GreenIoT Architecture for Internet of Things Applications

Yi-Wei Ma[1], Jiann-Liang Chen[2], Yung-Sheng Lee[2] and Hsin-Yi Chang[2]
[1] China Institute of FTZ Supply Chain, Shanghai Maritime University, China
[e-mail: yiweimaa@gmail.com]
[2] Department of Electrical Engineering, National Taiwan University of Science and Technology, Taiwan
[e-mail: lchen@mail.ntust.edu.tw]
*Corresponding author: Yi-Wei Ma

Received May 10, 2015; revised July 6, 2015; revised August 12, 2015; revised October 6, 2015; accepted October 28, 2015; published February 29, 2016

Abstract

A power-saving mechanism for smartphone devices is developed by analyzing the features of data that are received from Internet of Things (IoT) sensors devices to optimize the data processing policies. In the proposed GreenIoT architecture for power-saving in IoT, the power saving and feedback mechanism are implemented in the IoT middleware. When the GreenIoT application in the power-saving IoT architecture is launched, IoT devices collect the sensor data and send them to the middleware. After the scanning module in the IoT middleware has received the data, the data are analyzed by a feature evaluation module and a threshold analysis module. Based on the analytical results, the policy decision module processes the data in the device or in the cloud computing environment. The feedback mechanism then records the power consumed and, based on the history of these records, dynamically adjusts the threshold value to increase accuracy. Two smart living applications, a biomedical application and a smart building application, are proposed. Comparisons of data processed in the cloud computing environment show that the power-saving mechanism with IoT architecture reduces the power consumed by these applications by 24% and 9.2%.

Keywords: Internet of Things, Smart Device, Smart-Living, Power-Saving

1. Introduction

The rapid development of the Internet and the increasing use of information technology have increased quality of life. The rapid development of the Internet of Things (IoT) has led to the widespread use of smart devices and industrial systems. In the IoT environment, sensing devices collect environmental information, which is transmitted via a network to the middleware or cloud for computing and management. As the state of the environment changes, an IoT system dynamically controls the device of environmental equipment which ensures a high quality of service.

Although the performance of smartphone devices is improving, breakthroughs in energy technology have always been hard to achieve. Since its battery power is limited, using a smartphone for intensive computing and frequent communication is difficult. Accordingly, the effective management of the power consumed by smart devices has become an important issue. To reduce the power consumption of smart devices, applications are gradually migrating to the cloud computing environment. With the development of cloud computing technology, this approach practice can substantially improve the overall operational efficiency and reduce the power consumption for smartphone devices. As smart devices rapidly develop and as their computational power rapidly increase, an important question is whether all data must be transferred to the cloud.

In this work, a power-saving mechanism is implemented in a dynamic module in the IoT middleware. When an application is triggered, the bottom layer of the IoT device collects environmental data and sends them to the middleware via the device agent for processing. When an application collected the IoT sensing data and characteristics, the power-saving mechanism used above information to analysis and calculates the assessment grades. When the evaluated grade of power consumption is below a threshold, the system retains the data in the local of smartphone device for computing. If the grade exceeds the threshold, The data are transmitted to the remote cloud for computing. Power consumption is optimized by dynamic switching of computing location in a remote cloud or in a proximal smart phone. This work also proposes a feedback mechanism that improves the analysis accuracy.After the system has been used a certain number of times, accumulated power consumption data can be used to estimate the smart phone behavior for power consumption. Based on the result of the comparison, the system can dynamically adjust the threshold level.

2. Related Work

Such systems reduce power consumption using various strategies. **Table 1** presents the advantages and disadvantages of various strategies that are relevant to this study. Reference [1-3] proposed an IoT healthcare application for a cloud computing environment. Their research results showed that the application improves computing performance but at the cost of increased higher power consumption. Reference [4, 5] proposed an IoT healthcare application using cloud computing and Android OS. Although it reduced power consumption, its disadvantage was lack of flexibility and scalability. Reference [6-9] proposed dynamically adjusts CPU clock frequency, research result show that these study can be obtained advantage

of lower power consumption cost, but disadvantage of lack of flexibility and scalability. The proposed GreenIoT architecture in this work reduces power consumption and enhance flexibility.

Table 1. Literature Review

Literature Method	Advantage	Disadvantage
IoT Healthcare using Cloud Computing [1-3]	Higher performance of computing	Higher power consumption
IoT Healthcare using Cloud Computing and Android OS [4. 5]	Lower power consumption	Lack of flexibility and scalability
Dynamically Adjusts CPU Clock Frequency [6-9]	Lower power consumption cost	Lack of flexibility and scalability
Power-Saving IoT Architecture	Lower power consumption, and higher flexibility and scalability	Affected by device and applications

As wireless communications and mobile computing technology advance, various standards and protocols have been developed to satisfy various needs. The IoT concept was originally proposed in 1999. The overall architecture of an IoT supported smart identification and management system involves RFID, sensor networks, and the Internet. IoT is based on two important concepts, which are Internet-based development and object exchange information [10, 11].

The IoT and sensor networks have been developed and integrated using current internet technology, including software, networking, artificial intelligence, and automatic control. The integration involved is cross-disciplinary. IoT applies to all objects that can be accessed by the Internet and all objects are connected in series to each other. The technical architecture of IoT, which consists of four layers the application layer, the network layer, the sensor layer, and the public technology layer [12-14].

As the use of mobile multimedia services rapidly increases, the power consumption of handheld devices has become an important issue. Android uses a power management mechanism to reduce the power consumed by the Linux Kernel and to extend its standby time. When an application is started, the Application Framework connects to a Library that enables JAVA-based applications to use JNI technology for calling the C/C++ power management function and other functions. The Android application server mechanism uses the Android IBinder process and the watchdog timer parameter to monitor system power. This mechanism reduces unnecessary energy consumption [15-20].

3. Proposed GreenIoT Architecture

3.1 System Overview

In the proposed GreenIoT architecture for power-saving in IoT (**Fig. 1**), the information storage layer for IoT devices includes a monitoring module that can convert scattered bottom-layer IoT device information into a unified data format and apply an information monitoring and management mechanism during conversion.

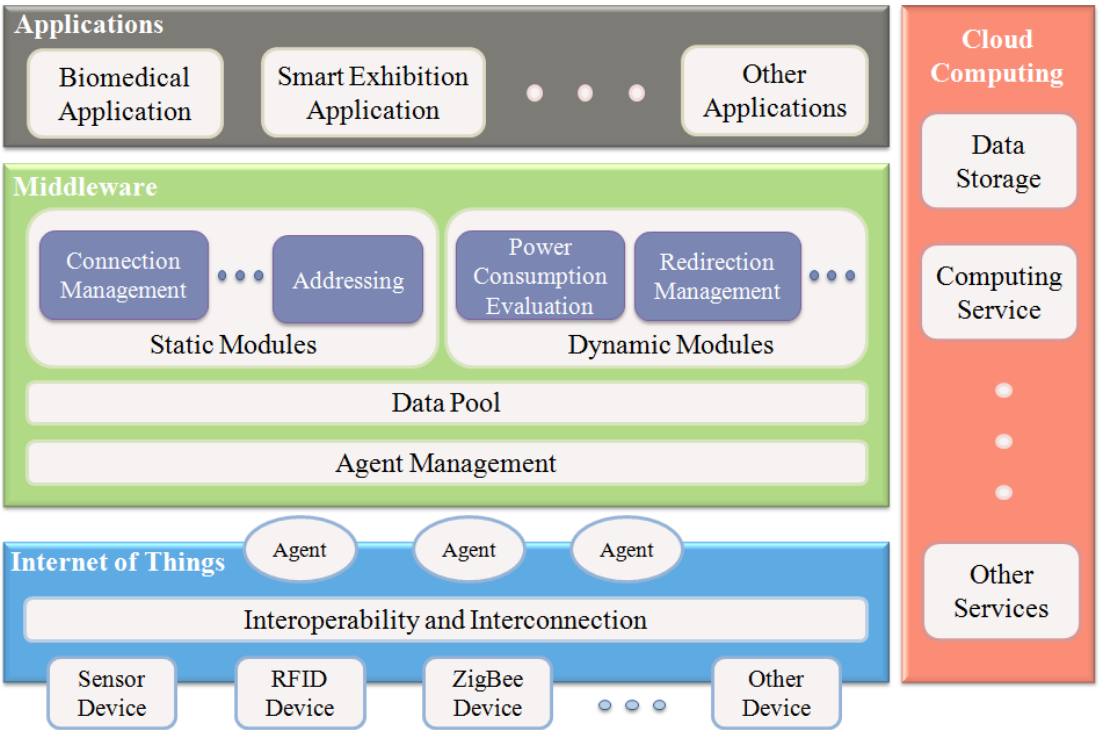

Fig. 1. GreenIoT Architecture

The power-saving mechanism of the proposed IoT architecture includes the following four layers: the IoT device layer, the IoT middleware layer, the application layer and the cloud computing layer. Each layer is described as follows.

3.1.1 IoT Device Layer

The IoT device layer is a bottom layer that transmits the data that are collected from IoT devices. The problem, however, is that IoT devices are designed without uniform standards, and the functions of IoT devices are always being improved and modified. The purpose of agent design is to solve interface problems. The agent in this layer transforms the IoT device functions into up layer and controls a part of this device that the agent manager can use directly.

3.1.2 IoT Middleware Layer

■ Agent Management

The proposed agent management architecture integrates various types and brands of IoT devices.

■ Device Agent Controller

The agent controller of devices is associated with a single IoT device that provides the first interface for communication between IoT devices and middleware software and records communication protocols that enable different IoT devices to be integrated and controlled. Communication between IoT devices of various types or brands can be controlled by the device agent controller with relevant transmission protocols. The device agent controller is also usedfor monitoring. For example, when an IoT device fails or has a fault, an error message is transmitted to the monitor processing module.

■ Redirector System

The redirector system supports communication between the upper-layer management interface and the lower-layer device agent controllers. The lower-layer device agent controllers oversee various IoT devices, so commands must be coded using different methods to communicate with IoT devices. Upper-layer management and the user interface only require a single access command or online command. The redirector system drives various lower-layer devices by sending single upper-layer commands to other IoT devices. The redirector system also converts the commands into a format that corresponds to the coding method according to communication protocols that are recorded in the device agent controller.

■ Feature Management

The feature capture module sends requests to the feature management module and supports the power consumption evaluation process.

■ Threshold Analysis

The threshold analysis module calculates the feature of devices in the application of IoT environment. The evaluation criteria depend on the data type, the amount of data, and the required computation.

■ Policy Decision

After the threshold analysis module has calculated the feature, this value is compared with the threshold to determine whether or not the data will be redirected to the cloud for computation.

■ Service Redirection

Service redirection is the main function of coordinating the various other functions. This module then sends the data to local or cloud computing systems.

■ Data Collection

The data collection module accesses data in the data pool. When the data collection module sends a request to the data pool, the requested data are extracted and returned.

■ Data Redirection

The data redirection module redirects the data to the cloud for computation or leaves it for local computation.

■ Data Reception

After the cloud computing service uses the data to make calculations, the data reception module receives the returned value. The redirection management module is a single interface in the middleware.

3.1.3 Application Layer

The proposed GreenIoT system has one or more IoT devices that collect information from the user or environment. Whether an event is triggered depends on the application scenario, each of which involves a different power consumption for computing and transmission.

3.1.4 Computing Layer

The proposed cloud computing layer includes the data storage, computing service and other services. To simplify the proposed system, the cloud computing layer is mainly responsible for computing.

3.2 Scanning Cycle

The proposed scanning cycle (**Fig. 2**) is an important mechanism of the system. When the application event is triggered, power consumption is evaluated by scanning the judgment and redirecting processes. If the evaluation mechanism module selected computing in local, the scanning cycle is classified as “Local”. If the evaluation mechanism module selected computing in remote, the scanning cycle is “Cloud”. The scanning cycle presents the various stages of every case. In this work, the proposed cycle time is 20s; a longer cycle time requires a lower frequency processing system.

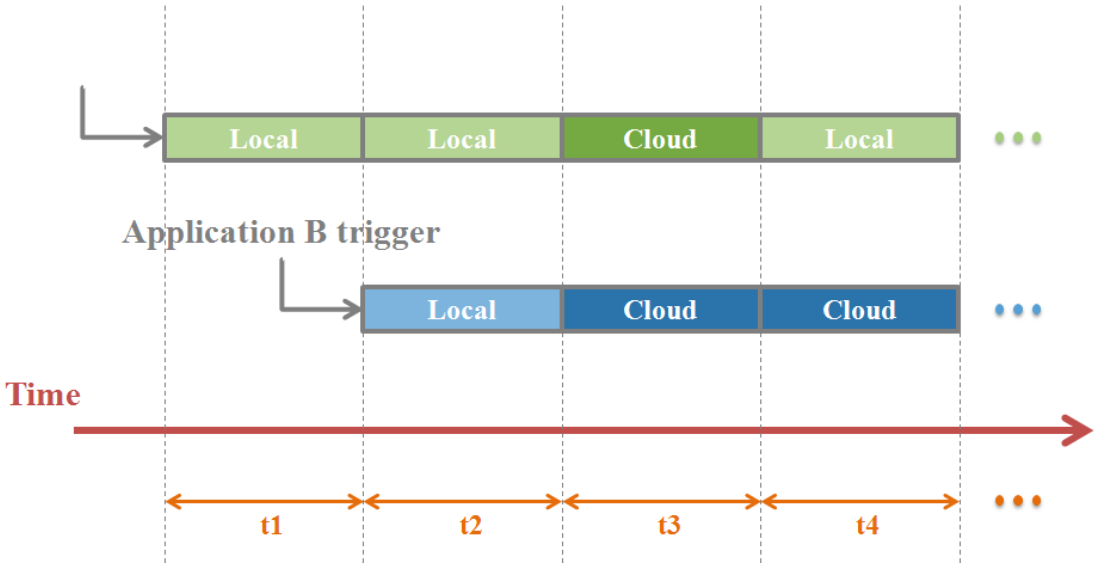

Fig. 2. Scanning Cycle

3.3 Process Sequence

The proposed process sequence (**Fig. 3**) shows the processing flow of the system. When the application event is triggered, the service evaluation begins, and other functions are coordinated. The detailed steps are as follows.

- Step 1: The service evaluation module coordinates other functions.
- Step 2: The service evaluation calls the feature capture module to get the feature requirements.
- Step 3: The feature capture module sends the feature requirements to the agent management module.
- Step 4: The agent management module returns the feature to the feature capture module.
- Step 5: The feature capture module returns the feature to the service evaluation module.
- Step 6: When the service evaluation module receives the feature, it sends a request to the threshold analysis module to calculate the grade of the feature.
- Step 7: Grades the feature and sends the grade to the policy decision module.
- Step 8: The policy decision module compares the results against the threshold and the grade and then returns the comparison results to the service evaluation module.
- Step 9: The service evaluation module begins the service evaluation.
- Step 10: The service redirection module receives data from the collection module.
- Step 11: The data collection module sends data requirements to the data pool.
- Step 12: The data pool returns the data to the data collection module.
- Step 13: The data collection module returns the data to the service redirection module.
- Step 14: The service redirection module redirects the data to the cloud computing service or the local computing service.

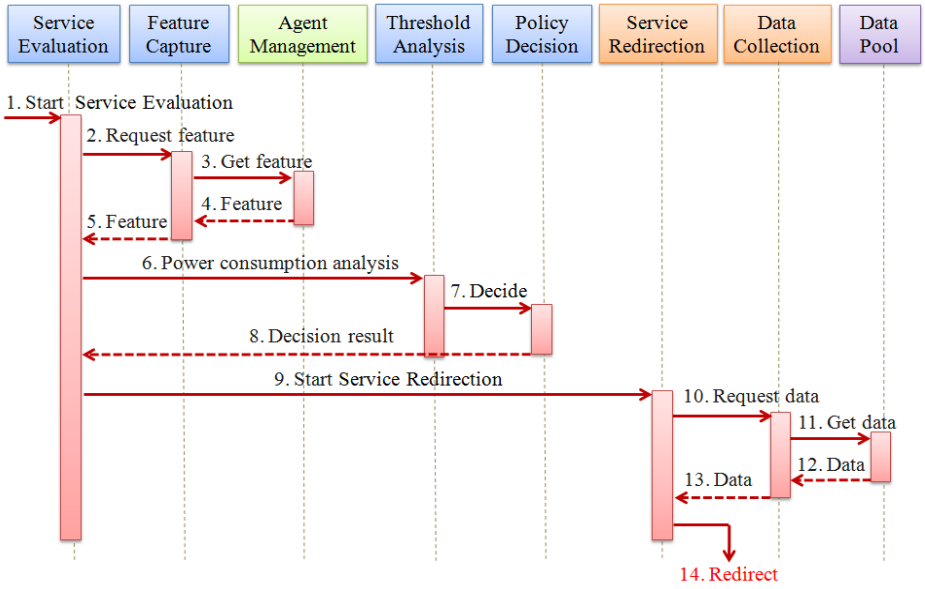

Fig. 3. Process Sequence

3.4 Evaluation Flowchart

The proposed evaluation flowchart (**Fig. 4**) shows the evaluation mechanism. When the evaluation mechanism begins, the variable *n* is set to the number of features. Each feature grade is then evaluated, and the grades are summed. After all features have been calculated, the total grade is compared with the threshold. If the total grade exceeds the threshold, the proposed data redirection module redirects the data to the cloud computing service; if not, it redirects the data to the local computing service.

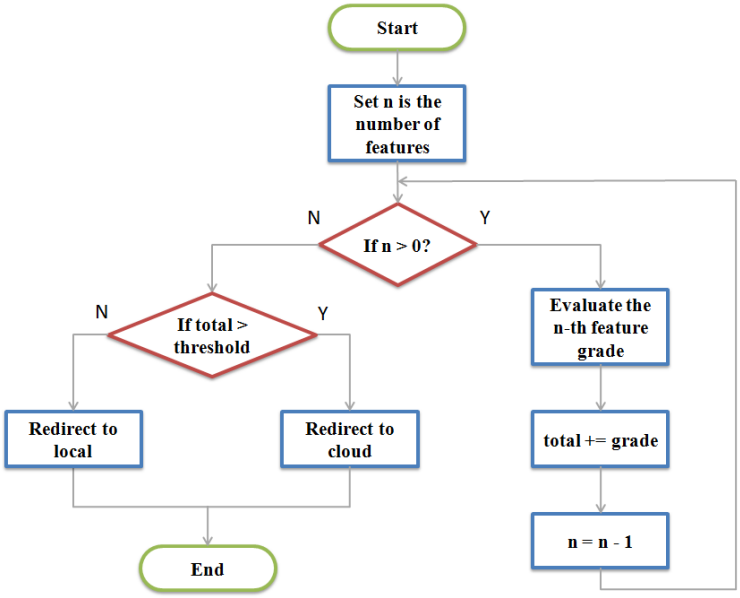

Fig. 4. Evaluation Flowchart

3.5 IoT Feature Relevance

The proposed IoT feature relationship is the relationship between the agent management and the application (**Fig. 5**). Application A and Application B use different IoT Sensors. Application A uses RFID, G-Sensor and Camera for smart home sensing and management. Application B uses GPS, NFC and humidity sensor for environmental monitoring purposes. Different sensors correspond to different features. For example, RFID corresponds to feature A while GPS corresponds to feature D.

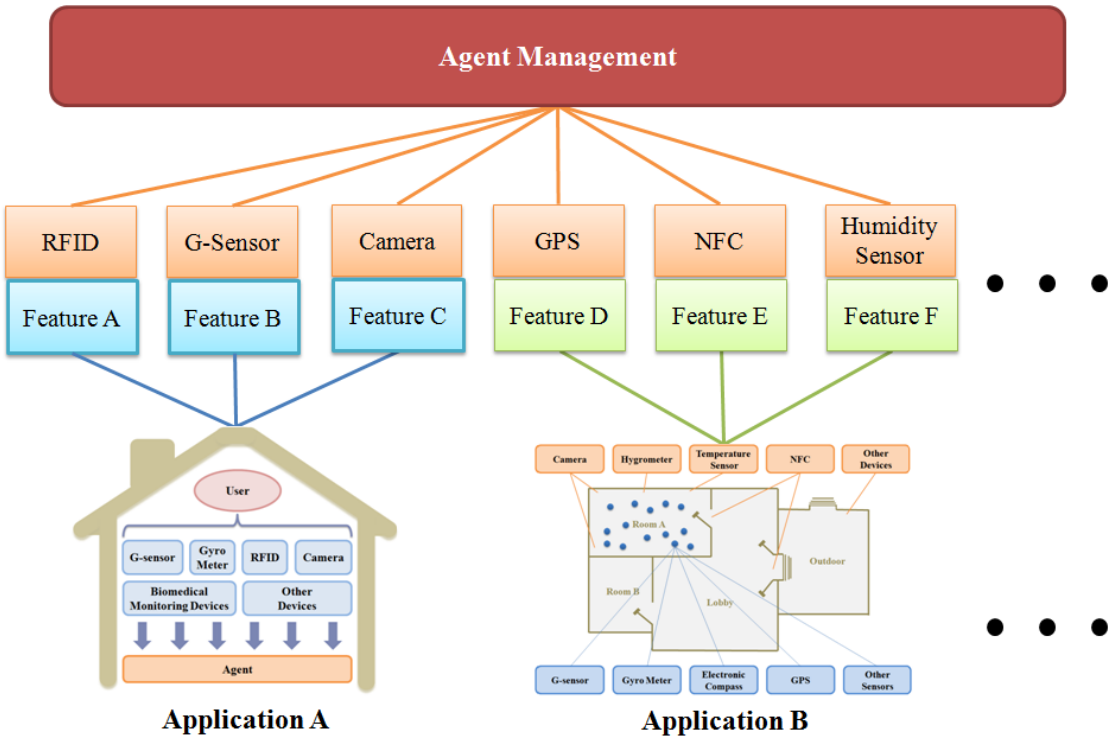

Fig. 5. IoT Feature Relationship

3.6 Evaluation Grade

This study proposed an Evaluation Grade Module for evaluating data characteristics, including type, amount and computing.

■ Type

The type category defines type as data, image or media. Data, which is the simplest type, is given a grade of 1, and media, which is the most complex type, is given a grade of 3. If data type is text, then the score is 1. If data type is image then the score into 2 points. If data type is media then the score into 3 points.

■ Amount

The amount category determines the number of data items. When the number of data exceeds 100, the grade will be increased by one; similarly, for each 50MB of data, the grade will increased by one. The system therefore calculates the grade based on the data size.

■ Computing

The computing category defines the computing property as real time and special case. If the data computing mode is general, then the score is 1. If the data computing mode is real time or special, then the score is 5.

3.7 Threshold Calculate

A Threshold Calculate Module is developed for calculating the threshold value of the application to be calculated in local or cloud computing. The threshold calculation formula is as follows:

$$G_{total} = \sum_{i=1}^{n} G_T + \sum_{i=1}^{n} G_A + \sum_{i=1}^{n} G_C \qquad (1)$$

G_T is grade point of data type obtained by use Evaluation Grade module.

G_A is grade point of data amount obtained by use Evaluation Grade module.

G_C is grade point of data computing mode obtained by use Evaluation Grade module.

G_{total} is total grade point after the calculation of equation (1).

$$G_{threshold} = (Min\ G_{total} + Max\ G_{total})/2 \qquad (2)$$

$Min\ G_{total}$ is the lowest of computing need.

$Max\ G_{total}$ is the highest of computing need.

$G_{threshold}$ is the Threshold value after the calculation of equation (2).

The proposed feedback mechanism (**Fig. 6**) corrects the threshold value. First, the threshold is set to an intermediate value between the low and high grades, so it may not be accurate. However, as the system is used, it records power consumed by the application in computing and communication. The feedback mechanism uses these data and the evaluation formula to correct the threshold.

Feedback

Grade	8	9	9.5	10	11
Computing		A		C	
Communication		D		B	

Fig. 6. Feedback

The proposed evaluation formula for the power consumed in computing and communication is simplified by using Android to specify each variable as follows. The variables are defined as follows.

- P_{total}: Total power consumption (mA).
- $P_{process}$: Process power consumption (mA).
- $P_{traffic}$: Traffic power consumption (mA).
- P_{radio}: Radio power consumption (mA).
- $P_{strength}$: The power consumption of signal strength (mA).
- $P_{scanning}$: The power consumption of radio scanning (mA).
- P_{cost}: The cost per byte of power consumption (mA).
- P_{active}: Defined by power profile (mA).
- P_{radio_on}: Defined by power profile (mA).
- $P_{radio\_scanning}$: Defined by power profile (mA).
- P_{radio_active}: Defined by power profile (mA).
- P_{cpu_active}: Defined by power profile (mA).
- P_{radio_active}: Defined by power profile (mA).
- T_{cpu}: The time of CPU use (ms).
- $T_{radio\_data\_up}$: The time of radio upload (ms).
- $T_{strength}$: The time of strength stay (ms).
- $T_{scanning}$: The time of scanning (ms).
- T_{mobile}: The time of mobile active (ms).
- $D_{received}$: Received data (Byte).
- D_{sent}: Sent data (Byte).
- D_{Bps}: Byte per second.
- $D_{radio:}$ The data of radio upload (Byte).

4. System Performance Analysis

4.1 System Implementation

An Android platform is used to simulate IoT middleware, and implements the power-saving mechanism on the HTC EVO 3D X515m. **Table 2** shows the specification of device.

Table 2. Specification of Device

HTC EVO 3D X515m	
OS	Android 4.0.3
CPU	1.2 GHz dual-core
Memory	1 GB RAM
Storage	1 GB ROM
Network	HSDPA 900/2100

The Android platform is used to simulate IoT middleware and to implement the power-saving mechanism.

4.1.1 Power Profile

The proposed power profile is a system file in the application framework layer. Since power profile includes power-related information about the hardware, the system uses this file to calculate the power consumption of the application. The proposed power profile is simplified by excluding useless information.

4.1.2 Evaluation and Feedback Mechanism

The pseudo code of the proposed evaluation mechanism (**Fig. 7**) and that of the feedback mechanism (**Fig. 8**) show how the mechanism is implemented. First, the evaluation mechanism initializes all variables. The evaluation mechanism calculates each feature and finds the sum. After the data are redirected, the power consumed in data processing are recorded in the system. The feedback mechanism compares the measured consumption with the evaluated consumption. If the compared value is greater than threshold, then the threshold will be corrected accordingly. The threshold is set to 0.3 since a smaller value would require frequent adjustment.

Evaluation Mechanism

```
Initialize n to 0
Initialize grade to 0
Initialize total to 0

Set n to the number of features

While n is greater than 0
    Set grade to the evaluate of the n-th feature
    Add grade to total
    Minus 1 to n

If total is greater than threshold
    Redirect to cloud computing service
else
    Redirect to local computing service
```

Fig. 7. Evaluation Mechanism Pseudo Code

Feedback

```
A and B is the value by actual measurement
C and D is the value by evaluation formula

If ((B - A) > x && B > C)
    threshold += 1
else if ((A - B) > x && A > D)
    threshold -= 1
```

Fig. 8. Feedback Mechanism Pseudo Code

4.2 Smart Living Application

Next, the proposed application was evaluated in a smart exhibition application and a biomedical if application, which are described below.

4.2.1 Biomedical Application

The proposed biomedical use case involves the detection of a fall by an elderly person. In this case, the g-sensor, a gyro meter, RFID and biomedical monitoring devices are deployed on the user, and a camera is deployed in the environment. The use of a g-sensor and a gyro meter reduce the required amount of computation. However, when the user falls, the outputs of the g-sensor and the gyro meter change drastically. The camera, RFID and biomedical monitoring devices then begin to collect user data, and the information computing requirement rises sharply. Therefore, the computational requirement will change.

The proposed biomedical simulation scenario is divided into three stages. The first stage lasts 40 minutes, and the second and third stages each last 10 minutes. In the first stage, the user is at home, so the system receives only a small amount of data from the g-sensor and the gyro meter, so computational computing requirements are low. In the second stage, the user is more active. The camera and the biomedical monitoring devices begin to operate, and image analysis is performed to determine whether the user has fallen. Data such as heartbeat, the readings of an oxygen meter and other biomedical data are used to analyze the state of the user, so this stage requires increased data transmission. In the final stage, the user falls, and the camera begins recording video and transmitting it to the hospital to provide essential information to the medical staff who treat the user.

4.2.2 Smart Exhibition Application

Another proposed use is measuring population density. In this scenario, a camera, hygrometer, temperature sensor and NFC are deployed in the environment, and a g-sensor, gyro meter, electronic compass and GPS are worn by the user. The system monitors the movements of a crowd and the density of people in the exhibition venue. If an area has too many visitors, then the system dynamically adjusts the environmental conditions in that area. Initially, the number of visitors to an open exhibition is generally low, so the computation requirement of the system is low, but as more visitors arrive over time, the computation requirement of the system rises rapidly. The proposed evaluation mechanism assigns the computing position to reduce power consumption.

The proposed smart exhibition simulation scenario is divided into five stages. In the first stage, only 200 visitors to the exhibition are present. In the second stage, which is 9 minutes later, many visitors arrive. In this stage, the system must handle substantial data from the g-sensor, the gyro meter, the camera and environmental sensors. The number of visitors declines in the following stages as the visitors gradually leave the exhibition hall. The number of visitors falls

from a peak of 450 to 100. In this scenario, the system saves power by dynamically adjusting its data processing method as the number of visitors changes.

4.3 Performance Analysis

This section describes the implementation and simulation results for the two proposed smart living applications the biomedical application and the smart exhibition application. Power consumption is compared with and without proposed mechanism. The scanning period is set to 20s, and each point on the curve represents a sampling time. The x-axis represents time over a period of one hour. The y-axis represents the power consumption, in units of mAh.

Fig. 9 shows the accumulated power consumption of the proposed biomedical application. Experimental results show that the power-saving mechanism reduces the power consumption by 24%.

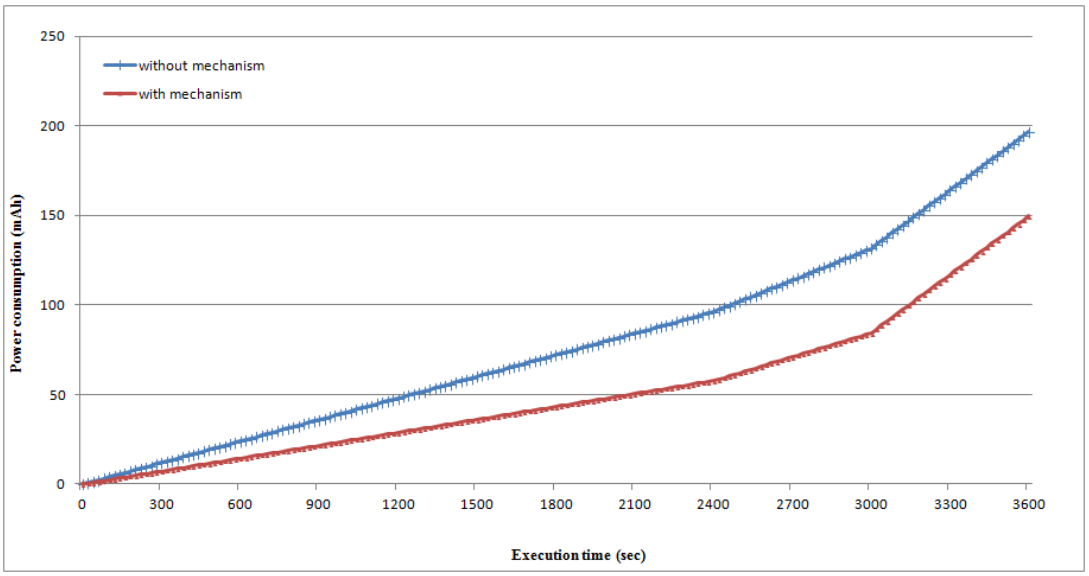

Fig. 9. Biomedical Accumulated Consumption

Fig. 10 shows the accumulated power consumption of the proposed smart exhibition application. Experimental results show that the power-saving mechanism reduces power consumption by 9.2%.

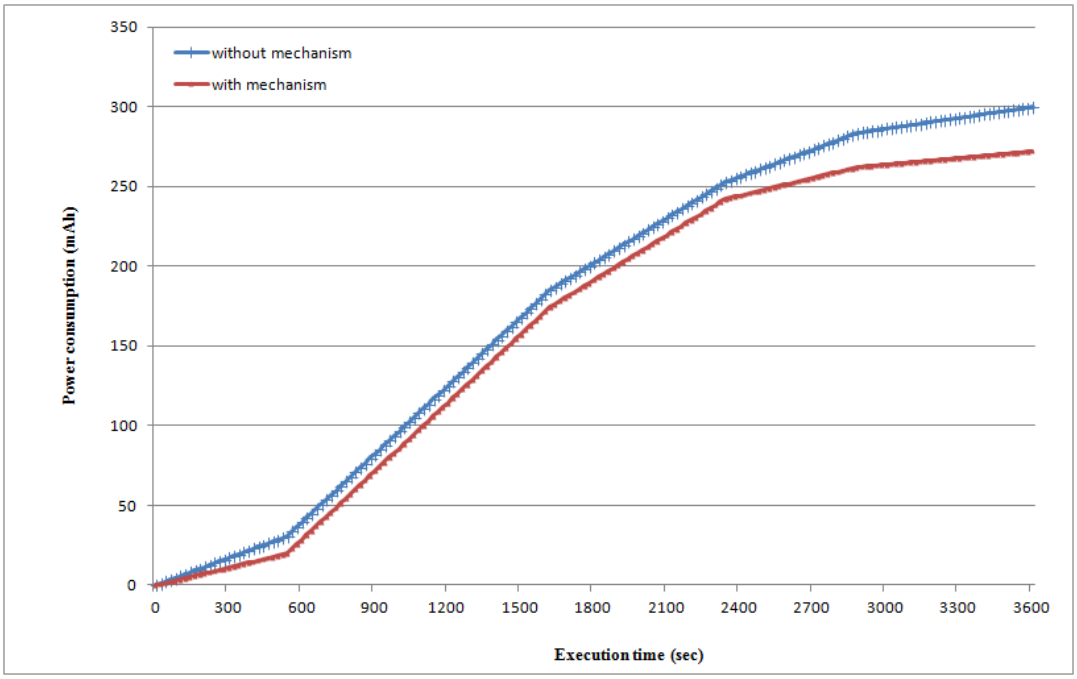

Fig. 10. Smart Exhibition Accumulated Consumption

5. Conclusion

This work proposed a power-saving mechanism in IoT architecture that is effective for two smart living applications. The proposed feedback mechanism continually revises the threshold to maintain accurate estimates and to maintain efficient power use. The simulation results show that the proposed power-saving mechanism effectively improves power usage. To increase accuracy in analyzing local or cloud computing, this work also proposed a feedback mechanism that automatically corrects the threshold value. The proposed power-saving mechanism with IoT architecture for dynamic switching between the processing of data in the cloud computing environment and in the IoT middleware effectively reduces power consumption.

References

[1] C. Doukas and I. Maglogiannis, "Bringing IoT and Cloud Computing towards Pervasive Healthcare," in *Proc. of the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp.922-926, 2012. Article (CrossRef Link).

[2] T. Liu and D. Lu, "The Application and Development of IOT," in *Proc. of the International Conference on Information Technology in Medicine and Education*, pp.991-994, 2012. Article (CrossRef Link).

[3] B.B.P. Rao, P. Saluia, N. Sharma, A. Mittal and S.V. Sharma, "Cloud Computing for Internet of Things & Sensing based Applications," in *Proc. of the IEEE International Conference on Sensing Technology*, pp.374-380, 2012. Article (CrossRef Link).

[4] C. Doukas, T. Pliakas and I. Maglogiannis, "Mobile Healthcare Information Management utilizing Cloud Computing and Android OS," in *Proc. of the International Conference on Engineering in Medicine and Biology Society*, pp.1037-1040, 2010. Article (CrossRef Link).

[5] R.S. Istepanian, S. Hu, N.Y. Philip and A. Sungoor, "The Potential of Internet of m-Health Things "m-IoT" for Non-Invasive Glucose Level Sensing," in *Proc. of the IEEE International Conference on Engineering in Medicine and Biology Society*, pp.5264-5266, 2011. Article (CrossRef Link).
[6] K. Nagata, S. Yamaguchi and H. Ogawa, "A Power Saving Method with Consideration of Performance in Android Terminals," in *Proc. of the International Conference on Ubiquitous Intelligence & Computing*, pp.578-585, 2012. Article (CrossRef Link).
[7] N.A. Maweri, K. Samsudin and F.Z. Rokhani, "Runtime CPU Scheduler Customization Framework for a Flexible Mobile Operating System," in *Proc. of the IEEE Student Conference on Research and Development*, pp.85-88, 2009. Article (CrossRef Link).
[8] A.A. Muhsen and R.F. Babiceanu, "Systems Engineering Approach to CPU Scheduling for Mobile Multimedia Systems," in *Proc. of the IEEE International Conference on Systems Conference*, pp.239-243, 2011. Article (CrossRef Link).
[9] H. Ma and R. Zimmermann, "Adaptive Coding with CPU Energy Conservation for Mobile Video Calls," in *Proc. of the IEEE International Conference on Multimedia and Expo*, pp.717-722, 2012. Article (CrossRef Link).
[10] X. Jia, Q. Feng, T. Fan and Q. Lei, "RFID Technology and its Applications in Internet of Things (IoT)," in *Proc. of the International Conference on Consumer Electronics, Communications and Networks*, pp.1282-1285, 2012. Article (CrossRef Link).
[11] H.H. Yen, "Novel visual sensor deployment algorithm in PTZ wireless visual sensor networks," in *Proc. of the IEEE Asia Pacific Conference on Wireless and Mobile*, pp.214-218, 2014. Article (CrossRef Link).
[12] Y. Huang and G. Li, "Descriptive Models for Internet of Things," in *Proc. of the IEEE International Conference on Intelligent Control and Information Processing*, pp.483-486, 2010. Article (CrossRef Link).
[13] Y. Zhu and T.N. Wang, "Research on the Visualization of Equipment Support Based on the Technology of Internet of Things," in *Proc. of the International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp.1352-1357, 2012. Article (CrossRef Link).
[14] X. Zhang and H. Li, "A Self-Reconfigurable Sensor Network Construction Research in the Paradigm of Internet of Things," in *Proc. of the International Conference on Computer Science & Service System*, pp.311-314, 2012. Article (CrossRef Link).
[15] R. Shrestha and A. Yao, "Design of Secure Location and Message Sharing System for Android Platform," in *Proc. of the International Conference on Computer Science and Automation Engineering*, pp.117-121, 2012. Article (CrossRef Link).
[16] K.C. Son and J.Y. Lee, "The Method of Android Application Speed Up by Using NDK," in *Proc. of the International Conference on Awareness Science and Technology*, pp.382-385, 2011. Article (CrossRef Link).
[17] Y. Zhang, M. Yang, Z. Yang, G. Gu, P. Ning and B. Zang, "Permission Use Analysis for Vetting Undesirable Behaviors in Android Apps," *IEEE Transactions on Information Forensics and Security*, vol.9, no.11, pp.1828-1842, 2014. Article (CrossRef Link).
[18] T.K. Kundu and K. Paul, "Improving Android Performance and Energy Efficiency," in *Proc. of the International Conference on VLSI Design*, pp.256-261, 2011. Article (CrossRef Link).
[19] Y.J. Kim, S.J. Cho, K.J. Kim, E.H. Hwang, S.H. Yoon and J.W. Jeon, "Benchmarking Java Application Using JNI and Native C Application on Android," in *Proc. of the International Conference on Control, Automation and Systems*, pp.284-288, 2012. Article (CrossRef Link).
[20] S. Liang, X. Du, C.C. Tan and W. Yu, "An effective online scheme for detecting Android malware," in *Proc. of the International Conference on Computer Communication and Networks*, pp.1-8, 2014. Article (CrossRef Link).

Yi-Wei Ma is a lecturer in China Institute of FTZ Supply Chain, Shanghai Maritime University. He received the Ph.D. degree in Department of Engineering Science at National Cheng Kung University, Tainan, Taiwan. His research interests include internet of things, digital home network, cloud computing, embedded system, multimedia p2p streaming and ubiquitous computing.

Jiann-Liang Chen was born in Taiwan on December 15, 1963. He received the Ph.D. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 1989. Since August 2008, he has been with the Department of Electrical Engineering of National Taiwan University of Science and Technology, where he is a distinguished professor now. His current research interests are directed at cellular mobility management and personal communication systems.

Yung-Sheng Lee (M'92) received the B.S. and M.S. degrees in Electrical Engineering from National Taiwan University in 1986 and 1988, respectively. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taiwan. His research interests include wireless communications, network architectures, automation systems and renewable energy systems.

Hsin-Yi Chang received the M.S. degree in Electrical Engineering of National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include digital home network, and software-defined network.