. Each light has its own address, with a length of six bytes, we can control the single light.

In programming, we first analyze the first two bytes of the frame data received by the server. That is the ID of the destination client. Then the server determines the function bytes sent by the phone. We divide the functional bytes into two modes. One is a function byte that does not require recalling, such as stop triggering, low light running, etc. Another is the function bytes that need to be recalled, such as the recall of direct current and alternating current. If the bytes are not required to be recalled, we would let server sent to the specified centralized controller directly. If is recalled bytes, then send to the centralized controller of destination and read the corresponding parameters from the centralized controller, transferring the call back forward data to a cell phone. The detailed flow diagram is shown as figure.2.

3.3 The Implementation of the Server Forwards Data.

The server can run quickly with multiple threads. In practical applications, there would be thousands of clients need to connect to server. In Java language multi-thread can solve this problem.

First, creating a management thread for the server to store all the client threads that are connected to the server. We according to the client ID to mark up the corresponding thread, then put the thread into management thread. Server analyses the destination address in downstream data, and then inquires manage threads, if it exists, then server send data to destination client; if does not exist, it returns failed messages to the client [4].

The diagram for the cloud server software is shown in figure.3.

3.4 Database of Cloud Server.

We designed the street light database used by MySQL, there are four tables in the database: administrators table, concentrator table, street light terminal table and log. These four tables can be associated with each other. Let's take the administrator table as an example [5].

We fill the administrator ID and concentrator ID in the administrators, we have registered administrator ID that can operate concentrator, so that when the cloud server detect that there is a client want to establish a connection, we should query the database, if the library has this ID then cloud server allow it to communicate with itself, otherwise, the client must be interrupted and would be notified it has no permission.

4. Conclusion

This software can run well in the streetlight management system. We can input the user name and password to load in management APP. After loading successfully, we can send commands. After testing, the server can forward the message correctly and the light can respond accordingly.

The system can make energy savings by adjusting the lighting power according to the different seasons and traffic conditions in real time. Though the interface of mobile phones, managers can easily view running status of the street light, which greatly improves the management efficiency, and saves a lot of manpower. The implementation of the cloud server in streetlight management system makes streetlight management more efficient and reduces the amount of electricity.

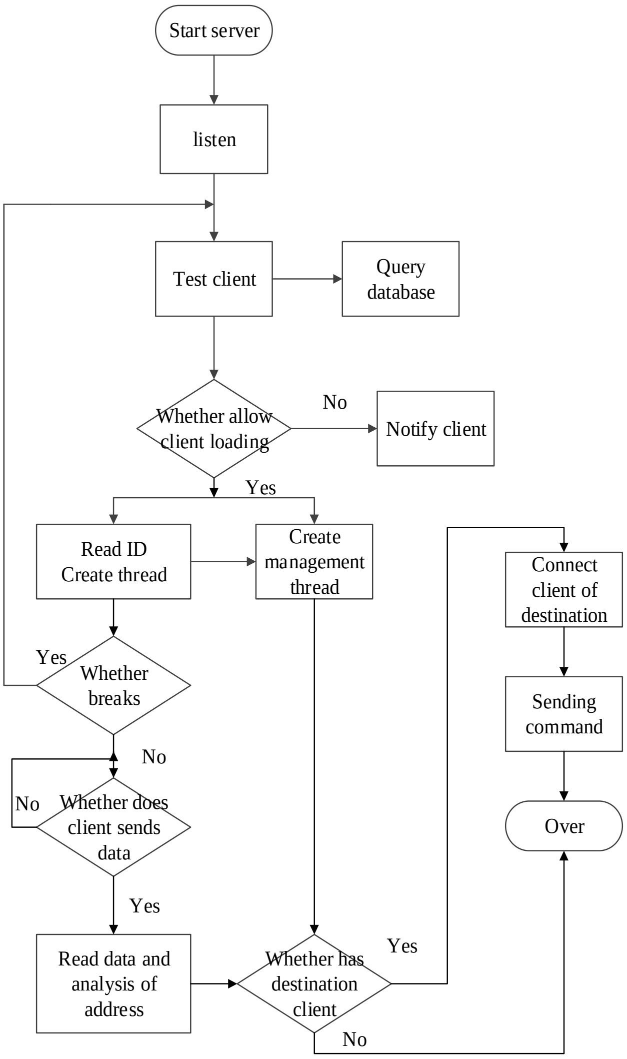

Figure 3. The flow chart of cloud downstream data server software program

References

[1]. Zhang Yan, Zhao Yingjuan, Yang Mei. Design and Implementation of street light control system based on WSN technology. Journal Of Northeast Dianli University, Vol. 31 (2011) No. 01, p. 84-87.

[2]. Kong Hao, Hu Yinbao. Application of Thread in the Communication Based on Socket. Journal of Lanzhou Polytechnic College, Vol. 17 (2010) No. 01, p. 23-27.

[3]. Chen Xiaoyan. Multi-thread socket communication based on Java. Journal of Hubei TV University, Vol. 28 (2008) No. 01, p. 157-158.

[4]. Zhou Huqing, Li Weimin, Zhang Changlong. Java typical modules and projects are in full combat. 1st ed. Beijing: Tsinghua University Press, 2012: p. 473-494.

[5]. Hu Baomei, Zhu Bo, Ning Jiaoxian, Li Jiujie, Zhu Jun, Yan Yonggang. Study of network hospital reservation and inquiry system for PACS based on Java and MySQL technologies. Journal of Sichuan University (Natural Science Edition), Vol. 48 (2011) No. 02, p. 327-332.