

Optimization of Remote Intake Pump Control System Using MQTT Protocol Based on PLCnext (Case Study : Pluit Reservoir)

Fransixus Wisnu Pranoto¹

¹Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

12141160042@student.polinema.ac.id

Abstract— This research was motivated by operational inefficiencies in the intake pump control system at Pluit Reservoir, caused by the 1.4 km distance between the Water Treatment Plant (WTP) and the pump house, as well as the failure of a previous fiber optic-based control system and the performance limitations of an HTTP-based system. This study aims to optimize the remote control system by comparing the performance of the Message Queuing Telemetry Transport (MQTT) protocol with the Hypertext Transfer Protocol (HTTP) on the PLCnext AXC F 2152 platform. The research method used was experimental, involving the construction of a control system prototype that represents field conditions and conducting tests under unstable internet network scenarios. The Quality of Service (QoS) parameters measured included throughput, packet loss, delay, and jitter. The results show that the MQTT protocol is significantly superior in terms of reliability and responsiveness. Under unstable network conditions, MQTT was able to suppress packet loss to below 1%, whereas HTTP experienced packet loss of up to 40%. Furthermore, the delay on MQTT was recorded to be 2-3 times lower than HTTP, making it far more responsive for realtime control applications. Although HTTP's throughput is inherently higher, MQTT's superiority in minimizing packet loss and delay proves that it is a more optimal and reliable solution for critical infrastructure control systems in environments with limited connectivity.

Keywords—AXC F 2151, HTTP, MQTT, Quality of Service, Optimization.

I. INTRODUCTION

The provision of clean water is a fundamental utility, and its reliable distribution hinges on the operational efficiency and robustness of control systems within water management infrastructure [1][2]. Water treatment plants (WTPs) are responsible for treating raw water sourced from reservoirs, springs, or rivers—often located in public or remote areas—before distributing it to consumers [3][4]. A critical component in this process is the intake pump, which draws raw water from the source and transports it to the treatment facility through pipelines. The remote and sometimes vulnerable location of these pumps poses significant challenges for operational monitoring and control, particularly in areas with limited infrastructure or security concerns [5].

A notable case illustrating these challenges is found at Waduk Pluit in North Jakarta, where the intake pump house is situated approximately 1.4 kilometers from the Water Treatment Plant (WTP). The absence of adequate operational facilities at the pump house requires personnel to commute manually between the WTP and the pump site to operate the pumps, resulting in significant inefficiencies in time and human resources. To address this, a fiber-optic-based remote control system was initially implemented [6]-[8]. While this system enabled real-time pump operation, it was highly vulnerable to frequent cable theft, leading to communication failures and a forced return to manual operation. This highlighted a critical weakness in systems reliant on exposed physical infrastructure[9].

Subsequently, an alternative system was developed using a Programmable Logic Controller (PLCnext AXC F 2152) integrated with an HTTP server via Node-RED [10]. Although this system successfully established communication between PLCs, it encountered limitations due to unstable internet connectivity at the pump house—a semi-underground concrete structure—and the inherent inefficiency of the HTTP protocol in low-connectivity environments [11][12]. HTTP, being a request-response protocol, is not optimized for scenarios with limited bandwidth, high latency, or intermittent connections, making it unsuitable for real-time remote control in such settings.

To overcome these persistent issues, a communication protocol that is efficient, lightweight, and reliable under unstable network conditions is essential. The Message Queuing Telemetry Transport (MQTT) protocol [13], which operates on a publish-subscribe model, is specifically designed for environments with limited connectivity and bandwidth. Its low overhead, support for asynchronous communication, and ability to function reliably in high-latency or unreliable networks make it an ideal candidate for remote industrial control applications.

This study aims to implement and evaluate the performance of MQTT for remote pump control in challenging environments like Waduk Pluit. By comparing key Quality of Service (QoS) parameters—such as throughput [13], packet loss [14], end-to-end delay [15], and jitter—between HTTP and MQTT protocols, this research seeks to demonstrate the

advantages of MQTT in enhancing system reliability and efficiency. The findings are expected to provide a validated, practical framework for deploying robust remote control systems in water management and other critical infrastructure applications, thereby improving operational continuity and security.

II. METHOD

This research employed an experimental method to optimize the remote intake pump control system for Pluit Reservoir by implementing and evaluating the MQTT protocol on the PLCnext platform. The study involved designing and building a functional prototype that simulated field conditions. The core of the experimental procedure was a direct performance comparison between the existing HTTP-based system and the proposed MQTT-based system under unstable internet network scenarios. The comparison was quantitatively assessed based on key Quality of Service (QoS) parameters: throughput, packet loss, end-to-end delay, and jitter.

A. Research Stages

Initial observation of the operational inefficiencies at Pluit Reservoir, specifically the 1.4 km distance between the Water Treatment Plant (WTP) and the pump house, and the vulnerabilities of the previous fiber optic and HTTP-based control systems. Review of existing journals and research on the application of the MQTT protocol for both monitoring and controlling systems to establish a theoretical foundation. Designing the new control system, including hardware wiring, MQTT protocol programming via the MQTT Client Library on the PLC, and creating control logic using Ladder Diagram and Function Block Diagram (FBD). Building the complete system based on the design to create a working prototype that accurately represents the technical conditions in the field. Conducting tests on the fully functional system to gather QoS performance data for both MQTT and HTTP protocols using applications like Wireshark and tcpdump. Collecting and analyzing the data to evaluate system performance, measure optimization, and compare the new MQTT system with the previous HTTP system. Drawing final conclusions based on the data analysis to determine the system's optimization and its feasibility for implementation at Pluit Reservoir

B. System Block Diagram

The block diagram for the optimized system using the MQTT protocol (as shown in Figure 1) featured a more direct and efficient architecture.

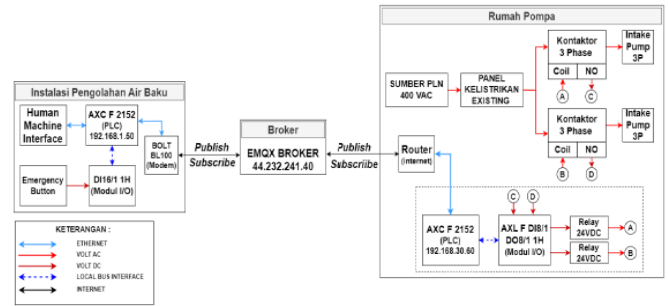


Figure 1. Block Diagram

The architectural design for the control system was defined through block diagrams for both communication protocols. The HTTP-based system, representing the legacy solution, operated on a client-server model where PLCs at the Master (WTP) and Slave (Pump House) locations communicated indirectly via a cloud-based HTTP server (ThingsBoard); data from the PLCs was read by local Node-RED instances, converted to JSON, and transmitted via HTTP POST and GET requests. In contrast, the proposed MQTT-based system utilized a publish-subscribe model centered on an EMQX MQTT broker. In this architecture, both the Master and Slave PLCs acted as MQTT clients that published messages to and subscribed directly to relevant topics on the broker, enabling a more efficient and direct form of communication that eliminated the need for intermediary servers and constant request-response cycles.

C. Human Machine Interface Design

A dedicated Human Machine Interface was developed natively on the PLCnext AXC F 2152 controller to provide operators with intuitive control and real-time monitoring capabilities, as shown in Figure 2

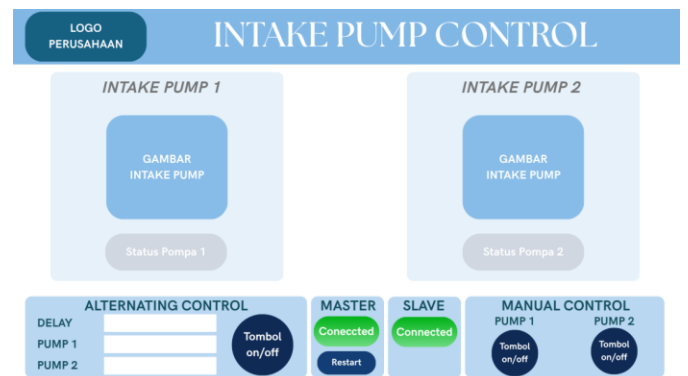


Figure 2 HMI Design

This main page displayed the real-time status of both pumps, indicated the connection status of the Master and Slave PLCs to the MQTT broker, and featured two control modes—Manual Control for direct operation and Alternating Control for automated timer-based switching between pumps. An emergency stop function was integrated to immediately halt all operations, ensuring operational safety.

D. Tools and Materials

The experimental setup utilized a range of specific hardware and software tools to construct and test the prototype system. The core hardware included two PLCnext AXC F 2152 controllers acting as Master and Slave, supplemented with digital I/O modules (AXL F DI8/1 DO8/1 1H and AXL F DI16/1 1H) for interfacing with sensors and actuators like contactors and relays. Power was supplied by 24 VDC power supplies, and internet connectivity for the Slave was facilitated by a 4G modem. The key software tools included PLCnext Engineer for programming the controllers, Node-RED for building the HTTP communication flows, Wireshark for network analysis, and PuTTY and WinSCP for accessing and managing files on the PLCs' Linux system. The cloud infrastructure comprised the EMQX MQTT broker and the ThingsBoard HTTP server.

E. Testing Parameters

The performance evaluation was based on measuring key Quality of Service (QoS) parameters: throughput (effective data transfer rate in kbps), packet loss (percentage of packets lost during transmission), end-to-end delay (time in milliseconds for a packet to travel from source to destination), and jitter (variation in delay). The testing procedure was designed to simulate real-world conditions; the Master and Slave devices were physically separated by 1.4 km, with the Slave connected via three different 4G ISP providers (A, B, and C) to emulate an unstable internet environment. Each protocol—HTTP and MQTT at QoS levels 0, 1, and 2—was tested for a duration of five minutes. Network traffic for HTTP was captured using Wireshark, while MQTT traffic was captured using the tcpdump command directly on the PLCs' terminals. The resulting packet capture (.pcap) files were then transferred and analyzed offline to calculate the QoS metrics, and the entire testing process was repeated twice to validate the consistency and reliability of the results.

III. RESULTS AND DISCUSSION

The experimental study successfully implemented and tested a prototype remote control system for intake pumps using the PLCnext platform. The core objective was to evaluate and compare the performance of the MQTT protocol against the traditional HTTP protocol under challenging network conditions that mimic the real-world environment of Pluit Reservoir. The following sections present the findings from the tests, which were conducted over two separate sessions to ensure data reliability, followed by a comprehensive discussion of the results.

A. Test Location

The tests were conducted in Malang City, Indonesia, to simulate the exact 1.4 km distance between the Water Treatment Plant (WTP) and the pump house at Pluit Reservoir. The Master device, representing the WTP with a stable internet connection, was located at Jalan Anggrek Garuda No. 47. The Slave device, representing the pump house with poor connectivity, was strategically placed in the basement of the Graha Polinema building, an underground concrete structure known for its weak and unstable 4G signal. This setup

accurately represented the operational challenges faced in the field.

B. Master Device Prototype

The Master device prototype was configured as the central control unit. Its architecture integrated a 24 VDC power supply, a PLCnext AXC F 2152 controller, an AXL F DI16/1 1H digital input module for acquiring signals, and an emergency push button. This assembly was housed on a DIN rail, forming a complete and functional control station for issuing commands and monitoring the system status via the built-in HMI, as shown in Figure 3.

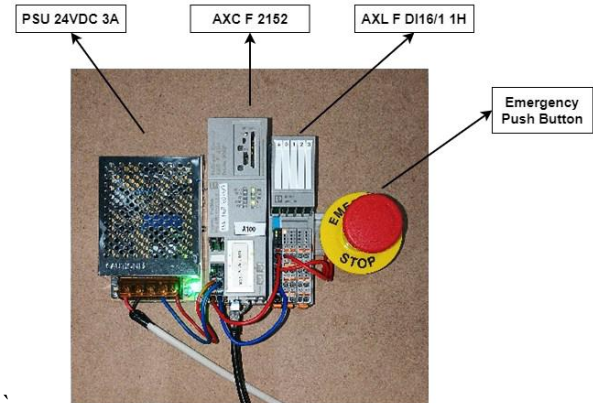


Figure 3 Master Device Prototype

C. Slave Device Prototype

The Slave device prototype was built to execute commands and relay pump statuses. It consisted of a more robust 24 VDC power supply, a second PLCnext AXC F 2152 controller, an AXL F DI8/1 DO8/1 1H digital I/O module, and two 3-phase contactors controlled by 24 VDC relays. This setup physically represented the pump house, capable of switching the pumps on and off based on commands received from the Master device, as shown in Figure 4.

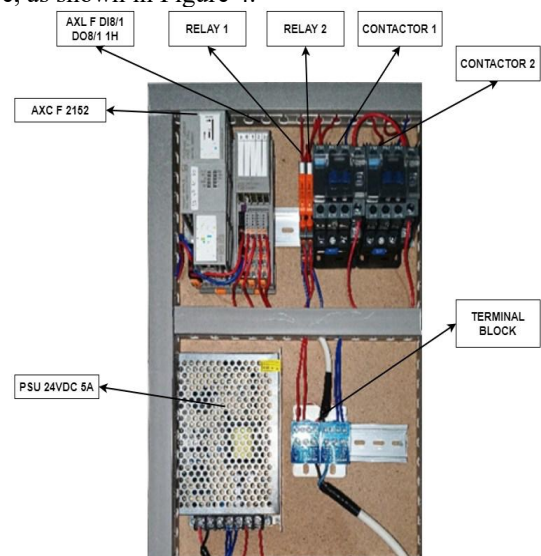


Figure 4 Slave Device Prototype

D. First Test Results

The first test was performed with the Slave device experiencing significant internet instability from all three ISPs (A, B, C). The results, consolidated from both Master and Slave devices, are presented in the table below and reveal a critical trend, as shown in Table I.

TABLE I
QOS PERFORMANCE METRICS

Parameter	Direction	HTTP	MQTT QoS0	MQTT QoS1	MQTT QoS2
Throughput (Kbps)	Master to Slave	6.481	1.500	0.867	0.609
	Slave to Master	4.508	1.533	1.389	0.655
Packet Loss (%)	Master to Slave	25.106	4.843	7.156	10.683
	Slave to Master	0.220	0.416	0.044	0.810
Delay (ms)	Master to Slave	774.708	232.881	255.998	257.398
	Slave to Master	804.340	367.439	382.287	399.492
Jitter (ms)	Master to Slave	82.920	5.445	11.554	26.565
	Slave to Master	110.459	9.640	18.949	55.752

E. Second Test Results

A second test was conducted after disabling the *Clean Session* feature in the MQTT protocol and during a period of marginally better internet stability at the Slave location. The results further solidified the findings from the first test, as shown in Table II.

TABLE II
QOS PERFORMANCE METRICS

Parameter	Direction	HTTP	MQTT QoS0	MQTT QoS1	MQTT QoS2
Throughput (Kbps)	Master to Slave	9.228	1.502	1.023	0.769
	Slave to Master	6.179	1.579	0.805	0.606
Packet Loss (%)	Master to Slave	40.359	0.535	0.465	0.356
	Slave to Master	0.224	0.133	0	0.114
Delay (ms)	Master to Slave	603.234	284.915	257.417	242.475
	Slave to Master	603.937	416.491	411.762	419.779
Jitter (ms)	Master to Slave	21.540	25.999	54.183	36.696
	Slave to Master	26.115	27.610	28.581	29.534

F. Overall Discussion

The results demonstrate a consistent and decisive advantage of the MQTT protocol over HTTP for real-time remote control applications in unstable network environments. While HTTP showed higher throughput, this metric is irrelevant for the small command-and-control messages used in this system. The critical parameters of reliability and responsiveness are defined by packet loss and delay.

The most striking result is the catastrophic packet loss suffered by HTTP (25.1% and 40.4% in Test 1 and 2, respectively) when sending commands from the Master to the Slave. This level of loss is untenable for a control system, as it would lead to missed commands and unpredictable pump behavior. In contrast, all levels of MQTT, especially QoS 1, maintained packet loss below 1% in the same direction, ensuring command delivery. This is because MQTT's lightweight, connection-oriented publish-subscribe model is inherently more efficient and resilient to packet drops than HTTP's heavy, stateless request-response model, which requires retransmitting entire packets upon failure.

Furthermore, MQTT significantly outperformed HTTP in end-to-end delay. HTTP's delay, often exceeding 600ms, is due to the overhead of establishing TCP connections and the request-response cycle for every communication. MQTT's persistent connection to the broker and immediate publishing mechanism slashed delays by over 50-60%, resulting in a far more responsive system crucial for real-time feedback and control.

The trade-off for MQTT's reliability is its lower throughput, which is a result of its smaller packet sizes and the acknowledgment mechanisms in QoS 1 and 2. The jitter performance was more variable; MQTT excelled in the highly unstable first test but showed higher jitter than HTTP in the more stable second test. This suggests that while MQTT is superior in bad conditions, its internal message queuing can introduce variability when the network is stable enough for HTTP to perform consistently.

IV. CONCLUSION

In conclusion, this study demonstrates that gateway antenna height is a primary factor influencing LoRaWAN performance in an urban environment. Elevating the antenna from 15 meters to 30 meters significantly expanded the reliable communication range for all Spreading Factors (SF) by reducing obstructions and minimizing signal fading. The optimal configuration was achieved at the 30-meter height, where SF7 proved best for short-range applications, SF9 for medium-range, and SF10 for the longest reliable distance. These findings highlight the critical importance of strategic antenna placement and SF selection for deploying reliable and efficient LoRaWAN networks in urban settings.

REFERENCES

- [1] M. P. Adji and M. Ali, "Penerapan Life Cycle Assessment untuk Kajian Dampak Lingkungan Pada Instalasi Pengolahan Air Bersih," *Jurnal Serambi Engineering*, vol. IX, no. 2, pp. 8905–8915, Apr. 2024.
- [2] B. Al Farisi and A. N. K. Movanita, "Air Bersih WTP Waduk Pluit Bisa Dikonsumsi, tapi Harus Dimasak Dahulu," *Kompas*, Mar. 02, 2023. [Online]. Available: <https://megapolitan.kompas.com/read/2023/03/02/18585931/air-bersih-wtp-waduk-pluit-bisa-dikonsumsi-tapi-harus-dimasak-dahulu>. [Accessed: Dec. 10, 2024].

- [3] V. Achmad et al., “Analisa QoS pada MQTT untuk website monitoring dan pengendalian pintu air,” *Jurnal Elektronika dan Otomasi Industri*, vol. 10, no. 3, pp. 379–386, Nov. 2023, doi: 10.33795/elkolind.v10i3.3865.
- [4] L. Nurfiqin et al., “Analisis Quality of Service (QoS) Protokol MQTT dan HTTP Pada Sistem Smart Metering Arus Listrik,” *REPOSITOR*, vol. 3, no. 1, pp. 121–130, Jan. 2021.
- [5] S. B. Bhaskoro, H. Supriyanto, B. B. Aji, B. Pamungkas, and P. M. Bandung, “Perbandingan Performansi Latency Protokol Komunikasi HTTP dan MQTT Pada Internet Of Things,” *JTI (Jurnal Teknologi Terapan)*, vol. 8, no. 2, pp. 82–89, Sep. 2022.
- [6] A. T. Alifibioneri, H. Nurwarsito, and R. Primananda, “Implementasi MQTT Websocket Pada Sistem Pendeteksi Detak Jantung,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 12, pp. 4217–4226, Dec. 2020.
- [7] M. R. Mufid et al., “Rancang Bangun Sistem Otomasi Mesin Pompa Air Tambak Menggunakan Protocol MQTT untuk Pengisian Air Tambak Saat Musim Kemarau,” *JTIT (Jurnal Teknologi Informasi dan Terapan)*, vol. 8, no. 2, Dec. 2021, doi: jt.it.v8i2.247..
- [8] M. F. Nadhiem, P. D. Ibnugraha, and A. P. Kurniawan, “Pengendali Raspberry Pi Jarak Jauh Menggunakan Protokol Mqtt Berbasis Web,” *eProceedings of Applied Science*, vol. 9, no. 4, p. 5626, Aug. 2023. [Accessed: Dec. 6, 2024].
- [9] E. Purnama, “Berkenalan Dengan Teknologi MQTT,” *Pujangga Teknologi*, Nov. 2018. [Online]. Available: <https://medium.com/pujanggateknologi/berkenalan-dengan-teknologi-mqtt-7e63cab9400d>. [Accessed: Dec. 6, 2024].
- [10] A. Jenar, "Analisis Perbandingan Protokol HTTP 2 dan HTTP 3 Dalam Perancangan Sistem Hibah Barang Kemenkes RI Berbasis WEB," Undergraduate Thesis, Universitas Hasanuddin, Makassar, 2022.
- [11] CD. Erivianto and A. Dani, “Training On The Use Of Programmable Logic Controller (PLC) To Improve Technician Competence At PT. Prima Multi Peralatan,” *Jurnal PEDAMAS (Pengabdian kepada Masyarakat)*, vol. 2, no. 3, pp. 762–769, May 2024.
- [12] M. Boers, *The PLCnext Runtime*, 2021. [Online]. Available: www.plcnext-runtime.com/print.html. [Accessed: Dec. 8, 2024].
- [13] Phoenix Contact, “AXC F 2152 - Controller,” Product Datasheet. [Online]. Available: <https://www.phoenixcontact.com/gb/products/2404267/pdf>. [Accessed: Dec. 9, 2024].
- [14] Phoenix Contact GmbH, “DB EN AXL F DI8/1 DO8/1 IH,” Mar. 2023. [Online]. Available: <http://phoenixcontact.net/product/2701916>. [Accessed: Dec. 9, 2024].
- [15] Phoenix Contact, *AXL F DI16/1 IH*, Product Datasheet. [Online]. Available: <https://www.phoenixcontact.com/gb/products/2688310>. [Accessed: Dec. 9, 2024].