Data Delivery Implementation on River Flow Control System Prototype Using HTTP

Mochammad Hannats Hanafi Ichsan¹, Sabriansyah Rizqika Akbar², Hamdan Malik Satriyo Aji³, Eko Setiawan⁴, Othman Mirizi Batubara⁵

¹,²,³,⁴,⁵Brawijaya University, Malang,
{¹hanas.hanafi, ²sabrian, ⁴ekosetiawan}@ub.ac.id
{³iammariyo25, ⁵othmanmirizi2}@student.ub.ac.id

Received 28 August 2023; accepted 13 December 2023

Abstract. Data transmission is one of the most essential components in a critical system. One of them is to control the river water canal. Previous research related to water canal monitoring has been carried out, but the data is still local. This research implements a water canal with one main river with three tributaries/river branches where the three tributaries have different area conditions. In this research, the process of sending data from the sensor node (Control Flow System) to the master node (Control station) to the cloud server to be displayed on a website. The microcontroller used for the main processor and as a data-sending device is the ESP8266 and the cloud server used is ThingSpeak. This research succeeded in implementing HTTP for the process of sending data. The HTTP protocol is suitable for the process of data delivery. Sending data on the Control Flow System to the Control Station was successfully sent optimally at a distance of 3 meters with the "Good" classification according to the TIPHON standard. Meanwhile, the average data is sent from the Control Station to the cloud server in about 44 seconds.

Keywords: River Flow Control, ESP8266, HTTP, River Canal, Cloud Server

1 Introduction

A flood is a disaster that occurs due to several factors. One of the causes of flooding is the runoff water from high areas to lower areas. Natural disasters can be unpredictable, and mitigation measures can reduce the damage's severity [1, 2]. Effective prevention strategies can reduce the victims and impacts of natural disasters. Thus, delivering information by accumulating data from trusted sources is very important. Effective and timely data delivery is necessary for community risk assessment and management [3, 4].

Floods occur not only in urban areas but also frequently in watershed/river branch areas [5]. The watershed is a very long stream from upstream to downstream. If rainfall in the upstream area is very high, the downstream area will be very vulnerable to flooding if it rains [6]. The river also has several tributaries. The risk of flooding from this problem can be minimized by controlling the water flow from upstream to one of the tributaries, where the water canals are used to control the amount of water flow discharge [7, 8]. Problems like this can be avoided by taking mitigation measures such as issuing direct warnings to the public, particularly those
who live in flood-prone areas [9, 10, 11]. Establishing a monitoring system has become essential for big city authorities in developing countries as computing technology has advanced [12].

Water Flow Control or canals transport water for irrigation and other human needs. While the development of more efficient modes of transportation has reduced the need for canals, they continue to play an essential role as transportation conduits and drivers of global trade. Water canals are widely used as flood control waterways in Indonesia [13, 14, 15]. Canals can be used as water discharge control channels that regulate incoming and outgoing water or as drain channels that flow beneath the road surface of several large cities so that the volume in the mainstream is not burdened or overflows, so that if flood water overflows, water will flow into this channel, and then into rivers or other bodies of water [16, 17].

Previous studies have researched controlling water canals from one main river to three tributaries. This research implemented Fuzzy Logic and Simple Additive Weighting to control water canals. However, the prototype is still wired for sending data between nodes (which contain sensors and microcontrollers) to the Master Node (algorithmic computing center) [18]. In actual river conditions, this is not possible because of the broad area of the watershed or river canal area. So wireless data transmission needs to be designed for research. Another thing is the monitoring process which can only be seen on the serial monitor, which makes the system need to be improved. Cloud technology and website can help users see objects' conditions. In previous studies, rainfall and water level sensors were implemented with the ESP8266 microcontroller for Sensor Nodes. Meanwhile, the Master Node uses the Arduino Mega.

One of the data transfer protocols that can be implemented on ESP8266 and Arduino Mega is Hypertext Transfer Protocol (HTTP) [19]. HTTP is a network protocol in the application layer of a hypermedia information system whose data transmission is distributed [20]. The HTTP POST request-response method can be used to communicate between wifi client nodes and server nodes [21]. The data sent to the server with the POST method is stored in the entity-body of the HTTP request. The server listens for connection requests from the client, parses each received message with the identified request target, and provides one or more responses to the received messages. So it is possible to send data between ESP8266 to Arduino Mega using the HTTP protocol [22]. In addition, the HTTP protocol was chosen because it is considered suitable for data delivery.

In addition to sending data, storing data is one of the critical factors in a monitoring system before data is displayed on a website. One of the webservers that are good for prototyping is ThingSpeak [23, 24]. This web server platform can receive data from clients using the HTTP protocol. After this platform receives the data, the website will display the data the webserver stores.

So this research will focus on sending data from the Sensor Node to the Master Node on River Flow Control. After that, the Master Node will send data to the web server and display data to a website. The data generated by the Sensor Node can be sent and appropriately read by the user, and the user can monitor the status of the river. So that the process of sending data must be appropriately studied so that data can be sent and stored on the web server using HTTP Protocol and can read well the actual condition of the Canal/River Flow Control.
2 Previous Research

In previous studies, research was carried out related to the control system of the water canal on water level and rainfall [18]. However, the process of sending data is still done conventionally. Figure 1 shows a Sensor Node at the river Flow Control System to control the water discharge height from upstream (river section A) to each downstream or tributary (river section B). Then the processing is done at the control station, located some distance from the river. Conventional communication using cabling is less likely to be implemented in actual conditions.

There are several previous research references related to flood natural disaster management research. Research (Ichsan et al., 2019) presents the design of a sluice control system using fuzzy and SAW. This study calculated the number of floodgate openings using water level and rain module sensor data. The servo module controls the sluice opening. The Arduino Mega 2650 module is used for processing and calculating the algorithm. However, this study used LED indicator lights to notify of flood hazards. As a result, we require a website that can display system monitoring results and send users warning notifications.

The other research conducted is monitoring flood, but this research only monitoring on one location only [23]. This research sent data to ThingSpeak and then sent into telegram. Only one person can access the data who is given access rights to access Telegram. Other people cannot access the information. The other research conducted remote sensing for sense the flood detection system, this process is effective in land area not in water canal [25].

The other related research is implementing Wireless Sensor Network AI Based for flood. This research only sense on one point or one location using LoraWAN [26]. This research only sensing water level. On the actual condition, rainfall playing high role on flooding by adding the water volume into the river.

With the development of technology and today's accessibility needs, the previous research that has been done is considered not suitable for river branch condition. So this research focuses on sending data from the system to a cloud-based storage area that can be monitored on a website.

3 System Design

Based on previous studies, the Flow Control System, or as a Sensor Node, has a mechanism for sensing environmental conditions using the YL-83 sensor as a rain level sensor and a Water Level Sensor to determine the water level in river sections. The microcontroller used is ESP8266 to process data transmission. Data transmission on the ESP8266 uses the HTTP protocol. The design diagram can be seen in Figure 2.
This research was conducted to select and implement the process of sending data from the Flow Control System (as a Sensor Node) and Control Station (as a Master Node). All nodes are connected to the same wifi. The wifi acts as a wifi server. The system has 3 Sensor Nodes on each river branch acting as a wifi client/station whose function is to retrieve and send sensor data and a sink node wifi server whose job is to collect data from clients and send Master Node data. Communication between the wifi client and the wifi server uses HTTP. The client uses the POST method for sending each piece of data. The POST method uses the entity-body in the HTTP Request to send data to the Master Node. On the wifi side, the server has a function to listen if the Sensor Node sends data.

After the wifi server receives the data, the data is sent to the Control Station for further processing with several algorithm processes for the control process. After that, the processing data is sent to the Flow Control system to open/close the channel. After the process, the data will be sent to the cloud server for storage and displayed on a website for visualization. The process of storing data on the cloud server is done on ThingSpeak. Several studies consider ThingSpeak to be good in performance for IoT data storage. Sending data from the Control Station to the Cloud also uses the HTTP protocol.

On the Control Station as the Master Node using Arduino Mega. The HTTP Post method sends data between the Flow Control System and Control Station. So that the Control Station only does the listening process. After the Control Station receives the data, the data is also sent using the HTTP protocol to the Cloud. After that, the data will be displayed on the website for data monitoring. This research focuses on measuring the suitability and performance of HTTP for sending data from the Flow Control System to the Control Station and Control Station to the Cloud Server and then automatically shown on the website.

4 Research Result

The testing process in this research was tested into 2 phases. In addition, the data display can be seen on the cloud server and website. The first testing scenario is data transmission between the Flow Control System and Control Station. The second testing scenario is the Control Station to the Cloud Server. Tests are carried out to measure the conditions that allow data transmission to occur. As well as the accuracy and timeliness of delivery are one of the benchmarks for the success of this research.
4.1 Website Interface

The Master Node sends data to the ThingSpeak web server. The web server displays sensor measurement data and time in the day, date, and time of data collection by the sensor. The graph will continue to be updated every time the measurement results are successfully sent to ThingSpeak, as shown in Figure 3. This web server is a storage area that is not publicly accessible and cannot be customized/not user-friendly. So we need a website for hosting and displaying the website to make it easy for users to read.

![Figure 3. Cloud Server (ThingSpeak)](image)

A website page has been developed and can be accessed at the https://s.ub.ac.id/rivermonitoringsystem. While Figure 4 is an image of the Monitoring System page that has been customized on the website. This page displays the conditions for the three controlled river sections. The conditions displayed by the website are the rain and water levels obtained from the Rain Level sensor and Water Level Sensor. Figure 3 shows the Channel Image and Label Field and the date that cannot be customized. As well as the nature of the data stored on ThingSpeak is private, so it cannot be accessed by people who do not have special access rights.
Figure 4 is a customized website display that has been more legible by users and accessible to everyone who accesses the website. The data displayed is in the form of a graph. On the vertical axis is the height data from the sensor, both the Water Level sensor for water level and rainfall for rainfall. In comparison, the horizontal axis is when the two sensors acquire the data.

4.2 First Testing Scenario, Data Delivery Performance from Flow Control System to Control Station

In this test, testing is carried out between each Flow Control System / Sensor Node to the Control Station / Master Node. Each Sensor Node is tested for performance based on the distance of data transmission, and there are three Sensor Nodes because they represent three tributaries. So that this test sends data from Node1-Master Node, Node2-Master Node, and Node3-Master Node. Each test was carried out 15 times for sending data at different intervals and under different prototype conditions. Each test was carried out at various distances, namely 3 meters, 30 meters, 50 meters, 80 meters, and 100 meters.

Based on the test results shown in Table 1, from 3 to 100 meters, data transmission is carried out successfully and sent from the Flow Control System to the Control Station. At a distance of 3 meters, the three nodes transmit data in a relatively similar manner, which is around 238 milliseconds. The data delivery testing trend can be seen in Figure 5. However, at a distance of 30, 50, 80, to 100 meters, almost all data transmission delays have the same delay in the range of 969 to around 1000 milliseconds. The Tiphon category for "Good" is 150-300 milliseconds, and "Poor" >450 milliseconds. Based on the delay standard from Tiphon for 3 meters, it is categorized as "Good," while for 30 to 100 meters, it is categorized as "Poor."
Table 1. Data Delivery From Flow Control System To Control Station Testing Result Recap

<table>
<thead>
<tr>
<th>Testing on</th>
<th>3 meter</th>
<th>30 meter</th>
<th>50 meter</th>
<th>80 meter</th>
<th>100 meter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Node 1 (ms)</td>
<td>240.6</td>
<td>965.6</td>
<td>978.4</td>
<td>988.7</td>
<td>999.1</td>
</tr>
<tr>
<td>Sensor Node 2 (ms)</td>
<td>242.1</td>
<td>960</td>
<td>979.4</td>
<td>988.5</td>
<td>987.3</td>
</tr>
<tr>
<td>Sensor Node 3 (ms)</td>
<td>243.8</td>
<td>967.7</td>
<td>980.3</td>
<td>989.6</td>
<td>999.7</td>
</tr>
<tr>
<td>Average (ms)</td>
<td>242.2</td>
<td>964.4</td>
<td>979.4</td>
<td>988.9</td>
<td>995.4</td>
</tr>
<tr>
<td>Overall Average (ms)</td>
<td><strong>834.1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3 Second Testing Scenario, Data Delivery Performance from Control Station to Cloud Server

The second test is carried out to measure the success of sending data from the Control Station to the Webserver until the data is displayed on the website. The data measurement model uses the timestamp on the Control Station and calculates the difference with the time the data is displayed on the website. The data collection process was carried out 45 times, and the data displayed is the condition of the three tributaries.

Based on the pictures, all tests were carried out successfully. The results of sending data have an irregular period with a minimum of 32 seconds and a maximum of 79 seconds, where the average is around 44 seconds. The irregular period can be seen in Figure 6—the irregular period caused by various factors in the Cloud that cannot be controlled.
5 Conclusion And Future Works

In this research, the HTTP protocol was implemented for data transmission on a data transmission prototype for water canal control. A prototype of the water canal has been carried out in previous research to measure rainfall and water level in river sections. Then the data is processed with several algorithms to determine the control of opening and closing the water canal. This research focuses on sending data based on the sensing and control results from the Flow Control System to the Control Station and the Control Station to the website. Sending data from the Flow Control System to the Control Station was carried out from a distance of 3 meters to 100 meters which were all successful, where the delay produced at a distance of 3 meters had "Good" results and a distance of 30-100 meters produced "Poor" results based on Tiphon standard. Sending data from the control station to the Cloud and the website is also done well, with an average of around 44 seconds.

For future work, sending data to a cloud server is considered long and not real-time. Therefore further studies need to be carried out on a cloud server environment that is more reliable and real-time. This research aims to examine data transmission protocols that are considered better. And can be implemented in water canals to assess the performance of the system built as a whole.

Acknowledgments. Thanks to Mrs. Aisshah Roesiana Dewi who has assisted in processing the test data using Tableau, so that the data presented can be visualized and read properly.

References


