Integration of Agricultural Wireless Sensor Networks to Web-of-Things Through an Edge-Computing-Enriched WSNs/WoT Gateway

Chenrui Yu, School of Mechanical Engineering, Shanghai, China
https://orcid.org/0000-0003-1245-3018

Liang Gong, School of Mechanical Engineering, Shanghai, China*
Rui Fang, School of Mechanical Engineering, Shanghai, China
Yixiang Huang, School of Mechanical Engineering, Shanghai, China
Wei Wu, School of Mechanical Engineering, Shanghai, China
Chengliang Liu, School of Mechanical Engineering, Shanghai, China

ABSTRACT

To connect agricultural wireless sensor networks (WSNs) to web services and applications, the agricultural WSNs/WoT gateway is empowered with local data management and network maintenance functions for downstream WSNs in addition to traditional upstream data collection. This work demonstrates a low-cost, highly scalable, rapidly deployable web of things (WoT) gateway with edge computing capabilities. First, an agricultural WSNs/WoT topology is architected, connecting a ZigBee WSNs to the Web for remote monitoring the local environmental and agronomical information, and simultaneously for managing the solar-powered WSNs for a prolonged lifespan according to the instant data scrawled from the cloud. Second, a WSNs/WoT gateway is designed with the hardware platform Raspberry Pi 3, which serves multiple needs for bidirectional information exchange and local WSNs management. Finally, experimentation demonstrates the proposed hardware platform and architecture can perform edge computing, and efficiently realize the up and down transmission and distribution of data stream.

KEYWORDS

Agricultural Sensing, Edge Computing, Energy-Aware Scheduling, Gateway, Raspberry Pi, WoT, WSNs

1. INTRODUCTION

Recent advances in sensors, actuators and wireless radio frequency (RF) technologies and their convergence with the Internet offer vast opportunities for the development and application of WSNs for agriculture (Pierce & Elliott, 2008). conventional agricultural WSNs merely collect local environmental or agronomical information and send them to the remote terminal, while contemporary smarter applications support web-based, bi-directional services with highly dynamic content and real-time data(Piromalis &

DOI: 10.4018/IJITN.309694
*Corresponding Author

Arvanitis, 2016; Sabri et al., 2012). Web-Services-enabled devices allow the usage of different existing Web technologies for the development of applications, and in such a more scalable and flexible way the agricultural WSNs evolve into Web-of-things (WoT) (Howell et al., 2018; Tran et al., 2017).

Under these circumstances, agricultural wireless sensor networks can be interconnected to the Internet to achieve the Web of Things through a WSNs/WoT gateway (Qian et al., 2010). Typically, in an upstream data flow, agricultural WSNs acquire the environmental and agronomical data such as radiation, temperature, humidity, air quality, soil parameters and plant images. Meanwhile, in the downstream data flow, Web-based commands such as ventilation, irrigation controls and data routing in the agricultural WSNs are distributed to the actuators and slave WSNs. The WSNs/WoT gateway plays a paramount role in an Internet-to-sensor bridging end, which coordinates the bidirectional data flow, enabling seamless and ubiquitous web communication between users and the agricultural WSNs. However, developing such hierarchical systems is a challenging task due to the fact the gateway takes critical responsibilities for local data processing, bidirectional communication coordination, and WSNs scheduling and management, and herein these functionalities implemented in a gateway device are referred to as edge computing, discriminating the gateway-oriented computation from that in WSNs or cloud servers (Morabito et al., 2018). Hence, diverse endeavors are needed to meet the building requirements of edge-computing-enriched WSNs/WoT gateway, such as (1) Bridging different communication networks by interacting with multiple cloud-based servers and heterogeneous sensor devices; (2) Ensuring greater flexibility in integrating newer applications while maintaining service isolation; (3) Ensuring a healthy balance between design requirements, specific performance indicators, and application management(Chang & Huang, 2016; Gagnes et al., 2006).

In this paper proposes, the authors illustrate the design and implementation of a gateway to fulfill information exchange in heterogeneous networks and enable bidirectional data streams between the remote server and local agricultural WSNs more transparently and effectively, highlighting its edge-computing functionalities for supporting energy-efficient WSNs management according to Web-based service push and pull subscription.

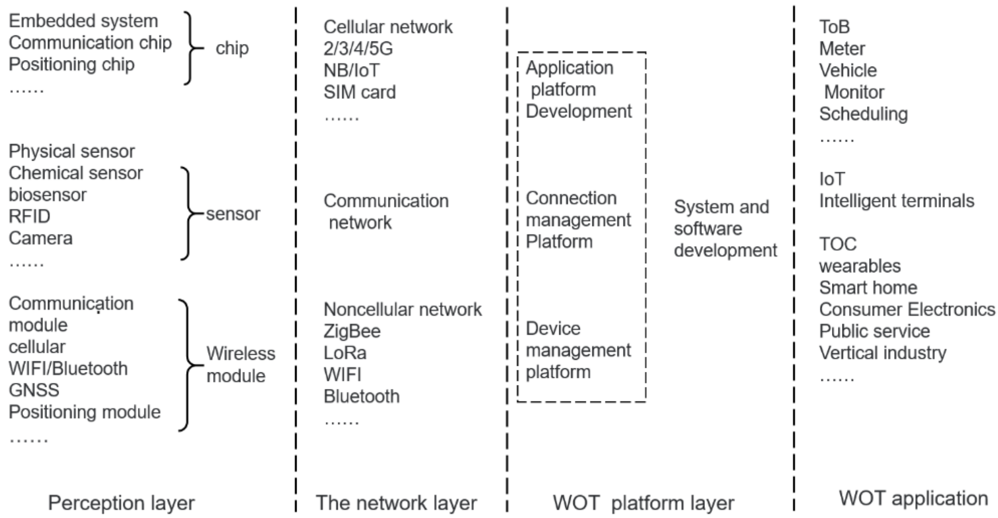
The rest of this paper is organized as follows. In section 2, the authors introduced the architecture of the Agricultural WoT. Section 3 describes the architecture and implementation process of the WSNs/WoT gateway. Section 4 shows the final experimental results of this study. Section 5 provides the conclusions of this study.

2. ARCHITECTURE OF THE AGRICULTURAL WOT

The rapid development of embedded technology, sensor network technology, and RFID technology has provided support for the interconnection among objects. A wide variety of smart things, complex transformations between different network protocols and application exclusivity have led to the complexity of inheritance among the WoT. Therefore, the Web of Things technology that combines Web technology and WoT technology has emerged (Gubbi et al., 2013; Khan et al., 2012). WoT integrates smart things, such as RFID tag objects, sensor nodes, etc., using web technologies to provide an open platform for the WoT. WoT application development can simplify the application integration of intelligent objects by making full use of mature Web development tools, programming languages and methods, and also can realize the fusion between the physical space of intelligent objects and the virtual space of web applications (Ferdoush & Li, 2014).

Most WoT applications are still closed-loop applications, and resource information lack sharing between applications, which is not conducive to promoting application innovation. WoT technology virtualizes objects into Web resources, providing the foundation for interoperability between objects, which enables interaction and collaboration among objects. WoT technology is open and flexible. This paper proposes and implements a WoT framework, which uses the web technology gateway and WoT service platform to open sensor capabilities to the application layer. The Classical architecture of WSNs/WoT is shown in Figure 1. The architecture mainly includes four parts.

Figure 1. The classical architecture of WSNs/WoT



This paper implements a WOT system based on multiple WSNs including the WSNs, the gateway, the server and the monitoring terminal. First, the gateway aggregates the data from agricultural Wireless Sensor Networks (WSNs), which is implemented by integrating a ZigBee coordinator as the data receiver. Second, the gateway performs data management such as data storing, compressing and energy-aware scheduling in an edge-computing fashion with a light-weighted database in the Raspberry Pi. Third, the cloud-server supplies a Human Machine Interface (HMI) for environmental monitoring in the agricultural application within the upstream data flow, meanwhile a web-API delivers instant weather information and forecast within the downstream data flow for the solar-powered WSNs to schedule the data-acquiring and data-sending tasks for long-term operation. The system architecture is shown in Figure 2.

More specifically, the Agriculture WOT consists of several wireless sensor nodes and accommodates five common environment sensors. It has two kinds of nodes; one is the router and another is the coordinator. There are several router nodes and just one coordinator node. Each node carries its own 3000mAh lithium battery, which is powered by solar power because the outdoor environment cannot provide a stable power supply. The agricultural WoT node is shown in Figure 3.

The sensors collect environment information data, and the WSNs router nodes transmit data to the coordinator node, which is integrated into the gateway. Every node integrates the wireless communication module, which has data transmission and routing functions. It is responsible for maintaining the network topology and managing routing information. Such nodes feature multi-hop, self-organizing and non-centered. The data collected by the sensor nodes are eventually converged to the sensor network coordinator through the sensor network. And then, the data is transmitted to Raspberry Pi, which is the main part of gateway hardware. The gateway interprets the received data according to the WSNs transport protocol and translates it into the data format recognized by traditional internet protocol. And the data will be forwarded to the cloud server eventually. Such a gateway device with built-in multiple WSNs coordinators implements AD HOC network management and its intercommunication with the Internet. Due to AD HOC's plug-and-play, strong mobility and good flexibility, it is very convenient to delete or add network sensor nodes at the WSNs (Carlson et al., 2013). The cloud-server supplies an HMI for environment monitoring in the agricultural application within the upstream data flow, meanwhile, a web API delivers instant weather information and forecast within the downstream data flow for the solar-powered WSNs to schedule the data-acquiring and

Figure 2. The proposed WSNs/WoT architecture with an edge-computing-enriched gateway

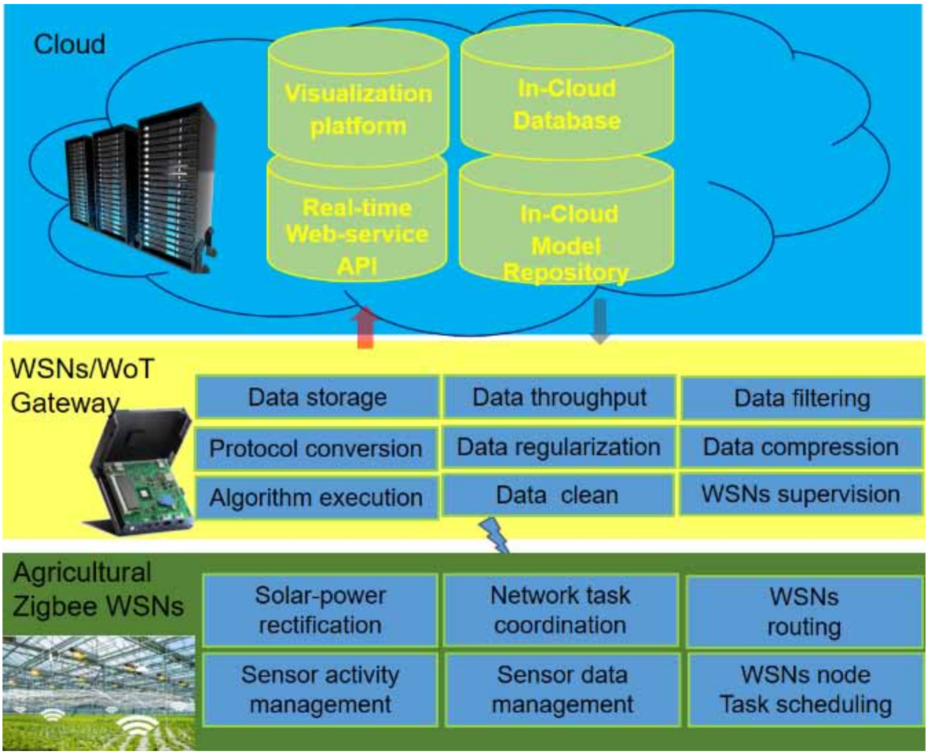
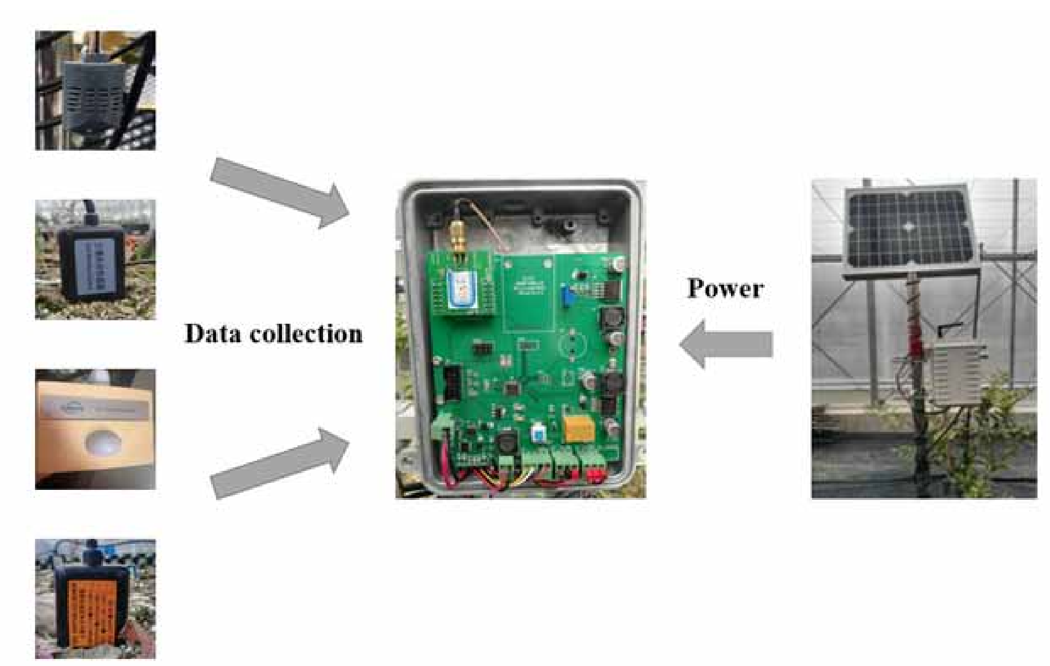


Figure 3. The agricultural WoT node



data-sending tasks for long-term operation. The web-API use the API interface service provided by Ali cloud. This paper focuses on the design of the gateway.

3. WOT GATEWAY DESIGN AND IMPLEMENTATION

3.1 Functionalities of a WSNs/WoT Gateway

As WoT designers push intelligence to the edge of the WoT, the WSNs/WoT gateway must serve multiple needs, such as protocol conversion and data transmission between multiple sensing networks and basic networks. In addition, the gateway also needs to have a device management function. The gateway supports WSNs communication interface and supports TCP/IP, WiFi, 3G/4G, GPRS communication interfaces and other protocol interfaces. Therefore, the WoT gateway features the following functionalities:

1. **Extensive access capabilities:** There are many transmission protocols currently used for short-range communications, each of which has its targeted application and lacks compatibility and uniform standards. Data access capability for heterogeneous networks is one of the gateway's essential functions.
2. **Protocol conversion:** The WoT gateway solves the communication problem between WSNs and the traditional Internet. Therefore, a unified information model for information exchange needs to be established to realize the conversion of communication protocols. This function standardizes the construction of agricultural WoT applications and achieves the goal of efficient application integration and data sharing.
3. **Data Interoperability:** Interoperation refers to the process of interactive operation of data and information sharing between two systems (Calvo et al., 2016). The interoperability between information systems is the basic requirement for system integration.
4. **Management functions:** The amount of heterogeneous data received by the gateway need to be properly managed. Simultaneously, it receives not only the node data of the sensor network, including the node's identification, parameters, status and other information, but also the control commands issued by the remote monitoring terminal. Therefore, the gateway needs to identify, classify, and process all kinds of data to achieve unified management (Al-Fuqaha et al., 2015).
5. **Edge computation:** Each WSNs/WoT gateway is designed to manage a local area WSNs. The sensory data from WSNs are stored in a light-weighted database and optimally selected for transmission. Meanwhile, an energy-aware task scheduling algorithm parses the Web-API package for solar radiation forecasting data to hibernate or invoke the WSNs node in real-time according to the node energy status and its energy-harvest prediction. Both of these strategies alleviate the server and WSNs computational workload in an edge computing manner.

The heterogeneity of application scenarios and applications brings lots of challenges to the abovementioned gateway design. It is critical to design and implements efficient algorithms and hardware platform for customizing the WSNs/WoT gateway. Here an open-source hardware platform Raspberry Pi 3 is adopted as a future-ready and easily customizable edge computing device that can be expanded and configured to match and scale to every requirement in a gateway context. The Raspberry Pi Card computer was originally designed for student computer programming education. Considering its powerful scalability and excellent performance, Raspberry Pi has become an important part of education, research and amateur network physical systems. It is based on the LINUX operating system and supports C, java, python and other languages. It provides Ethernet, USB, and HDMI interfaces, featuring processing, networking, and video decoding capabilities (Molano et al., 2015; Upton & Halfacree, 2014). It also exposes its General-Purpose Input-Output pins (GPIO), making it a simple matter to connect peripherals. Because the Raspberry Pi does not have a memory chip, all data, including the operating system, needs to be saved on a micro-SD card. Such a resilient hardware architecture can help create the secure, scalable and robust infrastructure needed to facilitate WSNs/WoT convergence, reduce risks and maximize overall working time. Samourkasidis A et al proved that Raspberry Pi can be used as a small, low-cost data repository that provides persistent data storage and data sharing services (Samourkasidis & Athanasiadis, 2017). Due to the simple and easy-to-use, low-cost features of the Raspberry Pi platform, it has swept the world in just a few years. The application of the Raspberry Pi in a variety of situations has gone far beyond its educational significance. These papers (Jain et al., 2014; Leccese et al., 2014; Noriega-Linares & Ruiz, 2016; Rudin et al., 2016; Segura-Garcia et al., 2015) apply the Raspberry Pi to experimental research, smart city, home automation, intelligent medical and other fields. In these papers(Flores et al., 2016; Kamath et al., 2019; Moshayedi et al., 2019; T. & I., 2020; Z. et al., 2020), Raspberry Pi is used in agricultural WoT systems. The authors can conclude from these works that the Raspberry Pi is powerful and cost-effective, and it is feasible to build a gateway platform for the Web of things. In this paper, the authors designed and implemented a lightweight, low-cost gateway system using a Raspberry Pi as a hardware-centric platform. It provides WIFI/ZIGBEE-based WSNs interfaces, data interoperability, data storage, and data sharing services.

3.2 Hardware

The WoT gateway uses the Raspberry Pi as the center to establish a gateway hardware platform, which the authors denote as a "network repeater". On one aspect, the network repeater, which can be connected to the Internet in various modes such as Ethernet, WiFi, or GPRS, is used as the gateway main hardware to implement wide access to communication protocols. The remote server can open ports to receive data. On the other aspect, the network repeater is connected to the embedded sensor network coordinator, which implements USB to TTL serial communication through the PL2303. The coordinator then performs parameter configuration and topology management on the sensor network. The function module is shown in Figure 4(a), and Figure 4(b) shows the photograph of the gateway. To better illustrate the functionality of the gateway, the authors match the functionality implemented by the gateway with the general overview functionality in Section 3.1, as shown in the following table 1.

The proposed sensor network coordinator selected C8051F340 as the master chip. The C8051F34x devices are fully integrated mixed-signal system-on-chip (Microcontroller Units) MCUs. It is fully compatible with the standard 8051 core, and instruction execution speed is greatly improved (Jassas et al., 2015). The coordinator includes four parts, the ZigBee module, the WiFi module, the voltage conversion module, and the serial communication module. The ZigBee module selects the DRF1605H module. This module can realize the conversion between ZigBee and serial port communication. The

Figure 4. Raspberry Pi based gateway associated with the WSNs coordinator (a) WSNs/WoT scheme and (b) Gateway photograph

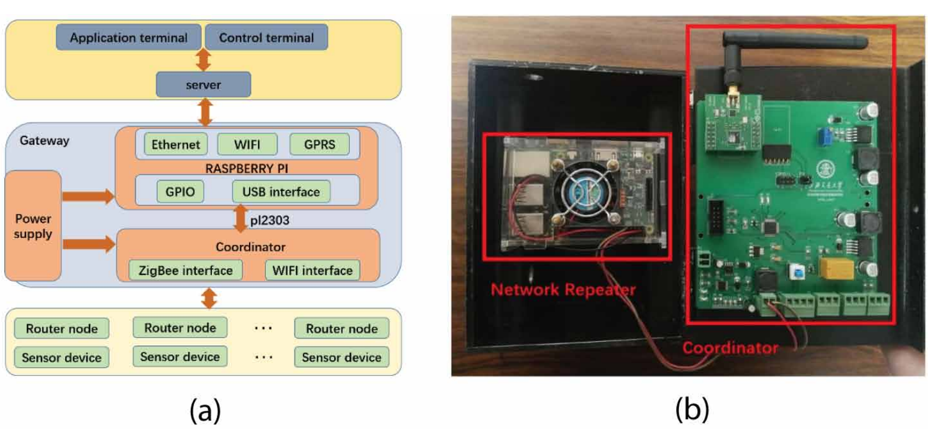


Table 1. Features of the WoT Gateway

Function	Description
Extensive access capabilities	For the WSNs side, the gateway provides Zigbee and wifi communication methods. For the web side, it provides three communication methods: Ethernet, WIFI, and GPRS.
Protocol conversion	For different protocols, there are corresponding interfaces in the gateway, and then the data is parsed and converted.
Data Interoperability	The authors can query the real-time and historical data of WSNs in the cloud server, and WSNs can also receive web scheduling commands.
Management functions	The gateway can implement various functions such as data storage and forwarding, WSNs address allocation and network topology, and energy scheduling calculation.
Edge computation	The authors combined WEB-resource information and local data to design energy-aware scheduling. This work is done in the repeater instead of in the cloud server.

antenna can be installed and the outdoor transmission distance can reach 1.6 kilometers. The WiFi module selects the TTL-WiFi module to realize the real-time transparent transmission between the serial port and WiFi module. The voltage conversion module converts the power supply voltage to the required voltage for each module of the development board. The serial communication module serves as the communication interface with the network repeater. The coordinator prototype realizes data collection and data transmission for sensor networks.

The sensor node also includes a ZigBee module, a WiFi module, a voltage conversion module, and a serial communication module, and the design implementation is also the same as the coordinator. The difference is that, in order to achieve solar power and energy-aware, the solar node voltage regulator module, relay module and battery energy measurement module are added to the sensor node. In addition, each sensor node carries a 3000mah lithium battery for powering and storing solar energy. The solar power supply voltage regulator module is used to connect to solar panels with unstable input voltage. The relay module controls the sleep and working state switching of the node. When the relay is turned on, the sensor and the wireless transmission module are energized. Otherwise, the power is turned off. The battery energy measurement module calculates the remaining battery power.

3.3 Software Development

The functionalities of the WoT gateway are as follows. (1) To aggregate the data from the agricultural sensor network and realize it by using the Zigbee coordinator as the data receiver. (2) To perform data management such as data storage and energy efficiency scheduling in an edge computing method. (3) To keep the real-time communication of WSNs/Web and maintain the stability of WSNs. The WoT gateway prototype software function module is shown in Figure 5.

In this system, each node has the function of data acquisition and independent routing. The data's format is shown in Table 2. The sensor network coordinator receives data from the WSNs. It identifies the data source address ID after judging whether the data's format is correct, and assigns a network repeater recognized address ID to the data again. The collated data is sent to the Raspberry Pi through the serial port. A timer is started in the program to eliminate the error redundancy periodically generated when receiving data errors. When the data is wrong, the receptive fields and send fields are cleared. Similarly, the data that the coordinator accepts from the network repeater is still processed, then address reassigned, and finally sent to WSNs.

The network repeater communicates with the serial port of the WSNs Coordinator prototype through the USB interface, and uses the Python language to write an application programming interface (API) based on the USB interface to realize the transmission of sensor data. First, accept data from the coordinator and parses the node data, such as the node ID, environment data, and battery voltage. And then, sqlite3, a lightweight integrated database module in python, creates data

Figure 5. Software design function module

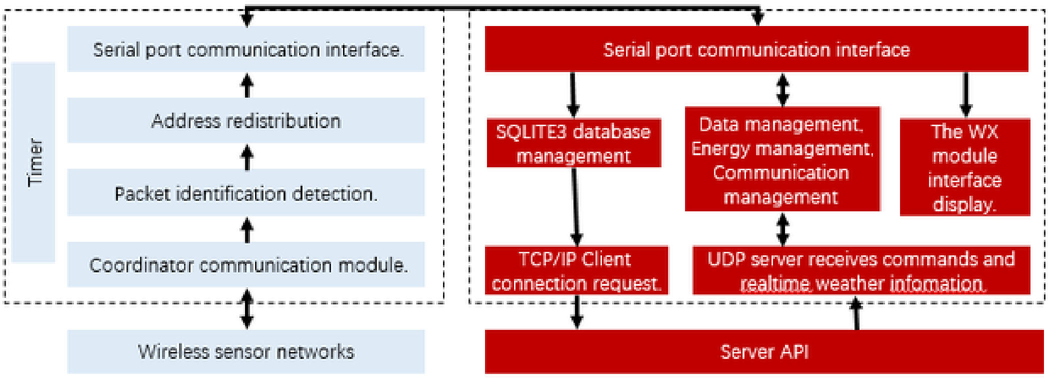


Table 2. The format of the data received by the coordinator

Byte	Data format specification
byte1	FD (Transmission type description)
byte2	Data length
byte3~byte4	Transmission target address
byte5~byte13	Transfer data
byte14~byte15	Source address ID

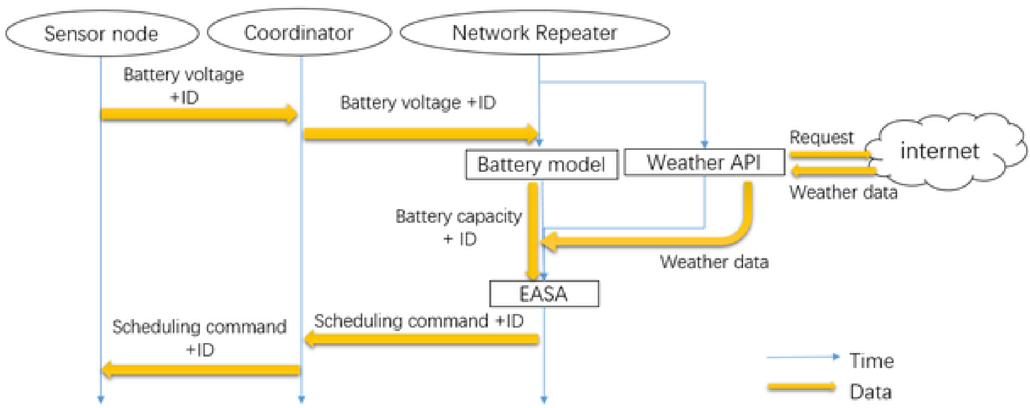
tables for normalized storage, and meanwhile, Wxpython, an image user interface module of python, realizes real-time display of node data on the network repeater. second, the program sends TCP/IP client connection requests to the cloud-server and export the data to the server from sqlite3 database.

In the Energy-Aware Scheduling task, the network repeater inputs battery voltage data into the battery model to obtain the remaining battery power. At the same time, it will obtain weather data through the weather API and enter it into the Energy-Aware Scheduling Algorithm(EASA) along with the battery model output data. Finally, the scheduling scheme is obtained and a control command is issued. The communication flow and data flow are shown in Figure 6. And the pseudo-code of EASA is shown in Algorithm 1.

4. EXPERIMENTATION AND RESULTS

The key components of the perceived node's energy consumption are data acquisition, wireless transmission, and the basic energy consumption of nodes to maintain operation. It is important to formulate the operation strategy of the node device according to the intensity of the sunlight and the remaining power of the battery in the energy-aware scheduling algorithm. The given design is to adjust the frequency of data acquisition and transmission, as described in the energy-aware scheduling algorithm. The authors measured the power consumption of the node in the high-power state, low power state, idle state, and sleep state, and the result is shown in table 3. More specifically, the high power, low power and idle states respectively indicate three different data acquisition and transmission frequencies for the node device. The Sleep state means that the node does not perform

Figure 6. The communication flow and data flow in the Energy-Aware Scheduling



Algorithm 1. The pseudo-code of the Energy-Aware Scheduling Algorithm

Data_WSNs[p] = serial.Read (port)
$Node_{num}$ = *Data_WSNs*[0]
for (i = 1; i < 1 + l_{env_len}; i++)
    Env_{info}[i] = *Data_WSNs*[i]
E_{res} = *Data_WSNs*[1 + l_{env_len}]
Input $Node_{num}$, Env_{info}, P_{polar}, E_{res} to sqlite3 database
soket.connect ('server_IP', server_port)
soket.send ($Node_{num}$, Env_{info})
conn = socket.accept()
Data_web[q] = conn.recv()
$Exec_{instr}$ = Data_web[k] // $Exec_{instr}$ is the control instructions for irrigation and blower.
serial.write (*Destination_address*, $Exec_{instr}$)
$Light_{solar}$ = Data_web[j]
for each node
    if E_{res} ³ E_{high}
        none
    else if $E_{res} < E_{high}$ && $E_{res} > E_{low}$
        if $Light_{solar}$ ³ $Light_{high}$
            command: none
        else if $Light_{solar} < Light_{high}$ && $Light_{solar} > Light_{low}$
            command: $T_{cycle} = n * T_{cycle}$
            // T_{cycle} refers to the interval time of Node sending data
            // It means that the node is running in a low power state
        else
            command: $T_{cycle} = m * T_{cycle}$ // m > n
            // It means that the node is running in idle state
    else
        command: Node sleep
end if
serial.write (node, command)

Table 3. Battery power consumption in different states

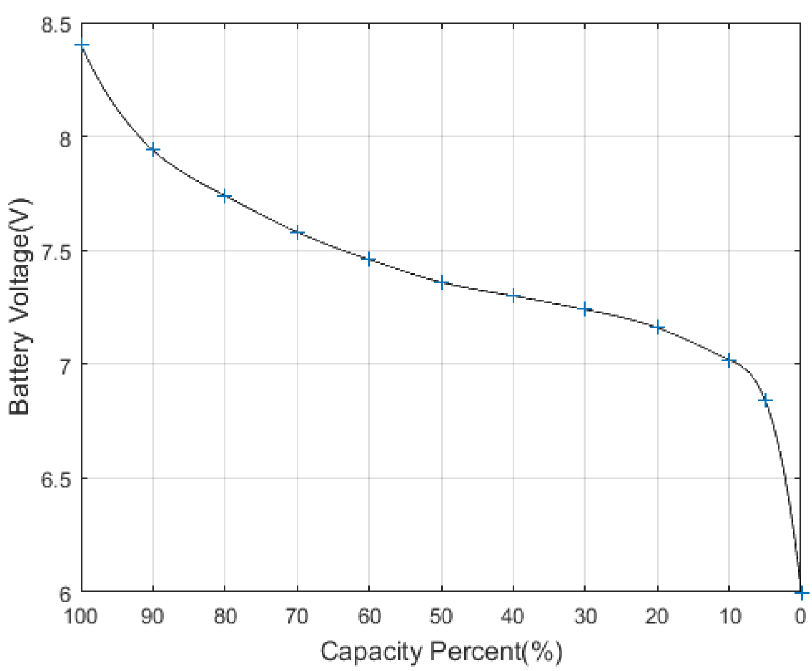
State	High power	Low power	Idle	Sleep
Power consumption(J)	0.68809	0.473952	0.35849	0.224493

active transmission activities, and only maintains a surviving state, ensuring that it can be woken up when the power reserve is sufficient. The data shows that the power consumption of the node in the low power state is 31.11% lower than that in the high-power state.

The overall energy consumption of a node is calculated indirectly from changes in battery capacity. The node device uses two parallel 4.2V lithium polymer batteries, which are mainstream in the market. Accurate battery capacity measurements are more complicated. The authors use the classic empirical model shown in Figure 7 to estimate the percentage of battery capacity remaining. The relationship between the voltage and the remaining capacity of a standard 4.2V lithium-ion battery can be represented by the curve in the empirical model. Therefore, the authors only need to measure the voltage of the battery to estimate the remaining capacity of the battery.

The authors measured and recorded the voltage value of the battery. The sampling period is 5 minutes. Here the authors extract the node battery voltage data collected from October 18 to October 23. Set the accumulated hours, which the start time is 0:00 on October 18, 2017, as the X-axis, and the battery voltage as Y-axis to establish a line graph shown in Figure 8(a). The vertical dotted line indicates 0 o'clock of the day. It can be seen that the battery voltage drops steadily from around 6 p.m. to 6 a.m. There is no light at this time, so the solar panel cannot generate energy and the battery is in a pure consumption state. The battery voltage rose in varying degrees between 6 a.m. and 6 p.m. This is because solar panels have different charging powers under different weather conditions. The weather conditions are shown in Table 4. The overcast, cloudy and light rain are all meteorological representations in the weather forecast. Overcast in meteorology refers to more than 90% covered by clouds. Cloudy refers to the cloud coverage of the sky is 30% ~90%. It can be seen that when the weather is light rain or overcast, the battery voltage is in a relatively low state. However, it can still be charged during noontime. In stable cloudy weather, the battery voltage changes in a certain periodicity. This means that even in cloudy weather, the node can maintain sufficient power. So, if there is less cloud, the battery can get more power. In energy-aware scheduling. The authors set the node to run at low power operation when the is lower than 60% and the light is insufficient. The

Figure 7. The Empirical model of battery capacity



authors fitted the data for the pure consumption phase of the battery between 6 p.m. and 6 am of the next day. The fitted curve is shown in Figure 8(b). The slope of the fitted curve is indicated in the upper left corner of Figure 8(b), and the meaning of the slope refers to the rate of consumption of the battery capacity. The first two fitted line segments are in a state where the battery capacity is below 60%, and the node device is operating in a low power consumption state. Correspondingly, the last three fitted line segments are in the high-power operation state of the device. The authors averaged the rate of consumption of the two states. The consumption rate of the high-power state is 1.2379, and the consumption rate of the low-power state is 0.8179, which is 33.93% lower than the consumption rate of the high-power state. This result is consistent with the 31.11% calculated in Table 3.

5. CONCLUSION

Compared to the traditional agricultural wireless sensor networks, the WSNs/WoT system constructed in this paper support web-based, bi-directional services with highly dynamic content and real-time data. In one way, the agricultural WSNs transfer data to the cloud, where it can be analyzed and transformed into actionable insight. On the other way, the gateway receives weather information from cloud-web-API, and performs energy-aware scheduling in an edge-computing fashion, which adjusts the working energy consumption of WSNs and maximizes the working time of the node. The application of edge computing technology spreads the burden of massive data processing of the server and communicates with WSNs in a faster and more real-time way. As both the WSNs and the WoT systems continue to evolve, their requirements are diverging from those of standard gateway. It is critical for using an open-source platform to design and implement efficient algorithms and hardware platforms for customizing the WSNs/WoT gateway to meet these challenges.

In summary, the authors have built a Raspberry Pi based WoT gateway that integrates a network coordinator. The results of the system operation show that the gateway operates stably, data communication is stable, reliability is high, and the display is light and portable. The Raspberry Pi

Figure 8. (a) Cumulative hour-battery voltage graph; (b) Partial data fitting curve

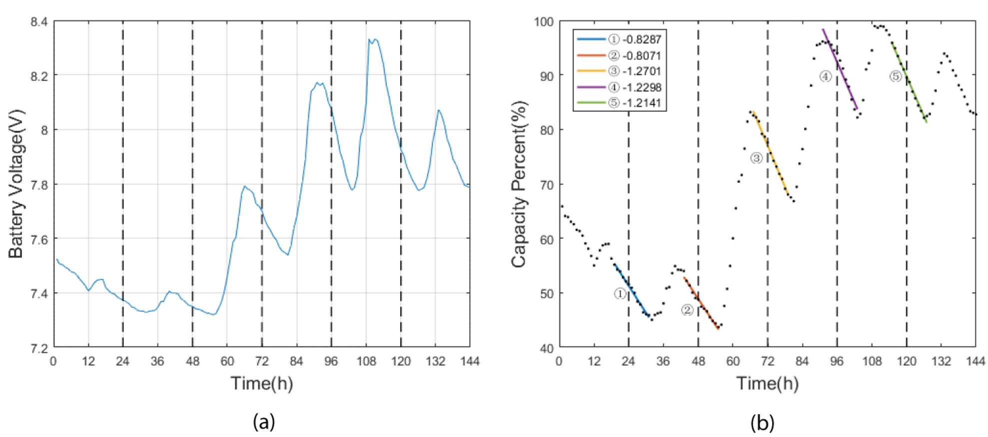


Table 4. Weather state from October 18 to October 23

Data	October 18	October 19	October 20	October 21	October 22	October 23
Weather	overcast	Light rain	Cloudy	Cloudy	Cloudy	Cloudy

is not a completely open-source device. But the work of this paper still shows superior performance. Gateways using the Raspberry Pi platform are easy to reproduce and customized. The Raspberry Pi represents an "open-source sharing spirit". Its excellent scalability gives people a great deal of room to play and generates a variety of colorful applications. Such as media centers, super calculators, drones, smart homes, intelligent robots and so on. In recent work, the Raspberry Pi has demonstrated its enormous application potential. Open-source hardware stimulates people's creativity, making DIY (do it yourself) change to DIT (do it together). The use of open source has accelerated the speed of application of the platform, accelerated the standardization of technology, derived a benign ecosystem, and promoted the full upgrade of technology products and services.

ACKNOWLEDGMENT

There have been many contributors for this to take shape and the authors are thankful to each of them. Thanks to Lu Yang and Hou Rui. Specifically for their technical support.

CONFLICT OF INTEREST

The authors of this publication declare there is no conflict of interest.

FUNDING AGENCY

This research was supported by Science and Technology Innovation Action Project of Shanghai Committee of Science and Technology [grant number 21N21900100].

REFERENCES

Leccese, F., Cagnetti, M., & Trinca, D. (2014). A Smart City Application: A Fully Controlled Street Lighting Isle Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX. Sensors (Basel), 14(12), 24408–24424. 10.3390/s141224408

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials*, *17*(4), 2347–2376. doi:10.1109/COMST.2015.2444095

Calvo, I., Gil-Garcia, J. M., Recio, I., Lopez, A., & Quesada, J. (2016). Building IoT Applications with Raspberry Pi and Low Power IQRF Communication Modules. *Electronics (Basel)*, *5*(3), 54. doi:10.3390/electronics5030054

Carlson, D., Altakrouri, B., & Schrader, A. (2013). *An ad-hoc smart gateway platform for the web of things*.

Chang, F., & Huang, H. (2016). A Survey on Intelligent Sensor Network and Its Applications. *J. Netw. Intell.*, *1*(1), 1–15.

Ferdoush, S., & Li, X. R. (2014). Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications. *Procedia Computer Science, 34,* 103-110. doi:10.1016/j.procs.2014.07.059

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems-The International Journal of eScience*, 29(7), 1645-1660. 10.1016/j.future.2013.01.010

Howell, S., Rezgui, Y., & Beach, T. (2018). Water utility decision support through the semantic web of things. *Environmental Modelling & Software*, *102*, 94–114. doi:10.1016/j.envsoft.2018.01.006

Jain, S., Vaibhav, A., & Goyal, L., (2014). Raspberry Pi based Interactive Home Automation System through E-mail *Proceedings Of The 2014 International Conference On Reliabilty, Optimization, & Information Technology,* (pp. 277-280).

Kamath, R., Balachandra, M., & Prabhu, S. (2019). Raspberry Pi as Visual Sensor Nodes in Precision Agriculture: A Study. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 45110–45122. doi:10.1109/ACCESS.2019.2908846

Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges *10th International Conference on Frontiers of Information Technology (FIT).*

Molano, J. I. R., Betancourt, D., & Gómez, G. (2015, 2015-01-01). *Internet of things: A prototype architecture using a raspberry Pi*.

Morabito, R., Petrolo, R., Loscri, V., & Mitton, N. (2018). LEGIoT: A Lightweight Edge Gateway for the Internet of Things. *Future Generation Computer Systems-The International Journal of eScience*, 81, 1-15. 10.1016/j.future.2017.10.011

Moshayedi, A. J., Roy, A. S., Liao, L. F., & Li, S. (2019). Raspberry Pi SCADA Zonal based System for Agricultural Plant Monitoring. *6th International Conference on Information Science and Control Engineering (ICISCE*), (pp. 427-433). doi: 10.1109/ICISCE48695.2019.00092. doi:10.1109/ICISCE48695.2019.00092

Muhammad, Z., Hafez, M. A. A. M., Leh, N. A. M., Yusoff, Z. M., & Hamid, S. A. (2020). Smart Agriculture Using Internet of Things with Raspberry Pi. *10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE).*

Pierce, F. J., & Elliott, T. V. (2008). Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington. *Computers and Electronics in Agriculture*, *61*(1), 32–43. doi:10.1016/j.compag.2007.05.007

Piromalis, D., & Arvanitis, K. (2016). SensoTube: A Scalable Hardware Design Architecture for Wireless Sensors and Actuators Networks Nodes in the Agricultural Domain. *Sensors (Basel)*, *16*(8), 1227. doi:10.3390/s16081227 PMID:27527180

Qian, Z., Ruicong, W., Qi, C., Yan, L., & Weijun, Q. (2010). IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things. *Proceedings 2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC 2010)*, (pp. 347-352). doi:10.1109/EUC.2010.58

Rudin, A., Audah, L., Jamil, A., & Abdullah, J. (2016). Occupancy Monitoring System for Campus Sports Facilities Using the Internet of Things (IoT). *Journal of Information Science and Engineering*, *28*(2), 295–316.

Samourkasidis, A., & Athanasiadis, I. N. (2017). A Miniature Data Repository on a Raspberry Pi. *Electronics (Basel)*, *6*(1), 1. doi:10.3390/electronics6010001

Segura-Garcia, J., Felici-Castell, S., Perez-Solano, J. J., Cobos, M., & Navarro, J. M. (2015). Low-Cost Alternatives for Urban Noise Nuisance Monitoring Using Wireless Sensor Networks. *IEEE Sensors Journal*, *15*(2), 836–844. doi:10.1109/JSEN.2014.2356342

Thaher, T., & Ishaq, I. (2020). *Cloud-based Internet of Things Approach for Smart Irrigation System: Design and Implementation*. Paper presented at the 2020 International Conference on Promising Electronic Technologies (ICPET).

Tran, N. K., Sheng, Q. Z., Babar, M. A., & Yao, L. N. (2017). Searching the Web of Things: State of the Art, Challenges, and Solutions. *ACM Computing Surveys*, *50*(4), 1–34. doi:10.1145/3092695

Upton, E., & Halfacree, G. (2014). *Raspberry Pi user guide*. John Wiley & Sons.

Chenrui Yu received a bachelor's degree from the School of Mechanical and Automotive Engineering, South China University of Technology, and is currently pursuing a master's degree in mechanical engineering from the School of Mechanical Engineering, Shanghai Jiaotong University. His research interest is agricultural robotics.

Liang GONG received a B.Sc. (Eng.) from the Department of Mechanical Engineering, Hefei University of Technology, Hefei, China, in 2002 and his Ph.D. degree in mechanical engineering from Shanghai Jiao Tong University (SJTU), China, in 2010. From 2007 to 2008, he worked as a visiting Ph.D. student with the Laboratory of Embedded Internet System, Luleå University of Technology, Sweden. From 2010 to 2012, he was a postdoctoral fellow with the Laboratory of Power Engineering, SJTU. He has taken charge of multiple corss-disciplinary projects from NSF of China, High-Tech Research and Development Program and held more than 50 patents in the ðeld of intelligent control and its applications in agriculture. He is currently an associate professor in SJTU. His research interests cover computational intelligence and cognitive science, agricultural robotics, wireless sensor networks, and motor drive power electronics. He is an associate editor of the AAAS journal Plant Phenomics.

Rui Fang received a M.Eng degree from school of mechanical engineering, Shanghai Jiao Tong University(SJTU), China, in 2020. Her research interest is the internet of things in agriculture.

Yixiang Huang (M'14) received the B.S. degree in power and energy engineering in 2002, the M.S. and Ph.D. degrees in mechatronics engineering in 2006 and 2010, respectively, all from Shanghai Jiao Tong University, Shanghai, China. Currently, he is an Associate Professor of Mechanical Engineering at Shanghai Jiao Tong University, China, where he studies the topics of intelligent maintenance, prognostics and health management(PHM), edge computing and machine learning. Previously, he worked with the NSF Industry/University Cooperative Research Center for Intelligent Maintenance Systems at University of Cincinnati, OH, USA. His current research focuses on big data analysis, sparse coding and dimensionality reduction for industrial applications and computational intelligence techniques and their applications in various industrial domains. He is a regular reviewer for a number of international journals.

Wei Wu received a M.Eng degree from school of mechanical engineering, Shanghai Jiao Tong University(SJTU), China, in 2020. His research interests are agricultural machine vision, database technology and information systems research.

Cheng-liang Liu received his Ph.D. in Mechanical Engineering from Southeast University (China), 1999. He came to Institute of Robotics in School of Mechanical Engineering, Shanghai Jiao Tong University and joined the faculty as an Assistant Professor later the same year. In the year of 2000, he was invited to work at University of Cincinnati and University of Wisconsin respectively as a senior visiting scholar. He was promoted to Full Professor in 2002, taking charge of Institute of Mechatronics. He teaches undergraduate- and graduate-level subjects related to mechanics and electronics. His research interests range from intelligent robot systems, MEMS application to precise agriculture, network-based monitoring, and GPS/GIS/RS-equipped apparatus/machinery. To his filed, he is a senior member and board chairperson of Chinese Mechanical Engineering Society, elected in 2001. He has served as a planning expert for China's National 14th five-year plan of National High-Tech Research and Development Program in the field of modern agriculture technology.