An Efficient IoT-based Smart Water Meter System of Smart City Environment

Raad AL-Madhrahi[1], Jiwa Abdullah[3]
Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

Nayef.A.M. Alduais[2]*, Hairulnizam B. Mahdin[4]
Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia
Johor, Malaysia

Abdullah B. Nasser[6]
Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, 26600
Pekan, Pahang, Malaysia

Abdul-Malik H. Y. Saad[5]
Division of Electronic and Computer Engineering, School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 JB, Johor, Malaysia

Husam Saleh Alduais[7]
Department of Architecture, Faculty of Built Environment, University of Malaya, 50603 Kuala Lumpur
Malaysia

Abstract*—Water is a precious need of our lives. Due to the rapid population and urbanization, water usage monitoring is a significant problem facing our society. One solution is to control, analyze, and reduce the water consumption of the houses. The emerging of the Internet of Things (IoT) concept lately in our lives has offered the opportunity to establish water usage-efficient smart devices, systems and applications for buildings and cities. Many studies have suggested designing an IoT-based smart meter system; however, the IoT sensor node has limited studies, especially in battery life. Therefore, this study aims to implement and analyze an efficient data collection algorithm for IoT-based smart metering applications in consideration with energy consumption. The system items used are Arduino Uno, Wi-Fi-ESP8266, and water flow sensors. The applied algorithm is an efficient data collection algorithm for water meter (EDCDWM) to reduce the number of packet transmissions. Arduino performed this system's implementation, while the simulation and analysis performed by MATLAB R2019b. The average percentage of energy saved by the applied algorithms of EDCDWM absolute change; and EDCDWM with relative differences in all nodes are around 60% and 93%, respectively.

Keywords*—Internet of things; smart water metering; energy consumption; smart city

I. Introduction

In many countries, water conservation is becoming increasingly necessary as countries face a widening gap between the ever-decreasing water availability due to climate change and the rising demand for population growth. Water efficiency implies less water consumption and searching for an alternative of conventional water meters to measure the quantity and quality of water. Water utilities build daily demand profiles and peaking factors to construct water delivery network infrastructure [1]. The role of smart metering is increasingly recognized by water utilities in demand management, customer service, work optimization and operating efficiency [2]. Today's advanced programs include water-efficient sensors and innovations such as the Internet of Things (IoT) based smart meters are increasingly highlighted [3]. There are significant advances in optimizing water-intensive processes and controlling activities where automatic leak detection and monitoring systems permit to locate and cut off the leaks flow automatically and even patch the leaks [4].

IoT is defined as a system where physical objects can become active participants, and resources can be accessed over the internet through the cloud to communicate with these objects [5][6]. It allows devices and networks to connect, share and store data with or without human interference on the cloud platform using the internet. To send data from a device to a cloud, it is required to add a new device on the cloud and get the device credentials which are the username and device ID. These credentials are used to set-up a connection between Wi-Fi and the cloud to send data to the cloud. After identifying the Wi-Fi in the cloud, it can receive the transmitted data by the Wi-Fi. The smart meter water usage offers a solution to overcome water consumption issues and monitors the amount of water used by each household or building to control their consumption level. Random consumption problems should be avoided when installing this smart water meter [7][8], and the water consumed can be monitored using the internet. The water supply may be terminated if people are not present in their homes, and water usage is decreased directly or indirectly [9].

This study presents an analysis and implementation of a smart water meter that uses an embedded device. Arduino microcontroller and Wi-Fi are used to introduce the 'Smart' feature in a traditional domestic water meter. The IoT-based smart water meter built in this research allows reading the meter without physically entering each house periodically. This was accomplished using the Arduino unit, which continuously monitors and records the water flow sensor's readings in its memory. The meter is connected to the internet using Wi-Fi, making the device part of the IoT. The user and

*Corresponding Author

the service provider would be able to display and read the consumed water along using this smart water meter. Via such a comprehensive record of events, the service provider can keep track of the trend of water use, allowing better and efficient forecasting and management of use. On the other hand, the payment system will become more transparent for clients.

The rest of the paper is organized as follows: Section II presents the motivation of this study and the main contributions. The related works are presented in Section III. Section IV describes the system design of our proposed approach. The proposed algorithm, experimental results and simulation are explained in Section V. Finally, Section VI presents the conclusion of the paper.

II. Motivation

As a result of rapid population and urbanization, water demand is rising for various purposes such as agriculture, industries, homes, hospitals, etc. Furthermore, the world is looking forward increasingly to adapting and introducing modern technology to enhance the quality of life and reduce the impact on the environment of human activities and patterns of consumption. However, rapid changes in lifestyle and increased paying capacity have had an effect on the usage of water and associated overheads on sewerage requirements [10][11]. Hence there is an urgent need to conserve as much water as possible with the integration of technology. Therefore, water usage-efficient operation requires water usage analysis constructively based on data collected from many applications. This study involves collecting sufficient analytical data, which requires the acquisition and transfer of data from multiple sources in real-time, high processing power and storage of massive data sets. The current system provides a unique twist to the smart water meters using IoT technology, where the smart meter data is stored in the cloud. Nevertheless, the IoT sensor node has limited resources, especially in battery life. Therefore, this project aims to reduce the energy consumption during the data sending process to the cloud, extending the sensor's node battery lifetime by implementing and analyzing an efficient data collection algorithm for IoT-based smart water metering application.

For a more high-efficient operation of water-usage, we have designed a smart water meter using IoT technology. As IoT-based devices have limited power resources and rely on batteries, therefore we proposed an efficient data collection algorithm to reduce the data transmission and thus we can extend the battery life. For evaluation, the developed system has been tested in many scenarios.

The main contributions of this research are as follows:

- Design an IoT-based smart water meter system.
- Implement an efficient data collection algorithm for IoT based smart water metering applications.
- Evaluate the performance of the data collection algorithm in terms of energy savings and the amount of data transmission.

III. Related Work

Many researchers have investigated smart water meters systems for efficient control of water in the built environment (Table I). In [12], the authors suggested an integrated (IoT) infrastructure for smart meter networks in smart cities. The proposed design used communication protocol, data format, data collection procedures, and decision system based on big data processing. Furthermore, in [13] the authors suggested a method to reduce the number of transmitted packets in order to extend the sensor’s node battery lifetime. So, the suggested method supports single and multiple sensors. In [14], the authors designed and implemented an innovative smart energy meter based on the internet of things. This system uses the IoT power meter, installed at the top of the traditional consumer power meter, which provides detailed electrical energy use information. Moreover, in [15] the authors studied various algorithms' performance, which reduces the number of transmitted packets through Wireless Sensor Network (WSN) by the nodes attached to it. The algorithms were tested by applying them to data from temperature, relative humidity, and light sensors. The data produced by these three types of sensors were collected by Intel Berkeley Research Laboratory (IBRL). In [16], the authors proposed a power dissipation model to determine fault detection cost at the edge device level. The proposed model was used to assess the impact of various data validation schemes on the power consumption of edge devices. The analysis revealed that defining a data validation scheme at the edge device level was a critical issue. In [17], the authors designed a smart energy meter platform and their system implemented and built included clouds, smart plugs, and gateways. The system was applied in industrial centers and buildings to sense energy consumption and the power system's parameters. Furthermore, [18] proposed implementing a smart energy meter with a very low-cost wireless sensor network and web application protocol that can read the units and send the measured data to users automatically to display their current reading of a smart energy meter. The system consists of a digital energy meter, ESP8266 Wi-Fi module, and web applications for the management system. In [19], the authors proposed a system that monitors energy consumption using the IoT. The system consists of an Arduino Uno board as a microcontroller that communicates through Ethernet using the IP address on their computers to receive information about energy usage. In [20], the proposed system designed by authors was able to read the water consumption and send the consumption to customers. The water meter's primary function is to secure, regulate, and track water supply. The system was designed from Microcontroller (ATmega328), GSM Module (SIM800), LCD Module (1602A), and Step-down Transformer. In [21], the authors proposed mobile applications using low-cost IoT hardware for smart metering.

TABLE I. SUMMARY OF THE RELATED WORK

Ref.	Application	Implementation	Sensor	Architecture	Update data strategy	Energy Consumption Analysis
[12]	Energy, Water, Gas	Simulation Spark SQL as a database system	Electricity, water, Gas	IoT	No	Yes
[13]	Environment	Simulation MATLAB	Humidity, Temperature, Atmosphere Pressure, Carbon Dioxide, Libelium Gas	IoT/WSN	Aggregation, EDCD algorithms	Yes
[14]	Energy	Prototype Serial port, opto-coupler MCT2E	Current	IoT	No	No
[15]	Environment	Simulation MATLAB	Temperature, Relative Humidity, Light	WSN	AR-B, MA, AR-YW, EDCD1	Yes
[16]	Environment	Simulation MATLAB	Current, VSNL	IoT/WSN	ER-NN, ER-HD, ED-NN	Yes
[17]	Energy	Prototype Arduino Uno	Current, voltage	IoT	No	No
[18]	Energy	Prototype ADSL modem	Power, Temperature, Humidity, Brightness, Gas leakage	IoT	No	No
[19]	Energy	Prototype Arduino Uno	Current	IoT	No	No
[20]	Water	Prototype Microcontroller (ATmega328),Proteus	Flow Sensor (Hall-Effect)	GSM	No	No
[21]	Water	Prototype Arduino SBC, Ethernet Stack,Flash Memory,LCD	Tamper-flag	IoT	No	No
[22]	Water	Prototype NodeMCU	YF-S201 water flow sensor	IoT	Yes	No
[16]	Water	Prototype LinkIt ONE development board	ultrasonic flow sensor, PT1000 temperature sensors	MCS, IoT	Yes	No

The system allowed both the Meter Reader and individual domestic/industrial customers to use standard smartphones to read and update meters to the utility's billing and payment portal/database. The proposed Smart Meter System was composed of the Electronic Interface Module (EIM) hardware component, which resides in conjunction with the basic water meter. In [22], the authors designed a smart water metering that curbed water pollution. The proposed system has been used combined with machine learning-based tools to identify excess water consumption with simple monitoring and visualization of the data via the cloud platform, where the system adopted the server-less architecture. In [23], the authors suggested the architectural system for the IoT based water meter. The Mediatek Server Sandbox was used as a cloud platform in the proposed framework for the cloud to interact with the water meter. The suggested framework used RESTful web services to communicate with the water meter and the IoT cloud and had the potential to analyze data at lower costs.

IV. SYSTEM DESIGN

In this paper, the proposed system aims to design an IoT-based smart water meter to monitor water consumption, taking into account the energy consumption of IoT sensor nodes. Fig. 1 shows the general structure of the proposed system.

The proposed system is designed by integrating a smart meter and Arduino Uno into the ESP8266 Wi-Fi module for various scenarios. The water flow sensor YF201B is used to measure water consumption. The key idea is that when water flows through the sensor, the magnetic rotor will rotate, and the rotation rate of the rotor will vary with the flow rate. The sensed data will be sent to the IoT-Fusion center via ESP8266 Wi-Fi. In addition, an effective water meter data collection (EDCDWM) algorithm has been applied to reduce the amount of data transmission. This process can reduce the energy consumption of IoT nodes. This is because the battery life of an IoT node is affected by the number of transmitted data packets. The EDCDWM algorithm has been described in detail in the next section.

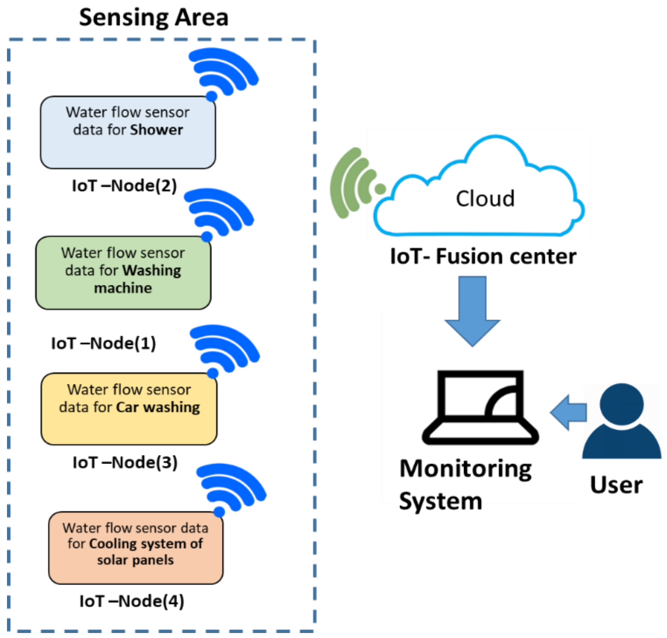
Sensing Area
Water flow sensor data for Shower
IoT –Node(2)
Water flow sensor data for Washing machine
IoT –Node(1)
Water flow sensor data for Car washing
IoT –Node(3)
Water flow sensor data for Cooling system of solar panels
IoT –Node(4)
Cloud
IoT- Fusion center
Monitoring System
User

Fig. 1. System Structure.

V. EDCDWC Algorithm

In the section, we explained the proposed EDCDWC algorithm to reduce the number of transmissions. The proposed algorithm EDCDWC used in this study is described in the following pseudocode where it was deduced based on the EDCD algorithm [6]. It works based on the absolute change in the total quantity of water usage measured by the water flow sensor.

```
// SetupPhase //
// Setup phase: In this phase, the IoT sensor board transmits one sample only.
Set $TM(t-1) \leftarrow 0$ // previous TotalMilliLitre
Call Algorithm1- MWF
$TM(t) \leftarrow MWF$
Send TM(t)  TO FC
Set $TM(t-1) \leftarrow TM(t)$
END
```

```
Algorithm1: Measured Water flow (MWF)
Inputs: $T_p, T_c, WF, TM(t-1)$
Output: $TM(t)$ //  TotalMilliLitres in current time (t)
Begin:
Set  CF = 6.8 // CF is sensor calibration factor
Read WF //water flow sensor value
If  WF  == 1 // water flow
    Set $T_D = T_c - T_p$   // T_p is time of previous scan (ms), T_c is current time (ms)
    Calculate $FR = \left(\frac{1000}{P_c * T_D}\right) * CF$ // FR is Flow Rate
    $FM = \left(\frac{FR}{60}\right) * 1000$  // FM is flowMilliLitres
    Set $TM(t) = TM(t-1) + FM$
end if
```

```
// EDCDWM-AC Algorithm//
Inputs: $Thr, TM(t-1),\ TM(t)$
Output:  T //  T=On/OFF transmite the data or not
Begin:
 Call SetupPhase // call setup phase for one time only
// Reduction phase//
Call Algorithm1- MWF
$TM(t) \leftarrow MWF$
 Compute Absolute change ($TM(t-1),\ TM(t)$ )
$AC \leftarrow Abs(TM(t) - TM(t-1))$
$If\ AC > Thr\ Then$
    Set $T \leftarrow 1$
Else:
    Set $T \leftarrow 0$
End if
// The decision to Transmit data
If $T = 1$ Then
    $Send\ \ TM(t)$ // transmit (ON) and send data to FC
    Set  $TM(t-1) \leftarrow TM(t)$
Else
    // Transmit (Off) // No
End If
Wait // and Call // Reduction phase//
End Algorithm
```

```
// EDCDWM-RD Algorithm//
Inputs: $Thr, TM(t-1),\ TM(t)$
Output:  T //  T=On/OFF transmite the data or not
Begin:
 Call SetupPhase // call setup phase for one time only
// Reduction phase//
Call Algorithm1- MWF
$TM(t) \leftarrow MWF$
 Compute Relative difference ($TM(t-1),\ TM(t)$ )
$AC \leftarrow Abs(TM(t) - TM(t-1))$
$RF \leftarrow \frac{AC}{(\ TM(t)+TM(t-1))\times 0.5)}$
$If\ RF > Thr\ Then$
    Set $T \leftarrow 1$
Else:
    Set $T \leftarrow 0$
End if
// The decision to Transmit data
If $T = 1$ Then
    $Send\ \ TM(t)$ // transmit (ON) and  send data to FC
    Set  $TM(t-1) \leftarrow TM(t)$
Else
    // Transmit (Off) // No
End If
Wait // and Call // Reduction phase//
End Algorithm
```

A. Validity and Reliability of the Sensed Data

In this study, a water flow sensor is connected to the pipe to measure the movement into the pipe, and the Hall effect of the water flow sensor converts the movement of water into pulses. If the valve of the pipeline is closed, the sensed output is zero (no pulse), but when the valve of the water is opened, the sensor converts the motion of the Hall effect into pulses and then converts it into a measured value. In addition, the calibration coefficient is used to represent the output of the Hall-effect flow sensor. As shown in Table II, this factor is adjusted for correct reading during the working period.

As the loop may not complete in exactly 1-second intervals, the number of milliseconds that have passed since the last execution is calculated to scale the output. The flow rate is also calculated based on the applied calibration factor, which scales the output of flow rate (FR) based on the number of pulses per second per unit of measure as shown in Eq. (1). In order to calculate the current flow of water, when the water stops flowing through the sensor, the time in this case will not increase. Accordingly, the current time of the water flow is equal to the time before the water flow stopped.

$$FR = \frac{(\ 1000/CT-TP)\times CP)}{SCF} \qquad (1)$$

TABLE II. Adjustment of Calibration Factor

Calibration factor	Actual value (ml)	Measured value (ml)
7.5	1500	1329
7.0	1500	1428
6.8	1500	1508
6.5	1500	1526
6.0	1500	1647
5.5	1500	1915

where, CT is the current time (ms), TP is the time of previous scan (ms), CP is the current pulse count and SCF is the sensor calibration factor.

Equation (2) is used to calculate how many milliliters have passed through the sensor each one second by dividing the flow rate (liters/minute) by 60, then multiplying by 1000. Thus, the final value of consumption can be obtained at any time. Equation (3) is used to calculate the total milliliters flowing through the sensor. Table III shows the initial values of the parameters of the water flow sensor.

$$FlowMilliLitres = \left(\frac{flowRate}{60}\right) \times 1000 \tag{2}$$

$$TotalMilliLitres = CurrentMillilitres + PreviousMilliLitres \tag{3}$$

TABLE III. PARAMETERS OF THE WATER FLOW SENSOR

Parameter	Initial value
Calibration factor	6.8
Byte pulse Count	0
Flow Rate	0.0
Flow Millilitres	0
Total Millilitres	0
old Time	0

B. Testing of Sensor

The sensor was tested by using a bottle of water with 1500ml capacity. Fig. 2 shows the testing process, where the bottle was filled four times. As shown in Fig. 2, the system consists of Arduino Uno, water flow sensor, a bottle of water, SD card memory, and a battery. These components have been connected to the system. During this process, the sensed values are 1511ml, 1491ml, 1500ml, and 1508ml, with an accuracy of up to 99% as shown in Table IV.

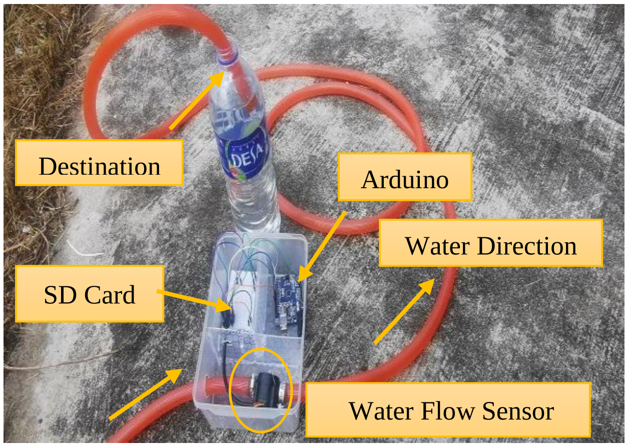

Fig. 2. Testing of Sensor.

TABLE IV. TESTING EXPERIMENTS

Experiment	Actual value (ml)	Measured value (ml)	Accuracy
E1	1500	1511	99%
E2	1500	1491	99%
E3	1500	1500	100%
E4	1500	1508	99%

C. Data Collection

To investigate the various approaches; absolute change, relative change and relative differences; the proposed algorithm was applied separately for the collected data from the washing machine, shower, car-washing, and cooling system scenarios as following:

- Scenario (1) - washing machine: The total number of transmissions is 2965. Fig. 3 shows the experiment of collecting data from the washing machine, while Fig. 4 presents the graph of collected data.
- Scenario (2) - shower: The total number of transmitted samples is 592. Fig. 5 shows the experiment of collecting data from the shower, while Fig. 6 presents the graph of collected data.

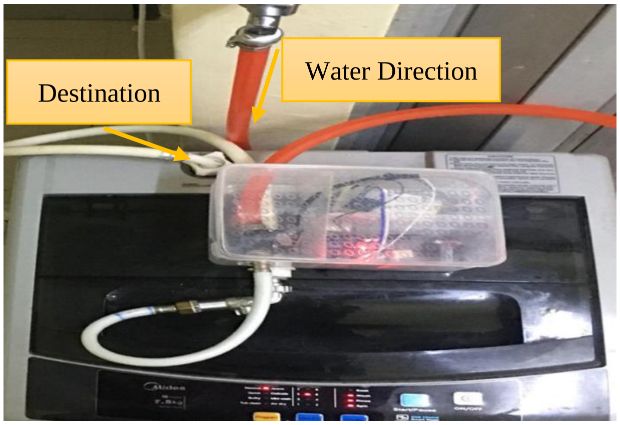

Fig. 3. Collection Data from Washing Machine.

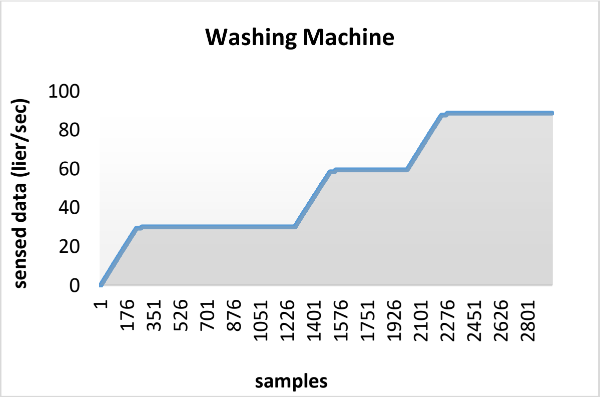

Fig. 4. Data of Washing Machine.

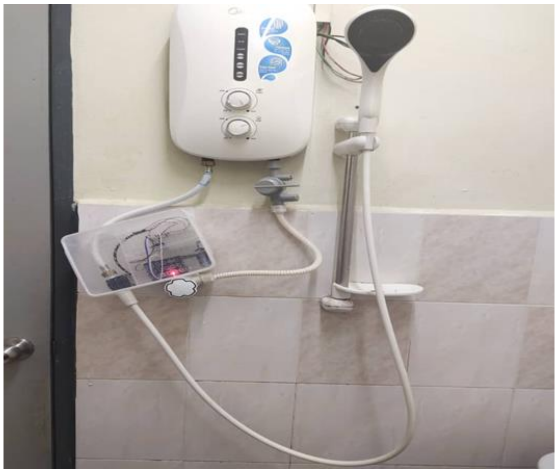

Fig. 5. Collection Data from Shower.

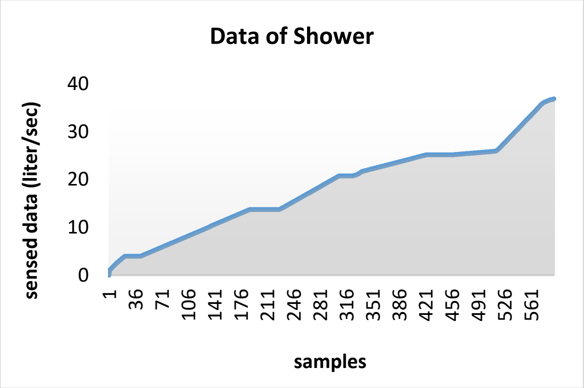

Fig. 6. Data of Shower.

- Scenario (3) - car washing-: The total number of transmitted samples is 1793. Fig. 7 shows the experiment of collecting data from car washing, while Fig. 8 presents the collected data.

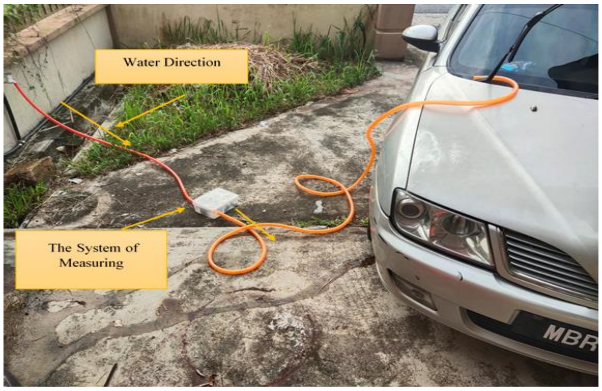

Fig. 7. Collection Data from Car Washing.

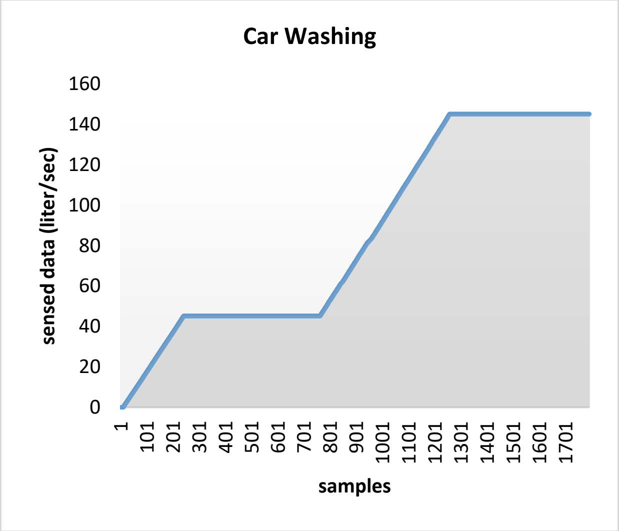

Fig. 8. Data of Car Washing.

- Scenario (4) - cooling system: The total number of transmitted samples is 14565. Fig. 9 shows the experiment of collecting data from the cooling system, while Fig. 10 presents the collected data.

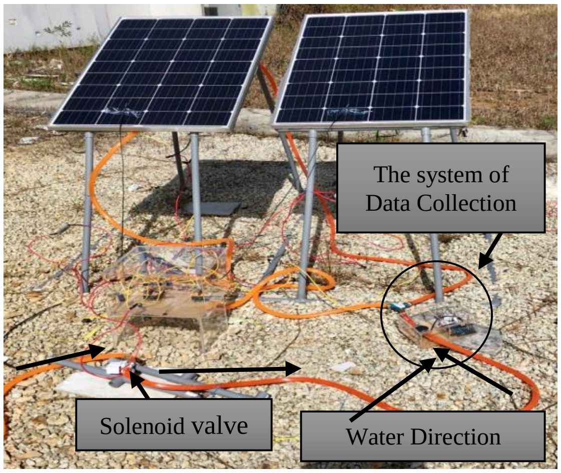

Fig. 9. Collection Data of Cooling System.

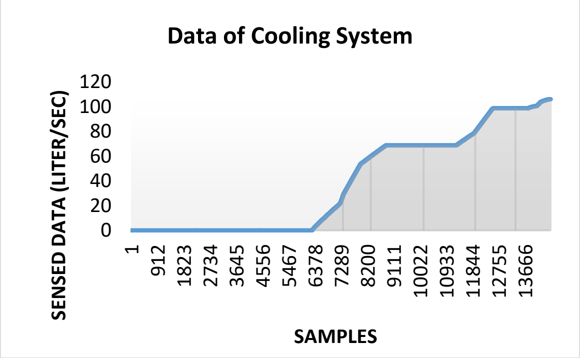

Fig. 10. Data of Cooling System.

D. Performance Evaluation of EDCDWM Algorithm

The EDCDWM algorithm with the applied threshold has been tested in various scenarios. In node (1), the data samples collected from the washing machine are 2965 samples. For the Node (2) and (3) (i.e. for shower and car washing scenarios, respectively): the data samples are 592 and 1793 samples, respectively. For the last node, i.e. Node (4), the data samples collected from the cooling system for solar panels are 14565 samples.

The algorithm was evaluated in terms of the number of transmitted data and energy consumption reduction during the transmission process. The experiment parameters are shown in Table V, where the applied threshold *Thr* is 0.03, and the cost of transmitting one byte is 59.2 µJ.

TABLE V. EXPERIMENT PARAMETERS

Parameter	Value
Thr	0.03
Algorithm	EDCDWM
Sensor	Water flow
Cost of transmitting one byte	59.2µJ
Size of one sample	4 bytes
Cost of transmitting one bit	59.2/8 µJ

Fig. 11 shows the comparison between three ways to send data from water flow sensors to the cloud, direct transmission and transmission by EDCDWM-AS and EDCDWM-RD algorithms. As it noticed that the transmitted processes were from four nodes for different scenarios. Fig. 10 shows the amount of data transmitted by different nodes/scenarios. The graph shows that the number of transmitted data directly was 2965 samples for node (1). In comparison, it was about 717 samples by using EDCDWM-AS algorithm, while the number of packet transmission was decreased sharply to reach 122 samples by applied EDCDWM-RD algorithm. For node (2), the number of transmitted samples reduced by applying the EDCGWM-AS algorithm is 150 samples. The number of transmissions has changed from 592 samples directly sent to 442 samples, and the transmitted data is reduced to 88 samples transmitted through the EDCWDWM-RD algorithm. For node (3), it is clear that the number of packets transmitted decreases by the applied EDCDWM-RD algorithm compared with the EDCDWM-AS and direct-transmission. The number of packets transmitted is 126, 732 and 1793 samples for EDCDWM-RD, EDCDWM-RD and direct-transmission. Respectively. Finally, for node (4), the number of packets transmitted is 177, 11813 and 14565 samples for EDCDWM-RD, EDCDWM-RD and direct-transmission, respectively.

Fig. 12 represents the energy consumption for the tested scenarios with the three transmission algorithms. From the figure, it is clear that EDCDWM-AS and EDCDWM-RD transmission schemes were able to reduce the energy consumption significantly compared with the direct transmission scheme. For example, EDCDWM-AS was able to reduce the energy consumption for node (1) by around 76%. In contrast, EDCDWM-RD was able to save energy by 96%. For the node (2) the consumption of energy is less than other nodes, and about 25% saving in energy was achieved by EDCDWM-AS, and 85% achieved by EDCDWM-RD. In node (3), the percentage of energy consumption was about 59% for EDCDWM-AS, and around 93% for EDCDWM-RD. The consumption of energy was decreased sharply in node (4), where about 81% reduction is obtained by EDCDWM-AS, and around 99% by EDCDWM-RD. Overall, from these shown experimental results, it is very clear that decreasing the amount of packet transmission results in a reduction in energy consumption as well as extending the meter battery life. Although EDCDWM-based schemes reduced data transmission, it did not affect the calculation of the actual water consumption value.

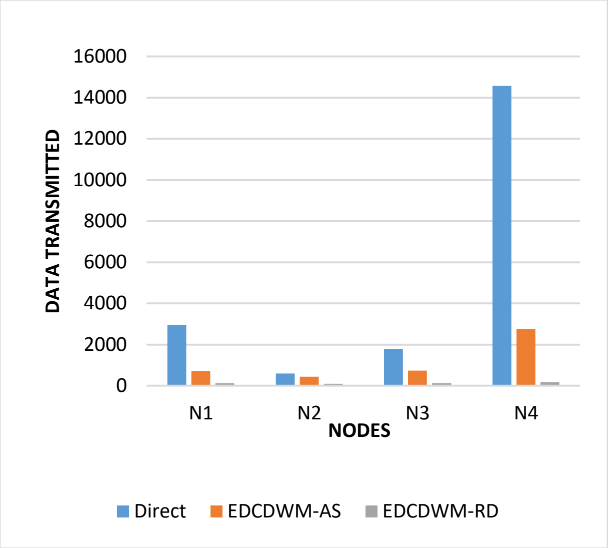

Fig. 11. Transmitted Data.

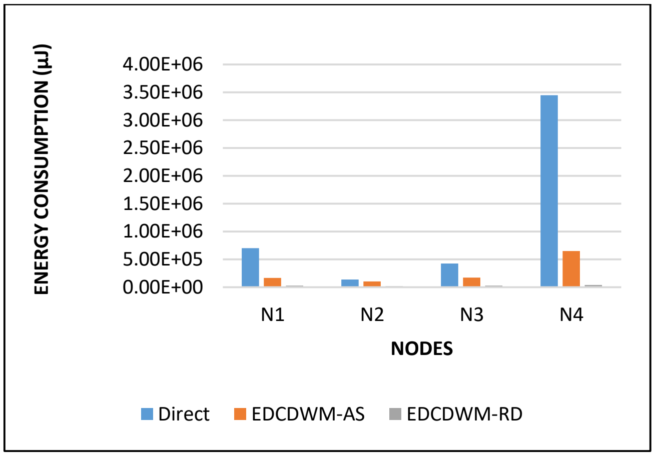

Fig. 12. Energy Consumption.

Fig. 13 provides the percentage of data transmitted for various sensor nodes to the cloud using EDCDWM-AS and EDCDWM-RD algorithms. The data reduction rates achieved by applying EDCDWM-RD to N1, N2, N3, and N4 are 96%, 85%, 93%, and 99%, respectively. In contrast, applying EDCDWM-AS to N1, N2, N3, and N4 achieved data reduction rates of 76%, 25%, 59%, and 81%, respectively. It can be seen from the results that EDCDWM-RD has better performance than EDCDWM-AS in terms of the data reduction rate of all nodes.

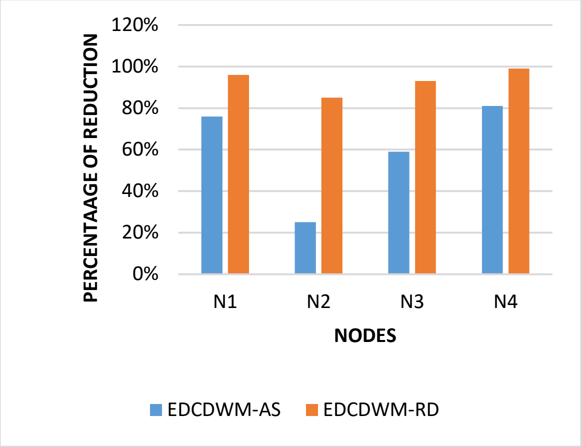

Fig. 13. The Percentage of Data Reduction.

E. System Implementation

The smart water meter is designed using NodeMCU with a Wi-Fi module and a water flow sensor as shown in Fig. 14. When the water passes through the sensor, the Wi-Fi is sending the value of flow rate and the total quantity of water to the Cloud platform, which is in this research Thinger.IO platform. Besides, two Wi-Fi modules have been used. The first one is used to send the measured values directly while the other one sends the measured values based on the EDCDWM schemes.

Fig. 15 shows the transmitted data to the cloud platform directly as one packet per second. In contrast, Fig. 16 shows the transmitted data using EDCDWM. In summary, it is clear that the applied algorithm is able to reduce the number of transmissions without any effect on the total quantity of the water usage.

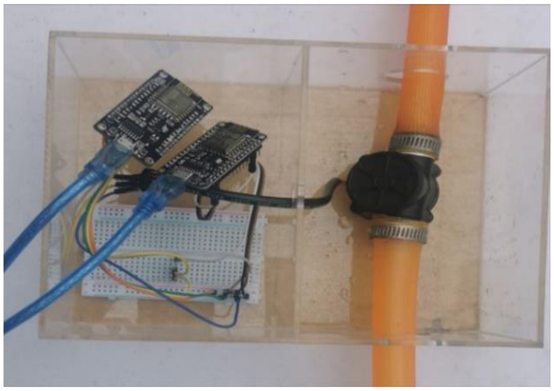

Fig. 14. Smart Water Meter based IoT.

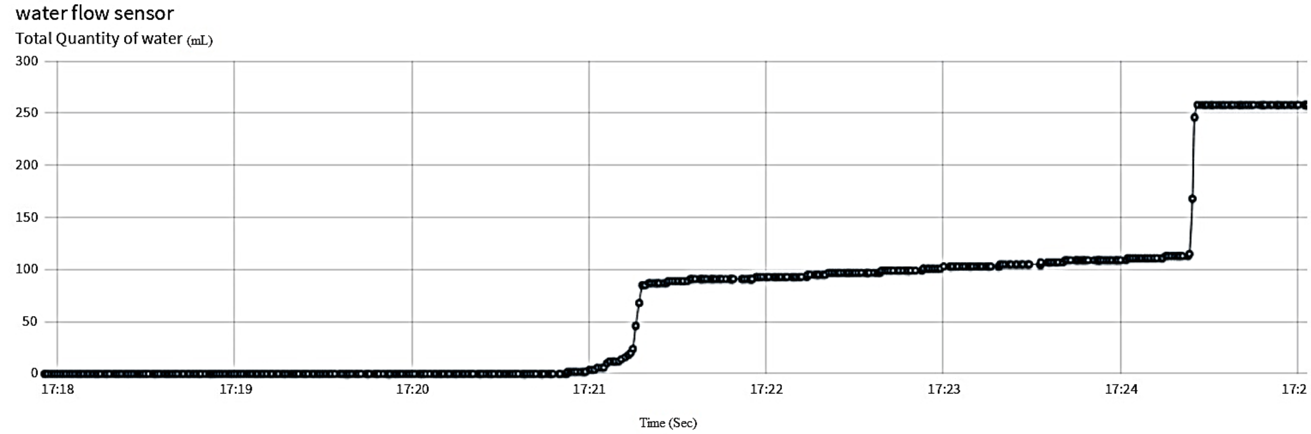

Fig. 15. Transmitted Data Directly.

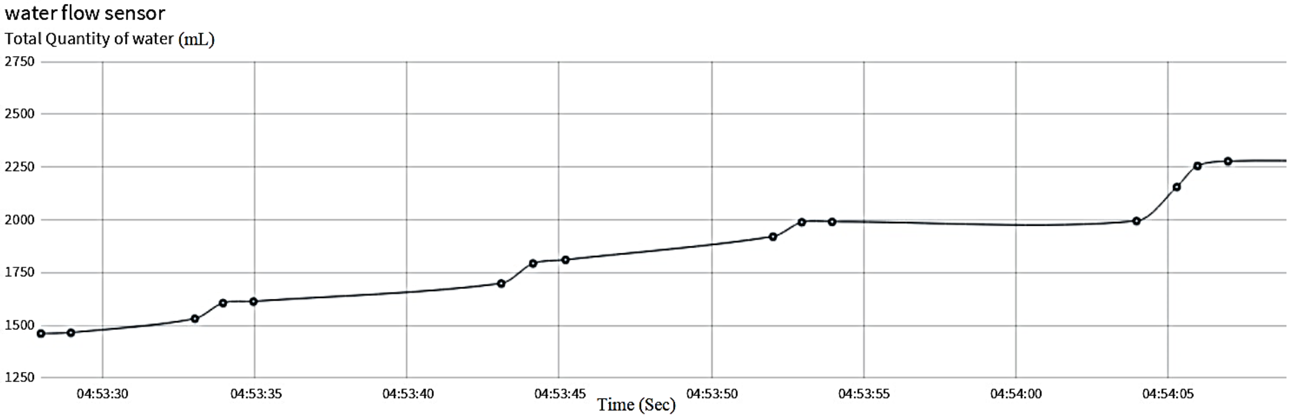

Fig. 16. Transmitted Data via EDCDWM.

VI. Conclusion

In this paper, an energy-efficient IoT-based smart water meter has been developed and tested with different scenarios. Energy saving was achieved through reducing the amount of data collection and transmission using the EDCDWM scheme. The performance of this applied data-collection scheme has been measured. From the experimental results, it is clear that the applied data-collection scheme is effective in reducing energy consumption by reducing the number of transmissions. The percentage of energy saved by using the EDCDWM-AS scheme in the nodes (1), (2), (3), and (4) are 76%, 25%, 59%, and 81%, respectively, with an average of 60% energy saving. The EDCDWM-RD scheme shows better results for the same nodes, where the percentages of energy saved for the nodes (1) - (4) are 96%, 85%, 93%, and 99%, respectively with an average of 93%.

As a future work, we will connect the smart meter with a valve to control the water flow when there is a leak. Furthermore, the system software will be improved to add more features, such as giving the users the ability to reset their recorded usages of the water in the cloud and paying their bills automatically.

Acknowledgment

“The authors would like to express their thanks to Universiti Tun Hussein Onn Malaysia (UTHM) for support. This research is supported by Universiti Tun Hussein Onn Malaysia (UTHM) through Tier 1 (vot H789).”

References

[1] Gurung, T.R., et al., Smart meters for enhanced water supply network modelling and infrastructure planning. Resources, Conservation and Recycling, 2014. 90: p. 34-50.

[2] Beal, C.D. and J. Flynn, Toward the digital water age: Survey and case studies of Australian water utility smart-metering programs. Utilities Policy, 2015. 32: p. 29-37.

[3] Alduais, N. A. M., Abdullah, N., Abdullah, J., Jamil, A., & Saad, A. M. H. (2020, April). Implementation and analysis of an updating data strategy on IoT-Waspmote Gases Testbed with 2.4 GHz XBee for Air Quality Application. In 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 268-272). IEEE.

[4] Haie, N. (2020). Transparent Water Management Theory (pp. 39-70). Springer Nature Singapore Pte Ltd.: Singapore.

[5] Alduais, N. A. M., Abdullah, I., & Jamil, A. (2018, October). An efficient data collection algorithm for wearable/mobile tracking system in IoT/WSN. In 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS) (pp. 250-254). IEEE.

[6] Abdul-Qawy, A. S., & Srinivasulu, T. (2018, January). Greening trends in energy-efficiency of IoT-based heterogeneous wireless nodes. In International Conference on Electrical, Electronics, Computers, Communication, Mechanical and Computing (EECCMC) (pp. 118-427).

[7] Marais, J., Malekian, R., Ye, N., & Wang, R. (2016). A review of the topologies used in smart water meter networks: A wireless sensor network application. Journal of Sensors, 2016.

[8] Fuentes, H., & Mauricio, D. (2020). Smart water consumption measurement system for houses using IoT and cloud computing. Environmental Monitoring and Assessment, 192(9), 1-16.

[9] D. Anandhavalli, K. S. Sangeetha, V. P. Dharshini, and B. L. Fathima, “Smart Meter for Water Utilization using IoT,” pp. 4–7, 2018.

[10] Suresh, M., Muthukumar, U., & Chandapillai, J. (2017, July). A novel smart water-meter based on IoT and smartphone app for city distribution management. In 2017 IEEE region 10 symposium (TENSYMP) (pp. 1-5). IEEE.

[11] Sundresh, H. D., & Priya, D. (2020). Design and Integrate IoT Sensors to RO Water Purifiers for Remote Monitoring and Allowing Users to Pay Per Usage on the Rented RO System. In Inventive Communication and Computational Technologies (pp. 647-651). Springer, Singapore.

[12] J. Lloret, J. Tomas, A. Canovas, and L. Parra, “An Integrated IoT Architecture for Smart Metering,” IEEE Commun. Mag., vol. 54, no. 12, pp. 50–57, 2016, doi: 10.1109/MCOM.2016.1600647CM.

[13] N. A. M. Alduais, J. Abdullah, A. Jamil, and L. Audah, “An efficient data collection and dissemination for IOT based WSN,” 7th IEEE Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEEE IEMCON 2016, 2016, doi: 10.1109/IEMCON.2016.7746084.

[14] S. Saha, S. Mondal, A. Saha, and P. Purkait, “Design and implementation of IoT based smart energy meter,” Proc. 2018 IEEE Appl. Signal Process. Conf. ASPCON 2018, pp. 19–23, 2018, doi: 10.1109/ASPCON.2018.8748696.

[15] M. I. Husni, M. K. Hussein, N. A. M. Alduais, J. Abdullah, and I. Marghescu, “Performance of Various Algorithms to Reduce the Number of Transmitted Packets by Sensor Nodes in Wireless Sensor Network,” Proc. 11th Int. Conf. Electron. Comput. Artif. Intell. ECAI 2019, 2019, doi: 10.1109/ECAI46879.2019.9042081.

[16] N. A. M. Alduais, J. Abdullah, A. Jamil, L. Audah, and R. Alias, “Effect of Data Validation Schemes on the Energy Consumptions of Edge Device in IoT /WSN,” 2018 14th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2018, pp. 77–81, 2018, doi: 10.1109/IWCMC.2018.8450460.

[17] M. H. Yaghmaee and H. Hejazi, “Design and Implementation of an Internet of Things Based Smart Energy Metering,” 2018 6th IEEE Int. Conf. Smart Energy Grid Eng. SEGE 2018, pp. 191–194, 2018, doi: 10.1109/SEGE.2018.8499458.

[18] W. Hlaing, S. Thepphaeng, V. Nontaboot, N. Tangsunantham, T. Sangsuwan, and C. Pira, “Implementation of WiFi-Based single phase smart meter for internet of things (IoT),” 2017 Int. Electr. Eng. Congr. iEECON 2017, no. March, pp. 8–10, 2017, doi: 10.1109/IEECON.2017.8075793.

[19] M. Kaur, L. Mathew, A. Alokdeep, and A. Kumar, “Implementation of Smart Metering based on Internet of Things,” IOP Conf. Ser. Mater. Sci. Eng., vol. 331, no. 1, 2018, doi: 10.1088/1757-899X/331/1/012015.

[20] K. Shrotriya, M. Jain, M. Mittal, L. Yadav, and N. Vijay, “Digital Water Meter Using Arduino,” IJEMR Int. J. Eng. Manag. Res., vol. 7, no. 2, pp. 276–279, 2017.

[21] M. Suresh, U. Muthukumar, and J. Chandapillai, “A novel smart water-meter based on IoT and smartphone app for city distribution management,” TENSYMP 2017 - IEEE Int. Symp. Technol. Smart Cities, pp. 2–6, 2017, doi: 10.1109/TENCONSpring.2017.8070088.

[22] A.Ray and S. Goswami, “IoT and Cloud Computing based Smart Water Metering System,” 2020 Int. Conf. Power Electron. IoT Appl. Renew. Energy its Control. PARC 2020, pp. 308–313, 2020, doi: 10.1109/PARC49193.2020.236616.

[23] N. Cherukutota and S. Jadhav, “Architectural framework of smart water meter reading system in IoT environment,” Int. Conf. Commun. Signal Process. ICCSP 2016, vol. 400019, pp. 791–794, 2016, doi: 10.1109/ICCSP.2016.7754253.