Architecture Image Studies
AIS Scientific Journal

Vol.6, Issue 4, pp.125-136, 2025
DOI: https://doi.org/10.62754/ais.v6i4.406

https://journals.ap2.pt/index.php/ais/index

IEC 61850-Based Data Modeling and Prototype Implementation for Hydrogen Refueling Stations Using libIEC61850 and Raspberry Pi

Sang-Ho Lee[1], Sung-Ho Hwang[2]

Abstract

Hydrogen refueling stations require stable communication protocols for real-time monitoring and control. The Modbus TCP, which is used in existing hydrogen refueling stations, has limitations owing to a lack of time synchronization and event search functions. This study addressed these issues by applying the IEC 61850 communication standard to hydrogen refueling stations. Using IEC 61850 data modeling techniques, we designed logical nodes and data objects for key equipment, including tube trailers, compressors, mid tanks, high tanks, dispensers, and valves. We developed a prototype that simulates a small-scale hydrogen refueling station environment using a Raspberry Pi and a network switch. Data modeling and communication were implemented using the libIEC61850 library, and time synchronization was performed using the IEEE 1588. We quantitatively analyzed the average and standard deviation of the transfer delay according to three data collection methods (individual, dataset, and reported) in a hydrogen refueling station using the IEC 61850 communication method. The experimental results were analyzed to compare the performances of the three data collection methods. Significant differences in transmission delay and standard deviation were observed depending on the collection method. The proposed framework can serve as the basis for the development of advanced hydrogen refueling station systems.

Keywords: *IEC 61850, Hydrogen refueling station, Data modeling, libIEC61850, Raspberry Pi.*

Introduction

With the spread of the hydrogen economy, the safe and reliable operation of hydrogen refueling stations has become essential. Currently, most hydrogen refueling stations use Modbus TCP-based communication methods; however, they lack time synchronization and event-recording functions, which limit their ability to perform advanced operations and real-time control. Therefore, this study aims to apply the IEC 61850 standard, which has been proven for real-time performance and reliability in the power industry, to hydrogen refueling stations and to examine its applicability by implementing data modeling and a prototype system.

Modbus TCP does not support accurate time synchronization and lacks the capability to manage timestamped event data such as hydrogen leak detection, compressor faults, pressure sensor anomalies, and emergency shut-off events. Although the existing Modbus TCP is simple to implement and widely used, it has several limitations [1].

-Lack of support for accurate time synchronization

-Insufficient event and disturbance record functions

-Absence of a standardized data model

-Lack of support for safe control

-Constraints on real-time performance and interoperability

Conversely, IEC 61850 is an international standard recognized for its real-time performance and reliability in the power utility field, providing the following functions:

[1] Department of Electronic and AI System Engineering, Kangwon National University, Samcheok, Korea, bondijun@hanmail.net, ORCID: https://orcid.org/0009-0008-6917-7273

[2] Department of Electronic and AI System Engineering, Kangwon National University, Samcheok, Korea, shhwang@kangwon.ac.kr, ORCID: https://orcid.org/0000-0002-2824-9399(corresponding author)

-Precise time synchronization based on IEEE 1588

-Timestamp-based event and data logging

-Standard data model based on logical nodes

-Support for real-time communication (Generic Object Oriented Substation Event (GOOSE), Manufacturing Message Specification (MMS), etc.) and automation control

-Interoperability between devices

Therefore, IEC 61850 is more suitable for hydrogen refueling station environments requiring highly reliable real-time control.

To satisfy these requirements, migration from Modbus TCP, which is widely used in hydrogen refueling stations, to IEC 61850 is necessary, and the scalability of future intelligent refueling station systems can be ensured.

The main equipment in hydrogen refueling stations performs various functions such as hydrogen gas leak detection, temperature and pressure measurements, current detection, and valve control. In this study, data modeling was performed based on the logical node and data object class standards defined in IEC 61850-7-4 for such equipment. Specifically, tube trailers, mid tanks, compressors, high tanks, dispensers, and valves were modeled, and a small-scale experimental environment was constructed using a Raspberry Pi and a network switch. The libIEC61850 library was used for communication implementation, TShark was used for data transmission verification, and the IEEE 1588 Precision Time Protocol was used for time synchronization.

Background and Related Work

IEC 61850 is an international standard developed for power utilities that provides an integrated framework encompassing data modeling, service definition, and communication protocol mapping. Previous studies have mainly focused on the power industry environment, and extremely limited studies have applied this standard to hydrogen refueling stations. Existing hydrogen refueling stations use Modbus TCP [2][3], and research on data modeling specific to hydrogen refueling stations and small-scale system implementations is limited. This study addressed the limitations of the previous research by attempting IEC 61850-based data modeling and prototype implementation. Fig. 1 depicts the hierarchical information modeling framework of IEC 61850, which begins with physical devices and progresses to logical devices, logical nodes, and data objects. Furthermore, it is designed to ensure real-time performance and reliability through Ethernet-based communication and to ensure interoperability.

Physical devices include various functional modules that are modeled as logical devices. Each logical device can perform various operations that are defined as logical nodes. The IEC 61850-7-4 standard defines 159 unique logical node classes. Logical nodes include data objects that represent application program functions. The logical node variables are represented as a set of common data classes (CDCs). The IEC 61850-7-3 standard defines the 40 CDCs. Each data object contains a set of elements called data attributes belonging to 12 functional constraints. Attributes contain values defined by common data attributes (CDAs) [4][5].

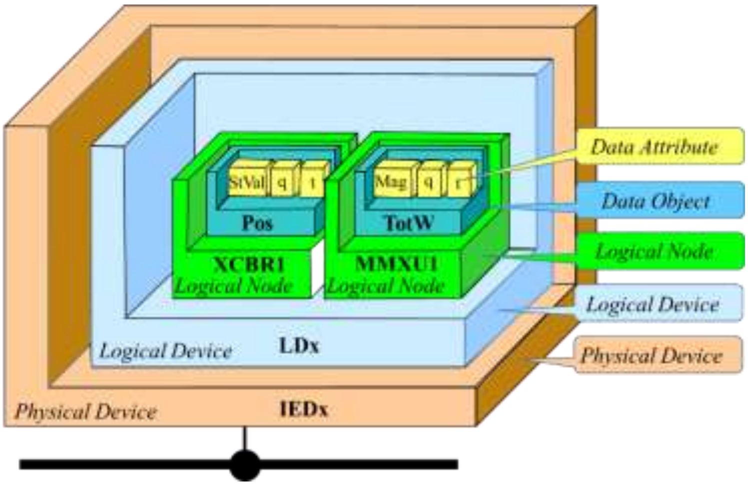


Fig. 1. Data modeling structure of IEC 61850

Previous studies have mostly limited the application of IEC 61850 to power plants and substations, with few examples of its application to new energy infrastructures, such as hydrogen refueling stations. In particular, a few studies have attempted to design customized data models for hydrogen refueling station equipment and implement testbeds based on them. This study aims to fill this gap by implementing a logic node design and a small-scale experimental system tailored to the hydrogen refueling station environment, thereby differentiating it from previous studies.

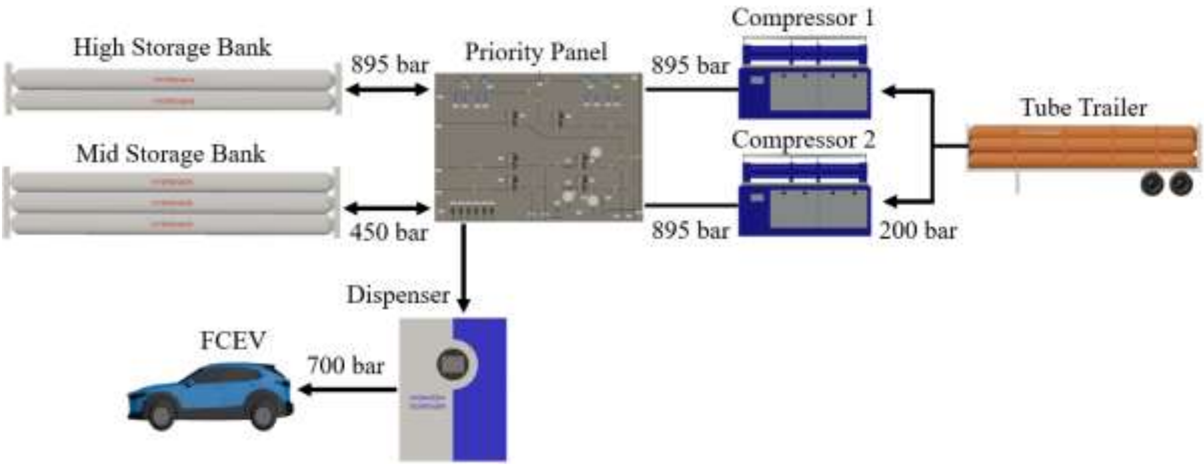


Fig. 2. Process Flow Diagram of a Samcheok Hydrogen Refueling Station

Fig. 2 shows the process flow and layout of the Samcheok hydrogen refueling station [2][3].

The tube trailer transports hydrogen to a hydrogen unloading station connected to an unloading hose. Hydrogen is fed from the tube trailer into the hydrogen compressor at 200 bar, pressurized, and stored in a gaseous hydrogen tank. The high tank stores hydrogen at 895 bar, and the mid tank stores hydrogen at 450 bar. When refueling a hydrogen electric vehicle, the dispenser uses the pressure difference to inject gaseous hydrogen at 700 bar into the vehicle.

The operating pressures and capacities of the equipment at the Samcheok hydrogen refueling station are listed in Table 1.

Table 1. Equipment Conditions [4]

Name	Pressure	Capacity
Tube trailer	200bar	4,833.045
Compressor	895bar	1,090 Nm^3/hr x 2 ea
Mid tank	450bar	1,189 L X 3 ea
High tank	895bar	553 L X 2 ea

Dispenser	700bar	2,180 Nm^3/hr

System Design and Implementation

Hydrogen refueling stations must monitor and control various physical parameters such as pressure, temperature, flow rate, and gas leak detection in real time. In particular, accurate time synchronization (IEEE 1588 support) and timestamp recording when events occur are essential. In addition, fast peer-to-peer communication between devices and reliable control command transmissions are required. To satisfy these requirements, IEC 61850-based data modeling and communication design are necessary for substation automation.

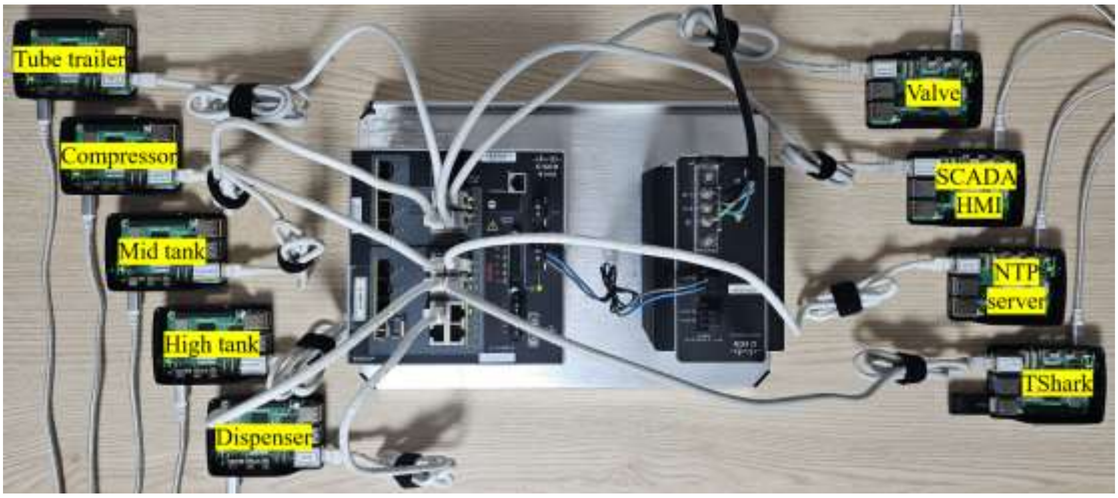


Fig. 3. Hardware System

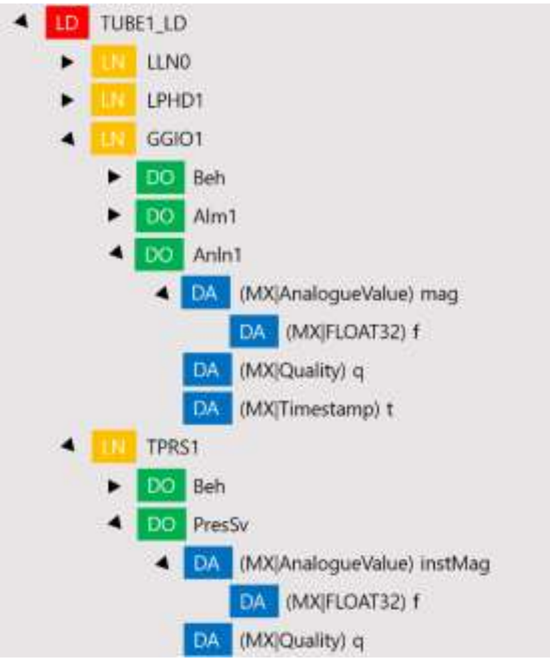
In this study, we designed an IEC 61850-based communication architecture for key equipment in hydrogen refueling stations. We simulated key equipment, such as tube trailers, compressors, mid tanks, high tanks, dispensers, and valves, using Raspberry Pi and configured the network using Cisco IE4000 switches. A time synchronization server using IEEE 1588 was configured to implement a structure similar to that of an actual industrial communication environment on a laboratory scale.

The hardware system is illustrated in Fig. 3. Six Raspberry Pi units were used to simulate the hydrogen refueling station equipment, one unit was used for the SCADA HMI functions, one unit was used as an IEEE 1588 server, and one unit was used as a TShark packet capture device, resulting in a total of nine Raspberry Pi units. The libIEC61850 open-source library was used to build an IEC 61850 client, and the dataset and Report Control Block (RCB) functions were set up to implement data collection and communication. All the Raspberry Pi devices were configured to maintain IEEE 1588-based time synchronization within the same network.

The experiment was conducted using a Raspberry Pi 5 Model B (8GB RAM) device running on a Raspbian operating system. IEEE 1588 Precision Time Protocol (PTP) servers and clients were configured to verify time synchronization. TShark was used to capture the transmitted and received MMS messages and analyze the reporting cycle and data transmission accuracy. All devices were directly connected to Gigabit Ethernet to minimize network latency.

The IEC 61850 information model abstracts actual devices to provide simplified descriptions and defines standardized syntax, semantics, and hierarchical structures for data exchange between heterogeneous devices and systems. In this standard, information modeling is implemented by defining logical nodes and data objects, where logical nodes comprise a set of data objects that perform specific functions. Combinations of multiple logical nodes form logical devices that can interact according to rules specified in the standard [6].

In accordance with the IEC 61850-7-4 standard, appropriate logical nodes and data objects were mapped onto each piece of equipment at the hydrogen refueling station. The tube trailer uses TPRS.PresSv for pressure data and GGIO.AnIn for gas concentration data. The compressor maps the pressure, temperature, and current data to the TPRS, TTMP, and GGIO nodes, respectively, and the mid tank, high tank, dispenser, and valve perform data modeling using appropriate logical nodes according to their functions.



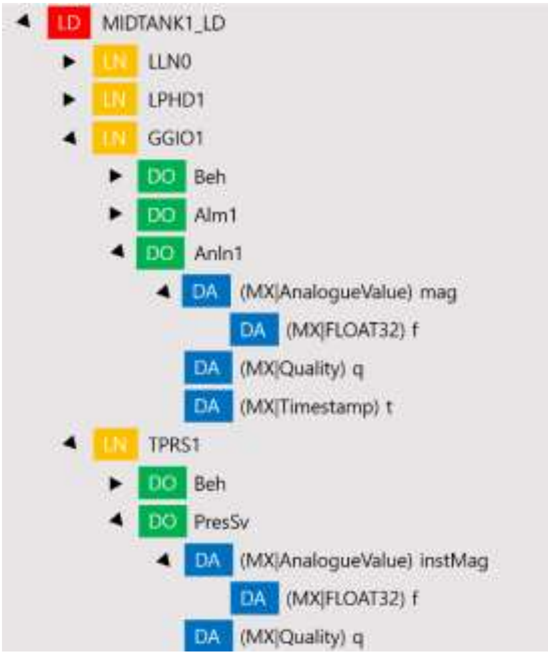


Fig. 4. IEC 61850 data modeling for tube trailer Fig. 5. IEC 61850 data modeling for mid tank

Data modeling of the tube trailer was performed, as shown in Fig. 4. It includes the existing LLN0 and LPHD logic nodes. The AnIn data object of the GGIO logic node is used to measure the hydrogen gas concentration, whereas the hydrogen gas pressure information is expressed through the PresSv object of the TPRS logic node.

Data modeling for the mid tank is shown in Fig. 5, which includes LLN0 and LPHD as bases, with the hydrogen gas concentration represented using GGIO. AnIn and hydrogen gas pressure information is represented using TPRS.PresSv.

The data modeling for the compressor is shown in Fig. 6 and includes LLN0 and LPHD. GGIO.AnIn was used for the hydrogen gas concentration and current measurements, TPRS.PresSv was used for the hydrogen gas inlet and outlet pressures, and TTMP.TmpSv was used for the inlet/outlet hydrogen gas and oil temperatures.

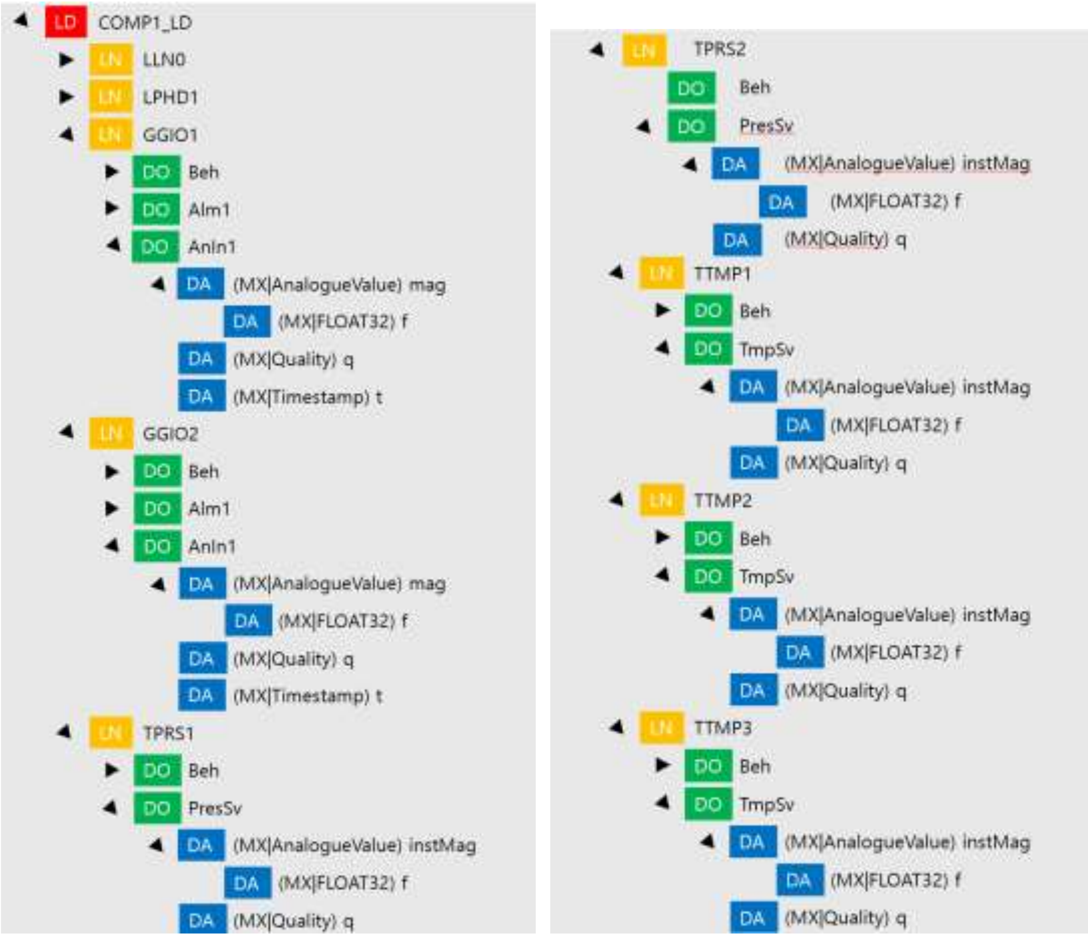
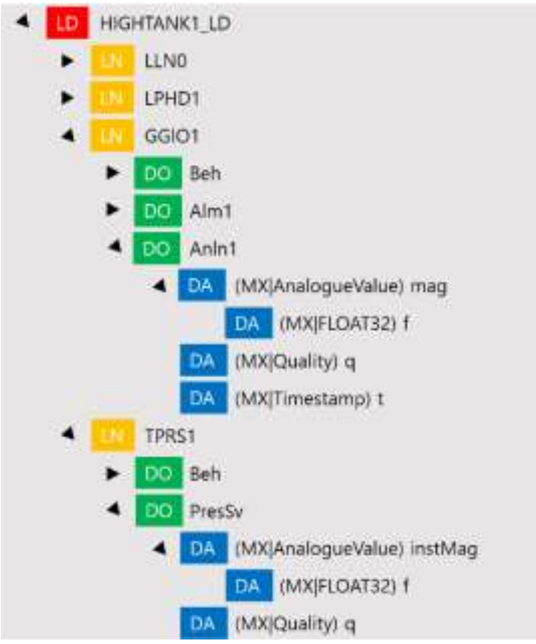


Fig. 6. IEC 61850 data modeling for compressor



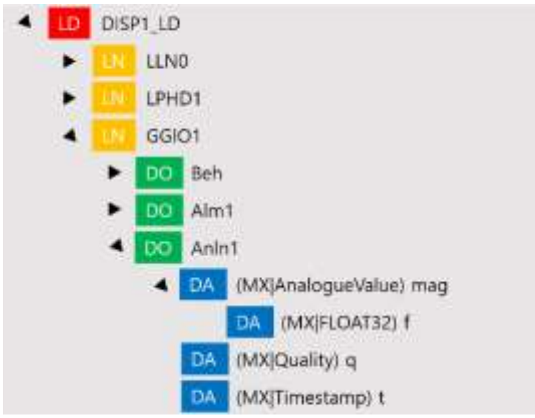


Fig. 7. IEC 61850 data modeling for high tank **Fig. 8. IEC 61850 data modeling for dispenser**

As shown in Fig. 7, the high tank includes the LLN0 and LPHD. Data modeling was configured to include GGIO.AnIn for hydrogen gas concentration measurements and TPRS.PresSv for hydrogen gas pressure data.

As shown in Fig. 8, the dispenser includes the LLN0 and LPHD. The hydrogen gas concentration was modeled using the AnIn object of the GGIO logic node.

Various types of valves are installed in the priority panel inside the hydrogen refueling station. The results of the data modeling for these valves are shown in Fig. 9. In addition to the LLN0 and LPHD logic nodes included in the modeling, the PosVlv data object of the KVLV logic node was used to represent the open/closed status of the valves.

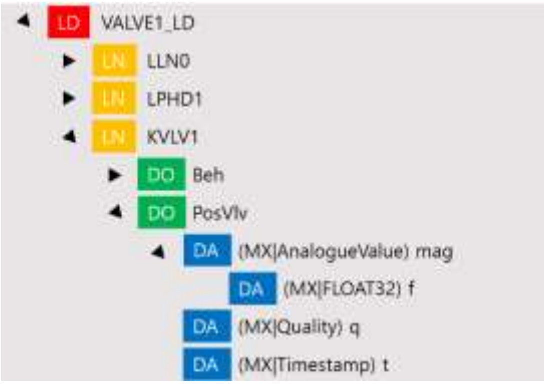


Fig. 9. IEC 61850 data modeling for valve

Performance Evaluation

In this study, a hardware system was used to measure the data transfer delay, and time synchronization was achieved using ptp4l, a time synchronization daemon implemented on a Linux system based on a Precision Time Protocol (PTP, IEEE 1588). The offset, frequency, and path delay output of ptp4l are key indicators that precisely represent the time synchronization status between the master and slave clocks via the PTP. In Fig. 10, the offset represents the time difference between the master and slave clocks in nanoseconds (ns), indicating the current operating speed of the slave clock. The frequency in Fig. 11 indicates the extent to which the slave clock adjusts its frequency to match that of the master clock, typically measured in parts per billion (ppb). The path delay in Fig. 12 refers to half of the round-trip communication delay time between the master and slave (one-way delay time) and is a correction value that considers the network transmission delay to accurately receive time information from the master. These three values play important roles in evaluating and tuning the accuracy of PTP-based time synchronization. In this study, time synchronization entered all stabilization stages (offset, frequency, and path delay) after 75 or more packets were generated.

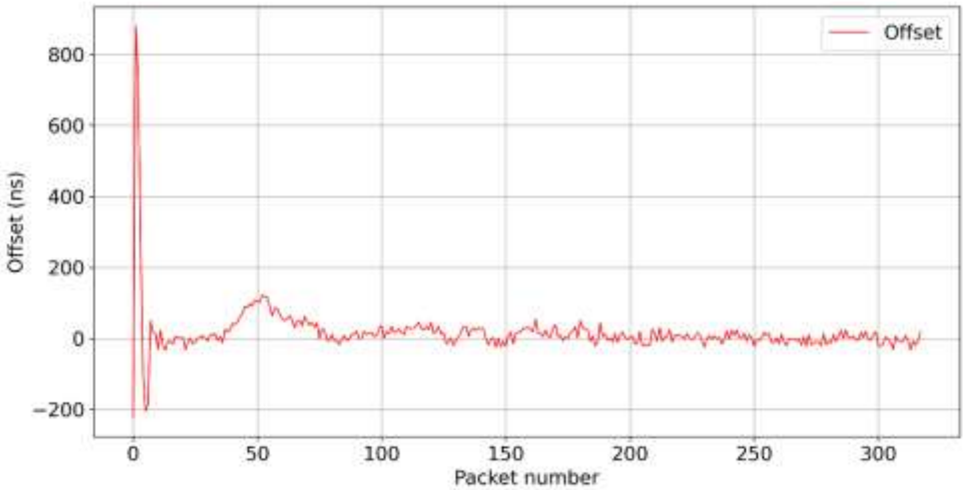


Fig. 10. PTP synchronization offset versus packet number

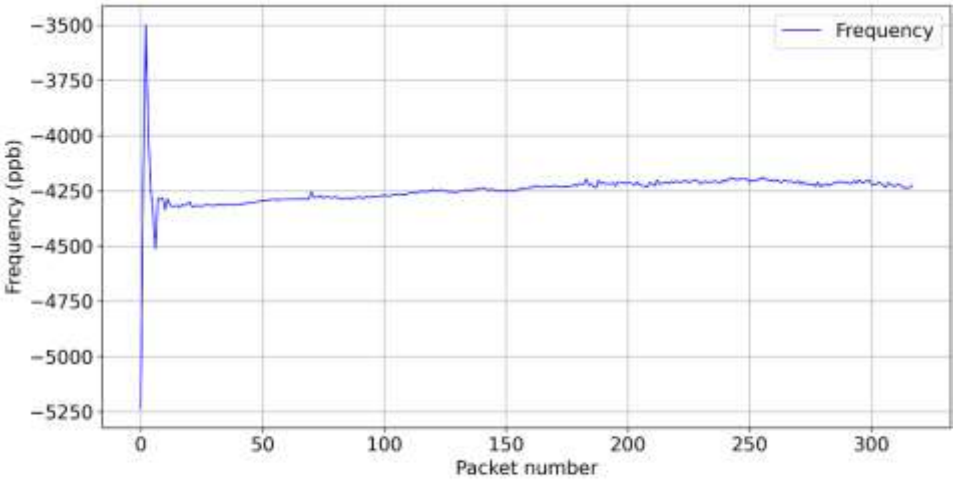


Fig. 11. PTP clock frequency adjustment versus packet number

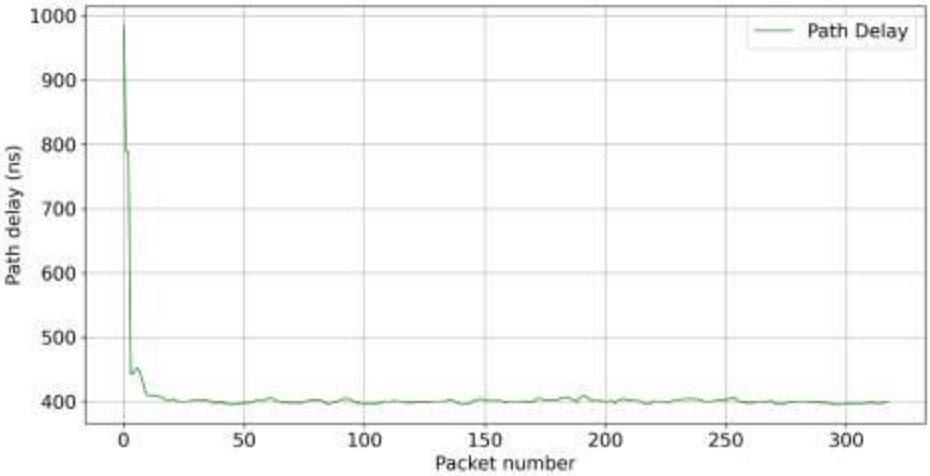


Fig. 12. PTP path delay variation versus packet number

In this experiment, the transfer delays were measured and compared for three data transmission methods (individual, dataset, and reported) in an IEC 61850-based communication structure.

The individual method involves the IEC 61850 client sequentially requesting and receiving sensing data from each device, resulting in repeated handshakes between data requests and responses. Consequently, the average transfer delay was the longest, suggesting that delays accumulate as the number of devices increases.

The dataset method involves an IEC 61850 client requesting and receiving multiple sensor data as a single dataset. This method minimizes the number of requests and allows multiple data to be included in the response packet, resulting in the shortest possible transfer delay. This demonstrates the effectiveness of the dataset method in environments with high real-time requirements.

The reported method is a structure in which the IEC 61850 server periodically transmits data without requests from the client, and the client-side software processes the data through an interrupt at the time of reception. However, because the Raspbian operating system installed on Raspberry Pi 5 B is not a real-time operating system, the reception delay is affected by the system scheduling, resulting in irregular fluctuations in the delay time. This method minimizes network traffic while maintaining a certain level of real-time performance, although the average delay time is longer than that of the dataset method.

These results are shown in the graph and table below, where the dataset method showed the lowest average transmission delay and dispersion, the reported method showed a pattern of high variability, and the individual methods showed the highest transmission delay values.

In a hydrogen refueling station environment requiring real-time performance, the dataset-based approach demonstrated the best performance, whereas the reported method is expected to improve the performance when used in a real-time OS environment. However, although the individual method has the advantage of a simple design, it may cause severe bottlenecks in transmission delays in large-scale systems; therefore, caution is required when using it.

The transfer delays for each device are shown in Figs. 13–18.

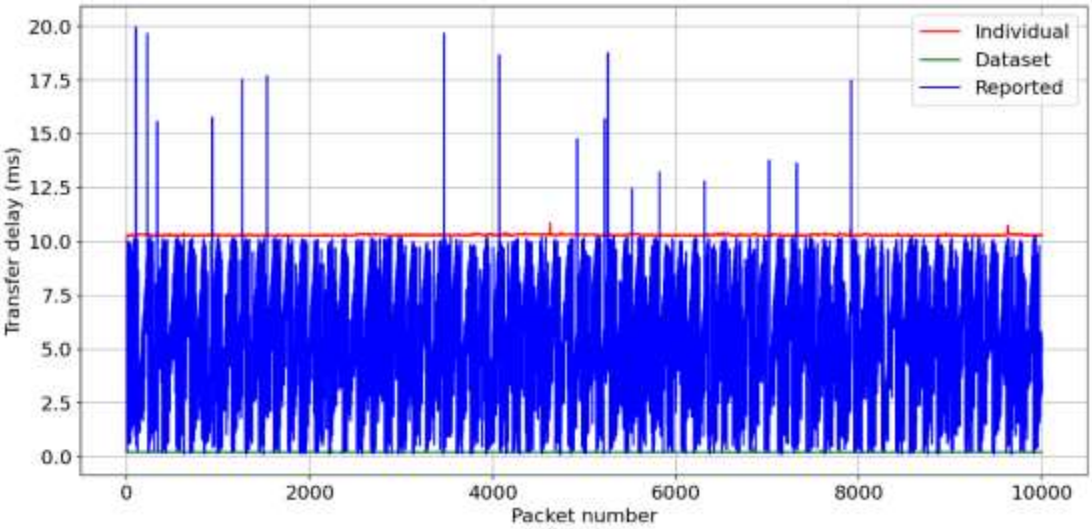


Fig. 13. Transfer delay comparison from tube trailer

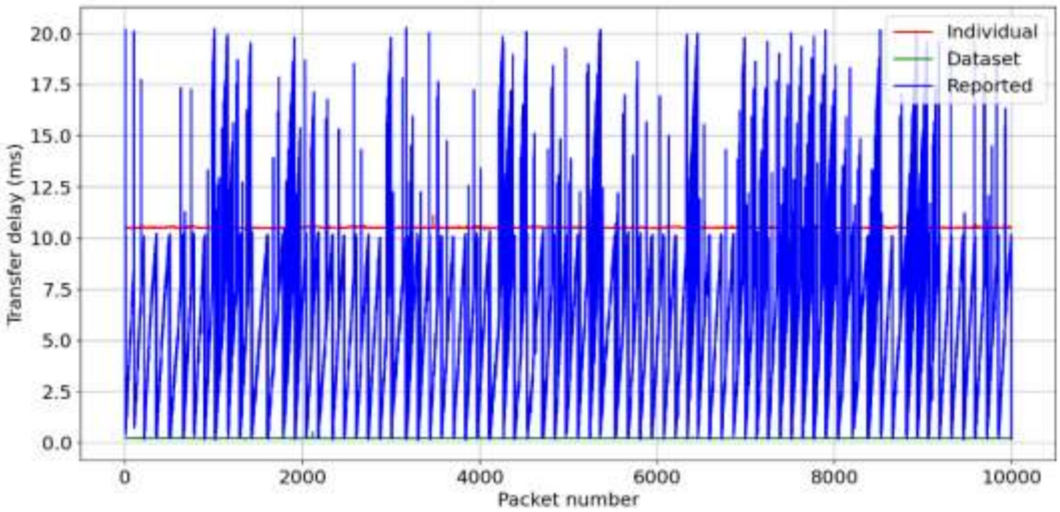


Fig. 14. Transfer delay comparison from mid tank

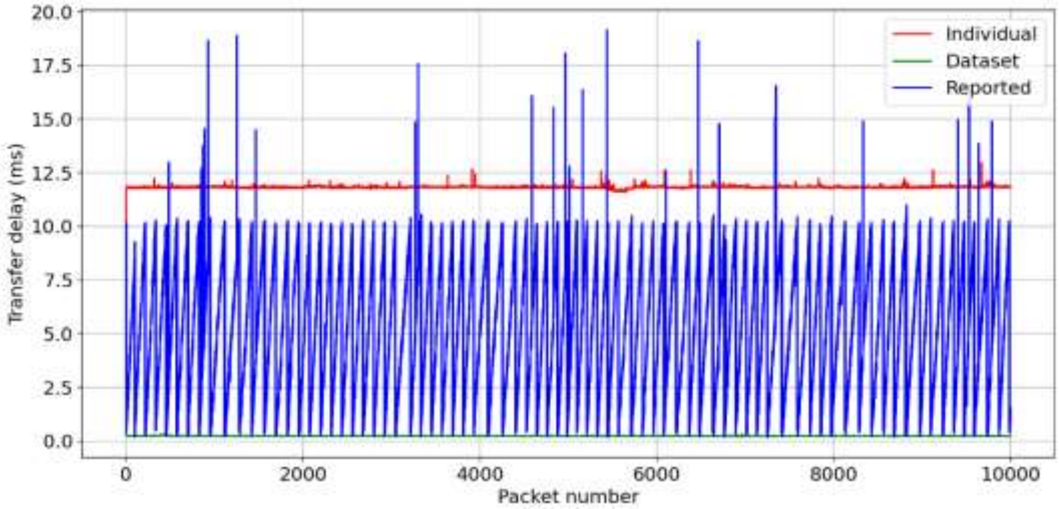


Fig. 15. Transfer delay comparison from compressor

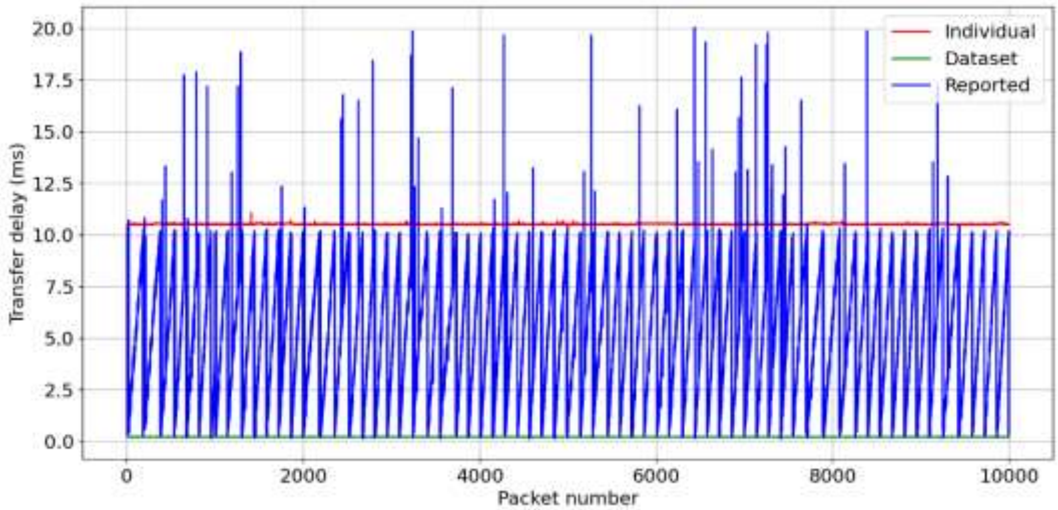


Fig. 16. Transfer delay comparison from high tank

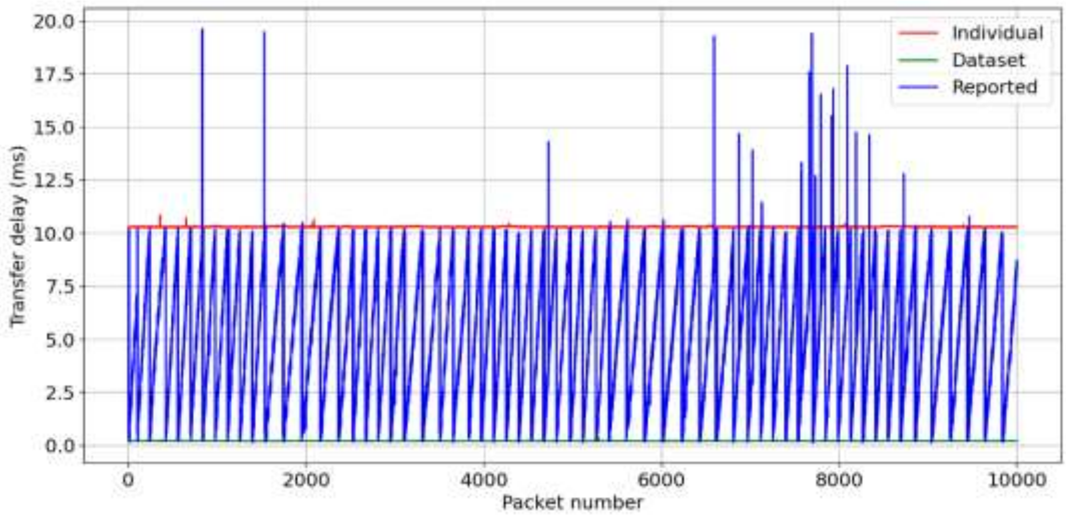


Fig. 17. Transfer delay comparison from dispenser

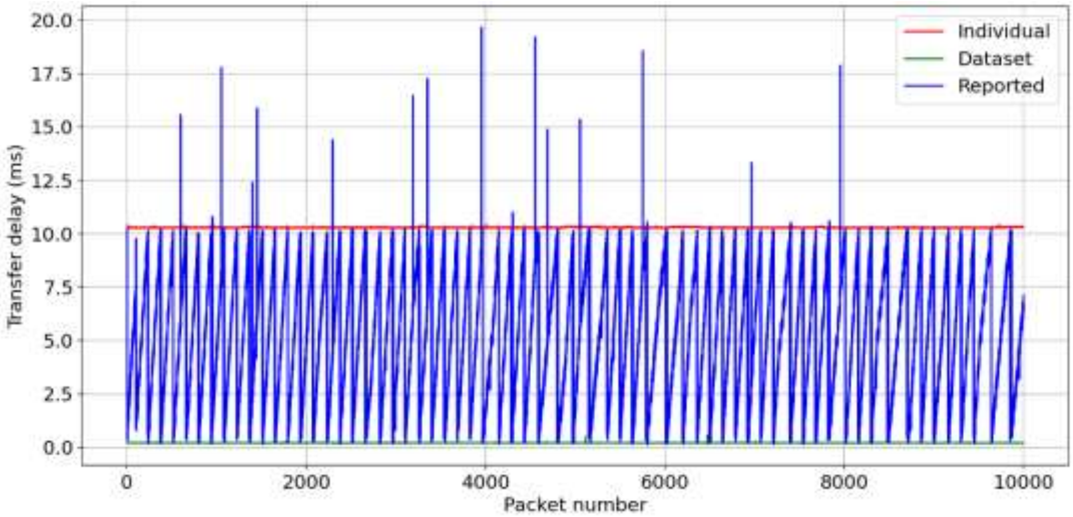


Fig. 18. Transfer delay comparison from valve

Table 2. Mean Values for Each Measurement Type

Component	Individual Mean (ms)	Dataset Mean (ms)	Reported Mean (ms)
Tube trailer	10.3024	0.2154	5.1821
Mid tank	10.5117	0.2168	5.8084
Compressor	11.8268	0.2449	5.2501
High tank	10.5240	0.2166	5.2287
Dispenser	10.2943	0.2123	5.1938
Valve	10.2950	0.2120	5.1984

Table 3. Standard Deviation Values for Each Measurement Type

Component	Individual Standard Deviation (ms)	Dataset Standard Deviation (ms)	Reported Standard Deviation (ms)
Tube trailer	0.1000	0.0038	2.9638
Mid tank	0.0997	0.0038	3.8573
Compressor	0.1084	0.0060	2.9452
High tank	0.0999	0.0040	3.0119
Dispenser	0.0992	0.0029	2.9270
Valve	0.0993	0.0060	2.9324

In this study, we analyzed the average and standard deviation of transfer delay according to three data collection methods (individual, dataset, and reported) at a hydrogen refueling station using the IEC 61850 communication method. The results are presented in Tables 2 and 3.

The individual method had an average transfer delay time of approximately 10.30–11.83 ms and a standard deviation of approximately 0.099–0.108 ms. This is because the individual method involves the client sequentially requesting and receiving data from each device. The transfer delay was generally high, with the compressor exhibiting the highest average transfer delay of 11.8268 ms. Its characteristics include an extremely small standard deviation (within approximately 0.1 ms) and a high transmission delay.

The dataset method had an average transmission delay time of approximately 0.212–0.245 ms and a standard deviation of approximately 0.0029–0.0060 ms. This method combined multiple sensor data into a single dataset that the client requested and received in bulk, showing the lowest transmission delay and standard deviation. Its characteristics include excellent transmission speed and temporal stability, rendering it suitable for high-speed and high-precision real-time monitoring.

The reported method has an average transmission delay of approximately 5.18–5.81 ms and a standard deviation of approximately 2.93–3.86 ms. The reported method is a structure in which the server periodically and automatically transmits the sensing data to the client. The transmission delay was higher than that of the dataset, but lower than that of the individual, and the standard deviation was exceedingly large, resulting in the highest delay variability. The reason for this high variability is that the operating system (Raspbian) of the Raspberry Pi 5 B is not a real-time OS; therefore, the timing of software interrupt handling is inconsistent, causing issues. Its average performance is average, but in systems requiring real-time performance, variability can be a risk.

The dataset method exhibited the best performance in terms of both average transmission delay and delay deviation and is considered the optimal method for real-time monitoring at hydrogen refueling stations. The reported method can be an effective transmission structure when the operating system is in real-time; however, caution is required in environments where performance prediction is difficult owing to high variability. The individual method has a simple transmission structure, but it has the highest latency and is unsuitable for large-scale systems.

These experimental results demonstrate that the IEC 61850-based communication structure can meet the real-time data processing requirements of hydrogen refueling stations and confirm that the system performance can vary significantly depending on the data collection method.

Conclusions

In this study, we performed data modeling of a hydrogen refueling station using a Raspberry Pi and libIEC61850, utilizing the logical nodes and data objects provided by IEC 61850. We constructed a small-scale laboratory model using a Raspberry Pi and a network switch. After performing IEC 61850 data modeling for the hydrogen refueling station, we collected data using three methods: individual, dataset, and reported. Using TShark, we confirmed that data traffic occurred accurately. The experimental results confirmed that the data were accurately collected between devices through the individual, dataset, and reported transmissions provided by IEC 61850. We calculated the mean and standard deviation for each collection method.

In this study, we proposed an IEC 61850 data modeling structure specialized for hydrogen refueling station environments. Furthermore, we proposed a prototype system implementation plan using low-cost hardware. Through experiments, we quantitatively evaluated the applicability and performance indicators of IEC 61850-based communication in hydrogen refueling station environments.

Future research directions include the following: The IEC 61850 data modeling approach using Raspberry Pi at hydrogen refueling stations described in this study can be useful for developing algorithms, applications, and prototypes that utilize IEC 61850 at hydrogen refueling stations.

References

[1] Patrick Montignies, Philippe Angays, and Laurent Guise, “IEC 61850 in the Oil and Gas Industries,” IEEE Industry Applications Magazine, Vol. 17, No. 1, pp. 36-46, January 2011. DOI:10.1109/MIAS.2010.939430

[2] Part 7 (7-1; 7-2; 7-3; 7-4): Modeling of the Data and Communication Services, IEC 61850.

[3] Part 8-1: Mapping of the Communication Services onto Ethernet, IEC 61850.

[4] Sung-Ho Hwang, “Implementation of a Hydrogen Leakage Simulator with HyRAM+,” The Journal of Convergence on Culture Technology (JCCT), Vol. 10, No. 1, pp. 551-557, January 2024. DOI:10.17703/JCCT.2024.10.1.551

[5] Sung-Ho Hwang, “Transmission for IEC 61850 Sampled Values Using Current and Voltage Sensors of Raspberry Pi,” The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 18, No. 6, pp.157-162, December 2018. DOI:10.7236/JIIBC.2018.18.6.157

[6] Singh, Shakti, Prachi Chauhan, Mohd Asim Aftab, Ikbal Ali, S. M. Suhail Hussain, and Taha Selim Ustun, "Cost Optimization of a Stand-Alone Hybrid Energy System with Fuel Cell and PV," Energies, Vol. 13, no. 5, pp.1-23, March 2020. DOI:10.3390/en13051295.