

Sending Smart Meter Data from a Data Concentrator Unit to a Head-End System via Ethernet

G Rajasekhara Yadav¹, R Appala Naidu², B Manikanta³, G Rajesh⁴, J Rama Ganesh⁵

^{1,2}Assistant Professor, Department of Electrical and Electronics Engineering, Avanthi Institute of Engineering & Technology, Tamaram, Makavarapalem, Anakapalle – 531113, Andhra Pradesh, India

^{3,4,5}UG Students, Department of Electrical and Electronics Engineering, Avanthi Institute of Engineering & Technology, Tamaram, Makavarapalem, Anakapalle – 531113, Andhra Pradesh, India

Abstract

This paper presents the design and implementation of a system for the efficient transmission of smart meter data from a Data Concentrator Unit (DCU) to a Head-End System (HES) over Ethernet. The smart meter used, an El Sewedy Electrometer EM101-2, communicates with the DCU using the DLMS/COSEM protocol. The DCU, built around a Rugged Board with an i.MX6UL-class processor, aggregates and processes data from multiple meters before forwarding it to the HES over a TCP/IP network, using MQTT, HTTP(S) or FTP as appropriate, giving real-time monitoring, control and analysis for utilities. Data security is maintained throughout with AES encryption and TLS/SSL protocols. Where the field link is wireless, the system operates in the 865–867 MHz band using frequency-hopping spread spectrum (FHSS) for interference resistance. The implementation was validated by measuring latency, signal strength, packet loss and data integrity between the DCU and HES; the results show that the combination of Ethernet backhaul, DLMS/COSEM data modelling and AES/TLS security gives a scalable, reliable link suitable for accurate, error-free utility billing.

Index Terms— Smart metering, Data Concentrator Unit (DCU), Head-End System (HES), DLMS/COSEM, Ethernet, MQTT, OBIS codes, AES encryption, Rugged Board.

I. Introduction

Smart metering uses advanced digital meters to monitor and record energy consumption in real time, giving both consumers and utilities accurate, detailed usage data without manual reading. A smart meter is typically connected to a Data Concentrator Unit (DCU), which aggregates readings from several meters and forwards them to a Head-End System (HES) for storage, validation and analysis. This automated chain supports dynamic pricing, demand response and predictive maintenance, and reduces both operational cost and the time between an event in the field and its visibility to the utility.

Energy metering has passed through several generations. Mechanical meters, introduced in the late 19th century, used a rotating disc whose speed was proportional to power consumed, but required manual reading and offered limited accuracy. Electromechanical meters, from the early 20th century, added electromagnetic induction for better

accuracy under growing demand. From the 1990s, digital smart meters replaced these electromechanical designs, adding remote data transmission and, for the first time, two-way communication that let a utility query a meter directly. Since the 2000s, smart meters have increasingly become part of IoT-enabled smart grids: they support time-of-day pricing, interact with home automation, solar panels and electric vehicles, and give both consumers and utilities real-time visibility into consumption.

This paper focuses specifically on the link between the DCU and the HES. The objectives are to establish a reliable communication path, secure the data in transit, and quantify network performance – latency, signal strength and data integrity – for a DCU that aggregates several meters and forwards their readings to the HES using MQTT, HTTP(S) or FTP over Ethernet, with AES encryption and TLS/SSL protecting the transmitted data.

II. Literature Survey

Smart-grid communication has been studied extensively at the technology and standards level [1], and wireless links for smart grids have been surveyed for their trade-offs in range, throughput and power [2]. Closer to the present work, DLMS/COSEM-based advanced metering infrastructure has been implemented and evaluated for field deployment [3], and DCUs specifically have been analysed for their architecture, function and performance as the aggregation point between meters and the head-end [4]. Big-data analytics applied to smart-grid telemetry has been shown to support real-time monitoring and decision-making once data reaches the HES [5], while IEC 62056 defines the data-exchange model for meter reading, tariff and load control that DLMS/COSEM implements [6]. Security has been treated as a first-class concern at every layer of the smart grid [7], consistent with the broader case for a smart grid as a platform for 21st-century power delivery [8]; IEC 61850 has separately been shown to shape protection and automation practice in this space [9], and secure communication and authentication mechanisms for smart-grid links have been surveyed in detail [10]. Building on this literature, the present work aggregates data from a DCU built on a Rugged Board platform and transmits it to the HES over Ethernet using DLMS/COSEM, MQTT and AES/TLS security, and reports latency, packet-loss and throughput measurements for that specific link.

III. Importance of the DCU and the HES

A. Data Concentrator Unit (DCU)

A Data Concentrator Unit sits between remote field devices – sensors, meters and control equipment – and the central management system, and is responsible for collecting, processing and forwarding their data (Fig. 1). It aggregates readings from many meters into a single stream, maintains reliable communication with the central system over wired, wireless or hybrid links, encrypts the data it forwards, and supports real-time monitoring and control by making current readings available to operators as soon as they arrive.

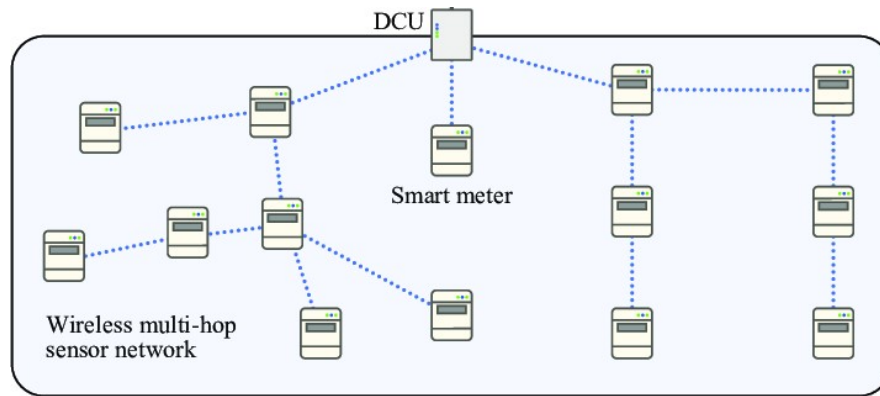


Fig. 1. Data Concentrator Unit (DCU) aggregating readings from a multi-hop meter network.

B. Head-End System (HES)

The Head-End System is the centralised software and hardware infrastructure that collects, manages and processes data forwarded by one or more DCUs (Fig. 2). It aggregates and organises readings from many remote devices, processes them in real time so that faults or anomalies can be acted on immediately, and provides reporting and analytics for forecasting, billing and optimisation. It also automates routine tasks such as meter reading, outage detection and remote device control, and feeds a downstream meter data management system for billing and reporting.

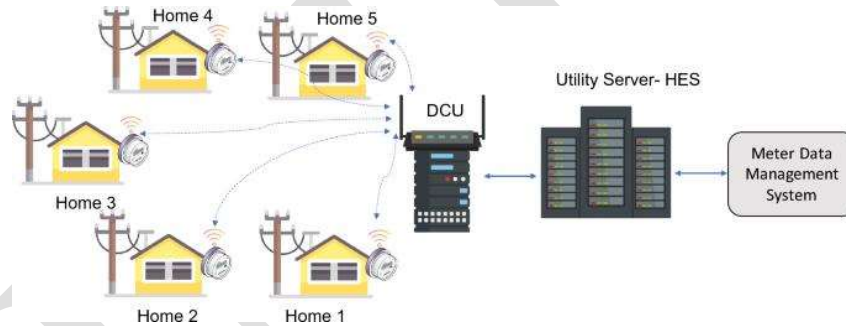


Fig. 2. Head-End System (HES) receiving aggregated readings from a DCU serving several homes.

Ethernet is well suited to the DCU–HES link because it offers high-speed, reliable and scalable transmission, which matters when a single HES may be receiving data from many DCUs simultaneously. Network security – encryption such as SSL/TLS together with firewalls – protects this link from unauthorised access, and efficient data aggregation and filtering at the DCU reduce the volume of redundant traffic that reaches the HES, which keeps bandwidth usage in check as the system scales.

IV. Hardware Platform

A. Rugged Board DCU

The DCU in this implementation is built around the Rugged Board for phyCORE-A5D2x (Fig. 3), a System-in-Peripheral development platform built on Microchip's SAMA5D2-family processor. At its core is the phyCORE-A5D2x System-on-Module, which integrates the processor, flash, power regulation, supervision and transceivers in a direct-solder form factor; the surrounding carrier board adds power input, buttons, connectors and Ethernet and microbus connectivity.

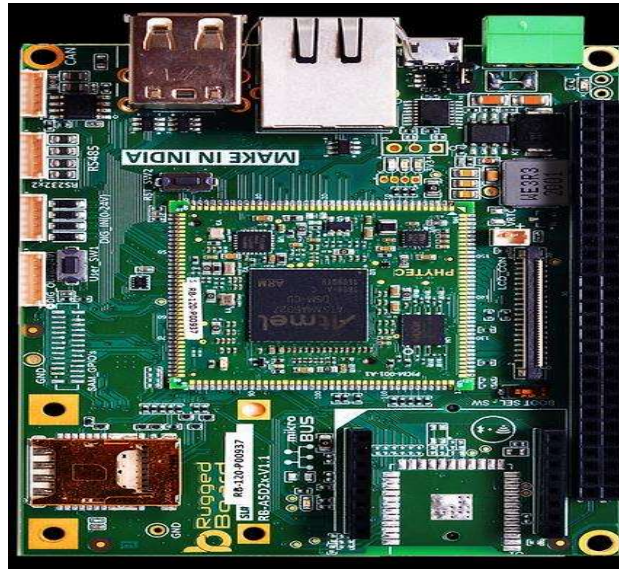


Fig. 3. Rugged Board for phyCORE-A5D2x used as the Data Concentrator Unit.

The board exposes one Ethernet port, RS-232, one isolated RS-485 port, one CAN port, four isolated digital inputs, four digital outputs, an LVDS display interface, a micro-SD slot, a SIM slot, USB 2.0 ports, a microbus socket and a 60-pin expansion header (Fig. 4). The Ethernet port provides the standard network interface used for the HES link; RS-232 and RS-485 give short- and long-distance serial links respectively for interfacing directly with meters or other field devices, with RS-485 additionally supporting multi-drop networks of several meters; CAN provides a fault-tolerant bus for industrial or automotive-style sensor and actuator networks; and the isolated digital I/O lines interface with external switches, sensors, relays and indicators while protecting the board from voltage spikes and noise.

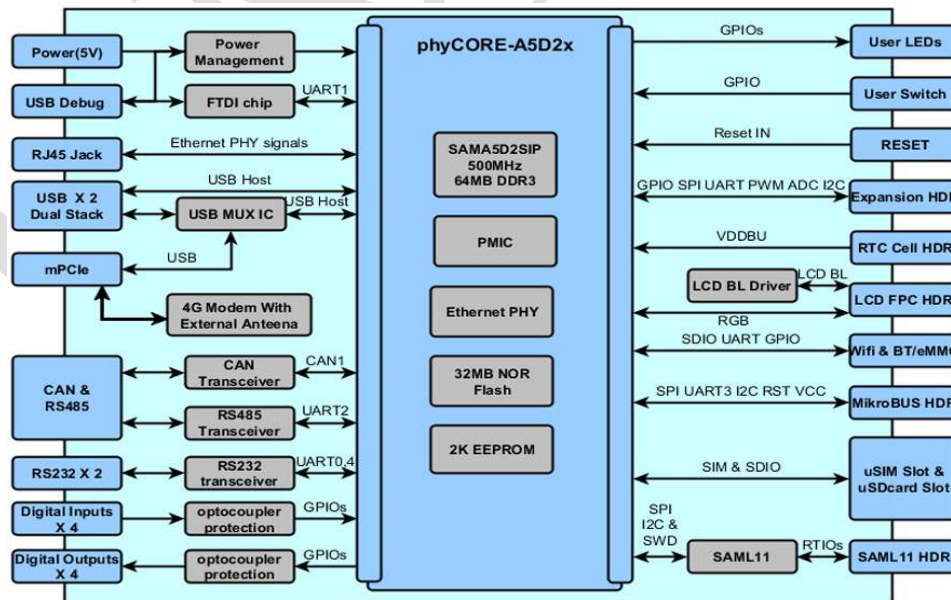


Fig. 4. Internal block diagram of the Rugged Board phyCORE-A5D2x.

B. Meter Interfacing and Data Processing

The Rugged Board's RS-232 and RS-485 interfaces collect data directly from smart meters: RS-232 suits short, point-to-point links, while RS-485 handles longer distances and multiple meters on one bus. Once collected, the onboard processor aggregates, analyses and formats the data – covering parameters such as energy consumption, voltage, current and power factor – and can handle protocols such as Modbus RTU over RS-232/RS-485 and Modbus TCP/IP over Ethernet, preparing the data in the appropriate protocol before it is sent onward.

C. Network Interface Card

Each meter node uses a Network Interface Card (Fig. 5) to connect to the network, wired or wireless, so it can exchange data with the rest of the system. The NIC handles the conversion between the meter's internal data representation and the format needed for network transmission, is identified on the network by a unique MAC address, and supports Ethernet speeds from 10/100/1000 Mbps up to gigabit rates depending on the deployment.

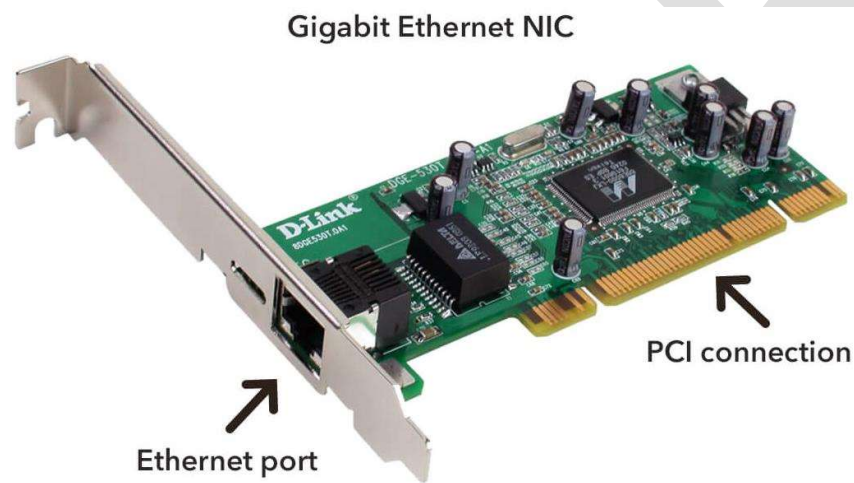


Fig. 5. Network Interface Card (NIC) used for meter-side network connectivity.

Once processed, meter data leaves the Rugged Board through its Ethernet port toward the HES over the LAN or WAN, using Modbus TCP/IP, HTTP or MQTT depending on the deployment. Because the Rugged Board is designed for rugged environments, this link remains stable in industrial plants, outdoor installations and vehicles. The GURUX configuration tool was used during development and commissioning to exercise the DLMS/COSEM link directly – it supports Ethernet, GSM and RS-485 media, so the same tool can be used to bring up and verify meters regardless of which physical interface they use.

V. DLMS/COSEM Protocol

The Device Language Message Specification (DLMS) and its Companion Specification for Energy Metering (COSEM), standardised as IEC 62056, is the messaging and object-modelling standard used for the meter-to-DCU and DCU-to-HES data exchange (Fig. 6). DLMS defines the messaging structure while COSEM supplies the object models used to represent electricity, gas, water and heat metering data, and together they replace the incompatible proprietary protocols that earlier metering systems relied on. The standard gives interoperability across meter manufacturers, encrypted and authenticated communication, and the scalability needed for large smart-grid deployments.



Fig. 6. DLMS/COSEM as the common communication standard across smart-grid applications.

A. OBIS Codes

Object Identification System (OBIS) codes are standardised, six-field identifiers (medium, channel, data type, billing period, measurement unit and measurement attribute) attached to each item of metering data, which let different meter models and manufacturers represent the same quantity consistently. Table I lists the OBIS codes used in this implementation.

TABLE I
Common OBIS Codes Used

OBIS Code	Description
1-0:1.8.0	Total active energy consumed
1-0:2.8.0	Total active energy exported
1-0:32.7.0	Voltage (Phase 1)
1-0:52.7.0	Voltage (Phase 2)
1-0:72.7.0	Voltage (Phase 3)
1-0:31.7.0	Current (Phase 1)

B. Communication Model

DLMS follows a client–server model in which the client (the HES or utility system) requests data from the server (the energy meter or DCU) over an application layer that structures the exchange using COSEM objects (Fig. 7). A session begins with association establishment, in which the client authenticates itself; the client then requests specific data items by OBIS code, the meter responds with the requested values, the HES processes and stores what it receives, and the session is finally released. Within this model the COSEM object model groups meter data logically: a Logical

Device Name identifies the meter, association objects manage authentication and communication state, and data objects hold the actual values – energy consumption, voltage, current and so on.

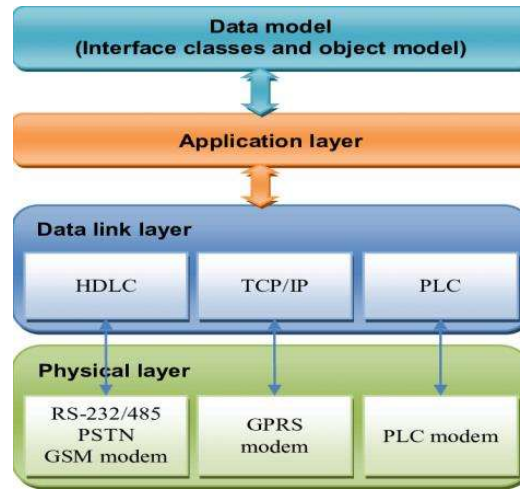


Fig. 7. Data model, application, data-link and physical layers used in DLMS/COSEM communication.

C. Security Levels

Because energy-consumption data is sensitive, DLMS defines four authentication levels: no authentication (Level 0); password-based authentication (Level 1); challenge-response authentication using cryptographic methods (Level 2); and certificate-based authentication (Level 3). In addition, AES is used to encrypt data during transmission, so that even where a session is intercepted the payload remains protected.

D. Implementation Challenges

Implementing DLMS/COSEM in practice raised several recurring difficulties: understanding the full protocol stack and its many configuration parameters across Ethernet and RF media; vendor-specific interpretations of the standard and of OBIS codes that complicate interoperability between meters from different manufacturers; communication-layer issues such as latency under network congestion, packet loss on wireless and PLC links, and the need for careful error handling to preserve data integrity; security concerns around cyberattacks, data privacy and the safe management of over-the-air firmware updates; and the cost of upgrading legacy systems and training personnel to operate a DLMS-based infrastructure.

VI. System Architecture and Design

Fig. 8 shows the overall data path. Each smart meter connects to its own NIC over UART; the NICs communicate wirelessly with a central NIC, which forwards the aggregated stream over UART to the DCU. The DCU exposes this data over Ethernet to an MQTT broker, from which a Windows PC – running the GURUX tool – receives it over UDP; a Python script then converts the GURUX output into MQTT messages suitable for onward distribution to the HES and other subscribers.

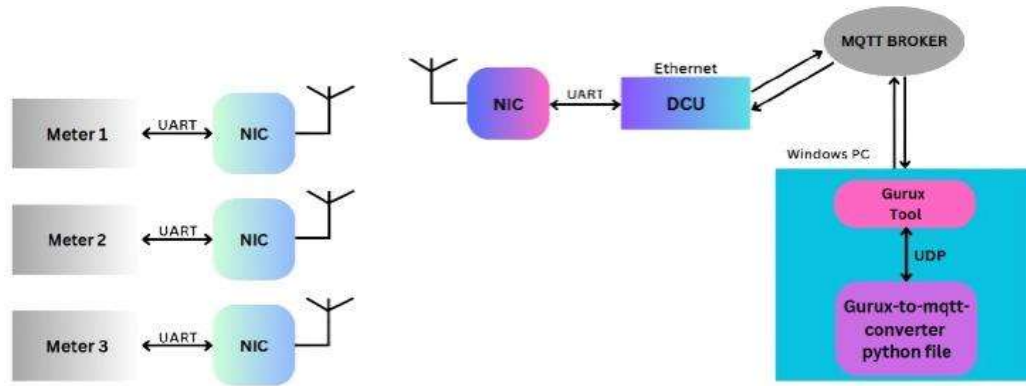


Fig. 8. Block diagram of the DCU-to-HES data path used in this implementation.

The energy meters are DLMS/COSEM-compliant devices measuring voltage, current, power and energy consumption, communicating with their NIC over UART. The NIC provides wireless connectivity between each meter and the DCU. The DCU aggregates the incoming streams, communicates with the NICs over UART, and forwards the combined data to the Windows PC over Ethernet. On that PC, the GURUX tool interprets the DLMS/COSEM data and passes it over UDP to a Python script, which republishes it as MQTT messages for efficient onward transmission to the cloud or other applications. Table II and Table III list the hardware and software used to realise this architecture.

TABLE II
Hardware Requirements

Component	Specification
Smart meters	DLMS/COSEM-compliant, UART communication
NIC	Wireless RF, GPRS, or PLC module
DCU	Supports RF, UART and Ethernet protocols
HES server	High-performance storage and processing unit
Windows PC	Runs the GURUX DLMS tool and MQTT converter

TABLE III
Software Requirements

Software	Purpose
GURUX DLMS tool	DLMS/COSEM data interpretation
Python (MQTT converter)	Converts DLMS data for MQTT transmission
MQTT broker (e.g. Mosquitto)	Handles real-time data distribution

Database (SQL)	Stores meter data
DLMS stack	Ensures protocol compliance for data exchange

VII. Implementation of DCU-to-HES Communication

A. Data Collection and Preprocessing

Smart meters communicate with the DCU over UART (RS-232/RS-485), RF or PLC modules, with the NIC in each meter handling the data transfer; the DCU issues DLMS/COSEM requests and the meters respond with the requested packets. Because raw meter data can be redundant, missing or unstructured, it is pre-processed before transmission: relevant OBIS codes, readings, timestamps and meter IDs are parsed out of the DLMS packets; binary values are converted into human-readable units such as kWh, kW, V and A; CRC values are validated to catch corrupted or incomplete packets; and, optionally, the data is compressed to reduce the bandwidth needed for transmission.

B. Ethernet Communication Setup

The DCU and HES are each assigned a static IP address for reliable addressing; Cat5e/Cat6 or fibre-optic Ethernet connections are installed and configured over TCP/IP; and connectivity is verified with ping tests, with firewall and router settings checked to ensure data packets are allowed to pass.

C. DLMS/COSEM over Ethernet

The DLMS/COSEM stack (the GURUX DLMS library in this implementation) is installed and configured, and OBIS codes are defined for the parameters of interest – voltage, energy, power factor and so on. Each exchange is carried as a DLMS Application Protocol Data Unit (APDU) containing a source address (meter or DCU ID), a destination address (the HES IP), a command type (read or write), the relevant OBIS code and a security header for encryption. The DCU then sends its request to the meter, receives the response, formats and encrypts the data, and forwards it to the HES over TCP/IP, UDP (for faster, best-effort transmission) or MQTT (for cloud-based HES deployments); the HES decrypts and processes what it receives.

D. Secure Data Transmission

Because meter data is sensitive, several protections are applied end to end: AES-128/256 encryption of DLMS packets together with TLS/SSL over TCP/IP; role-based access control and certificate- or key-based authentication to prevent unauthorised access; CRC checks and digital signatures to verify data integrity and detect tampering; and HTTPS or MQTT-TLS with IP whitelisting for cloud-based HES deployments.

E. Working Principle of DCU–HES Communication

Fig. 9 summarises the end-to-end flow from field meters to the utility's distribution system state estimator. Smart meters record consumption and store it locally; each meter's NIC forwards this data to the DCU over UART or RF. The DCU aggregates readings from multiple meters and organises them into standardised DLMS/COSEM packets. It then transmits the packets to the HES over Ethernet, GPRS or another communication module, using secure protocols such as DLMS over TCP/IP or MQTT for cloud-based communication. On arrival, the HES verifies the readings, performs validation, storage and analysis, and forwards the processed data to billing systems, monitoring dashboards or, as shown here, a distribution system state estimator feeding the utility's control centre.

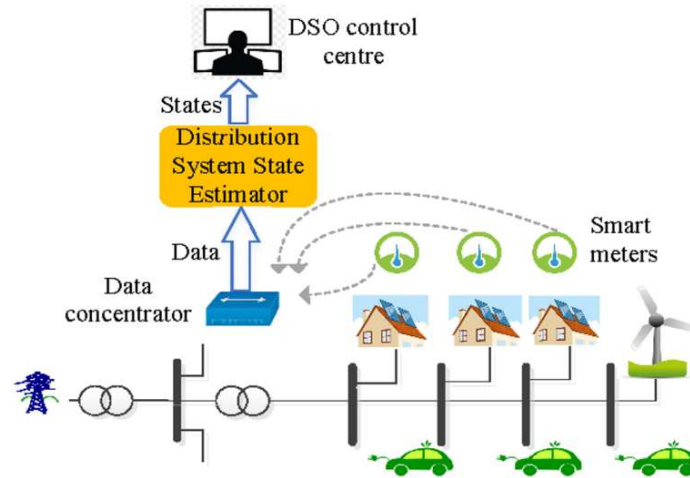


Fig. 9. Working principle of DCU–HES communication feeding a utility distribution system state estimator.

VIII. Testing and Results

Testing verified accurate data collection from the meters, seamless transmission to the HES, efficient processing and storage on arrival, and the security and reliability of the system as a whole. The setup used PuTTY for serial communication with the Rugged Board, the GURUX tool for DLMS/COSEM exchanges, Wireshark for packet-level inspection, and Python scripts for data validation and for the GURUX-to-MQTT conversion; unit, integration, performance, security and error-handling tests were all carried out.

A. Functional and Test-Case Results

The DCU was verified for its ability to establish communication with multiple smart meters, request and receive DLMS/COSEM data by OBIS code, preprocess and format that data, forward it to the HES over Ethernet, and retry failed transmissions; the HES was verified for receiving and storing DCU data, decoding DLMS packets and extracting OBIS values, and validating data integrity and accuracy. Table IV summarises the resulting test cases, all of which passed.

TABLE IV
Test Cases for DCU and HES

Test Case	Expected Result	Status
Smart meter connectivity	DCU detects all connected meters	Passed
Data request from DCU	Meter responds with DLMS packet	Passed
Data parsing	Correct OBIS code extraction	Passed
Ethernet transmission	HES receives and processes data	Passed
Error handling	DCU retries failed requests	Passed

B. Performance Analysis

Performance was assessed against data transmission speed, packet loss rate, system latency and CPU/memory utilisation, alongside data accuracy, network throughput, uptime and error rate. Table V lists the measured values against their expected thresholds; the system met or beat every threshold, giving fast, reliable transmission with low latency and minimal packet loss.

TABLE V
Performance Test Results

Parameter	Measured	Threshold	Status
Data transmission speed	250 ms	< 500 ms	Good
Packet loss rate	0.8%	< 2%	Acceptable
System latency	1.2 s	< 2 s	Optimal
CPU usage (DCU)	40%	< 60%	Efficient

C. Data Transmission Efficiency

Transmission efficiency depends mainly on available network bandwidth, whether data is compressed before sending, and which protocol is used. Table VI compares the three transmission methods tested: Ethernet with UDP gave the fastest transmission time but a slightly lower packet-success rate; Ethernet with TCP/IP gave the highest reliability at a modest cost in latency; and MQTT was efficient for cloud-based communication but introduced a small additional delay.

TABLE VI
Data Transmission Speed Comparison

Transmission Method	Time Taken	Packet Success Rate
Ethernet (TCP/IP)	250 ms	99.2%
Ethernet (UDP)	180 ms	98.5%
MQTT	300 ms	97.8%

D. Error Analysis

The errors observed during testing were packet loss from network congestion, addressed with automatic retries; data corruption from transmission interference, addressed with error-correction coding; timeout errors from delayed meter responses, addressed by increasing timeout limits; and authentication failures from incorrect security keys, addressed by resetting security credentials. Wireshark was used to debug Ethernet packet flow and tune network parameters, and the GURUX tool was used to trace incorrect DLMS responses and verify OBIS codes in request frames; automatic retry logic and redundant backup servers were introduced to recover from lost packets and HES failures respectively.

IX. Challenges and Future Enhancements

Development surfaced challenges across hardware, protocol, network and security layers (Fig. 10): interfacing and synchronising multiple smart meters over UART and NIC modules without data collisions, within the DCU's limited processing headroom for real-time encryption; the inherent complexity of DLMS/COSEM, including its multiple data models and the effort needed to map OBIS codes consistently across meters that format their data slightly differently; Ethernet packet loss and processing latency under simultaneous transmissions from several meters, and the trade-off between low-latency UDP and more reliable TCP; the computational overhead of High-Level Security authentication and AES encryption on a resource-constrained DCU; and integration work to get the GURUX tool reliably forwarding UDP packets to the MQTT broker, alongside the routine debugging of DLMS packet-parsing errors.

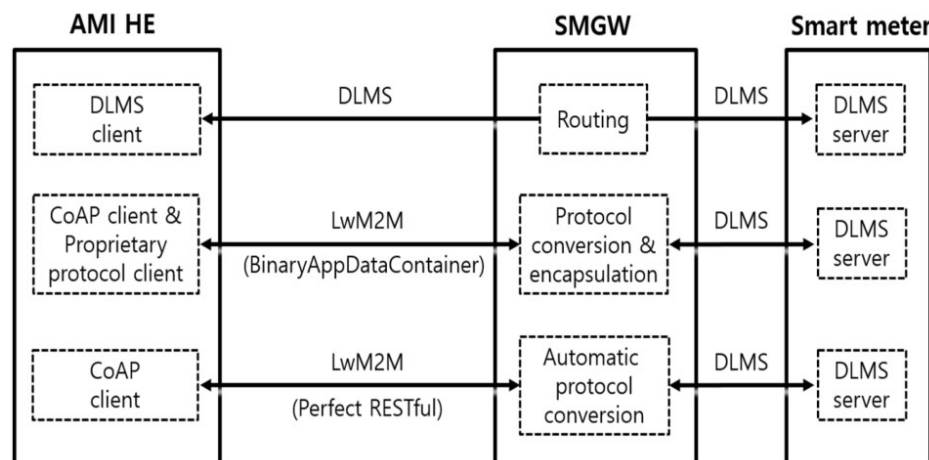


Fig. 10. Protocol conversion and routing at the smart-meter gateway, illustrating the hardware and communication issues addressed during implementation.

Planned enhancements target three areas. Scalability: multi-threaded processing in the DCU, a higher-performance processor, and batch rather than single-meter polling, so the same architecture can serve more meters. Security: a Public Key Infrastructure for stronger authentication and encryption, end-to-end encryption from meter to HES, and a dynamic key-management system. Connectivity: GSM, LoRa or Wi-Fi modules to complement the Ethernet link, used in a hybrid arrangement where Ethernet remains primary and wireless serves as an automatic-failover backup, improving overall network redundancy.

X. Conclusion

This work implemented and evaluated a DCU-to-HES communication system for smart metering, built around a Rugged Board DCU, DLMS/COSEM over Ethernet, and MQTT-based delivery to the HES. OBIS-coded data exchange gave structured, standardised categorisation of meter readings – voltage, current, power and demand – suitable for large-scale deployment, while AES encryption and High-Level Security authentication protected the link against unauthorised access. Testing showed reliable, low-latency transmission with minimal packet loss and full recovery from the fault conditions exercised, and the difficulties encountered during implementation – around multi-meter synchronisation, protocol interoperability, network reliability and processing overhead on a constrained DCU

– point directly to the scalability, security and connectivity enhancements identified for future development. Overall, the results support Ethernet and DLMS/COSEM as a scalable, reliable and secure basis for smart-metering data transmission and accurate, error-free billing.

References

- [1] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, “Smart grid technologies: Communication technologies and standards,” *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, 2011.
- [2] A. Mahmood, N. Javaid, and S. Razzaq, “A review of wireless communications for smart grid,” *Renewable and Sustainable Energy Reviews*, vol. 41, pp. 248–260, 2015.
- [3] J. Kim, H. Lee, and Y. Kim, “Implementation of DLMS/COSEM-based smart metering system for advanced metering infrastructure,” *Sensors*, vol. 19, no. 5, p. 1156, 2019.
- [4] R. Gupta, R. Kumar, and R. Pal, “Data concentrator unit for smart metering systems: Architecture, functions, and performance analysis,” *IEEE Access*, vol. 8, pp. 159423–159439, 2020.
- [5] Y. Zhang, L. Wang, and X. Wang, “Big data analytics in smart grid: Enabling real-time monitoring and decision making,” *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1804–1811, 2017.
- [6] IEC 62056, “Electricity metering – Data exchange for meter reading, tariff and load control,” International Electrotechnical Commission, 2021.
- [7] C. Modi, D. Patel, B. Borisaniya, H. Patel, and M. Rajarajan, “A survey on security issues and solutions at different layers of smart grid,” *Computer Science Review*, vol. 9, pp. 236–254, 2013.
- [8] S. M. Amin and B. F. Wollenberg, “Toward a smart grid: Power delivery for the 21st century,” *IEEE Power and Energy Magazine*, vol. 3, no. 5, pp. 34–41, 2005.
- [9] A. Apostolov, “Impact of IEC 61850 on smart grid protection and automation,” *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2248–2255, 2016.
- [10] M. I. Tariq, A. A. Khan, and J. Qadir, “Secure communication and authentication in smart grid,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 1013–1035, 2019.