Towards an IoT-based Water Quality Monitoring System with Brokerless Pub/Sub Architecture

Alif Akbar Pranata[1,2], Jae Min Lee[1], Dong Seong Kim[1]
[1]Department of IT Convergence Engineering, School of Electronic Engineering
Kumoh National Institute of Technology, Gumi, South Korea
[2]Department of Informatics, Faculty of Mathematics and Information Technology
Institut Teknologi Kalimantan, Balikpapan, Indonesia
Email: [1]{*alifakbar, ljmpaul, dskim*}*@kumoh.ac.kr*, [2]*alifakbar@itk.ac.id*

***Abstract*—This paper investigates a real-time water quality monitoring system by using a proposed brokerless publisher-subscriber (pub/sub) architecture framework. On the system, sensors sense the water measurement metrics, including temperature, pH, and dissolved oxygen level. All collected data are stored in a database and computed stochastically for further analysis on water quality. A complementary experiment compares the proposed pub/sub architecture and MQTT, a lightweight protocol on which IoT mostly uses, to show better performance of the proposed architecture in case of network latency and throughput for diverse message payload size, thus suggesting the future IoT implementation of the system. To complete the experiment, the relationship among temperature, pH, and dissolved oxygen is analyzed, and the experiment summarizes that water temperature is inversely proportional to pH and dissolved oxygen value.**

***Keywords*—Water quality; Wireless Sensors Networks; Internet of Things; Pub/sub architecture; Temperature; pH; Dissolved oxygen**

I. INTRODUCTION

According to the United Nations in 2005, an estimated 1.1 billion people worldwide lack clean water and 2.6 billion lack access to basic sanitation [1]. Hence, 2005-2015 was designated the International Decade for Action: "Water for Life". Clearly, the international community has recognized that water distribution must be carefully monitored and controlled in terms of water quality and quantity.

Water quality is the thought measure of the water suitability to be used for a particular routine based on selected physical, chemical, and other characteristics. To determine water quality, scientists first measure and analyze characteristics of the water such as temperature, dissolved mineral content, number of bacteria, and so on. Selected characteristics are then compared to numeric standards and guidelines to decide if the water is suitable for a particular use. These standards and guidelines for water quality are established to ensure that public water supplies are as safe as possible for designated uses such as drinking, recreation, or agricultural irrigation.

To maintain the water quality, recent trend of wireless technologies can be used to provide relatively inexpensive and continuous monitoring of water inside its container. They encompass wireless sensor networks (WSN) as an emerging communication technology, consisting of a large number of small sensors with low power transceivers. The deployment of WSN is easy, flexible, and inexpensive. The short-range wireless networking communication technology based on ZigBee protocol has lots of advantages, such as low power, low cost, short time-delay, and good reliability [2][3]. Due to all mentioned advantages, WSN can be an effective tool for gathering data in a variety of environments including water quality monitoring.

Fig. 1: MQTT architecture.

Motivated by those problems and challenges, this paper describes an implementation of a real-time water monitoring for the aforementioned purposes by measuring temperature, acidity (pH), and dissolved oxygen for determining the quality of water. The monitoring system alleviates today's emerging wireless communication technologies, including WSN and towards IoT by using pub/sub architecture. Sensors are placed underwater and the system then collects data from the sensors network nodes which communicate wirelessly using Zigbee. Data are stored in a central database as well as presented in real-time to users. Moreover, measurement data are computed stochastically, determining standard deviation and confidence interval, to analyze water quality during measurement time. For the pub/sub procedure, this paper proposes a brokerless pub/sub architecture to suggest the system development towards IoT application. This water monitoring implementation with pub/sub architecture contributes to the main paper uniqueness. Involving the proposed pub/sub procedure with wireless communication using ZigBee protocol, the evaluation shows better performance after being compared with a standard pub/sub MQTT architecture. Eventually, an additional experiment was conducted to study a relationship of water temperature, pH, and DO. It confirms that an increase in water temperature induces a decrease of water pH and DO level, showing a relevant conclusion to other related works [4][5].

The remaining parts of the paper are organized as follows. Section II relates this paper to other previous works. The detail of IoT-based water quality monitoring system is explained in section III, and followed by performance analysis of the experiment in section IV. Finally, section V concludes the paper.

II. Related works

Real-time water quality monitoring has been implemented widely by researchers. However in this paper, the monitoring system is developed uniquely with a brokerless pub/sub architecture. In addition, this paper proves the relationship between temperature, pH, and DO, which also has been investigated, mainly in physiochemistry and hydrology field.

A robust real-time water quality monitoring with smart sensors has been developed by Cloete *et al.* [6]. They used some physiochemical parameters: flow, temperature, pH, conductivity, and oxidation reduction potential. The system used a microcontroller which communicates wirelessly with some other nodes to circulate the data and present through a user interface. The system was also utilized by notification alarm to warn users when a specific parameter is at a perilous level. However, the system was not equipped with data logging to record the measurement history, which is solved in this paper. Other contributions on water quality monitoring have been made with various target applications and purposes [7][8][9].

In this era, application of water quality monitoring is mainly developed towards IoT, where every single device can do self-configuration and communicate without human intervention. To support IoT, such application is also equipped with pub/sub architecture for message exchange among nodes. Typically, pub/sub architecture alleviates a broker to be an intermediary device between publisher and subscriber, as in the work by Wong and Kerkez [10], which employed MQTT-supported web services for their water quality monitoring system. However, researchers innovate a brokerless pub/sub nowadays, removing the broker from the architecture to accelerate the pub/sub performance by securing the communication and enhancing the quality of service [11][12].

Lastly, some works related to physiochemistry and hydrology reveals that there is a link among water properties, *e.g.*, temperature, dissolved oxygen, and pH [4][5], which motivates this paper. For higher temperature, the tendency of pH and dissolved oxygen level is lower. This phenomenon happens because in higher temperature, there might be about 10^{-7} moles of H+ per liter of water and this yields in the pH and oxygen level of water drop.

III. IoT-based Water Monitoring System

A. Pub/Sub Architecture

The aim of this work is to monitor water quality within a particular time and area by using a dedicated publish and subscribe communication approach. The intended pub/sub approach is brokerless and was designed to boost the system performance in regards to low end-to-end latency and high sensor throughput with message payload size variance. To grasp the idea of widely-known pub/sub architecture with a broker before deeply understand about the proposed approach, this section preliminarily describes the most popular approach in IoT application using MQTT in brief.

Message Queue Telemetry Transport (MQTT) is a lightweight, ISO-standard publish-subscribe protocol which is intentionally developed for open and simple device communication at a premium network bandwidth and/or small code footprint. It is agnostic to the payload content and support one-to-many message distribution as of the basic characteristic of pub/sub architecture. In MQTT, a broker runs an important role as the medium of message exchange among publishers and subscribers. It collects topics of which particular subscribers are interested in, notifies publishers to send the interesting topics, and replies the topic messages to the intended subscribers. The pictorial explanation of MQTT architecture is provided in figure 1, which briefly denotes three simple processes: publish, subscribe, and message.

Algorithm 1 The Subscribe Algorithm

Input:
- Topic definition: Temp, pH, DO,
- Array of topics in subscription list: $topics[] \leftarrow \{\}$,
- Message topic header containing interesting topic.

Output:
- Stored all interesting topics in subscription list.

procedure SUBCRIBE()
  if $!topics[topic]$ **then**
    $topics[topic] \leftarrow NULL$;
  end if
  $topics[topic] \leftarrow$ the topic header and content;
end procedure

Algorithm 2 The Publish Algorithm

Input:
- Topic in array of topics at subscription list: $topics[topic]$,
- Message which is about to be published: $message$,
- Message length $\leftarrow l$.

Output:
- Published message for all requested topics.

procedure PUBLISH()
  if $!topics[topic]$ **then**
    return $false$;
  end if
  if $message == topics[topic]$ **then**
    $l \leftarrow topics[topic].length()$;
  else
    $l \leftarrow 0$;
  end if
  while l **do**
    Publish $message$ for the requested topic;
  end while
end procedure

This paper utilizes Arduino to be the communicating nodes of the monitoring system, and for MQTT, it is suitable to be implemented with Arduino as it is line with the Arduino functionality itself, offering only a small size of transmission bandwidth for a network system. This affects to a small transfer overhead and minimized network traffic. Based on this fact, this paper chooses MQTT architecture to be compared with the proposed brokerless pub/sub architecture to analyze both water quality monitoring performance, specifically for the networking perspective using Arduino device communication. The MQTT for this paper selects HiveMQ [13] as its broker due to its open and full support to MQTT.

Moving to the the proposed approach, the main idea is to omit the existence of broker inside the architecture. Communication process lets the publisher replace the broker responsibilities by adding a subscription list. The list contains an array field of interesting topics for all the upcoming topic subscriptions from subscriber. More detail explanation of the proposed pub/sub approach is given in algorithm 1 and 2. To follow general flow of publish and subscribe mechanism, algorithm 1 begins with the subscribe procedure, and followed by the publish procedure in algorithm 2.

When subscribing, the algorithm 1 firstly checks whether the array field in subscription list for the requested topic is empty. By emptying this field, the topic message then can be stored in the dedicated array, containing message ID and the message itself. Meanwhile, after the subscription list is fully ready with all the requested topic, publishing procedure in algorithm 2 can be performed. Initially, every time a particular message is about to be transmitted, the algorithm checks whether the topic is the appropriate one (the interesting topic for the subscriber). If not, the message will not be sent to subscriber, but otherwise, the message will then be sent. Specifically, to send the message, the algorithm determines which message is about to be sent, and sets the length of message to be equal to the length of array field containing the requested topic in subscription list. After both lengths match, message is sent periodically byte-per-byte, until it reaches the maximum length. By this way, the approach ensures the message to be fully published to the subscriber.

B. Device Architecture

In this paper, several devices were used and assembled carefully to form an architecture depicted in figure 2. The architecture follows pub/sub architecture, in addition to a general hardware architecture using Zigbee protocol, to demonstrate the uniqueness of the proposed system. The system is intended to record three water quality measurement metrics: temperature, pH, and dissolved oxygen (DO) and thus, three kinds of sensors were used to support each metric. In addition, other devices were used as media for wireless data communication to a PC. These devices are two Arduino Uno boards with an embedded ZigBee wireless module for each. Hereafter in hardware architecture, the two boards are each called relay node and gateway node. The task of each node is explained as follows.

- Relay node is to forward data from sensors to the gateway or another relay, if any, through self-organizing wireless network. This relay node also acts publisher in pub/sub architecture.
- Gateway node is to transmit data collected by the sensors to the PC monitoring platform, either through wireless network or wired network. This gateway node also acts subscriber in pub/sub architecture.

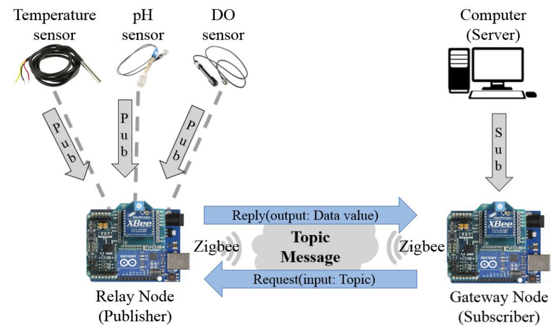

Fig. 2: Hardware with brokerless pub/sub architecture.

As seen in figure 2, sensors are placed underwater inside a water container. It can be one or more than one container in different place to locate the sensors. A relay node is installed in each container, from where the sensors read and collect the measurement data. Here, the communication between acquisition and relay node is wired. Subsequently, the relay node forwards the data of sensor measurement to the gateway node via wireless communication. It is assumed that the number of relay and gateway node is only one in the proposed system, but more nodes can be deployed to accommodate larger area of sensor coverage, as of the characteristic of IoT. Using ZigBee as a wireless transmission protocol, all data are ensured to be received by gateway before being presented to users. The ZigBee protocol is chosen as all of its advantages support the proposed system, *i.e.*, it provides low bandwidth consumption, reliable and self-configuring network, and small-size packet data backing. Finally, users can access the data on the PC connected to the gateway through USB serial cable.

IV. Performance Analysis

A. Evaluation Setup

TABLE I: Sensor Table

Metric	Variable	Unit	Sensor name	Sensor cost
Temperature	Temp	^oC	DS18B20	\$9.95
pH	pH	-	PH SEN0161	\$29.50
Dissolved Oxygen	DO	mg/L	SEN-11194	\$249.95
			Total	\$289.40

TABLE II: Device Table

Item	Device name	Cost
Relay node	Arduino Uno R3 DEV-11021	\$24.95
Gateway node	Arduino Uno R3 DEV-11021	\$24.95
Wireless module	Xbee Pro Module S1 (802.15.4)	\$37.95
Wireless shield	Xbee Shield Module for Arduino	\$11.97
	Total	\$99.82

The proposed system uses several devices for network node communication, which are tabulated in table I. Moreover, it uses several sensors as tabulated in table II to read the water quality data. Three sensors were used as mentioned: temperature sensor, pH sensor, and DO sensor for water quality testing. The three sensors were connected to an Arduino Uno as the relay node/publisher with an embedded ZigBee wireless module. To show the testing results to users, another Arduino Uno with a ZigBee wireless module also was used as the gateway node/subscriber.

Both tables also include the price of each to prove that the system was implemented with a low price. Compared to a same-purpose prototype by Rao *et al.* [7] and another commercial device for water quality monitoring [9], the proposed system has a lower cost and may be suggested to be developed commercially and more cheaply for IoT applications in the future. Table III compares the production cost of three aforementioned systems. Although the proposed system is simple which covers a small water area, the price of those three systems will increase in linearity when deployed more largely, keeping the cost of the proposed system lower than that of the other two.

TABLE III: Implementation Cost for Practical System

Commercial device [9]	**Rao *et al.* [7]**	**The proposed system**
$3,040	$1,040	$389.22

To monitor the water quality, the experiment was run during seven days. In each day, data were sampled in thirty minutes. The sensors were dipped underwater inside an aquarium, which was placed inside a room with a standard room temperature ($20{\sim}25^{o}C$). At the time the sensors had been dipped into the water, the measurement evaluation can be started directly by powering the devices.

Furthermore, the experiment compares the proposed pub/sub architecture and MQTT protocol to show the system performance in view of networking perspective. The comparison considers network end-to-end latency and throughput with varied message payload size, which are explained in detail as follow.

- Latency defines the time required for the two network nodes: relay and gateway, to send the data each other. As the system alleviates the pub/sub architecture, it is also can be defined as the measurement time from when the subscriber (gateway node) sends a topic subscription message until the publisher (relay node) replies with the appropriate published message. The latency testing was run ten times with varied message payload size, from fifty to five hundred Byte, for both pub/sub architecture. for each size, the latency in millisecond (ms) was captured and all of them are plotted as seen in figure 6.
- For throughput, this paper determines the number of successful message per particular time circulating between nodes over the network. The experiment diversifies payload size as well and calculates how many messages can be transmitted in one second. The $delay()$ statement in Arduino code was set to one millisecond and by this way, it may possibly achieve a maximum one thousand messages per second. The throughput testing was run ten times with varied message payload size, from fifty to five hundred Byte, for both pub/sub architecture. For each size, all data are plotted as seen in figure 7. The measurement unit is the number of message itself.

B. Results and Analysis

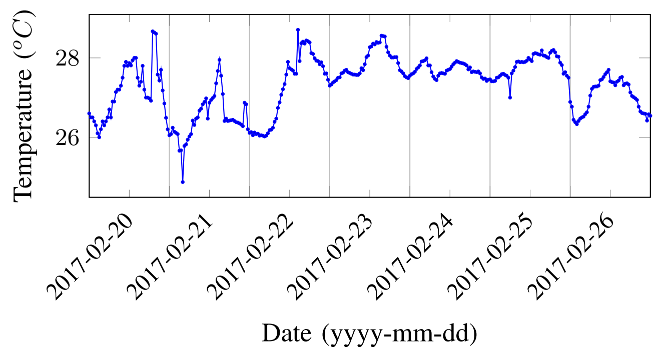

Fig. 3: Temperature measurement result during seven days.

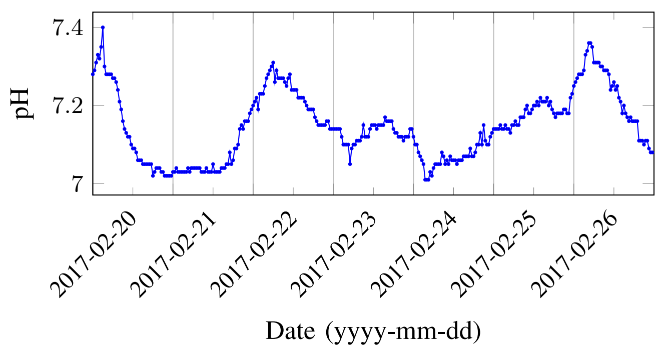

Fig. 4: PH measurement result during seven days.

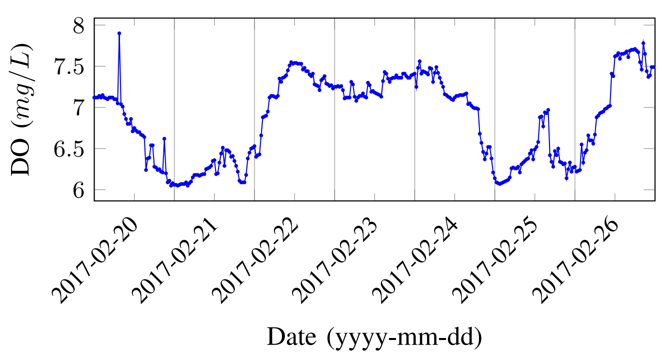

Fig. 5: DO measurement result during seven days.

The evaluation was performed to measure the water quality. When the sensors have been dipped underwater, users can directly power the devices. The system was set to capture the data in real-time every thirty minutes for main measurement, but hereafter in this paper, capture time is modified to check another performance of the system. The sensors were tested inside a water aquarium and the testing lasted for seven days. Measurement data can be observed in real-time and to clearly show the data for this paper, all data are plotted into a graph for each measurement metric. The graphs are shown in figure

TABLE IV: Stochastic Data of Measurement Result

Metric	**Day** / **Measurement**	**Value (7 Days)**
Temperature	Standard deviation	0.703
	Confidence interval	**27.318 ± 0.075**
pH	Standard deviation	0.087
	Confidence interval	**7.174 ± 0.009**
DO	Standard deviation	0.509
	Confidence interval	**6.687 ± 0.054**

3 to figure 5 for temperature, pH, and DO, respectively. In the figures, all metrics are plotted against time, which is separated using minor x grid line for each day. From the graphs, it is clearly seen that all metrics fluctuates greatly over times because the y axis is set in a relatively small range.

Based on the measurement results, stochastic data can be calculated to identify the water behavior. The behavior refers to how the water temperature, pH, and DO change over times, by determining standard deviation and confidence interval of measurement results. Firstly, data average ($\bar{x}$) is calculated. Then, standard deviation (σ) can be calculated as

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}, \tag{1}$$

where N is the number of days and x_i is the i-th data. Prior to knowing confidence interval, confidence coefficient is fixed for this paper, that is 95% or 1.96 based on z-table [14], and margin of errors (E) can be expressed as

$$E = \frac{(1.96 \times \sigma)}{\sqrt{N}}. \tag{2}$$

In mathematical model, confidence interval is data mean plus or minus margin of errors.

$$CI = \bar{x} \pm E. \tag{3}$$

In this paper, standard deviation and confidence interval are calculated for total seven days. Table IV summarizes the stochastic data for all sensors. From the table, some verdicts can be inferred:

- Standard deviation (STD) shows a heterogeneity of data and helps to infer how the measurement metrics (temperature, pH, and DO) change. The bigger the STD value, the more gap of every single data value to data mean. For all days, temperature, pH, and DO show an insignificant change. It is noticeable from STD value of each metric which comes under the value of 1. Based on STD definition, the measured metrics data has a relatively close distance to the mean of each metric. Observing further, temperature and DO have bigger fluctuation because of bigger STD, while pH produces lower STD, meaning that it fluctuates more slightly that other metrics.
- Confidence interval (CI) implies a range of two values, upper and lower limit, and it is dependent to the margin of errors to denote its upper and lower limit. The bigger the margin of errors, *i.e.* the bigger the CI, the more inaccuracy of sample data value to be inside of CI. In context of measurement result as in table IV, each measurement metric has a relatively low margin of error. It results on a narrow interval of 95% confidence coefficient. For example, according to the temperature measurement results, the room temperature where the aquarium resides (20∼25^oC) is not within 95% confidence interval for the measured temperature mean of all days.

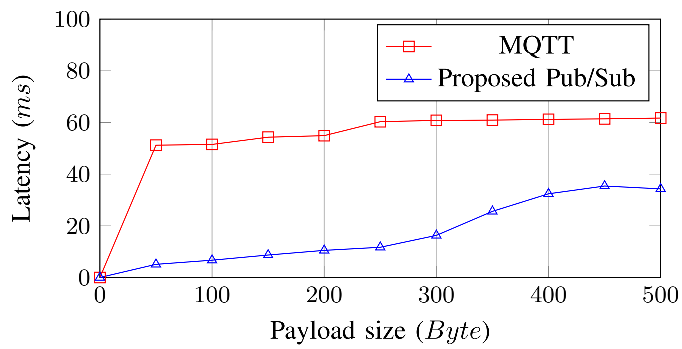

Fig. 6: Latency comparison of two methods for varied payload size.

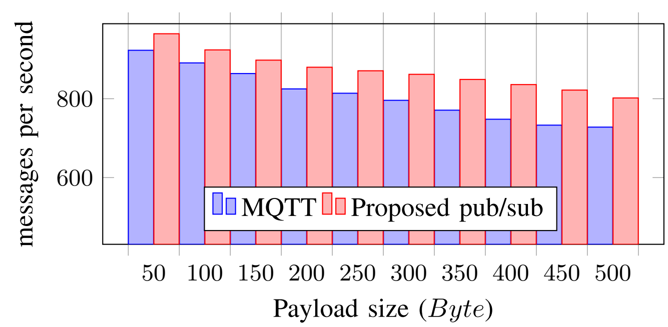

Fig. 7: Throughput comparison of two methods for varied payload size.

From networking perspective, Arduino latency and throughput were tested to compare the performance between the proposed pub/sub architecture and MQTT. The experiment adjusts message payload size for both latency and throughput testing. Figure 6 illustrates the latency in ms against payload size. Latency increases in line with the raise of payload size in general as more size require more network resources. It is clearly seen that the proposed pub/sub architecture has lower latency for all payload size; only up to 35 ms latency for the proposed pub/sub against over 60 ms for MQTT. Brokerless pub/sub of the proposed system minimizes the network resource and thus reduces the latency.

For throughput comparison as illustrated in figure 7, the proposed pub/sub also shows a better performance over MQTT. It can successfully send more messages per second compared to MQTT performance for all measured payload size. Overall, the delivered message per second drops over the increase of payload size. It is because network bandwidth is allocated for all messages based on its payload size, and apparently, the bigger payload size demands more bandwidth allocation to send every single message.

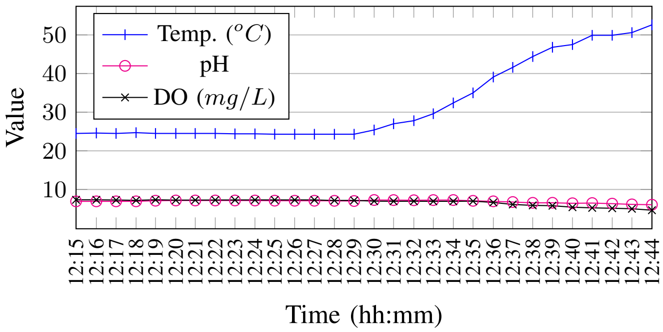

Fig. 8: Experimental results show higher temperature against lower pH and DO value.

The last experiment was conducted to test the relationship among temperature, pH, and DO of water. Precisely, temperature was adjusted to study how pH and DO change related to this adjustment. Temperature was chosen as an adjustment metric as it is the easiest metric to be recognized with normal treatment, *i.e.*, only using a bare hand is enough to feel a slight and/or dramatic change of water temperature. For evaluation scenario, data were sampled every minute and it lasted for ten minutes. The ten minutes duration was divided into two parts, with five minutes for each part. During the first half, a static water temperature was checked and for the second half, the water temperature inside the aquarium was raised on purpose.

The result for this experiment is illustrated in figure 8. From the figure, it is observable that during the first half, water temperature, oxygen, and pH value were only slightly changing, or it can be said that they were remained unchanged in general. However, after five minutes, it is noticeable that when the temperature raises, oxygen and pH value of the water drops gradually. This experiment result is in line with several previous works [4][5], which reported that:

- The higher the water temperature, the stronger the possibility for water to ionize H+ from its molecule and hence forces the pH level drop.
- Also, a raised temperature decreases oxygen solubility inside the water, and hence forces the oxygen level drop.

To sum up, the experiment reveals that the water temperature is inversely proportional to dissolved oxygen and pH.

V. Conclusion

This paper describes an implementation of a real-time water quality monitoring system for three measurement metrics: temperature, pH, and DO with ZigBee protocol for nodes communication. The ZigBee communication code was modified to accommodate a brokerless pub/sub architecture towards IoT applications in the future, which certifies the uniqueness of this paper compared to other similar works. The system monitors water quality, stores data into a database, and computes the data for standard deviation and confidence interval during measurement days for data analysis. The evaluation also compares the proposed pub/sub architecture with traditional MQTT architecture to investigate the system performance in regards to networking perspective. From this comparison, it is proven that the proposed architecture has a better performance than the traditional MQTT architecture, in case of network latency and throughput for varied message payload size.

Furthermore, an experiment was conducted to show a relationship among the three metrics, and the result reveals that the water temperature is inversely proportional to pH and dissolved oxygen value. The system helps to tackle the preliminary problems and due to the low production cost, it suggests future commercial production of the system for the sake of clean consumable water.

Acknowledgment

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2017-H8601-16-1011) supervised by the IITP(Institute for Information and communications Technology Promotion) and BK21+ Project in Kumoh National Institute of Technology.

References

[1] "International decade for action 'water for life' 2005-2015." [Online]. Available: http://www.un.org/waterforlifedecade/sanitation.shtml

[2] C. H. Tseng, "Coordinator traffic diffusion for data-intensive zigbee transmission in real-time electrocardiography monitoring," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 12, pp. 3340–3346, Dec. 2013.

[3] S. K. Chen, T. Kao, C. T. Chan, C. N. Huang, C. Y. Chiang, C. Y. Lai, T. H. Tung, and P. C. Wang, "A reliable transmission protocol for zigbee-based wireless patient monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 1, pp. 6–16, Jan. 2012.

[4] R. Manasrah, M. Raheed, and M. I. Badran, "Relationships between water temperature, nutrients and dissolved oxygen in the northern gulf of aqaba, red sea," *OCEANOLOGIA*, vol. 48, no. 2, pp. 237–253, Apr. 2006.

[5] J. Ashton and L. Geary, "The effects of temperature on ph measurement," *Reagecon Delivering the Correct Result, Shannon, Co.Clara, Ireland*, Apr. 2011.

[6] N. A. Cloete, R. Malekian, and L. Nair, "Design of smart sensors for real-time water quality monitoring," *IEEE Access*, vol. 4, pp. 3975–3990, 2016.

[7] A. S. Rao, S. Marshall, J. Gubbi, M. Palaniswami, R. Sinnott, and V. Pettigrovet, "Design of low-cost autonomous water quality monitoring system," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Aug. 2013, pp. 14–19.

[8] A. N. Prasad, K. A. Mamun, F. R. Islam, and H. Haqva, "Smart water quality monitoring system," in *2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, Dec. 2015, pp. 1–6.

[9] M. Zennaro, A. Floros, G. Dogan, T. Sun, Z. Cao, C. Huang, M. Bahader, H. Ntareme, and A. Bagula, "On the design of a water quality wireless sensor network (wqwsn): An application to water quality monitoring in malawi," in *2009 International Conference on Parallel Processing Workshops*, Sept. 2009, pp. 330–336.

[10] B. P. Wong and B. Kerkez, "Real-time environmental sensor data: An application to water quality using web services," *Environmental Modelling & Software*, vol. 84, pp. 505 – 517, 2016.

[11] M. A. Tariq, B. Koldehofe, and K. Rothermel, "Securing broker-less publish/subscribe systems using identity-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 518–528, Feb. 2014.

[12] S. Shitole and A. D. Gujar, "Securing broker-less publisher/subscriber systems using cryptographic technique," in *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, Aug. 2016, pp. 1–6.

[13] "Hivemq." [Online]. Available: http://www.hivemq.com/

[14] "Ztable." [Online]. Available: http://www.statisticshowto.com/tables/z-table/