Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

JURNAL TEKNIK INFORMATIKA
Homepage : http://journal.uinjkt.ac.id/index.php/ti

Starlink-Based IoT Network Performance Evaluation for Water Quality Monitoring in Remote Environments

Kasyful Amron[1*], Dany Primanita Kartikasari[2], Tiara Calista Kusumawardani Atarian[3], Archie Vian Nizam Efendi[4], Rayhan Egar Sadtya Nugraha[5], Maritza Aliyya Devy[6]

[1,2,3,4,5,6]Faculty of Computer Science, University of Brawijaya
[1,2,3,4,5,6] Jl. Veteran No.10-11, Ketawanggede, Lowokwaru, Malang, Indonesia

Article:
Accepted: February 10, 2026
Revised: December 01, 2025
Issued: April 30, 2026

***Correspondence Address:**
kasyful@ub.ac.id

ABSTRACT

Continuous water quality monitoring in remote and infrastructure-limited regions is constrained by the lack of reliable communication networks to support real-time IoT data transmission. Starlink offers a promising alternative due to its independence from terrestrial infrastructure, yet empirical evidence on its reliability as an end-to-end IoT communication backbone remains limited. This study therefore presents the design, implementation, and empirical network performance evaluation of a Starlink-based IoT system for real-time water quality monitoring. The proposed system integrates an ESP32 microcontroller with pH, total dissolved solids (TDS), and turbidity sensors, transmitting sensor data via a Starlink satellite link to a backend platform using the MQTT protocol with AES-128-GCM application-layer encryption. Received data are processed in Node-RED, stored in InfluxDB, and visualized through a Grafana real-time dashboard. Network performance was evaluated through five independent test iterations under both TCP and UDP transmission modes, measuring latency, jitter, packet loss, and throughput as key indicators of satellite link reliability for continuous IoT data transmission. The results demonstrate stable and reliable satellite connectivity, with latency consistently within 33–36 ms, jitter below 10 ms, zero packet loss across all configurations, and UDP throughput reaching up to 32.8 Mbps. TCP throughput was constrained to approximately 3.4–4.1 Mbps due to congestion control behavior over high-latency satellite links, a finding with direct implications for transport protocol selection in satellite-based IoT deployments. These results confirm that Starlink-based connectivity provides communication quality well in excess of the demands of periodic MQTT-based sensor transmission, demonstrating its feasibility as a reliable communication backbone for IoT-based water quality monitoring in environments where terrestrial infrastructure is unavailable or unreliable.

Keywords : *Satellite IoT; water quality monitoring; ESP32; Starlink; telemetry.*

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

1. INTRODUCTION

Access to safe and reliable water sources remains a major challenge in remote and infrastructure-limited regions. Communities in these areas often rely on unverified water sources whose quality may fluctuate due to environmental changes, contamination, or natural disturbances. The lack of continuous water quality monitoring can also increase public health risks, especially during emergency situations when clean water supply systems are disrupted. Recent developments in the Internet of Things (IoT) provide new opportunities to address this issue by enabling low-cost sensor systems that can continuously monitor water quality parameters [1]. However, the implementation of IoT-based monitoring systems depends heavily on the availability of reliable communication infrastructure to transmit data from monitoring locations to processing centers.

Several previous studies have developed IoT-based water quality monitoring systems using various communication technologies, including LoRaWAN [2], Wi-Fi and Bluetooth [3], and cellular networks [4]. In many remote regions, cellular networks have limited coverage, which may disrupt continuous data transmission. While long-range communication technologies such as LoRa can extend coverage and maintain low power consumption, large-scale deployment requires additional gateways, increasing system complexity and installation costs [5]. These limitations highlight the need for a communication solution that can operate independently of terrestrial infrastructure, particularly in remote or geographically isolated environments.

Satellite-based communication emerges as a compelling alternative to address these connectivity constraints. Low Earth Orbit (LEO) satellites offer significant advantages in global coverage, while having a lower latency [6] and lower transmission loss compared to Geostationary Earth Orbit (GEO) [7]. As a practical implementation of LEO satellites, the Starlink satellite network delivers high throughput with a low latency of 20–40 ms [8]. Starlink also demonstrates greater resilience to physical disruptions, such as floods or earthquakes, that can damage terrestrial communication infrastructures [9]. This resilience, combined with its ease of terminal deployment and automatic satellite tracking, makes Starlink suitable for sustaining IoT connectivity for rural monitoring deployments and scenarios where terrestrial networks are disrupted.

Despite offering solutions to coverage limitations, existing studies indicate that satellite-based IoT implementations present distinct challenges in terms of network performance. Prior research reports that Starlink exhibits non-congestive packet loss and latency variation related to satellite handover events [10-12], raising concerns about its reliability for continuous IoT data transmission. Furthermore, many existing evaluations of satellite-based IoT focus primarily on simulation or theoretical modeling, while empirical studies using real IoT hardware, operational LEO satellite networks, and end-to-end application protocols remain limited [13–15]. Moreover, network performance in IoT deployments is not determined solely by transport-layer behavior, application-layer factors such as messaging protocol selection and transmission mode can influence the communication characteristics. Yet empirical studies that systematically assess Starlink network behavior under realistic IoT traffic conditions, including evaluation across both TCP and UDP transmission modes, remain scarce. Most existing work also does not contextualize network measurements within a complete operational IoT stack, leaving a gap in understanding how satellite connectivity performs when integrated with a complete IoT application stack.

To address this gap, this study designs and implements an IoT-based water quality monitoring system utilizing Starlink satellite connectivity as the primary communication link. The system integrates a microcontroller-based sensing node equipped with water quality sensors to collect environmental measurements and transmit data to a backend monitoring platform through the MQTT messaging protocol with AES-128-GCM encryption. The received data are processed and stored in a time-series database and subsequently visualized through a real-time monitoring dashboard. The objectives of this research are as follows:

1. To design and implement an IoT-based water quality monitoring node capable of transmitting sensor data securely

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

through a Starlink satellite communication link; and
2. To evaluate the network performance of the Starlink-based communication link for real-time IoT data transmission, including latency, jitter, packet loss, and throughput.

By integrating a practical IoT sensing system with an operational LEO satellite network, this study provides empirical insights into the feasibility and performance characteristics of Starlink-based connectivity for water quality monitoring in remote regions.

2. METHODS

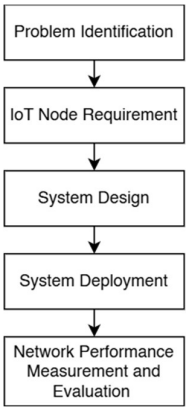

Figure 1. *Research methodology*

Figure 1 illustrates the research methodology adopted in this study to develop and evaluate a Starlink-based IoT water quality monitoring system. The methodology begins with problem identification, focusing on connectivity limitations in remote and infrastructure-limited areas where continuous water quality monitoring is required. This is followed by an IoT node requirement analysis stage, in which functional requirements for the IoT node are defined. Based on these requirements, the system design stage specifies the architecture of the IoT node and the backend data processing pipeline. The designed system is then deployed using an ESP32-based sensor node and Starlink satellite connectivity. After deployment, network performance measurements are conducted to evaluate the reliability of satellite-based communication.

2.1. IoT Node Requirement

The sensor node is designed to monitor pH, turbidity, and total dissolved solids (TDS) as key water quality parameters [16-17]. These parameters represent fundamental physical and chemical characteristics of water. The pH parameter indicates water acidity or alkalinity, which affects chemical stability and suitability for consumption. Turbidity reflects the presence of suspended particles that may indicate contamination, while TDS represents the concentration of dissolved substances influencing taste, conductivity, and overall water quality. The combination of these parameters provides a practical basis for continuous water quality monitoring.

Table 1. *Functional Requirement of IoT Node*

Functional Requirement	Description
Sensor data acquisition	The sensor node is capable of periodically acquiring water quality parameters using pH, TDS, and turbidity sensors.
Data formatting and encryption	The sensor node is capable of transmitting sensor data in JSON format encrypted using AES-128-GCM before being sent to the MQTT broker.
Wi-Fi connectivity	The sensor node is capable of managing Wi-Fi connectivity to the Starlink router, including automatic reconnection when the connection is lost.
MQTT management	The sensor node is capable of establishing and automatically reconnecting to the MQTT broker, subscribing to control topics, and receiving remote downlink commands to modify transmission intervals, broker addresses, and encryption settings without device reconfiguration.
Automatic system recovery	The sensor node is capable of automatically restarting the system when no data transmission occurs within a predefined time threshold.

Based on these sensing requirements, the functional requirements define the essential capabilities that must be fulfilled by the sensor node. These include periodic sensor data acquisition, basic data processing and formatting, and data transmission to MQTT brokers. In addition, the sensor node is required to support data security mechanisms, network connectivity management, bidirectional communication using the MQTT protocol, and automatic system recovery to ensure stable operation without continuous physical

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

intervention. A detailed description of the functional requirements is summarized in Table 1.

2.2. System Design

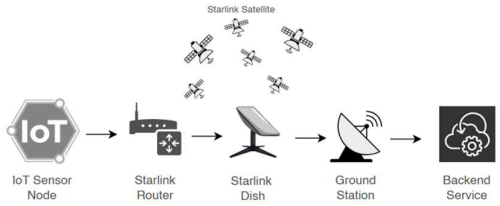

Figure 2. *End-to-End Communication Architecture*

The design integrates an IoT sensor node, Starlink-based communication, and a backend processing platform as a single end-to-end prototype system, as illustrated in Figure 2. The IoT node is responsible for acquiring water quality parameters and preparing data for transmission, while Starlink communication provides network connectivity. On the backend side, the system design defines a data processing pipeline consisting of message brokering, data processing, time-series storage, and visualization.

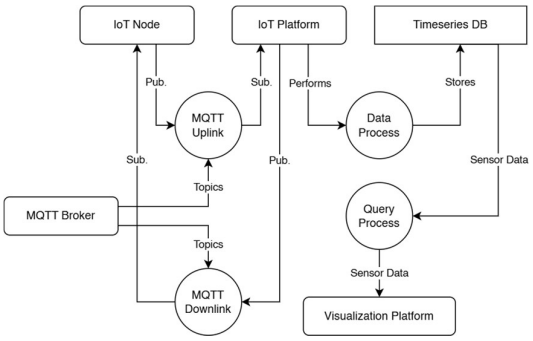

Figure 3. *Data processing pipeline*

The data processing pipeline consists of an MQTT broker, Node-RED, and InfluxDB, as illustrated in Figure 3. Sensor data acquired by the IoT node are transmitted to the backend system via a Starlink satellite Internet connection using the MQTT protocol over TCP/IP. The MQTT broker functions as the primary data acquisition layer, handling asynchronous message delivery and ensuring reliable data reception under variable satellite network conditions. Incoming MQTT messages are processed using Node-RED, which subscribes to predefined topics and parses the received JSON payloads. Within Node-RED, the data undergo validation, timestamping, and enrichment with device identifiers before being forwarded to the storage layer. Processed data are stored in InfluxDB to support efficient storage and retrieval of historical water quality data as well as network performance metrics, enabling subsequent analysis and visualization.

For visualization, Grafana was integrated with InfluxDB to provide real-time dashboards displaying pH, TDS, turbidity, and selected system performance metrics. This visualization method enables continuous remote monitoring and historical analysis of water quality, ensuring that system behavior and environmental changes can be observed effectively in remote or infrastructure-limited deployment scenarios.

Data security is implemented using AES-128-GCM symmetric encryption at the application layer. A 128-bit encryption key and a 96-bit initialization vector (IV) are statically provisioned on the ESP32 firmware using the Arduino Cryptography Library. Prior to transmission, the JSON payload containing sensor measurements is encrypted on the node, and the resulting ciphertext along with a 128-bit GCM authentication tag is published to the MQTT broker. On the backend side, Node-RED performs decryption using the same pre-shared key, allowing payload integrity verification through the GCM authentication tag before the data is forwarded to InfluxDB. This approach ensures confidentiality and integrity of sensor data in transit over the satellite link. Key distribution is managed through static provisioning, which is appropriate for a controlled prototype deployment where physical access to devices is restricted.

2.3. System Deployment

The IoT sensor node was assembled into a compact and protective enclosure to support deployment in rural and outdoor environments. The enclosure was designed to protect electronic components from dust, moisture, and mechanical disturbances, allowing stable operation without continuous physical supervision. During deployment, the sensor probes were positioned to enable direct contact with the water samples while the main electronic components remained isolated within the enclosure.

Sensor calibration was performed prior to deployment to ensure measurement accuracy.

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

The pH sensor was calibrated using a two-point calibration procedure with standard buffer solutions of pH 4.0 and pH 7.0, which stores computed offset values in the ESP32 EEPROM for persistent calibration across power cycles. The TDS sensor was calibrated using a standard NaCl reference solution of 1382 ppm. Turbidity values are derived from a polynomial voltage-to-NTU conversion based on the sensor manufacturer's characterization curve. Temperature was assumed constant at 25°C during calibration and measurement, as no active temperature compensation module was integrated in this prototype. To verify system operation under controlled conditions, the deployed node was tested using water samples prepared with the calibration solutions to represent different water quality parameter levels.

2.4. Network Performance Measurement and Evaluation

After the system was developed, network performance testing was conducted to characterize the communication behavior of the Starlink network as the IoT communication backbone. Performance measurements focused on four primary metrics: packet loss, latency, jitter, and throughput. Packet loss was measured to determine the reliability of data delivery from the IoT node to the backend system. Latency was evaluated to observe end-to-end transmission delay, jitter was evaluated to measure variations in packet transmission delay, while throughput was measured to characterize the maximum data transmission.

Network performance monitoring within Node-RED is implemented through a ping-pong mechanism for Round-Trip Time (RTT) measurement and throughput testing for bandwidth characterization. Ping tests are initiated via a dashboard button that sends a timestamped message to the node; the node then responds with a pong message containing the original timestamp, the ESP32 processing time, and system metrics such as RSSI and heap memory. A Node-RED function node subsequently calculates the RTT latency and jitter. Simultaneously, throughput tests are executed by having the node transmit continuous packet streams for a configured duration. This generates a summary message including total bytes transmitted and the calculated throughput, which is processed by a Node-RED handler to perform validation, packet loss rate calculation, and storage in InfluxDB.

Each test iteration consisted of a 100-second continuous packet transmission session, with five independent iterations performed per configuration. Two packet size configurations were evaluated: a default payload size reflecting realistic IoT sensor data transmission, and a fixed-size packet configuration targeting a 10 Mbps load to characterize the maximum throughput capacity of the satellite link. Both connection-oriented (TCP) and connectionless (UDP) transmission modes were evaluated to provide a comprehensive characterization of the Starlink link behavior under different transport protocol conditions. All test sessions were conducted under consistent environmental and weather conditions, with the Starlink terminal placed in the same location throughout the measurement campaign, to minimize external variability between iterations.

3. RESULTS AND DISCUSSION

This section presents the experimental results obtained from the deployment of the water quality monitoring. The discussion is organized into four parts: the implemented IoT sensor node and its firmware behavior, the backend data processing pipeline including remote configuration and control, the real-time data visualization dashboard, and the network performance evaluation of the Starlink-based communication link. These components demonstrate the feasibility of an end-to-end satellite-based IoT system for continuous water quality monitoring in remote regions.

3.1. IoT Node Sensor

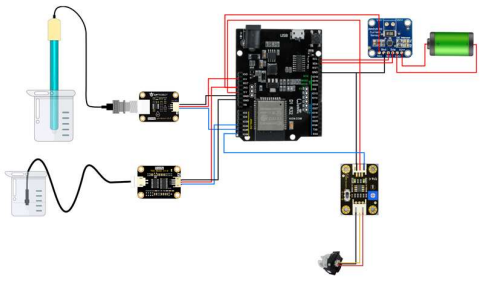

Figure 4. *Pin mapping diagram*

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

Table 2. *Sensor Specifications*

Sensor	Parameter	Function
Gravity pH (SEN0161-V2)	pH	Acidity/ alkalinity indicators
Gravity TDS/EC (SEN0244)	TDS (ppm)	Total dissolved solids
Gravity Turbidity (SEN0189)	Turbidity (NTU)	Suspended particulates

The water-quality sensing node serves as the primary data acquisition unit in the proposed prototype system. As illustrated in Figure 4, the node integrates pH, TDS, and turbidity sensors connected to an ESP32 microcontroller through analog input interfaces to enable in-site water quality measurement. This pin mapping configuration ensures proper signal acquisition and compatibility with the microcontroller's analog-to-digital converter. The selected sensors and their corresponding measured parameters are summarized in Table 2, which presents the function of each sensor in representing key physical and chemical indicators of water quality.

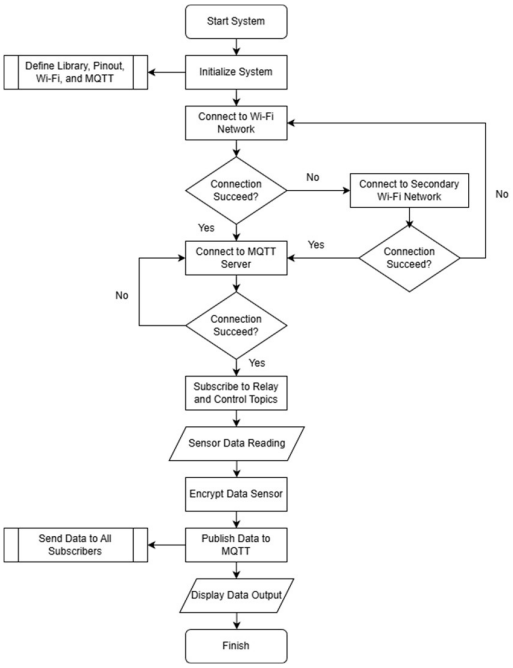

Figure 5. *Firmware node flowchart*

The firmware workflow begins with a comprehensive initialization sequence upon device power-up, where the system configures the ADC pins, the I2C communication bus, and the pH, TDS, and Turbidity sensors. Following hardware setup, the device initiates a Wi-Fi connection strategy by attempting to join a primary SSID; if unsuccessful, it automatically switches to a secondary SSID. Should both connection attempts fail, the system triggers an automatic restart to restore functionality. Once a network connection is established, the firmware connects to the primary MQTT broker and, if a dual-broker mode is enabled, establishes a simultaneous connection to a secondary broker. In this stage, the device subscribes to various control topics, including parameters for delay, broker addresses, and encryption settings.

In the main execution loop, the firmware continuously monitors Wi-Fi and MQTT connectivity, triggering reconnection routines if any link is dropped. Data acquisition cycles periodically read values from the pH, TDS, and turbidity sensors, and the collected data is transmitted to the backend through the MQTT messaging protocol. When security features are enabled, the JSON payload is encrypted using AES-128-GCM prior to transmission to ensure secure data delivery. The resulting hardware assembly is shown in Figure 6, IoT sensor node enclosed within a protective casing designed to shield the electronic components from dust, moisture, and mechanical disturbances during outdoor deployment.

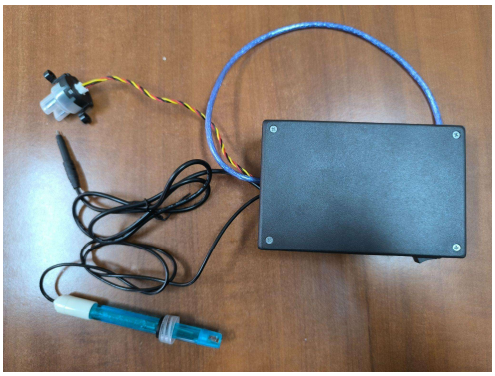

Figure 6. *Deployed IoT Sensor Node*

3.2. Backend Processing Pipeline

The backend data processing pipeline manages data ingestion, storage, processing, and visualization. The pipeline is built using Node-RED as the MQTT subscriber and orchestrator of data flow between system components, with sensor and performance data stored in an InfluxDB time-series database.

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

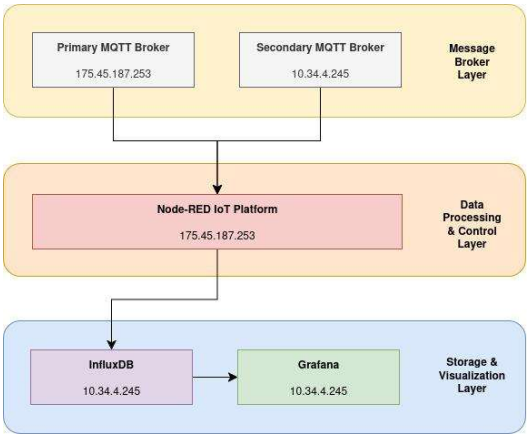

Figure 7. *Backend processing pipeline*

Figure 8 shows the Node-RED control pipeline used to manage both data processing and remote device control. In this pipeline, Node-RED subscribes to MQTT topics to receive sensor data and control commands, handling data parsing, routing, and forwarding to the storage layer while supporting bidirectional communication by publishing control messages back to the IoT node. These control messages enable remote configuration such as adjusting data transmission intervals, toggling encryption settings, and managing broker connections, allowing the sensor node to be reconfigured remotely without physical access.

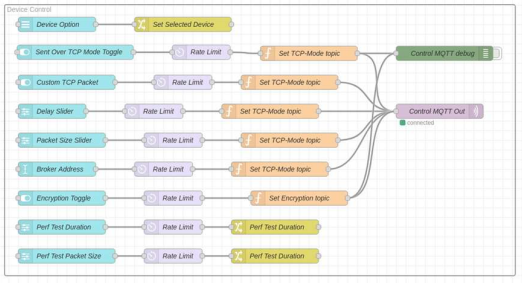

Figure 8. *Node-RED control pipeline*

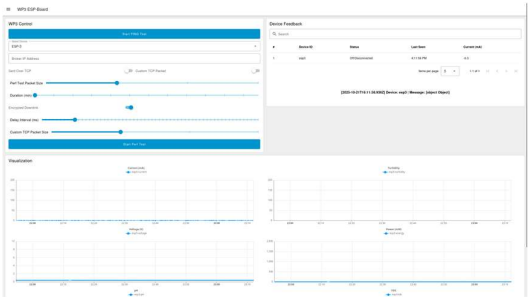

Figure 9. *Node-RED control dashboard*

The Node-RED dashboard provides a web-based interface centered around a Device Controls panel, featuring sliders for adjusting transmission intervals between 1,000 and 60,000 ms and text input fields for updating broker addresses. Users can also modify TCP packet sizes and execute commands such as status queries, ping tests, and throughput evaluations. All control widgets are integrated with a rate limiter that restricts outgoing commands to a maximum of one message per second to ensure system stability and prevent bandwidth overutilization.

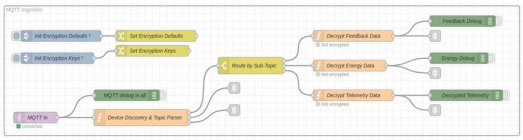

Figure 10. *Node-RED MQTT pipeline*

The MQTT ingestion pipeline subscribes to the wildcard topic wp3/+/# to aggregate messages from multiple devices. Upon reception, messages are routed through a parser function that extracts the device ID and message type, mapping them into telemetry messages undergo decryption and sensor data parsing before being forwarded in parallel to InfluxDB, CSV file writers for archival, and visualization links for real-time monitoring. There are also feedback messages that are directed to a debug terminal and notification panel for immediate operator monitoring.

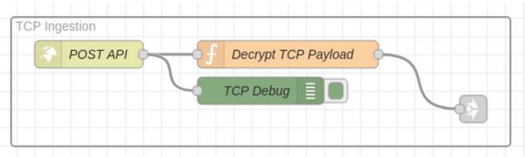

Figure 11. *Node-RED TCP ingestion pipeline*

To support network performance characterization, the system provides a dedicated HTTP endpoint that receives fixed-size packet streams transmitted by the IoT node during throughput test sessions. This endpoint operates independently from the primary MQTT-based sensor. An HTTP input node captures incoming packet payloads, which are processed by a function node for sequence validation, byte counting, and timestamp recording. A Node-RED handler then calculates the achieved throughput based on total bytes received over the test duration and stores the results in InfluxDB alongside other network

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

performance metrics. An HTTP response node returns a status code to the transmitting node to confirm successful packet reception, enabling packet loss rate calculation at the application layer.

3.3. Data Visualization Dashboard

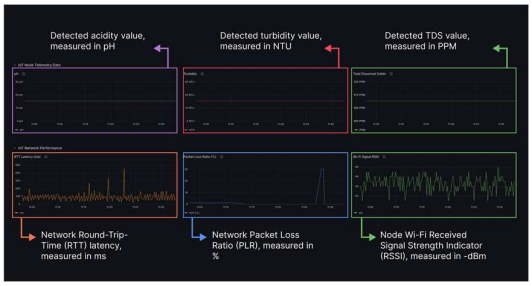

Figure 12. *Data visualization on Grafana*

The Grafana dashboard retrieves data from the InfluxDB bucket and presents visualization sections as shown in Figure 12. The IoT Node Telemetry Data section visualizes water quality parameters including TDS in parts per million, pH levels, and turbidity in Nephelometric Turbidity Units. While the Network Performance Metrics section presents latency RTT in milliseconds, packet loss rate, and Starlink Wi-Fi signal strength as RSSI in dBm. Each visualization panel employs Flux query language with time-range filtering and device ID-based grouping to enable multi-device monitoring.

3.4. Network Performance Analysis

Network performance monitoring within Node-RED is implemented through a ping-pong mechanism for Round-Trip Time (RTT) measurement and throughput testing for bandwidth characterization. Ping tests are initiated via a dashboard button that sends a timestamped message to the node; the node then responds with a "pong" message containing the original timestamp, the ESP processing time, and system metrics such as RSSI and heap memory. Simultaneously, throughput tests are executed by having the node transmit continuous packet streams for a configured duration. This generates a summary message including total bytes transmitted and the calculated throughput (bps/kbps/mbps), which is processed by a Node-RED handler to perform validation, packet loss rate calculation, and storage in InfluxDB. These entries include appropriate tags to facilitate historical trend analysis and correlation with varying network conditions.

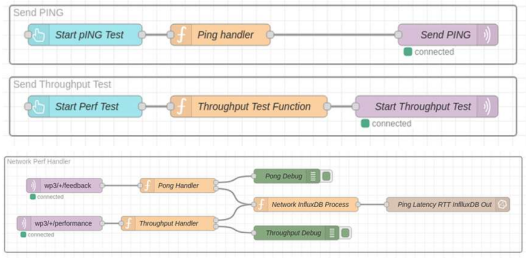

Figure 13. *Node-RED Measurement Pipeline*

Network performance evaluation was conducted through comprehensive testing protocols encompassing TCP and UDP link. Table 3 presents the aggregated network performance metrics obtained from five independent test iterations, providing statistical validation of network behaviour under varying protocol configurations.

Table 3. *Network Measurement Result*

Test Type	Iteration1	Iteration2	Iteration3	Iteration4	Iteration5	Mean	Stardart Deviation
Throughput TCP default (Mbps)	3.53	3.18	4.40	5.23	4.0	4.07	0.77
Throughput TCP 10 Mbps (Mbps)	3.44	3.42	3.44	3.51	3.0	3.36	0.20
Throughput UDP default (Mbps)	27.5	26.4	21.7	26.3	30.7	26.52	3.27
Throughput UDP 10 Mbps (Mbps)	29.5	32.2	28.6	28.5	32.8	30.32	2.02
Jitter (ms)	9.33	8.61	7.56	8.56	8.67	8.55	0.61
Latency (ms)	36.5	36.5	33.0	33.8	35.6	35.08	1.56
Packet loss (%)	0	0	0	0	0	0	0

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (172-182)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.50142

Throughput measurements indicate that the network is sufficient to support continuous data transfer for sensor-based water quality monitoring, where message payloads are relatively small but transmitted periodically. UDP throughput results demonstrate that the Starlink link is capable of delivering substantially higher data rates under connectionless transmission, reaching a mean of 26.52 Mbps under default payload conditions and 30.32 Mbps under the 10 Mbps target load configuration, with a peak observation of 32.8 Mbps. The low standard deviation across UDP iterations (2.02–3.27 Mbps) indicates consistent link behavior without significant throughput fluctuation between sessions. These figures confirm that the raw bandwidth capacity of the Starlink link is well in excess of what is required for continuous IoT sensor data transmission, where individual MQTT payloads typically range from tens to a few hundred bytes.

TCP throughput results, however, reveal a markedly different behavior. Both the default and 10 Mbps target load configurations produced similar mean throughput values of 4.07 Mbps and 3.36 Mbps respectively, despite the difference in configured load. This convergence can be attributed to the inherent behavior of TCP congestion control mechanisms over high-latency links. With a measured RTT of approximately 35 ms, the TCP congestion window expands more slowly compared to low-latency terrestrial networks, effectively limiting achievable throughput regardless of the configured target load, a well-documented phenomenon known as the bandwidth-delay product limitation. Additionally, the processing capacity of the ESP32 microcontroller in generating and transmitting continuous payloads introduces a node-side bottleneck that further constrains TCP throughput.

Latency measurements consistently fall within the range of 33.0–36.5 ms across all iterations, yielding a mean of 35.08 ms with a standard deviation of 1.56 ms. This level of stability indicates predictable and bounded transmission delay, which is essential for time-sensitive IoT monitoring applications where delayed sensor readings can affect the timeliness of observation and response. The low jitter mean of 8.55 ms further confirms that packet arrival intervals remain consistent, reducing the risk of out-of-order delivery or buffer overflow at the receiving end. Critically, a packet loss rate of 0% was recorded across all five iterations for both TCP and UDP configurations, demonstrating that the Starlink link maintains complete data delivery integrity under the tested conditions. For IoT applications where each sensor reading represents a discrete and irreplaceable measurement, zero packet loss eliminates the need for retransmission mechanisms or data recovery strategies at the application layer, simplifying node firmware design and reducing communication overhead.

From an application standpoint, these results collectively indicate that Starlink-based satellite connectivity is capable of supporting the full operational requirements of a continuous IoT water quality monitoring system. The bandwidth available under both TCP and UDP configurations substantially exceeds the demand of periodic MQTT-based sensor transmissions, providing significant headroom for scaling to higher sensor sampling rates, additional node deployments, or richer data payloads without approaching link saturation. The stability of latency and jitter further supports the use of event-driven communication patterns and real-time dashboard visualization, both of which depend on predictable message delivery timing. It is important to note, however, that these results were obtained under stable weather conditions and a fixed terminal placement; performance under adverse atmospheric conditions or during Starlink network congestion periods may differ and warrants further investigation.

CONCLUSION

This study demonstrates the feasibility of implementing a Starlink–based prototype for real-time water quality monitoring. The system integrates an ESP32-based sensing node equipped with pH, TDS, and turbidity sensors with AES-128-GCM application-layer encryption, MQTT-based data delivery, InfluxDB time-series storage, and Grafana real-time visualization into a complete end-to-end operational prototype. By integrating a low-cost ESP32-based senThe successful deployment demonstrates that continuous acquisition, secure transmission, and remote visualization of water quality parameters can be achieved under real operating conditions without dependence on terrestrial communication infrastructure, confirming the first research objective. Sensor

calibration using standard buffer solutions and the implementation of automated reconnection and remote reconfiguration mechanisms further demonstrate that the system can operate stably in environments where physical intervention is impractical.

Network performance evaluation conducted through five independent iterations under TCP and UDP transmission modes confirms the second research objective with quantitatively significant results. Zero packet loss was recorded across all iterations, latency remained consistently within 33.0 – 36.5 ms, and jitter was maintained below 10 ms, collectively indicating bounded and predictable transmission behavior suitable for continuous IoT monitoring workloads. UDP throughput reached a peak of 32.8 Mbps with a mean of 30.32 Mbps, while TCP throughput was constrained to a mean of 3.36–4.07 Mbps. A limitation attributable to TCP congestion control behavior over high-latency satellite links rather than insufficient link capacity, with direct implications for transport protocol selection in satellite-based IoT deployments.

These results demonstrate that Starlink-based connectivity provides communication quality well in excess of the demands of periodic MQTT-based sensor transmission, with substantial headroom for scaling to higher sampling rates or denser node deployments. It should be noted, however, that all measurements were conducted under stable weather conditions and fixed terminal placement; performance under adverse atmospheric conditions or network congestion remains uncharacterized. Future research may address this limitation by evaluating system performance under varying environmental conditions, implementing hybrid satellite-terrestrial architectures for improved resilience and extending the backend with intelligent anomaly detection to advance toward automated water quality assessment. The integration of energy harvesting and adaptive transmission strategies would further improve deployment sustainability, particularly given the relatively high power demand of Starlink terminals compared to conventional IoT gateway technologies.

REFERENCES

[1] M. Centenaro, C. E. Costa, F. Granelli, C. Sacchi, and L. Vangelista, "A Survey on Technologies, Standards and Open Challenges in Satellite IoT," IEEE Communications Surveys Tutorials, vol. 23, no. 3, pp. 1693–1720, 2021, doi: https://doi.org/10.1109/COMST.2021.3078433.

[2] W. A. Jabbar et al., "Development of LoRaWAN-based IoT system for water quality monitoring in rural areas," Expert Systems with Applications, vol. 242, p. 122862, May 2024, doi: https://doi.org/10.1016/j.eswa.2023.122862.

[3] R. Bogdan, C. Paliuc, M. Crisan-Vida, S. Nimara, and D. Barmayoun, "Low-Cost Internet-of-Things Water-Quality Monitoring System for Rural Areas," Sensors, vol. 23, no. 8, p. 3919, Jan. 2023, doi: https://doi.org/10.3390/s23083919.

[4] J. V. Abrajano, J. Nabua, J. Apanay, C. F. Pena, and K. A. Botangen, "IoT-Based Water Quality Monitoring System in Philippine Off-Grid Communities," 9th International Conference on Business and Industrial Research (ICBIR), pp. 0973–0978, May 2024, doi: https://doi.org/10.1109/icbir61386.2024.10875694.

[5] Y. Lalle, M. Fourati, L. C. Fourati, and J. P. Barraca, "Routing Strategies for LoRaWAN Multi-Hop Networks: A Survey and an SDN-Based Solution for Smart Water Grid," IEEE Access, vol. 9, pp. 168624–168647, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3135080.

[6] M. Jia and Q. Guo, "Editorial: Intelligent Cognitive Internet of Integrated Space and Terrestrial Things," *Mobile Networks and Applications*, Nov. 2019, doi: https://doi.org/10.1007/s11036-019-01406-4.

[7] I. Shayea et al., "Integration of 5G, 6G and IoT with Low Earth Orbit (LEO) Networks: Opportunity, Challenges and Future Trends," Results in Engineering, vol. 23, pp. 102409–102409, Jun. 2024, doi: https://doi.org/10.1016/j.rineng.2024.102409.

[8] X. Cao and X. Zhang, "SaTCP: Link-Layer Informed TCP Adaptation for

Highly Dynamic LEO Satellite Networks," IEEE Conference on Computer Communications, pp. 1–10, May 2023, doi: https://doi.org/10.1109/infocom53939.2023.10228914.

[9] Y. Zuo, M. Yue, H. Yang, L. Wu, and X. Yuan, "Integrating Communication, Sensing and Computing in Satellite Internet of Things: Challenges and Opportunities," IEEE Wireless Communications, vol. 31, no. 3, pp. 332–338, Mar. 2024, doi: https://doi.org/10.1109/mwc.019.2200574.

[10] Cao, X. and Zhang, X., "SaTCP: Link-Layer Informed TCP Adaptation for Highly Dynamic LEO Satellite Networks," IEEE Conference on Computer Communications, pp.1–10, 2023, doi: https://doi.org/10.1109/infocom53939.2023.10228914.

[11] Michel, F., Trevisan, M. and, Giordano, D. and Bonaventure, O., "A First Look at Starlink Performance," 22nd ACM Internet Measurement Conference (IMC'22), pp.130–136, 2022, doi: https://doi.org/10.1145/3517745.3561416.

[12] C. C. Chan, A. Al-Hourani, J. Choi, K. M. Gomez, and S. Kandeepan, "Performance Modeling Framework for IoT-over-Satellite Using Shared Radio Spectrum," Remote Sensing, vol. 12, no. 10, p. 1666, May 2020, doi: https://doi.org/10.3390/rs12101666.

[13] C. C. Chan, A. Al-Hourani, J. Choi, K. M. Gomez, and S. Kandeepan, "Performance Modeling Framework for IoT-over-Satellite Using Shared Radio Spectrum," Remote Sensing, vol. 12, no. 10, p. 1666, May 2020, doi: https://doi.org/10.3390/rs12101666.

[14] T. Hong, X. Yu, Z. Liu, X. Ding, and G. Zhang, "Narrowband Internet of Things via Low Earth Orbit Satellite Networks: An Efficient Coverage Enhancement Mechanism Based on Stochastic Geometry Approach," Sensors, vol. 24, no. 6, p. 2004, Mar. 2024, doi: https://doi.org/10.3390/s24062004.

[15] N. Mohan et al., "A Multifaceted Look at Starlink Performance," Proceedings of the ACM Web Conference 2024, vol. 12, pp. 2723–2734, 2024, doi: https://doi.org/10.1145/3589334.3645328.

[16] E. Hidayana, E. Setiawan, and A. I. Juniani, "Classification of water quality based on dissolved solids and turbidity parameters with the utilization of total dissolved solids sensor and turbidity sensor," Journal of Soft Computing Exploration, vol. 5, no. 3, pp. 231–239, Jul. 2024, doi: https://doi.org/10.52465/joscex.v5i3.376.

[17] R. Reddy, N. Salomi K, B. Sai, S. Pushpa, and E. Raju, "Analysis of Water Quality Parameters Across Diverse Sources," Journal of Pharma Insights and Research., vol. 2, no. 3, pp. 210–216, Jun. 2024, doi: https://doi.org/10.69613/3jxm7e23.