A Project/Dissertation Review-1 Report

on

Energy Efficiency in Internet of Things

Submitted in partial fulfillment of
the requirement for the award of the
degree of

BTech. In CSE with AIML

Under The
Supervision of

Mr. Praveen Mishra
Associate Professor

Submitted By

Om Prakash Kumar
19SCSE1180056

Rahul Kumar Chaubey
19SCSE1180053

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
OCTOBER,
2021

Abstract

Internet of Things (IoT) is an emerging technology and energy consumption is one of the important issues. It is crucial because the devices are energy constrained. Battery operated sensors, actuators and everyday objects are connected to the internet. Significant progress has been made in this paradigm. This paper presents the issues and ways to minimize the energy consumption in IoT environment.

The Internet of Things is a new paradigm. It combines technologies such as ubiquitous computing, pervasive computing, internet protocol, sensing technologies and communication technologies etc., the term was coined by Kevin Ashton in 1999. He imagined the world where, Internet is connected to the physical world to enhance comfort, security and control of our lives. Since millions of devices are connected to the internet and the devices are energy constrained, energy is an important factor in IoT. In order to increase the life time of a sensor node, energy must be saved at different levels. This paper presents an overview of energy efficiency in IoT environment, and some ways to energy conservation

List of Tables

List of Figures

Acronyms

IOT	**Internet Of Things**
LEACH	**Low Energy Adaptive Clustering Hierarchy**
PEACH	**Power-efficient and Adaptive Clustering Hierarchy**
PEGASIS	**Power Efficient Gathering in Sensor Information System**
TEEN	**Threshold sensitive Energy Efficient sensor Network Protocol**
EEABR	**Energy Efficient Ant-Based Routing**

CHAPTER-1
Introduction

Internet of Things (IoT) is an emerging technology and energy consumption is one of the important issues. It is crucial because the devices are energy constrained. Battery operated sensors, actuators and everyday objects are connected to the internet. Significant progress has been made in this paradigm. This paper presents the issues and ways to minimize the energy consumption in IoT environment.

The Internet of Things is a new paradigm. It combines technologies such as ubiquitous computing, pervasive computing, internet protocol, sensing technologies and communication technologies etc., the term was coined by Kevin Ashton in 1999. He imagined the world where, Internet is connected to the physical world to enhance comfort, security and control of our lives. Since millions of devices are connected to the internet and the devices are energy constrained, energy is an important factor in IoT. In order to increase the life time of a sensor node, energy must be saved at different levels. This paper presents an overview of energy efficiency in IoT environment, and some ways to energy conservation.

Algimantas proposed the energy efficient SSL protocol which ensured the maximum bandwidth and required level of security with minimum energy consumption. They explained the basic concept of the SSL protocol and proposed adaptive SSL protocol. They implemented the following security objectives in SSL protocol: confidentiality, integrity and availability. In order to achieve the security objective of SSL protocols, proper cryptography techniques were applied

Formulation of Problem

Based on the above literature, the issues related to energy efficiency in IoT can be summed up in the following manner. 3.1 Idle Listening Node in active mode is a major source of energy consumption. It is important to reduce wasted energy resources. It needs not to be in an active state. The awaiting ready to transmit data while not receiving or sending packets is called idle listening. There are different approaches to reduce the overall active time. The sleeping sensor nodes switch back to active mode after a certain time span or after the processing of a wake-up signal.

Collision occurs if nodes receive multiple data packets at the same time. Due to this, the received data is useless. Transmission process has to be repeated while energy is dissipated. Collision increases latency as well. These transactions could consume quite a lot of energy. 3.3 Over Hearing High density sensor nodes lead to interferences with neighbour nodes during data conveyance. This is called over hearing. The nodes within reach have this particular problem. This leads to burn up energy resources owing to receiving and processing useless information. 3.4 Reduction of protocol overhead The protocol header information depletes energy resources. Techniques for the reduction of the protocol overhead are adaptive transmission periods, cross-layering approaches and optimized flooding. 3.5 Traffic Fluctuation Traffic can lead to congestion or high delays. If the network is working on its maximum capacity, congestion raises to extremely high level.

Tool and Technology Used

These are some of the ways to implement energy conservation in IoT based on the study. 6.1 Node activity management There are two parts in node activity namely, sleeping scheduling and on-demand node activity. Sleep scheduling is the way to set the node to a sleeping mode and determine the time to wake up. This saves energy in idle time spans. Here, certain periods are determined in which the individual node is in sleeping mode. On-demand node activity is not scheduled, but the node is by default in an active state with a simple functionality. If a wake-up signal is broadcasted, the neighbouring nodes within the area switch to active mode. After activation, the data transmission takes place. As the start-up signal does not have to be decoded, all surrounding nodes are switched on and for the most unnecessarily. Devices are deployed and clusters are formed. An activity scheduling scheme is also proposed for sensing coverage. This is done in rounds. In each round, a node selects a random timeout and listens to messages from neighbours before it expires. The messages contain decision to be active or not. A small amount of energy is needed to pass this type of messages. By reducing energy consumption at the node level, energy efficiency can be increased. 6.2 Data Aggregation and Transmission Process The cost of transmitting data is higher than the data processing. It is beneficial to aggregate data within clusters. Clusters can reduce the amount of data, as the cluster heads are in-charge of monitoring and processing queries. It eases several of the energy dissipating effects. In this process, data which is coming from different sources is combined into a single data packet. This helps to reduce redundancy and to minimize the number of transmission. Wireless transmission of data uses a large portion of the total energy. Incorporating power control into the transmission process could yield higher energy savings. In short range applications, the transmission power, which is directly related to the data rate, and the circuit power should be carefully balanced to achieve high Energy Efficiency. In order to achieve energy efficiency during transmission, optimization algorithms are used. 6.3 MAC Protocol Energy is very important in handling IoT devices. Better design of MAC protocol is one of the ways to use energy efficiently. The important attribute of MAC Protocol is energy efficiency. MAC protocol is considered as a sub layer of the data link layer. It defines the rules to transmit the frame. If there are many nodes, the MAC protocol coordinates the channel access.

CHAPTER-2
Literature Survey

A good topology structure could improve efficiency of routing protocol. The data link protocol provided a basis for data fusion, target location and other aspects, thus prolong the survival time of the whole network. The authors explained energy collecting technologies like vibration energy, solar energy and wind energy. To sum up, the authors discussed the current situation of energy supply and management techniques of the wireless sensor network. Gang Wu provided an energy efficient approach in both physical layer and deployment aspects. They also proposed a basic principle of energy efficient optimization. There are various energy related algorithms such as multi-level water filling or bi-section algorithms for optimization. Hui Suo discussed the status of key technologies including encryption mechanism, communication security, protecting sensor data and cryptographic algorithms. They reviewed the above mentioned key technologies and adopted hop encryption protection. Finally, they analyzed security characteristics and requirements from four layers including perception layer, network layer, support layer and application layer. Julien Beaudaux proposed a strategy to enable heterogeneous MAC duty-cycle configuration among nodes in the network. To implement the idea, the nodes were divided into disjoint subsets, each of them standing for a given duty-cycle configuration. In the proposed solution, the authors explained routing role and sleep depth. The former made MAC and routing layers cooperate, so that each node was able to know number of nodes above in the routing tree. The latter relied on applicative criteria to separate the nodes into different sleep-depth, represented as disjoint virtual layers. Kyungmin Kim suggested energy efficient and reliable data automatic repeat request scheme to minimize transmission delay and energy consumption at the same time. The scheme consisted three aspects namely duplication retransmission prevention, congestion control and error notification. Mallikarjun Talwar outlined routing techniques and protocols for IoTs. Initially, the author explained the characteristics of routing protocols and some of the key challenges. Finally, the author explained a wide range of routing protocols like RPL, OLSR, AODV and PRoPHET. Marcus et al. proposed a new MAC protocol, called PaderMAC, for wireless sensor networks. PaderMAC principle was implemented using Tinyos and the MAC layer Architecture.

CHAPTER-3
Project Design

Code And Screenshot Of User Module

```
                        hr=i2cDevRead(0xd0,0x02);
                        day=i2cDevRead(0xd0,0x03);
                        date=i2cDevRead(0xd0,0x04);
                        month=i2cDevRead(0xd0,0x05);
                        year=i2cDevRead(0xd0,0x06);

        //---STORE UPDATED VALUE TO RTC_RAM---//
                        i2cDevWrite(0xd0,0x08,sec);
                        i2cDevWrite(0xd0,0x09,min);
                        i2cDevWrite(0xd0,0x0a,hr);
                        i2cDevWrite(0xd0,0x0b,day);
                        i2cDevWrite(0xd0,0x0c,date);
                        i2cDevWrite(0xd0,0x0d,month);
                        i2cDevWrite(0xd0,0x0e,year);
}


void clock_start(void)
{
        if(clock_flag==1)
        {
        //---READ VALUES FROM RTC_RAM(BUFFER)----//
                        sec=i2cDevRead(0xd0,0x08);
                        min=i2cDevRead(0xd0,0x09);
                        hr=i2cDevRead(0xd0,0x0a);
                        day=i2cDevRead(0xd0,0x0b);
                        date=i2cDevRead(0xd0,0x0c);
                        month=i2cDevRead(0xd0,0x0d);
                        year=i2cDevRead(0xd0,0x0e);

        //---UPDATE IT TO RTC Control REGISTER---//
                        i2cDevWrite(0xd0,0x00,sec);
                        i2cDevWrite(0xd0,0x01,min);
                        i2cDevWrite(0xd0,0x02,hr);
                        i2cDevWrite(0xd0,0x03,day);
                        i2cDevWrite(0xd0,0x04,date);
                        i2cDevWrite(0xd0,0x05,month);
```

```
main()
{
        unsigned int light_Status;
//------LDR VARIABLES----------//
        float f;        //---Value Read From LDR
//      char buf[10];//----LDR:Storing String For Sending Through UART


//-----RTC VARIABLES---------//
        unsigned char hr=0x00, min=0x00, sec=0x00,day=0x00,date=0x00,month=0x00,year=0x00;
        i2cDevWrite(0xd0,0x08,sec);
        i2cDevWrite(0xd0,0x09,min);
        i2cDevWrite(0xd0,0x0a,hr);
        i2cDevWrite(0xd0,0x0b,day);
        i2cDevWrite(0xd0,0x0c,date);
        i2cDevWrite(0xd0,0x0d,month);
        i2cDevWrite(0xd0,0x0e,year);
        Init_UART();
        LED1=0;
        while(1)
        {
                f=Read1_ADC_MCP3204(2);
                //TxStr_UART("LDR");
                //sprintf(buf, "%.2f", f);//Converting FLOAT to String and Storing it in "buf"
                //TxStr_UART(buf);
                if(f>4.0)
                {
                        LED1=1;//Active High LED
                        light_Status=fun();
                        if(light_Status>=3500)
                        {
                                TxChar_UART('1');
                                clock_start();
                                RTC_display();
                        }
                        else
                        {
```

Code Of Admin Module

```
struct pubnub *p = pubnub_init( PUBLISH_KEY , SUBSCRIBE_KEY ,&pubnub_sync_callbacks,sync);
json_object *msg ; //Creating Pointer to store *NEW* json_object
msg=json_object_new_object();
//json_object_object_add(msg, "DATA", json_object_new_string(arr));
if(arr[0]=='2')
{
        json_object_object_add(msg,"ENVIRONMENT:",json_object_new_string("Beautiful Day"));
        json_object_object_add(msg,"Light State:",json_object_new_string("OFF"));
}
if(arr[0]=='1')
{
        json_object_object_add(msg,"ENVIRONMENT:",json_object_new_string("Dark Night"));
        json_object_object_add(msg,"Light State:",json_object_new_string("ON"));
}
if(arr[0]=='0')
{
        json_object_object_add(msg,"ENVIRONMENT:",json_object_new_string("Dark Night"));
        json_object_object_add(msg,"Light State:",json_object_new_string("OFF"));
json_object_object_add(msg,"WARNING !!!",json_object_new_string("Light Is Not Working"));
}
time[0]=arr[1];time[1]=arr[2];time[2]=':';time[3]=arr[3];time[4]=arr[4];
time[5]=':';time[6]=arr[5];time[7]=arr[6];time[8]='\0';
        json_object_object_add(msg,"Bulb On Time",json_object_new_string(time));
//      json_object_object_add(msg," ",json_object_new_string("$"));


        pubnub_publish(
                /* struct pubnub */ p,
                /* channel */ "project",
                /* message */ msg,
                /* default timeout */ -1,
                /* callback; sync needs NULL! */ NULL,
                /* callback data */ NULL);

        json_object_put(msg);

        if (pubnub_sync_last_result(sync) != PNR_OK)
```

Architecture Diagram