A System Model for Energy Efficient Green-IoT Network

Sarder Fakhrul Abedin, Md. Golam Rabiul Alam, Rim Haw, Choong Seon Hong
Department of Computer Engineering
Kyung Hee University, South Korea
{saab0015, robi, rhaw, cshong}@khu.ac.kr

***Abstract*— The notion of IoT has ignited umpteen possibilities of different heterogeneous devices to blend within a network. In IoT, especially sensor devices are mostly deployed in an extremely resource constrained environment and thus pose the necessity to extend the capability and life expectancy of these kind of devices in terms of energy consumption. Since, the devices in IoT have limited energy sources they often run on battery with a certain energy capability, the deployment of green communication and system model in IoT has been a core challenging issue. This paper addresses the energy efficiency issues across diverse IoT driven networks by proposing a system model for G-IoT and energy efficient scheme for the IoT devices to extend the life expectancy of the whole IoT network.**

Keywords— Green-IoT; REST; Energy efficiency; Heterogeneous devices; Device virtualization

I. Introduction

The envisioned concept of IoT includes intelligent communication between different communicable or non-communicable heterogeneous objects with less human intervention and thus foster the context awareness characteristics within a framework consisting of smart objects. Those objects should be able to grasp the desired characteristics like Automation, Intelligence, Dynamicity, and Zero-Configuration [9]. Some of the key challenges in IoT related research have been identified by the researchers till now which are Heterogeneity, Scalability, Interoperability, and Security and Privacy [2].

A recent study [8] by Gartner projects that, around 26 Billion devices will be connected to the network by the year of 2020. These devices will produce a lot of electronic waste and will also consume a significant amount of energy in order to execute different tasks. This will eventually pose a challenge in near future to reduce the energy consumption and will also demand for new ways of developing a green communication across the network. The vision has become more sparse and promising since the promotion of green IoT(G-IoT). As a result, reduction in context of energy consumption and the deployment of green IoT, should be considered as the future challenges in IoT along with the current projected challenges in Fig.1 .Besides energy consumption is acute in different heterogeneous IoT devices as it actively relates to cost and availability of the IoT network. Thus energy consumption has become a core issue in future internet and different algorithmic approaches have been initiated for different effective solutions like complementing hardware or different system-based approaches. The most highlighted energy efficient methodologies includes different power-down mechanisms, defining systems with two or multiple states using ALG-P algorithm, scheduling with deadlines or minimizing response time [7].The main contribution of this paper is : to propose a system model which is apt to resolve the challenges of IoT

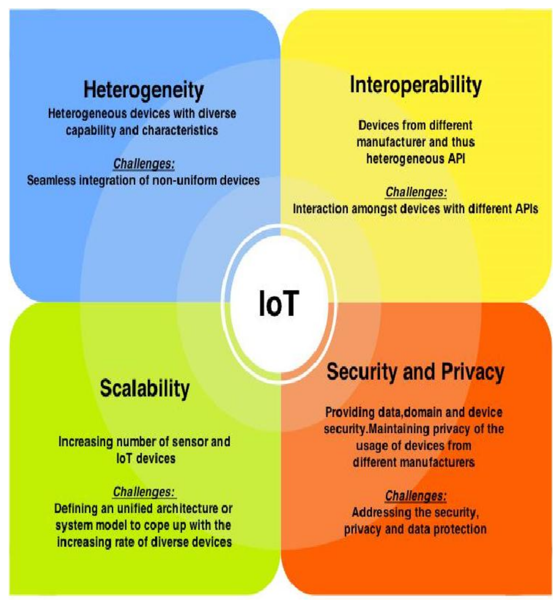

Fig.1. IoT research issues and challanges

related research and an energy efficient mechanism for heterogeneous IoT devices that conforms to the proposed IoT system model to promote G-IoT and increase the longevity of the IoT network.

II. Related Work

In [1] the authors have provided a vivid knowledge on the trending IoT based architecture and European research projects. They have cited the state-of-the-art reference model and architecture for IoT namely IoT-A Architectural Reference Model(ARM) and IoT6 architecture design. The main objective of IoT-A is mainly devoted to provide a generic ARM to discourage the emergence of silo application based IoT architecture rather to encourage to provide a number of ways to derive an IoT architecture. This ARM will let the architects to focus on distinct applications provided with a shared reference model. Side by side the IoT6 architecture design leverages different IPv6 functionality. This architecture includes the assets of IPV6 protocol and also reusing those within IoT6 architecture model.

In [2] a distributed architecture called Distributed Internet-like Architecture for Things (DIAT) has been proposed to address the interoperability and heterogeneity issue in IoT. The proposed layered architecture virtualizes the physical devices and defines cognitive functionalities at different layers. A detailed description has been provided to tackle the heterogeneity, scalability and interoperability issues.

An IoT system is comprised of different heterogeneous appliances and applications. In [3] the authors have proposed a conceptual system skeleton to reflect upon the flexibility and extendibility issues by following the Service Oriented Architecture (SOA) and Model-View-Controller (MVC) design pattern in IoT driven system. A large scale IoT system has been divided into subsystems namely IoT unit to decentralize the capability of the system and thus has proposed distributed brokers to gather data or information from different sensors and actuator within the IoT unit. Besides, a message scheduling algorithm has been proposed in light of MVC design pattern to stabilize the HTTP derived message stability in IoT system to reduce missed deadline of messages.

Energy efficiency in diverse IoT network has been a core challenging issue and nowadays G-IoT is quite a promising field of research. Different energy efficient routing protocols thus has been proposed and in [4] the authors have proposed an IoT system model and energy efficient routing protocol called Pruned Adaptive IoT Routing (PAIR) for heterogeneous IoT. The protocol establishes a routing path by considering the energy reservoir of the nodes in the system to transmit data to sink nodes which is electrically powered. A utility function has been implemented to aggregate the routing cost for each involved nodes for data transmission and represented through a pricing model with a utility function.

Ambient intelligence System (AMI) possess a profound impact on reducing the energy consumption of different sensor devices and actuators in IoT network. The implementation of the AMI technology in heterogeneous IoT environment is quite tedious where sensors and actuators are actually adjacent to each other results hindrance in accessibility and scalability issue of an IoT system. Moreover, the concept of implementing REST methodology in IoT provides easy integration and accessibility to the heterogeneous devices.In [5] the authors proposed a middleware for home automation network and a RESTful interface for heterogeneous IoT based home appliances. Stateless principle of REST guarantees high scalability and reduction in server complexity in their proposed middleware. In [6] the author has also accentuated the importance of using the IETF constrained RESTful environments (CoRE) for resource constrained devices or sensors in a network.

III. System Model

The proposed system model in Fig. 2 is comprised of conventional device hardware which represents the physical hardware for example sensors, home appliances. These devices will be connected with the embedded web server. Embedded webserver hosts RESTful web service to communicate with the cloud server for virtualization of objects. Virtual objects will be transferred to server application for service lookup and will host the virtualized object service executable applications. Application server interface will host the skeleton of the server side code and client will not be able to access the server side code directly rather client will communicate with the application server application through an interface. The admin can directly access the Server application directly without the use of server interface.

Nowadays a lot of open source software packages are used for different devices in the system but some of the open source software might be malicious and the untested material may contain bugs that can severely jeopardize the whole network by breaching the security. In order to find a solution in this

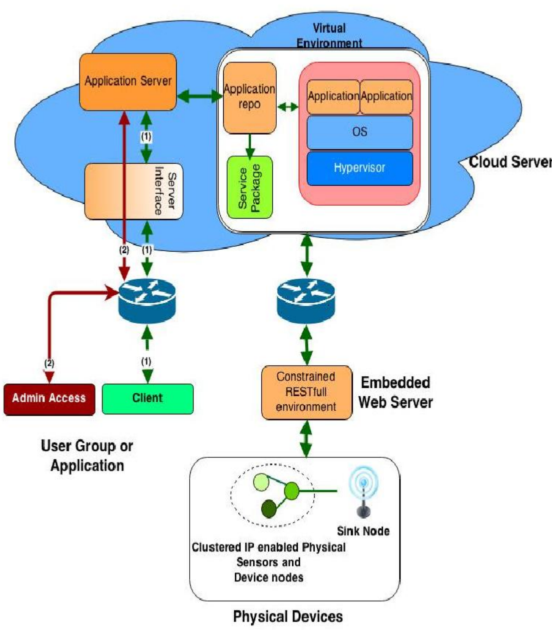

Fig. 2. IoT system model

paper we are going to introduce the concept of a test platform in IoT system model. In our system model, inside cloud server, in case of new service implementation, the server application will test the service package inside the application repository. The application repository will test the service package before deploying it in the environment.

A. Physical Devices

IoT network is comprised of both IP enabled sensor and physical devices. Sensor devices have low memory and data processing capability and on the contrary some of the physical devices have better processing capability than the sensor devices. So, it is feasible to categorize all the devices through their capability. In our scenario we devise a cluster of low powered and less capable sensor devices with IP address which act as relay nodes. On the other hand, devices with better connectivity and capability act as sink nodes that communicate with the embedded devices. Relay nodes produces the data and send it to the sink node so that the sink nodes can redirect the aggregated flow of data to the embedded web server for further processing.

B. Embedded Web Server

Embedded web server will host IETF constrained RESTfull environment. This server will exploit its limited connectivity through Ethernet shield. In the REST architectural style, the data and functionalities are considered as resources and can be accessed directly using Uniform Resource Identifiers (URI).This architectural style is based on simple client/server architecture with some constraints and designed to use a stateless communication protocol. The RESTfull applications are simple, lightweight and fast and ideal choice for resource constrained environment. Primary task of this web server will be to bridge the gap between physical devices and virtual objects. HTTP protocol will be used for communication between physical devices and embedded web server since HTTP can also be used between constrained devices, servers and devices in the constrained environment or even across the Internet. This server environment will receive the device specification, for example product id/MAC id, assigning device IP address and will notify the cloud server for creating a virtual object in the cloud virtual environment with the specifications or the semantic description of the device that includes device features and capabilities as parameter as well.

Another task of the embedded web server is to schedule the physical and sensor devices. In an IoT network some of the devices remain idle most of the time and those devices only react upon request. Scheduling will let this type of devices to become energy efficient and the energy consumption will be directly proportional to device utilization.

C. Cloud Environment

The cloud environment includes the following components:

1) Virtual Environment

Virtual environment provides the support of virtualization of the physical objects in the cloud server. It will host the virtual objects and composite virtual objects as application(s) hosted inside the virtual machine with enhanced processing capabilities.

The virtual environment also has the responsibility to mesh up multiple objects for initiating composite objects. Composite objects are made according to the service request of the user or based on situation in the system. Cyber object are also tested in the application repo and the test result is conveyed to the server for performing the deployment in the virtual environment.

2) Server Application and Interface

Application server has the ability to communicate to the client through the interface of can also be accessed directly by the admin for any necessary modification. The server also keeps track of the services and devices that are currently available within the IoT network and thus works as a registry. Activity in the virtual environment and physical environment will be monitored and will be notified to the authorized user. Server interface protects the server from directly accessing through the client application. Only the admin can directly access and modification of the server.

3) Application Repo

Application repo or repository is responsible for service look up for objects and based on the search result, the deeming perfect service package will be installed to the object. Application repo hosts the test services for different applications. Before deploying an application in real case and within the virtual environment, application repo runs a test of that applications' capability and functionalities and also look for any malicious or faulty codes in the service package according to the service request. If the test fails, the search for more capable service goes on in the service package until the test phase is successfully completed. Application repo sends the test result of new application to the application server and this result is analyzed by the server for sending request to the virtual environment to add the tested application.

D. User group and application

User application is the client side application. The applications are categorized into two subsets based on the authentication policy.

1) Admin access application

Admin can directly access the application server to perform necessary modification and can monitor the overall system performance.To make necessary changes in the network or in the server the admin interface can directly access the server.

2) Client application

The client can send request(s) to the application server thorough application interface and can use the objects that are hosted as application in the virtual machine.

IV. ENERGY EFFICIENT SCHEDULING ALGORITHM

The proposed energy efficient algorithm in Fig. 4 is solely devoted to schedule the duty cycle of different sensors and appliances. The algorithm is framed in the proposed system model so that the model can incorporate with the scheduling algorithm and can also serve its' true purpose. The algorithm has three core stages such as On-duty, Pre-off duty and Off- duty and Fig. 3 illustrates the activity diagram of the scheduling algorithm.

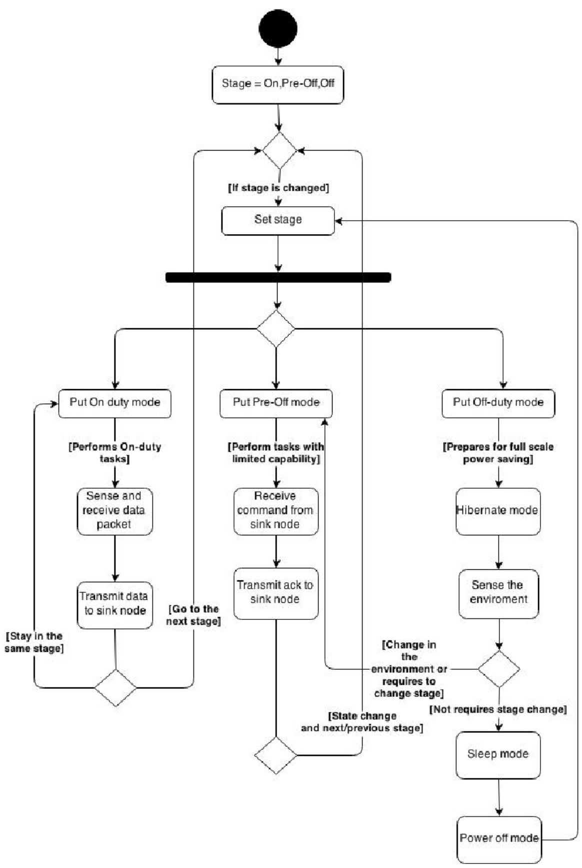

Fig. 3. Energy efficient algorithm activity diagram

1) On duty:

In this state device will be performing with its full-fledged capability. A device within the network will sense, receive and transmit the data accordingly. Apart from that, usually the devices specially the sensor devices have limited processing capability to process data in a resource constrained environment. But in the proposed system model the sensors and devices will only either act as a relay node or sink node based on the devices capability and the whole processing task of the devices and sensors is performed in the virtual environment in the cloud server. This enhancement in the network will reduce the tasks of the devices and will fuel for better data processing opportunity to ensure the QoS of any sort of services.

2. Pre-off duty:

Algorithm 1: Energy efficient scheduling

```
initialize Sleep_timer SL_t,sleep_energy E_sl,sensing_period S_t,
        consumed_energy E_elec,energy_reservoir e_i, maximum_energy E_max ,
        read_value,transmission_value,command_value,temp_value,
        Received data packet D_Rk ,Transmitted data packet D_Tk
while stage ≠ current_stage && e_i ≤ E_max
    Read stage
    Select Case of stages
    Case 1: current_stage ='on _duty'
        while(read_value > -1)
            D_Rk = read_value                //sensing and reading sensor value
        End of loop
        while(transmission_value > -1)
            D_Tk = transmission_value        //transmitting sensor value
        End of loop
    Case 2 : current_stage = 'pre_off'
        Read new_stage
        if new_stage ≠ current_stage
            current_stage = new_stage
            Transmit ACK to sink node
        else
            current_stage= 'off'
    Case 3: current_stage= 'off'
        Hibernate mode with sensing capability
            while S_t > 0
                if temp_value = read_value
                    S_t=0
                    current_stage = 'pre_off'
                else
                    set E_sl                 //enter sleep mode
                    while SL_t > 0
                        power down mode with SL_t =0
                    End of loop
            End of loop
    End case
End of loop
```

Fig. 4. Energy efficient scheduling algorithm

This state will be activated after On-duty state when the device will be idle for sometimes. This state is bidirectional can switch state to both On-duty and Off-duty when it requires to do so. During Pre-off stage, a device can only receive and transmit the necessary commands from the sink node. If the sink node sends any new request, the Pre-off stage will be changed to on duty to perform its tasks. In a nutshell, the devices will be activated but with a limited capability of receiving and transmitting. On the other hand, if the sink node sends command to the device to go to sleep/hibernate/power off mode the Pre-off stage will be changed to the corresponding mode.

3. Off duty

This stage holds three states to save energy in different circumstances and the energy efficiency of the entire cluster or network mainly depends on this state. This state can perform three tasks as follows,

a) Hibernate:

Hibernate state is the state where the device will have small power to only sense the environment before going into more energy efficient state. No transmission or reception of data will be occurring in this state and only sensing can trigger the next energy efficient task or can move the state of the device to previous state. In this state the device will use only the least amount of power and for the devices that have the renewable energy capability will recharge by that time which will extend the device's life expectancy.

b) Sleep

Sleep is a power saving state that has the ability to quickly allow the device to use resume the full-power operation. The device will immediately stop working in this state but will start again when the device is required to resume again.

c) Power off

Power off is the most energy efficient state that will put the device into deep sleep. The consumption of energy is supposed to be 0 at this point since all sort of energy consumed tasks of the device will be stopped by that time. The sink node will directly trigger the device when any necessary task should be performed. The embedded webserver will be responsible for putting the device into right state. If the device is only required to sense the environment and any change in the environment that requires to be received and transmitted to sink node, the device will be in the Pre-off state first and if it requires the full capability of the device the embedded webserver will send command through corresponding sink node to the device to go On duty mode.

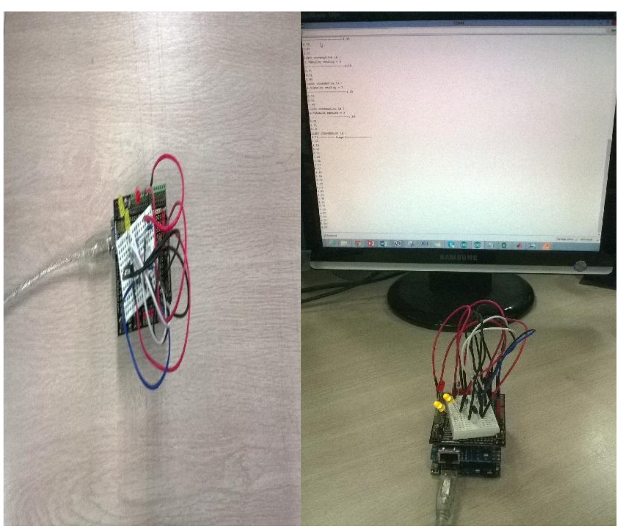

Fig. 5. IoT test bed environment

V. PERFORMANCE EVALUATION

The performance of the proposed algorithm was evaluated using a small real IoT test bed environment in Fig.5 comprised of Arduino Atmega 328P, Ethernet Shield, LM35 temperature sensor, three (3) LED(s), Ambient Light Sensor.

A. Energy Efficient Scheduling

The proposed scheduling algorithm is divided into three stages and we have evaluated all the three stages in the test environment.

Fig. 6 illustrates the power consumptions in three different stages which are On-Duty, Pre-Off Duty, and Off-Duty. The energy consumption in On-duty stage is comparatively higher than the other two stages because the environment runs with full capability and ample power consumption occurs for fully fledged device capability. On the contrary, the power consumption in Pre-off and Off-duty are much lower because of the reduced task load of the devices and sensors in these two stages. Devices and sensors

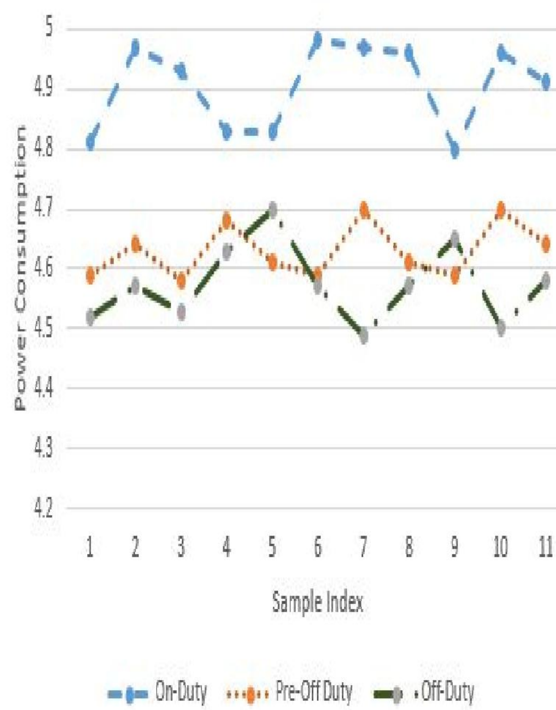

Fig. 6. Power consumption(in Joules) of the IoT environment in three stages

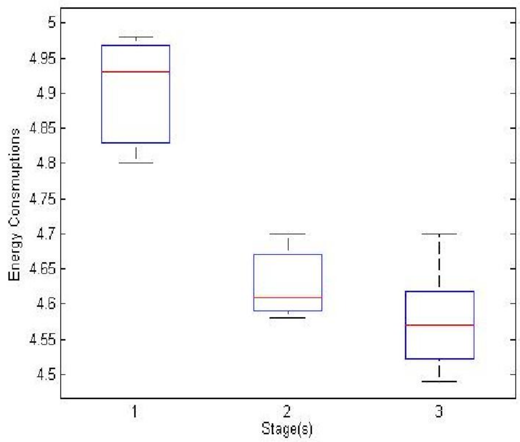

Fig. 7. Energy consumtion range(in Joules)

perform with limited capability and thus the power consumption is trivial than On-duty mode. The power consumption of the environment measured in Joules by taking charge value 1 coulomb.

Fig. 7 depicts more evidence on the efficiency of the proposed energy efficient scheme and illustrates the average power consumption in three different stages where On-duty, Pre-off duty and Off-duty are noted by stage 1, stage 2 and stage 3 respectively. The overall energy consumption of the environment is quite decent in all the three stages because of the proposed IoT system model. The task load and responsibility of the physical sensors and devices are less in physical layer which has positive impact in terms of power consumptions for data processing within the proposed system model. The system model virtualizes the IoT devices and the processing of tasks in the devices which are relocated from the physical layer to application layer.

VI. CONCLUSION

The proposed IoT system model promotes G-IoT by incorporating with the proposed energy efficient scheme which ensures less power consumptions with a well-defined stages of devices within an IoT network. The performance evaluation supports the necessity of the proposed energy efficient scheme and system model to endorse green communication in IoT. In future, we will integrate our proposed system model and scheduling algorithm with Fog computing.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2014R1A2A2A01005900). Dr. CS Hong is the corresponding author.

REFERENCES

[1] Srdjan Krco,Boris Pokric,Francois Carrez,”Designing IoT Architecture(s):A European Perspective”,IEEE World Forum on Internet of Things(WF-IoT),pp. 79-84,March 2014

[2] Chayan Sarkar,Akshay Uttama Nambi S.N. , R. Venkatesha Prasad,Abdur Rahim,”A Scalable Distributed Architecture Towards Unifying IoT Applications”,pp. 508-513,March 2014

[3] Jenq-Shiou Len,Chi-Feng Chen,Kun-Che Hsu, “Improving Heterogeneous SOA-based IoT Message Stability by shortest Time Scheduling”,IEEE Transactions on Services Computing,vol. PP,Issue :99,,17 May,2013

[4] Sharif M.A. Oteafy,Fadi M. Al-Turjman,Hossain S. Hassanein, “Pruned Adaptive Routing in the Heterogeneous Internet of Things”,IEEE Global Communication Conference(GLOBECOM), pp. 214-219, December,2012

[5] Thibaut Le Guilly,Petur Olsen,Anders P. Raven,Jesper Brix Rosenkilde,Arne Skou, “HomePort:Midleware for Heterogeneous Home Automation Networks”, IEEE Pervasive Computing and Communications Workshops(PERCOM Workshops), pp.-627-633 ,March 2013

[6] Zach Shelby, “Embedded Web Service”, IEEE Wireless Communications, vol. 17,issue :6,pp. 52-57,December 2010

[7] Susanne Albers,”Energy-Efficient Algorithms”,Communications of the ACM,vol. 53,no. 5,May 2010

[8] “Gartnet Says the Internet of Things installed Base Will Grow to 26 Billion Units By 2020 ”, available at: www.gartner.com/newsroom/id/2636073

[9] Gyu Myoung Lee, Jeong Yun Kim. “The internet of things: A problem statement”, Information and Communication Technology Convergence (ICTC),2010 International Conference on, pages 517–518. IEEE,November 2010