ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster Universitario en Ingeniería Industrial (MII)

Design and Implementation of Test Cases for IDIS Compliance in UMEME's OneSait Smart Metering Project

Autor
Guillermo Varas Yuste

Dirigido por
Néstor Rodríguez Pérez

Madrid
Augosto 2024

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Design and Implementation of Test Cases for IDIS Compliance in UMEME's OneSait Smart Metering Project

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2023/2024. es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Guillermo Varas Yuste Fecha: 30/ 08/ 2024

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Néstor Rodríguez Pérez Fecha: ……30/ ……08/ ……2024

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
(ICAI)

Máster Universitario en Ingeniería Industrial (MII)

Design and Implementation of Test Cases for IDIS Compliance in UMEME's OneSait Smart Metering Project

Autor
Guillermo Varas Yuste

Dirigido por
Néstor Rodríguez Pérez

Madrid
Agosto 2024

Resumen

La evolución de las infraestructuras energéticas hacia sistemas inteligentes ha llevado a la implementación de soluciones avanzadas de medición inteligente, que no solo permiten un monitoreo más eficiente de los recursos energéticos, sino que también mejoran la precisión en la facturación y optimizan la gestión de la red. Este proceso es crucial en países en desarrollo como Uganda, donde el acceso a la electricidad y la eficiencia de su distribución juegan un papel vital en el desarrollo económico y social. En este contexto, UMEME, el principal distribuidor de electricidad en Uganda, ha iniciado el Proyecto de Medición Inteligente OneSait, un esfuerzo significativo para modernizar su red eléctrica mediante la instalación de medidores inteligentes que cumplan con estándares internacionales como DLMS/-COSEM e IDIS.

Capítulo 1: Introducción

La tesis comienza destacando la importancia de las redes inteligentes y la medición avanzada en la mejora de la eficiencia, fiabilidad y seguridad de las redes eléctricas. Se establece la motivación principal del trabajo: garantizar que los medidores inteligentes implementados en el proyecto OneSait cumplan con los estándares internacionales de interoperabilidad y seguridad. En este sentido, la tesis se enfoca en diseñar y ejecutar casos de prueba que validen la conformidad de los medidores inteligentes con los estándares IDIS, asegurando que estos dispositivos puedan operar eficientemente dentro de la infraestructura de UMEME. Además, se introduce la relevancia de la ciberseguridad en estos sistemas, dada la creciente interconexión y la vulnerabilidad a posibles ataques cibernéticos.

Capítulo 2: Estado del Arte

En este capítulo se realiza una revisión exhaustiva de los protocolos DLMS/-COSEM e IDIS, que son fundamentales para la interoperabilidad y la comunicación eficaz en los sistemas de medición inteligente. DLMS/COSEM es un con-

junto de protocolos estándar que define la estructura de los mensajes y el modelo de datos utilizado por los medidores inteligentes, mientras que IDIS proporciona especificaciones adicionales para asegurar la interoperabilidad entre dispositivos de diferentes fabricantes. El capítulo también aborda los desafíos de implementación, tales como la gestión de recursos, la compatibilidad con plataformas existentes y la seguridad de los datos transmitidos. Además, se examinan las soluciones propuestas en la literatura para superar estos desafíos, incluyendo la integración con tecnologías como G3-PLC para mejorar la eficiencia y la fiabilidad de la comunicación en redes inteligentes.

Capítulo 3: Paradigma del Proyecto de UMEME

Este capítulo ofrece una visión detallada del sector eléctrico en Uganda, subrayando la importancia de la modernización de la infraestructura eléctrica a través de proyectos como OneSait. Se analiza el papel de UMEME en la distribución de electricidad y cómo la implementación de medidores inteligentes puede transformar la gestión de la red eléctrica en Uganda. Se destacan los desafíos específicos que enfrenta UMEME, como las pérdidas de energía, la precisión en la facturación y la necesidad de un monitoreo en tiempo real para mejorar la eficiencia operativa. El capítulo también explora cómo el proyecto OneSait se alinea con los objetivos nacionales de desarrollo sostenible, mejorando no solo la infraestructura energética, sino también la calidad de vida de los ciudadanos ugandeses.

Capítulo 4: Sistemas de Prepago

Aquí se exploran los sistemas de prepago actuales en Uganda y cómo los medidores inteligentes pueden mejorar estos sistemas. Se discuten las soluciones de prepago avanzadas, incluyendo la automatización y las opciones de pago digital y móvil, que ofrecen una mayor conveniencia para los consumidores y aseguran ingresos más fiables para UMEME. El capítulo también aborda cómo los estándares DLMS/COSEM e IDIS pueden integrarse en los sistemas de prepago para mejorar la interoperabilidad y la seguridad, facilitando la implementación de soluciones de prepago que sean tanto eficientes como seguras.

Capítulo 5: Pruebas de Casos de Uso IDIS

Este es uno de los capítulos centrales de la tesis, donde se detallan los casos de uso diseñados para evaluar la conformidad de los medidores inteligentes con los estándares IDIS. Se describen en profundidad los escenarios de prueba, que incluyen desde el registro de medidores hasta la actualización remota de firmware, pasando por la gestión de la carga y la supervisión de la calidad del suministro. Cada caso de uso es desarrollado y probado en un entorno simulado utilizando MATLAB, lo que permite replicar condiciones operativas reales y evaluar el rendimiento de los medidores bajo diferentes circunstancias. Los resultados de estas pruebas no solo validan la conformidad con los estándares, sino que también identifican áreas de mejora en la implementación de los medidores.

Capítulo 6: Ciberseguridad en la Medición Inteligente

Este capítulo aborda la creciente preocupación por la ciberseguridad en los sistemas de medición inteligente. Se realiza un análisis detallado de las vulnerabilidades específicas de los medidores inteligentes, como la posibilidad de accesos no autorizados, filtraciones de datos y ataques que puedan comprometer la integridad del sistema. Se proponen diversas medidas para mitigar estos riesgos, incluyendo la implementación de cifrado avanzado, autenticación robusta y la gestión segura de actualizaciones de firmware. Además, se exploran estrategias para la detección de intrusiones y la segmentación de la red, lo que podría prevenir ataques y limitar el daño en caso de que se produzcan. Estas recomendaciones son fundamentales para asegurar la resiliencia de la infraestructura de UMEME frente a amenazas cibernéticas.

Capítulo 7: Conclusiones

En las conclusiones, se sintetizan los principales hallazgos de la tesis, destacando la validación de la conformidad de los medidores con los estándares IDIS como un logro clave. Se subraya la importancia de las recomendaciones de ciberseguridad propuestas y su potencial impacto en la seguridad y fiabilidad de la red inteligente de Uganda. Además, se discuten las implicaciones más amplias de la modernización de la red eléctrica de Uganda, no solo en términos de eficiencia y fiabilidad, sino también en cómo estas mejoras pueden contribuir al desarrollo económico y social del país. Finalmente, se propone que los resultados obtenidos en esta investigación puedan servir de modelo para futuras implementaciones de redes inteligentes en otros países en desarrollo, proporcionando un marco sólido para la modernización energética global.

Summary

The evolution of energy infrastructures toward smart systems has led to the implementation of advanced smart metering solutions that not only allow for more efficient monitoring of energy resources but also improve billing accuracy and optimize grid management. This process is crucial in developing countries like Uganda, where access to electricity and distribution efficiency play a vital role in economic and social development. In this context, UMEME, the main electricity distributor in Uganda, has initiated the OneSait Smart Metering Project, a significant effort to modernize its electrical grid through the installation of smart meters that comply with international standards such as DLMS/COSEM and IDIS.

Chapter 1: Introduction

The thesis begins by highlighting the importance of smart grids and advanced metering in improving the efficiency, reliability, and security of electrical networks. The main motivation of the work is established: to ensure that the smart meters implemented in the OneSait project comply with international interoperability and security standards. In this regard, the thesis focuses on designing and executing test cases to validate the compliance of the smart meters with IDIS standards, ensuring that these devices can operate efficiently within UMEME's infrastructure. Additionally, the relevance of cybersecurity in these systems is introduced, given the increasing interconnection and vulnerability to potential cyberattacks.

Chapter 2: State of the Art

This chapter provides an exhaustive review of the DLMS/COSEM and IDIS protocols, which are fundamental for interoperability and effective communication in smart metering systems. DLMS/COSEM is a set of standard protocols that defines the structure of messages and the data model used by smart meters, while IDIS provides additional specifications to ensure interoperability between devices from different manufacturers. The chapter also addresses implementation challenges,

such as resource management, compatibility with existing platforms, and data security during transmission. Moreover, the solutions proposed in the literature to overcome these challenges are examined, including integration with technologies like G3-PLC to improve communication efficiency and reliability in smart grids.

Chapter 3: UMEME's Project Paradigm

This chapter offers a detailed overview of the electrical sector in Uganda, emphasizing the importance of modernizing the electrical infrastructure through projects like OneSait. It analyzes UMEME's role in electricity distribution and how the implementation of smart meters can transform grid management in Uganda. The specific challenges faced by UMEME are highlighted, such as energy losses, billing accuracy, and the need for real-time monitoring to improve operational efficiency. The chapter also explores how the OneSait project aligns with national sustainable development goals, improving not only energy infrastructure but also the quality of life for Ugandan citizens.

Chapter 4: Prepayment Systems

Here, the current prepayment systems in Uganda and how smart meters can improve these systems are explored. Advanced prepayment solutions are discussed, including automation and digital and mobile payment options, which offer greater convenience for consumers and ensure more reliable revenue for UMEME. The chapter also addresses how DLMS/COSEM and IDIS standards can be integrated into prepayment systems to enhance interoperability and security, facilitating the implementation of prepayment solutions that are both efficient and secure.

Chapter 5: IDIS Use Case Tests

This is one of the core chapters of the thesis, detailing the use cases designed to evaluate the compliance of smart meters with IDIS standards. The test scenarios are described in depth, ranging from meter registration to remote firmware updates, including load management and supply quality supervision. Each use case is developed and tested in a simulated environment using MATLAB, allowing for the replication of real operational conditions and evaluation of meter performance under different circumstances. The results of these tests not only validate compliance with standards but also identify areas for improvement in the implementation of the meters.

Chapter 6: Cybersecurity in Smart Metering

This chapter addresses the growing concern for cybersecurity in smart metering systems. A detailed analysis of the specific vulnerabilities of smart meters is conducted, such as the possibility of unauthorized access, data breaches, and attacks that could compromise the integrity of the system. Various measures are proposed to mitigate these risks, including the implementation of advanced encryption, robust authentication, and secure firmware management. Additionally, strategies for intrusion detection and network segmentation are explored, which could prevent attacks and limit damage in case they occur. These recommendations are fundamental to ensuring the resilience of UMEME's infrastructure against cyber threats.

Chapter 7: Conclusions

The conclusions synthesize the main findings of the thesis, highlighting the validation of meter compliance with IDIS standards as a key achievement. The importance of the proposed cybersecurity recommendations and their potential impact on the security and reliability of Uganda's smart grid is emphasized. Furthermore, the broader implications of modernizing Uganda's electrical grid are discussed, not only in terms of efficiency and reliability but also in how these improvements can contribute to the country's economic and social development. Finally, it is suggested that the results obtained in this research could serve as a model for future implementations of smart grids in other developing countries, providing a solid framework for global energy modernization.

To my family and friends,
for their great unconditional support,
for all those long nights of work,
and for their valuable advice.

None of this would have been possible without you.

A un panal de rica miel,
dos mil moscas acudieron,
que por golosas murieron,
presas de patas en él.

Félix María de Samaniego

Acknowledgements

I would like to express my deepest gratitude to both ICAI and the University of Strathclyde for providing me with the incredible opportunity to pursue my studies in such a stimulating and supportive environment. The guidance and education I have received from both institutions have been instrumental in shaping this thesis and my development as an engineer.

I would also like to extend my sincere thanks to Minsait for the invaluable experience and opportunities provided during my internship. The welcoming atmosphere, support, and professional treatment I received throughout my time at Minsait have contributed greatly to the success of this project. It has been an honor to work with such a dedicated and innovative team.

Finally, I would like to thank everyone who has supported me along this journey. Your encouragement and assistance have been greatly appreciated and have helped me immensely throughout this process.

Guillermo Varas Yuste
ICAI
August 2024

Contents

List of Figures

Acronyms

AES	Advanced Encryption Standard
AMI	Advanced Metering Infrastructure
COSEM	Companion Specification for Energy Metering
CPI	Consumer Price Index
CTT	Conformance Test Tool
C-UNB	Cooperative Ultra-Narrowband
DDoS	Distributed Denial of Service
DLMS	Device Language Message Specification
EAPP	East African Power Pool
ECDH	Elliptic Curve Diffie-Hellman
ERA	Electricity Regulatory Authority
FiT	Feed-in-Tariff
GCM	Galois/Counter Mode
HDLC	High-Level Data Link Control
HES	Head End System
HFO	High Fuel Oil
HLS	High Level Security
ICAI	Insitituto Católico de Artes e Industrias
IDIS	Interoperable Device Interface Specifications
IDS	Intrusion Detection System
IEA	International Energy Agency
IEEE	Institute of Electrical and Electronics Engineers
ITA	International Trade Administration
IP	Internet Protocol
NS	Network Simulator
OIML	International Organization of Legal Metrology
PKI	Public Key Infrastructure
PLC	Powerline Communication
QOS	Quality of Service
QTAM	Quarterly Tariff Adjustment Methodology
RBAC	Role-Based Access Control
TCP	Transmission Control Protocol
UEGCL	Uganda Electricity Generation Company Limited

UETCL	Uganda Electricity Transmission Company Limited
USH	Uganda Shilling
VLAN	Virtual Local Area Network
VPN	Virtual Private Network

Chapter 1

Introduction

In the current landscape of global energy systems, smart grid technologies and the deployment of smart metering solutions have become critical components in enhancing the efficiency, reliability, and security of electricity distribution networks. These innovations, particularly in the realm of smart metering, offer significant improvements over traditional electricity metering by enabling real-time monitoring, remote data transmission, and automated control of electrical devices. At the core of these advancements lies the need for robust communication standards that allow seamless interoperability between diverse systems and devices, regardless of manufacturer. The Device Language Message Specification/Companion Specification for Energy Metering (DLMS/COSEM) protocol, coupled with the Interoperable Device Interface Specifications (IDIS), provides this critical standardization. By adhering to these protocols, metering devices are capable of communicating effectively, allowing for consistent performance across a variety of environments and applications.

This thesis is motivated by the growing importance of these standards in the context of modern electricity distribution systems, specifically within Uganda's largest electricity distributor, UMEME. UMEME's OneSait Smart Metering Project represents a major initiative aimed at modernizing the country's energy infrastructure, focusing on the deployment of smart meters that will enhance billing accuracy, energy management, and operational efficiency. Given the critical role that smart meters play in the system, ensuring their compliance with international standards is paramount to the project's success. This thesis explores the challenges and opportunities related to DLMS/COSEM and IDIS standards, with a particular focus on their application within UMEME's infrastructure.

One of the primary goals of this thesis is to design and implement a set of test cases aimed at ensuring IDIS compliance for the smart meters used in UMEME's project. These test cases simulate real-world scenarios to verify that the meters adhere to the required communication protocols and perform reliably across various use cases. The test cases cover essential functionalities such as meter registration, tariff programming, on-demand and periodic readings, connection and disconnection, firmware upgrades, load management, and more. By implementing these simulations, the thesis seeks to validate that the meters are ready for deployment in a real-world setting, providing UMEME with the confidence that their smart meters will perform to standard under operational conditions.

In addition to the primary objective of ensuring IDIS compliance, this thesis also addresses the growing concern of cybersecurity within smart metering systems. As these systems become increasingly interconnected, they become more vulnerable to cyberattacks, which can compromise sensitive data, disrupt operations, and even endanger the integrity of the power grid. Given these risks, this thesis includes an exploration of cybersecurity threats specific to smart metering infrastructures and proposes solutions to enhance the security of UMEME's system. This includes investigating potential encryption strategies, secure communication protocols, tamper detection mechanisms, and the implementation of advanced authentication techniques to safeguard the system from external threats.

The first step in achieving these objectives involves extensive research into the DLMS/COSEM and IDIS standards. Understanding the intricacies of these protocols is essential to designing effective test cases that will accurately evaluate the meters' compliance. This research serves as the foundation upon which the entire simulation framework is built, guiding the development of the test cases and ensuring that they accurately reflect the real-world demands placed on the smart meters.

Following the research phase, a comprehensive series of test cases is designed to evaluate the performance of UMEME's smart meters. These tests are designed to simulate a wide range of scenarios, from the initial registration of the meter in the network to more complex functions such as remote firmware upgrades and load management. Each test case is carefully crafted to assess a specific aspect of the meter's functionality, ensuring that it can perform reliably in a variety of situations. These tests are implemented in a controlled simulation environment, allowing for detailed analysis of the meters' performance under different conditions.

One of the major challenges faced in this simulation process is ensuring the proper communication between the smart meter and the Head-End System (HES). To achieve this, separate MATLAB sessions are used to simulate the TCP/IP communication between the meter (acting as the client) and the HES (acting as the server). This approach allows for accurate simulation of the real-world communication flows that occur within smart metering infrastructures, enabling a thorough evaluation of the meters' ability to exchange data with the HES in real time.

Once the simulations are complete, the results of the tests are analyzed to determine the meters' compliance with IDIS standards. This analysis is crucial in identifying any areas where the meters may fall short of the required standards, allowing for targeted improvements to be made before the meters are deployed in the field. By ensuring that the meters meet these rigorous standards, this thesis contributes to the overall reliability and security of UMEME's OneSait Smart Metering Project, helping to pave the way for a more efficient and sustainable energy infrastructure in Uganda.

Another key aspect of this thesis is the focus on cybersecurity. As smart meters become more integrated into the digital landscape, the potential for cyberattacks increases, making it essential to safeguard these devices from potential threats. This thesis explores the vulnerabilities present in smart metering systems, such as weak encryption, inadequate authentication methods, and the lack of secure firmware management. In response to these vulnerabilities, a range of cybersecurity measures is proposed, aimed at strengthening the defenses of UMEME's infrastructure. These measures include the implementation of robust encryption algorithms, the use of advanced authentication protocols, and the development of secure firmware update processes that minimize the risk of unauthorized access or tampering.

In addition to the technical challenges addressed in this thesis, the broader implications of smart metering technology are also considered. The adoption of smart meters is expected to have a significant impact on Uganda's energy sector, particularly in terms of improving billing accuracy, reducing energy theft, and enhancing overall energy efficiency. By providing real-time data on energy consumption, smart meters enable more accurate billing, allowing consumers to pay for the exact amount of electricity they use. This not only benefits consumers but also helps UMEME to reduce losses due to inaccurate meter readings or energy theft. Moreover, the ability to remotely monitor and manage energy consumption allows UMEME to better balance supply and demand, improving the overall stability of the grid.

The deployment of smart meters also represents an important step toward achieving Uganda's energy goals, particularly in the context of its commitment to sustainable development. By improving the efficiency of the electricity distribution network, smart meters can help to reduce the environmental impact of energy consumption, supporting Uganda's efforts to transition to a more sustainable energy future. Furthermore, the implementation of advanced prepayment systems, which allow consumers to pay for electricity in advance, offers additional benefits in terms of revenue assurance and customer convenience. A brief summary of the objectives is enumerated in the following section.

Objectives

1. **Research and Analysis of DLMS/COSEM and IDIS Standards:**

   The primary objective is to deeply understand the DLMS/COSEM and IDIS standards, which form the foundation of modern smart metering systems. The research involves analyzing these protocols, ensuring their alignment with smart grid technologies, and identifying how they can be effectively applied in the OneSait Smart Metering Project. This objective is crucial to establish a robust basis for the design and testing of compliant meters in UMEME's infrastructure.

2. **Design and Implementation of IDIS-Compliant Test Cases:**

   Developing a series of test cases to evaluate whether the meters comply with IDIS standards is the second critical objective. This involves the creation of simulation environments that replicate real-world scenarios to validate the performance of smart meters under various use cases. Each test case is designed to assess specific functionalities, such as registration, tariff programming, and data reading, ensuring the meters can meet the practical demands of a smart grid.

3. **Simulation of Use Cases to Assess Smart Meters:**

   Another important objective is to simulate various smart meters and their responses to the designed use cases. By running these simulations, the project evaluates how meters behave in realistic conditions. This objective is key to ensuring that the tested meters are ready for deployment in Uganda's electrical network, as it highlights both strengths and potential areas for improvement before field implementation.

4. **Exploring and Proposing Cybersecurity Measures:**

   Given the increasing importance of cybersecurity in smart infrastructures, this thesis also explores vulnerabilities within smart metering systems, including weak encryption, inadequate authentication, and risks of tampering. The objective is to recommend stronger security measures, such as advanced encryption protocols and enhanced authentication methods, to protect UMEME's infrastructure from potential cyber threats.

In conclusion, this thesis aims to contribute to the successful implementation of UMEME's OneSait Smart Metering Project by ensuring that the meters deployed are fully compliant with IDIS standards and equipped with the necessary cybersecurity measures. Through the design and implementation of comprehensive test cases, the thesis seeks to validate the performance of UMEME's smart meters and ensure that they are ready for deployment in the field. Additionally, by addressing the growing concern of cybersecurity, the thesis provides valuable insights into how UMEME can protect its infrastructure from potential threats, ensuring the long-term success of the smart metering project.

This work not only addresses the technical challenges associated with implementing smart metering systems but also highlights the broader impact of this technology on Uganda's energy sector. By enhancing the efficiency, reliability, and security of electricity distribution, smart meters have the potential to transform the way energy is managed and consumed in Uganda, contributing to a more sustainable and prosperous future.

Chapter 2

State of the Art

Smart grid technologies and smart metering systems have become integral components of modern energy management infrastructures. These systems are in great need of robust standards and protocols to ensure interoperability, security, and efficiency. The Device Language Message Specification/Companion Specification for Energy Metering (DLMS/COSEM) is a pivotal protocol suite in this domain, further enhanced by the Interoperable Device Interface Specification (IDIS). This section delves into the current state of the art, exploring the DLMS/COSEM and IDIS protocols, their implementation challenges, security considerations, data transfer mechanisms, and validation processes. An example of a typical smart meter communication topology is shown in figure 2.1.

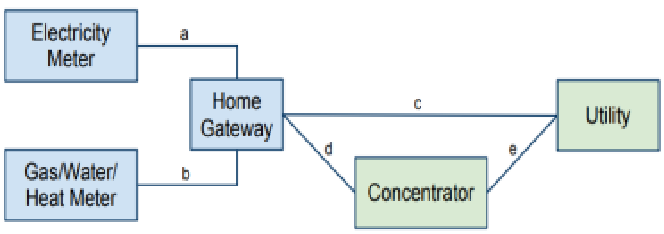

Figure 2.1: Smart Meter Communication Topology [1]

2.1 DLMS/COSEM Protocol

2.1.1 Overview of DLMS/COSEM

DLMS/COSEM is a standardized protocol suite designed to facilitate communication and data exchange in smart metering systems. DLMS defines the messaging structure, while COSEM specifies the data model used by smart meters, supporting various communication media such as PLC, RF, and cellular networks. This protocol suite is versatile and can be adapted to different deployment scenarios, enhancing the efficiency and reliability of smart grids. Thobekile Ngcobo (2022) [8] highlights that the DLMS/COSEM protocol's flexibility allows it to be implemented in various environments, making it a cornerstone for modern smart metering systems.

The comprehensive nature of DLMS/COSEM ensures that it can handle various functions required in smart metering, such as data collection, remote configuration, and real-time monitoring. The protocol's layered architecture allows for easy integration and scalability, which is crucial for large-scale deployments. Moreover, the standardized approach of DLMS/COSEM facilitates interoperability between devices from different manufacturers, ensuring seamless communication and data exchange within the smart grid infrastructure. The client-server model, provided by the DLMS protocol official document [2] is presented in figure 2.2. In greater detail, specifying the layers in figure 2.3. Additionally, the COSEM application process layers are presented in figure 2.4.

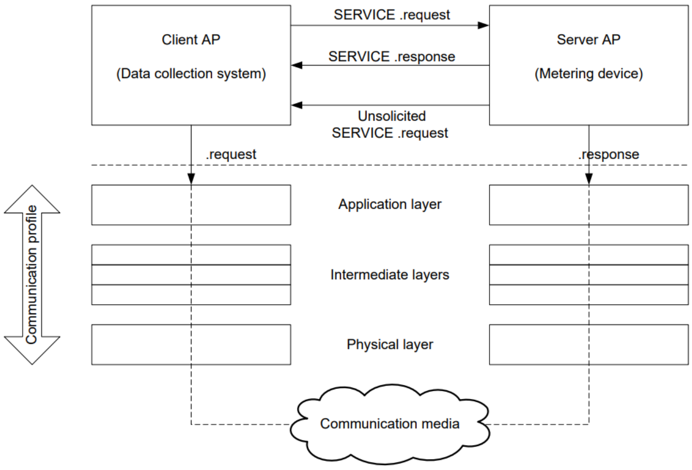

Figure 2.2: DLMS Client-Server Model [2]

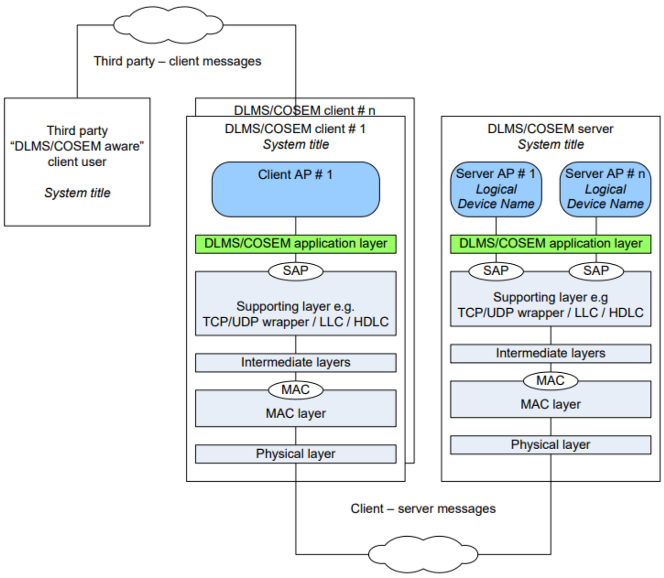

Figure 2.3: DLMS Client-Server General Layers [2]

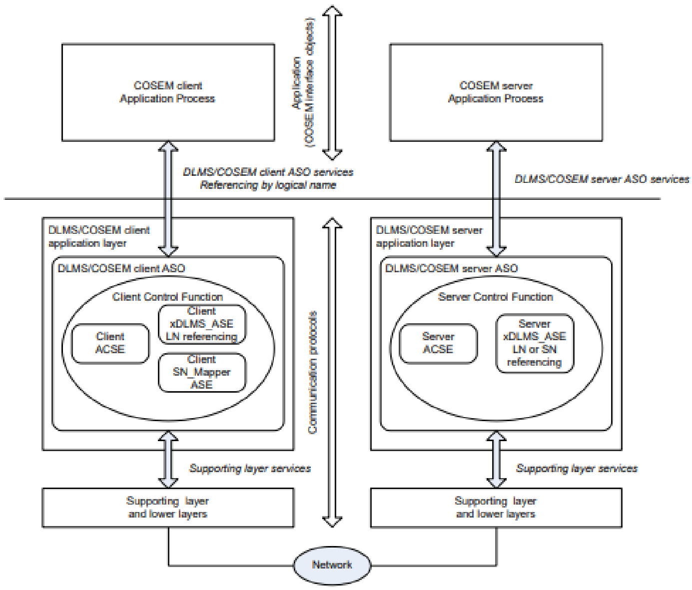

Figure 2.4: DLMS Client-Server Application Layers [2]

2.1.2 Integration with G3-PLC

The integration of DLMS/COSEM with G3-PLC has been shown to significantly enhance communication efficiency and reliability in smart grids. G3-PLC is a communication standard that uses power lines for data transmission, offering a reliable and cost-effective solution for smart metering. The combination addresses challenges like data transmission reliability and network congestion, thereby improving the overall performance of smart metering systems. Ngcobo [8] discusses how G3-PLC's robust error correction and efficient bandwidth utilization complement DLMS/COSEM, making it an ideal choice for enhancing smart grid communication infrastructure.

This integration enables the deployment of smart metering systems in areas with limited communication infrastructure, leveraging existing power lines for data transmission. It also reduces the overall cost of deployment and maintenance by eliminating the need for additional communication infrastructure. The use of G3-PLC with DLMS/COSEM ensures that data transmission is secure and reliable, which is critical for the accurate and timely collection of metering data. Additionally, the ability to use power lines for communication makes it easier to deploy smart metering systems in remote and rural areas, thus expanding the reach and benefits of smart grid technologies.

2.2 Implementation Challenges and Solutions

2.2.1 Practical Implementation

Implementing DLMS/COSEM in smart meters involves various practical challenges, including resource management, compatibility with existing hardware and software platforms, and ensuring interoperability. Grodan Štruklec [9] provides comprehensive guidance on addressing these challenges, highlighting the importance of meticulous planning and execution to achieve seamless integration and efficient data exchange. According to Štruklec, one of the main challenges is ensuring that the smart meters can handle the computational and memory requirements of DLMS/COSEM without compromising performance [9].

To address these challenges, developers need to optimize the firmware and hardware of smart meters to support DLMS/COSEM efficiently. This includes selecting appropriate microcontrollers, memory configurations, and communication modules that can handle the protocol's demands. Additionally, developers must ensure that the smart meters are compatible with existing infrastructure and can be easily integrated into the utility's backend systems. This involves thorough testing and validation of the meters to ensure they meet all the necessary standards and requirements.

Moreover, achieving interoperability between devices from different manufacturers requires adherence to strict standards and protocols. Developers must ensure that their implementation of DLMS/COSEM conforms to the specifications laid out by the DLMS User Association. This involves regular compliance testing and certification to verify that the smart meters can communicate seamlessly with other devices in the smart grid. By following these guidelines, developers can overcome the challenges associated with implementing DLMS/COSEM in smart meters and ensure a smooth and efficient deployment.

2.2.2 Simulation and Validation

Abhijeet Sahu (2019) [10] discusses the development and validation of a Cooperative Ultra-Narrowband (C-UNB) module for the NS-3 network simulator. This implementation demonstrates the feasibility of using C-UNB for smart metering communication, offering practical insights into simulating and validating DLMS/COSEM communication strategies. The NS-3 network simulator is a tool that allows developers to model and simulate the behavior of communication networks, providing valuable data on performance metrics such as latency, throughput, and packet loss.

By using NS-3, developers can create detailed simulations of smart metering communication networks, incorporating various factors such as network topology, traffic patterns, and environmental conditions. This enables them to identify potential issues and optimize the network design before deploying the smart metering system in the field. Sahu (2019) highlights that simulating the communication network allows for extensive testing of different scenarios and configurations, ensuring that the final implementation is robust and reliable.

There are several useful programs and applications to simulate the operation of communication lines and their protocols, more examples can be seen in [11].

2.3 Security Considerations

2.3.1 Cybersecurity Risks

Security is a paramount concern in smart metering systems due to the sensitive nature of the data involved and the critical role of these systems in the smart grid infrastructure. Wang (2022) [1] evaluates the cybersecurity risks associated with DLMS/COSEM smart meters using fuzzing testing methods. Fuzzing is a technique used to identify vulnerabilities by inputting random or unexpected data into the system and observing its behavior. The study identifies vulnerabilities in the High-Level Data Link Control (HDLC) layer, such as buffer overflow/underflow issues, which could lead to abnormal responses and denial-of-service attacks. These findings underscore the importance of robust security measures in DLMS/COSEM implementations.

To mitigate these risks, developers need to implement comprehensive security measures, including encryption, authentication, and intrusion detection systems. Encryption ensures that data transmitted between smart meters and the utility's backend systems is secure and cannot be intercepted by unauthorized parties. Authentication mechanisms verify the identity of devices communicating within the network, preventing unauthorized access and tampering. Intrusion detection systems monitor the network for suspicious activity and alert administrators to potential security breaches.

Additionally, regular security assessments and updates are crucial to maintaining the integrity of the smart metering system. Developers must stay informed about the latest cybersecurity threats and vulnerabilities and implement patches and updates as necessary to protect the system. Wang [1] emphasizes the need for a proactive approach to cybersecurity, ensuring that potential threats are identified and mitigated before they can cause harm. By adopting these security practices, developers can increase the protection of smart metering systems from cyberattacks and ensure the reliability and integrity of the data they collect.

Chapter 5 will include a later section about the current metering data security protocol implemented in UMEME's framework, and a proposal to improve and implement additional security layers to enhance data privacy and prevent cybersecurity risks.

A hierarchical key management scheme for AMI systems based on the DLMS/-COSEM standard is proposed by Mohammadali (2014) [12]. This method enhances security by ensuring efficient and secure communication between smart meters and utility providers, addressing the challenges of key distribution and management in large-scale AMI systems. The proposed scheme involves a multi-level key hierarchy, where different keys are used for different levels of communication, ensuring that a compromise at one level does not affect the entire system.

At the top level, a master key is used to generate and distribute session keys to individual smart meters. These session keys are used for encrypting and decrypting communication between the meters and the utility's backend systems. The hierarchical structure ensures that each meter has its unique session key, preventing attackers from gaining access to multiple meters if they manage to compromise one. Additionally, the use of session keys limits the exposure of the master key, further enhancing the security of the system.

Mohammadali [12] presents that the hierarchical key management scheme also includes mechanisms for key renewal and revocation. Key renewal ensures that session keys are periodically updated, reducing the risk of long-term exposure in case of a compromise. Key revocation allows the utility to invalidate compromised keys and issue new ones, ensuring that the security of the system is maintained even if some keys are compromised. Implementing such a robust key management scheme is essential for protecting smart metering systems from unauthorized access and ensuring secure communication.

Lieskovan (2019) [13] provides a comprehensive survey of the security challenges in smart grid networks, focusing on DLMS/COSEM. The authors discuss various security mechanisms, authentication methods, encryption techniques, and vulnerabilities in smart meters and concentrators. It also outlines historical cyberattacks on smart grids and proposes future security solutions. Lieskovan emphasizes the importance of a multi-layered security approach, combining various security measures to protect smart metering systems from different types of threats [13].

The survey [13] emphasizes that one of the primary security challenges in smart grids is ensuring the integrity and confidentiality of the data collected by smart meters. To address this challenge, developers must implement end-to-end encryption and secure data storage solutions. End-to-end encryption ensures that data is encrypted at the source (smart meter) and only decrypted at the destination (utility backend system), preventing unauthorized access during transmission. Secure data storage solutions protect the data at rest, ensuring that even if an attacker gains physical access to a meter, they cannot access the stored data without the proper keys.

Another significant challenge is securing the communication network. Lieskovan (2019) discusses the importance of using secure communication protocols and regular network monitoring to detect and respond to anomalies. Network segmentation can also be used to isolate different parts of the network, limiting the potential impact of a breach. By combining these measures, developers can create a robust security framework that protects smart metering systems from a wide range of cyber threats. There are several security levels when it comes to the communication between the client and the server as figure 2.5 shows.

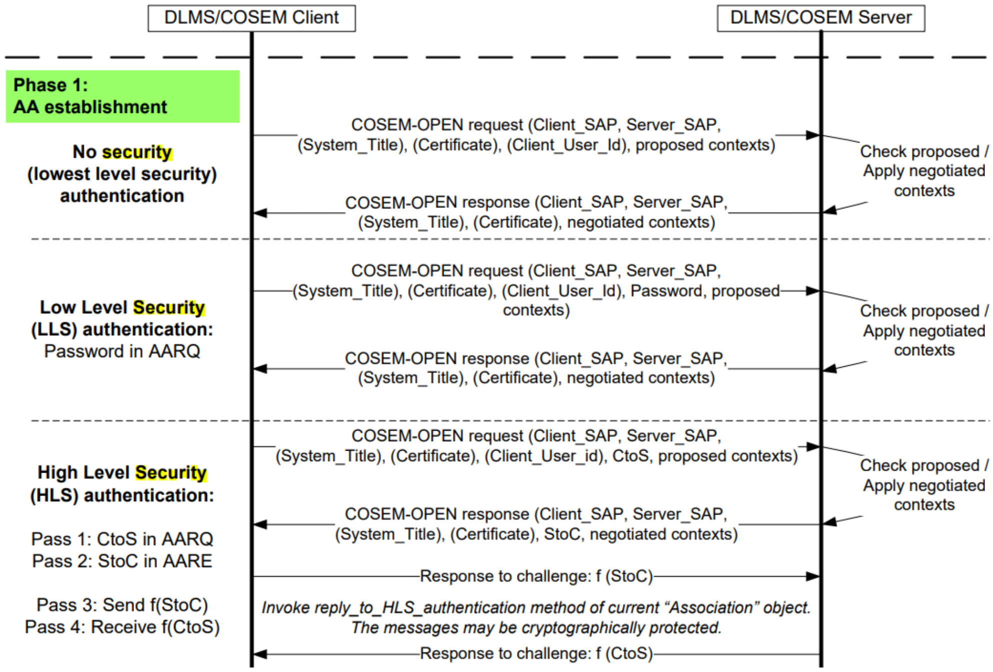

Figure 2.5: DLMS Client-Server Security Levels [2]

The survey also emphasizes the need for ongoing security training and awareness programs for utility staff. Regular training ensures that staff are aware of the latest security threats and best practices, enabling them to respond effectively to potential incidents. Additionally, security policies and procedures should be regularly reviewed and updated to reflect the evolving threat landscape. By adopting a proactive and comprehensive approach to security, utilities can protect their smart metering systems and ensure the integrity and reliability of their data.

2.4 Data Transfer and Processing

2.4.1 Packet Transfer Methods

Efficient data transfer is crucial for the reliability and performance of smart metering systems. Kheaksong (2014) reviews various packet transfer methods using

DLMS/COSEM protocols in smart grid applications. The study examines the efficiency and reliability of different packet transfer techniques, providing insights into optimizing data transmission for smart metering systems. One of the key findings is that the choice of packet transfer method can significantly impact the overall performance of the system, including factors such as data integrity, transmission speed, and error handling [14].

The study [14] shows several packet transfer methods, including unicast, multicast, and broadcast. Each method has its advantages and limitations, depending on the specific requirements of the smart metering application. For example, unicast is suitable for targeted communication between individual meters and the utility's backend system, ensuring reliable and secure data transfer. However, it may not be efficient for large-scale deployments due to the high number of individual transmissions required.

Multicast, on the other hand, allows for the efficient distribution of data to multiple meters simultaneously. This method is particularly useful for broadcasting firmware updates or configuration changes to a group of meters. However, it requires robust error correction mechanisms to ensure that all meters receive the data correctly. Broadcast is the most efficient method for sending data to all meters in the network, but it is also the least secure, as the data can be intercepted by any device within the broadcast range.

Kheaksong [14] emphasizes the importance of selecting the appropriate packet transfer method based on the specific requirements of the smart metering system. By optimizing the data transfer mechanisms, utilities can ensure reliable communication and efficient operation of their smart metering infrastructure. Additionally, the study provides recommendations for enhancing the performance of packet transfer methods, such as implementing advanced error correction techniques and optimizing network protocols.

2.4.2 Data Processing Setups

Wulfing (2022) evaluates different setups for processing data using the DLMS/-COSEM protocol in smart metering systems. The paper focuses on the efficiency, reliability, and scalability of various data processing configurations, offering practical recommendations for optimizing smart meter data handling. One of the primary challenges in data processing is managing the large volume of data generated by smart meters, which can overwhelm traditional data processing systems [15].

To address this challenge, Wulfing (2022) suggests implementing distributed data processing architectures that can handle the data processing workload across multiple nodes. This approach improves the scalability and reliability of the system, ensuring that it can handle the growing volume of data as the smart metering infrastructure expands. Additionally, distributed processing can reduce latency by processing data closer to the source, providing more timely insights for decision-making.

The study [15] also highlights the importance of data aggregation and filtering techniques to manage the data volume effectively. By aggregating data at intermediate nodes, utilities can reduce the amount of data that needs to be transmitted to the central processing system, thereby reducing network congestion and improving overall efficiency. Filtering techniques can be used to remove redundant or irrelevant data, ensuring that only valuable information is processed and stored.

Another critical aspect of data processing is ensuring data integrity and accuracy. Wulfing (2022) emphasizes the need for robust validation and verification mechanisms to detect and correct errors in the data [15]. This includes implementing checksums and other error-detection algorithms to identify corrupted data and using redundancy to recover from data loss. By ensuring the integrity and accuracy of the data, utilities can make more informed decisions and improve the reliability of their smart metering systems.

2.4.3 Comparative Analysis of Communication Protocols

Feuerhahn (2011) conducts a comparative analysis of DLMS/COSEM with other communication protocols such as SML and IEC 61850. This analysis highlights the respective advantages and limitations of each protocol in smart metering applications, providing valuable insights for making informed decisions about protocol selection. The study emphasizes that the choice of protocol can significantly impact the performance, scalability, and interoperability of smart metering systems [16].

DLMS/COSEM is widely recognized for its comprehensive data model and robust communication capabilities, making it a popular choice for smart metering applications. It supports a wide range of functionalities, including data collection, remote configuration, and real-time monitoring. Additionally, DLMS/COSEM's layered architecture and support for various communication media make it highly adaptable and scalable.

SML, on the other hand, is a simpler protocol that is primarily used for data exchange in smart meters. It is known for its lightweight implementation and ease of use, making it suitable for smaller-scale deployments. However, SML lacks the comprehensive data modeling capabilities of DLMS/COSEM, which can limit its applicability in more complex smart metering systems.

IEC 61850 is a communication standard widely used in substation automation and smart grid applications. It provides a comprehensive framework for data exchange and system integration, supporting advanced functionalities such as real-time control and protection. While IEC 61850 is highly versatile and powerful, its complexity and high implementation costs can be a barrier for smaller utilities and deployments.

Feuerhahn (2011) [16] concludes that the choice of protocol should be based on the specific requirements of the smart metering application. For large-scale deployments requiring comprehensive data modeling and advanced communication capabilities, DLMS/COSEM is the preferred choice. For simpler, cost-effective implementations, SML may be more suitable. For applications requiring advanced real-time control and protection, IEC 61850 offers the necessary capabilities.

By understanding the strengths and weaknesses of each protocol, utilities can make informed decisions that align with their specific needs and constraints. This comparative analysis provides a valuable framework for evaluating different communication protocols and selecting the most appropriate one for smart metering projects.

2.5 Validation and Compliance Testing

In the field of smart metering and energy management, validation and compliance testing are essential to ensuring systems meet industry standards and operate securely and efficiently. This section addresses the critical processes involved in implementing DLMS/COSEM protocols and IDIS standards, which are crucial for maintaining interoperability and security in smart metering systems.

2.5.1 ValiDLMS Framework

Validating and securing DLMS/COSEM implementations is essential to ensure compliance with industry standards and to identify potential security vulnerabilities. Mendes (2018) presents the ValiDLMS framework, an open-source solution for validating and securing DLMS/COSEM implementations. The framework combines fuzzing techniques and security analysis to detect bugs and non-conformance issues, providing practical tools and methods for ensuring protocol compliance and identifying security vulnerabilities [3].

The ValiDLMS framework is designed to be used in conjunction with Wireshark, a widely-used network protocol analyzer. By leveraging Wireshark's powerful packet capture and analysis capabilities, ValiDLMS can perform detailed inspections of DLMS/COSEM communication, identifying any deviations from the protocol specifications. This enables developers to quickly identify and address issues, ensuring that their implementations are fully compliant with the DLMS/-COSEM standards. The framework schematic can be seen in figure 2.6.

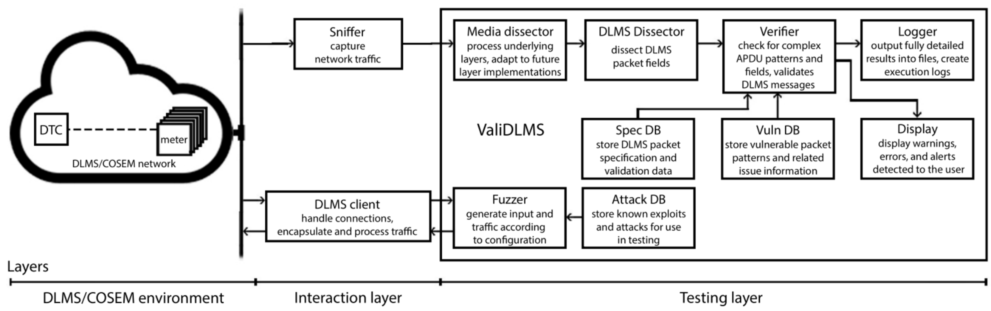

Figure 2.6: ValiDLMS Framework [3]

Fuzzing techniques used in ValiDLMS involve sending random or unexpected inputs to the DLMS/COSEM implementation and observing its behavior. This helps identify vulnerabilities such as buffer overflows, input validation errors, and other security weaknesses. By systematically testing the implementation with a wide range of inputs, ValiDLMS can uncover hidden bugs and security issues that might not be detected through conventional testing methods.

In addition to fuzzing, ValiDLMS incorporates security analysis tools that examine the implementation's security features, such as encryption, authentication, and access control. These tools help ensure that the implementation adheres to best practices for securing smart metering communication. Mendes (2018) emphasizes that by using ValiDLMS, developers can achieve a higher level of confidence in the security and reliability of their DLMS/COSEM implementations.

2.5.2 Compliance Testing

Chien (2023) details the implementation of an OIML R46 communication unit for DLMS/COSEM Security Suite 1 and the process of passing the CTT V3.1 compliance test. This study emphasizes the importance of security features and compliance testing in ensuring reliable smart meter communication. The OIML R46 standard specifies the requirements for the accuracy, reliability, and security of smart meters, making compliance with this standard essential for utilities [17].

The compliance testing process involves a series of rigorous tests to verify that the smart meter meets all the specified requirements. This includes testing the accuracy of the meter's measurements, the reliability of its communication, and the robustness of its security features. By passing the CTT V3.1 compliance test, the smart meter demonstrates its adherence to the highest standards of performance and security.

Chien (2023) highlights the challenges associated with achieving compliance, such as ensuring that the implementation is free from bugs and vulnerabilities, and that it performs reliably under various conditions. The study provides detailed insights into the testing procedures and best practices for achieving compliance, offering valuable guidance for developers and utilities.

By conducting comprehensive compliance testing, utilities can ensure that their smart meters are reliable, secure, and compliant with industry standards. This not only enhances the performance and security of the smart metering system but also builds trust with customers and stakeholders, ensuring the long-term success of the deployment.

2.6 IDIS Protocol and Standards

The IDIS protocol enhances DLMS/COSEM by providing specific use cases and guidelines to ensure interoperability among smart meters from different manufacturers. Their first package was published in 2010 and it included optimized use cases for narrowband power line communication networks. The package 2, published in 2015 included internet protocol support for TCP/IP networks (G3-PLC, 2G, 3G and 4G), which has been mentioned in previous sections, with its respective connection to DLMS/COSEM. [18]. The latest release, Package 3, includes enhancements for meter reading, management, and power quality functions, along with improved security measures [4]. These enhancements are vital for maintaining the relevance and efficacy of the IDIS protocol in evolving smart grid environments.

Package 3 introduces several new features designed to improve the functionality and security of smart metering systems. These include advanced meter reading capabilities that allow for more accurate and detailed data collection, improved power quality monitoring to ensure the reliability and stability of the electrical supply, and enhanced security measures to protect against unauthorized access and cyberattacks. By incorporating these features, Package 3 ensures that smart meters can meet the increasing demands of modern smart grid applications.

The specific system architecture modules supported by package 3 are shown in figure 2.7. These being the ones in green (I3, I3.1 and I2). For I1 and IE-M IDIS Package 3 defines the required functionality, but the choice of the physical interface is left to the manufacturer [4]. I5, I4, I3.2 and the local interface are not in scope of the package. Moreover, the interface I3.1 supports only PLC technologies that use IPv4/6 communication. The IP and higher communication layers for I3.1 are identical to those for I3. The COSEM client can be situated in either the HES or the DC.

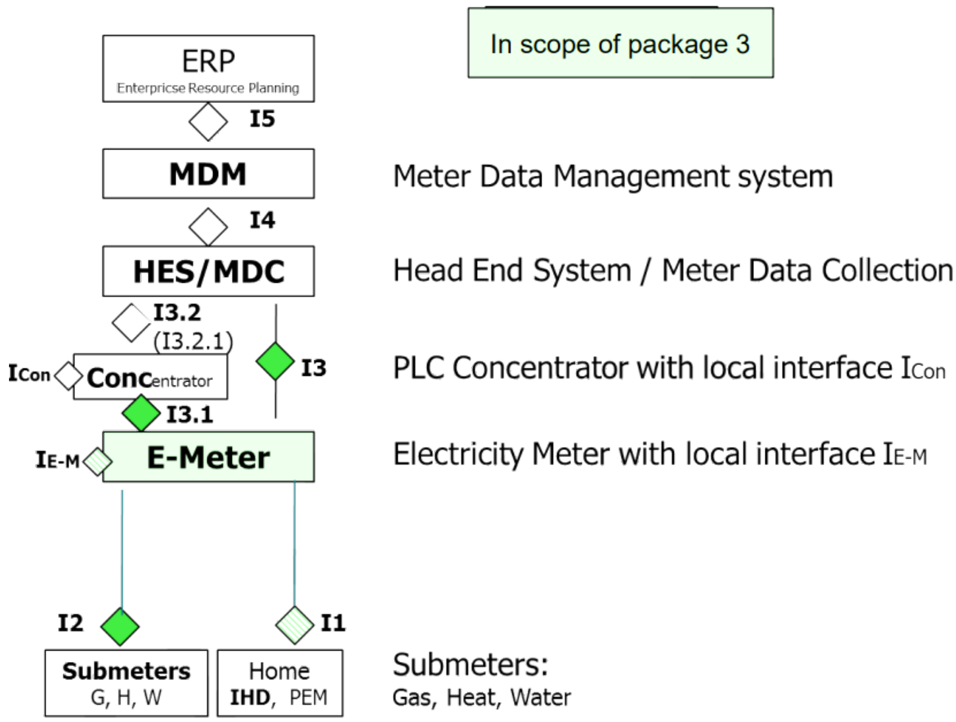

Figure 2.7: System Architecture Supported by IDIS Package 3 [4]

The IDIS protocol also includes specific use cases that outline the expected behavior and performance of smart meters in various scenarios. These use cases provide a clear framework for developers and utilities, ensuring that all devices conform to a consistent set of standards. This not only simplifies the development process but also ensures that all smart meters can interoperate seamlessly, regardless of the manufacturer. The use cases cover a wide range of functionalities, from basic meter reading and billing to advanced features such as remote disconnection and load management.

The IDIS Association website offers comprehensive information about the IDIS Companion Specification, including its benefits, the certification process, and the evolution of the standards [19]. This resource is invaluable for stakeholders and companies seeking to understand and implement IDIS-compliant smart metering solutions.

The IDIS Companion Specifications, based on the International Electrotechnical Commission (IEC) 62056 DLMS/COSEM standards, detail specific use cases and options to ensure complete interoperability of smart meter devices. These specifications are backed by a robust, independent certification process provided by the IDIS Association. In November 2021, the IDIS Association merged with the DLMS User Association to further promote interoperability and open standards. The DLMS User Association will expand the DLMS certification program by incorporating the IDIS methodology, introducing new initiatives for multi-utility smart metering profiles (heat, water, and gas), and offering comprehensive interoperability and compatibility testing services. Additionally, DLMS will maintain the existing services previously offered by IDIS [19]. This is the main reason DLMS has been and will be so present in this document, IDIS, originally based in DLMS/COSEM, now joined forces to further expand and keep their proposal, use cases and benefits.

The IDIS certification process is rigorous, involving extensive testing to ensure that devices meet the required standards for interoperability and performance. Certified devices are tested for their ability to communicate effectively with other IDIS-compliant devices, ensuring that they can be integrated seamlessly into existing smart grid infrastructure. The certification process also includes security testing to ensure that devices can protect against common cyber threats and vulnerabilities. These tests will be simulated and presented later in the paper for the purpose of demonstrating the usefulness of the standard when it comes to interoperability.

The IDIS Association also provides ongoing support and updates to ensure that the standards remain relevant and up-to-date with the latest technological advancements. This includes regular updates to the Companion Specification, incorporating new features and improvements based on feedback from industry stakeholders. By maintaining a robust and evolving set of standards, the IDIS Association ensures that smart metering systems can continue to meet the needs of utilities and consumers alike.

2.7 Qualitative Testing

Qualitative testing is essential for ensuring that smart meters comply with DLMS/COSEM standards. The study published on Electrical India (2024) discusses various tests conducted on DLMS/COSEM ICS compliant energy meters, covering authentication mechanisms, instantaneous parameters, load profiles, and security features [20]. Such rigorous testing ensures that smart meters perform reliably and meet necessary standards.

The qualitative testing process involves a series of comprehensive tests designed to evaluate the performance and functionality of smart meters. These tests include verifying the accuracy of meter readings, ensuring that the meters can handle various load profiles, and testing the reliability of communication with the utility's backend systems. By conducting these tests, utilities can identify and address potential issues before deploying the meters in the field.

Authentication mechanisms are a critical aspect of smart meter security, ensuring that only authorized devices and users can access the meter's data and functionality. The study highlights the importance of implementing robust authentication methods, such as digital certificates and secure key exchange protocols, to protect against unauthorized access. By ensuring that smart meters have strong authentication mechanisms, utilities can safeguard the integrity and confidentiality of their data.

Instantaneous parameters, such as voltage, current, and power, are essential for monitoring the performance of the electrical supply and identifying potential issues. The study evaluates the accuracy and reliability of these measurements, ensuring that the meters can provide accurate and timely data. This is crucial for maintaining the stability and reliability of the smart grid, allowing utilities to respond quickly to any issues that arise.

Load profiles provide valuable insights into the consumption patterns of consumers, helping utilities to optimize their energy management strategies. The study tests the meters' ability to accurately record and transmit load profile data, ensuring that utilities can rely on this information for billing and demand forecasting. By ensuring that smart meters can handle a wide range of load profiles, utilities can better manage their resources and improve the efficiency of their operations.

Security features are also a critical aspect of qualitative testing, ensuring that smart meters can protect against cyber threats and unauthorized access. The study evaluates the meters' encryption and authentication mechanisms, ensuring that they meet the required standards for security. By implementing robust security features, utilities can protect their smart metering systems from cyberattacks and ensure the integrity of their data.

2.8 Conclusion

The current state of the art in DLMS/COSEM and IDIS protocols highlights the critical aspects of implementing, securing, and optimizing smart metering systems. From ensuring interoperability and robust security to optimizing data transfer and validating compliance, the reviewed studies provide comprehensive insights and practical solutions for deploying effective smart metering systems. These insights are essential for developing and implementing test cases to verify IDIS protocol compliance in UMEME's OneSite smart metering project, ensuring reliable and secure operations within the smart grid infrastructure.

Chapter 3

UMEME's Project Paradigm

UMEME, Uganda's largest electricity distribution company, plays a crucial role in the country's energy sector, managing the distribution and sale of electricity to millions of Ugandans. As the primary electricity distributor, UMEME is responsible for ensuring the reliable supply of electricity, maintaining the distribution infrastructure, and implementing innovative solutions to enhance service delivery and efficiency.

The OneSite smart metering project is one of UMEME's key initiatives aimed at modernizing Uganda's electricity distribution network. This project involves the deployment of advanced smart meters across UMEME's service area to improve billing accuracy, reduce energy losses, and enhance overall operational efficiency. By leveraging state-of-the-art technology, the OneSite project aims to transform the way electricity is monitored and managed, contributing significantly to the reliability and sustainability of Uganda's electricity supply.

3.1 Uganda's Electricity Market

3.1.1 Overview of the Electricity Sector

Uganda's electricity sector has seen significant transformation over the past two decades, marked by substantial increases in generating capacity and efforts to expand access to electricity. From a modest capacity of around 320 megawatts (MW) in 2002, Uganda's electricity generation has surged to over 1,346 MW as of early 2023, largely due to the development of large hydroelectric projects such as the Karuma and Isimba dams [21].

Historical Context and Development

The historical development of Uganda's electricity sector can be traced back to the early establishment of the Uganda Electricity Board (UEB) in 1948, which was responsible for generating, transmitting, and distributing electricity across the country. The sector has since evolved with various reforms, including the unbundling of the UEB into three distinct entities: Uganda Electricity Generation Company Limited (UEGCL), Uganda Electricity Transmission Company Limited (UETCL), and Uganda Electricity Distribution Company Limited (UEDCL) [21]. This unbundling aimed to improve efficiency and attract private investment.

Key Players in the Market

The primary entities involved in Uganda's electricity market include:

1. **Uganda Electricity Generation Company Limited (UEGCL):**

   Oversees the generation of electricity.

2. **Uganda Electricity Transmission Company Limited (UETCL):**

Manages the transmission of electricity from generation sites to distribution networks.

3. **UMEME:**

   The primary electricity distributor responsible for distributing and selling electricity to end-users across Uganda.

The electricity market in Uganda is primarily dominated by renewable energy sources, with hydroelectric power contributing approximately 80% of the total generating capacity [21]. The remaining renewable energy sources include solar photovoltaic (PV) systems and biomass, which play a crucial role in diversifying the energy mix and enhancing sustainability [21].

Energy Mix and Consumption Patterns

In 2021, traditional biomass, including wood and charcoal used primarily for cooking, accounted for around 87% of Uganda's total final energy consumption [21]. Petroleum products, mainly used for transportation, made up about 11% [21], while electricity constituted a mere 2% of the final energy consumption [21]. Households are the largest consumers of energy, representing 61% of the total final consumption, followed by industry (22%), transportation (7%), and commercial and public services (9%) [21].

Challenges and Opportunities

One of the major challenges in Uganda's electricity sector is the low electrification rate, particularly in rural areas. As of 2022, only about 20% of the population had access to electricity from the national grid, while an additional 10% relied on solar home systems [21]. The scattered and sparse settlement patterns of the predominantly rural population increase the complexity and cost of extending the grid [21].

Investment in transmission and distribution infrastructure has not kept pace with the growth in generation capacity, resulting in underutilization of the generated power. The government's decision to avoid subsidizing electricity consumption has led to high tariffs, which further restricts electricity consumption and poses challenges to financing grid extensions and maintenance [21].

Future Prospects

Uganda's National Energy Policy 2023 outlines ambitious goals to enhance energy access and diversify the energy mix. The policy aims to increase the share of the population with access to electricity to 80% by 2040 and raise per capita electricity consumption to 3,668 kilowatt hours (kWh) from the current 100 kWh [21]. This will involve substantial investments in both grid and off-grid solutions, alongside regulatory reforms to encourage private sector participation and improve the overall efficiency and sustainability of the electricity sector.

By continuing to build on its renewable energy base and addressing the challenges in transmission and distribution, Uganda aims to secure a reliable, affordable, and sustainable energy future for its population.

3.1.2 Regulatory Environment

The Electricity Regulatory Authority (ERA) [22] plays a central role in regulating Uganda's electricity sector. Established in 2000 under the Electricity Act of 1999, ERA's mandate includes overseeing the generation, transmission, distribution, sale, export, and import of electrical energy in Uganda. The authority's primary objectives are to guide the liberalization of the electricity industry, manage licensing, set tariffs, ensure safety standards, and promote the efficient and reliable supply of electricity.

Role and Responsibilities of ERA

ERA is responsible for creating a conducive environment for investment in the electricity sector by ensuring that the regulatory framework supports both public and private sector participation. The main responsibilities of ERA are:

1. **Licensing:**

   ERA is responsible for managing the issuance of licenses for electricity generation, transmission, distribution, and sales. The licensing process involves rigorous evaluation to ensure that projects meet technical, financial, and environmental standards. Applicants must provide comprehensive details about their projects, including technical specifications, financial viability, and environmental impact assessments. This thorough vetting process ensures that only projects with fairly strong credentials receive licenses, fostering a reliable and efficient electricity sector [23].

2. **Tariff Setting:**

   ERA's role in tariff setting is central to ensuring that electricity prices in Uganda reflect the true cost of service delivery while being fair to consumers and attractive to investors. The tariff setting process involves detailed cost of service studies, public hearings, and stakeholder consultations, making it a transparent and participatory process [24]. The primary goal is to balance the interests of all stakeholders, including consumers, investors, and the government, while ensuring the financial sustainability of the electricity sector.

   The process begins with the determination of the Base Tariff, which is calculated at the start of each calendar year. The Base Tariff is set to cover the total revenue requirement of the electricity sector, ensuring that utilities can meet their operating expenses, maintenance costs, and provide a reasonable return on investment. The Base Tariff remains constant throughout the year, providing a stable pricing framework for consumers and investors alike [24].

To determine the Base Tariff, ERA uses a comprehensive tariff model that incorporates various cost elements, including generation costs, transmission costs, distribution costs, and operational and maintenance expenses. The model also takes into account the projected energy demand, system losses, and collection rates. According to [24, p. 36], this model includes input sheets for monthly data, bulk energy, power purchase, distribution, generation, and transmission costs, as well as economic and commercial costs.

One of the critical components of the tariff model is the inclusion of macroeconomic parameters such as exchange rates, inflation rates, and fuel prices. These parameters are crucial because they can significantly impact the costs incurred by utilities, especially those costs associated with power purchase agreements denominated in foreign currencies. The "Quarterly Tariff Adjustment Methodology" document [25, p. 7] outlines how these macroeconomic parameters are incorporated into the tariff model to adjust the Base Tariff on a quarterly basis.

The Quarterly Tariff Adjustment Methodology (QTAM) is designed to account for fluctuations in macroeconomic parameters that occur throughout the year. This methodology ensures that the tariffs remain responsive to changes in the economic environment, thus protecting both the consumers and the utilities from unforeseen financial burdens. The QTAM includes three main adjustment factors: the Fuel Adjustment Factor, the Exchange Rate Adjustment Factor, and the Inflation Adjustment Factor [25].

The Fuel Adjustment Factor accounts for changes in the cost of fuel used by thermal power plants. Since these costs can fluctuate significantly due to changes in international fuel prices, the adjustment factor ensures that the tariffs reflect these variations. The calculation of this factor involves assessing the difference between the actual fuel costs incurred and the base fuel costs assumed at the beginning of the year. This difference is then adjusted for system losses and applied to the tariff [25, p. 8].

The Exchange Rate Adjustment Factor addresses the impact of fluctuations in the foreign exchange rate on the costs of power purchases and other operational expenses incurred in foreign currencies. Since many power purchase agreements and equipment costs are denominated in US dollars, any depreciation of the Uganda Shilling can significantly increase these costs. The exchange rate adjustment factor calculates the impact of these currency fluctuations and adjusts the tariff accordingly [25, p. 11].

The Inflation Adjustment Factor accounts for changes in the domestic inflation rate, which can affect the operating and maintenance costs of utilities. This factor ensures that the tariffs keep pace with inflationary pressures, thus maintaining the financial viability of the utilities. The inflation adjustment is based on the Consumer Price Index (CPI) published by the Uganda Bureau of Statistics [26] and other relevant economic indices [25, p. 15].

To implement these adjustments, ERA uses a structured and transparent process. At the beginning of each calendar year, ERA approves the Base Tariff, taking into account the utilities' revenue requirements and the macroeconomic parameters. The Base Tariff is then adjusted quarterly based on the updated values of the macroeconomic parameters. The adjusted tariffs are published by ERA, ensuring transparency and accountability in the process. These quarterly adjustments ensure that the tariffs remain fair and reflective of the current economic conditions, thereby protecting both the consumers and the utilities from financial instability [25, p. 19].

3. **Regulatory Compliance:**

   ERA enforces compliance with electricity regulations and standards through continuous monitoring and evaluation. The authority conducts regular inspections and audits to ensure adherence to safety and operational standards. This rigorous oversight helps to maintain high industry standards, ensuring that all licensed entities operate within the legal and regulatory framework. Non-compliant entities are subject to corrective measures and sanctions, which may include fines or revocation of licenses. ERA's compliance and enforcement manual [27] outlines the procedures for maintaining these high standards, which are crucial for ensuring the reliability and safety of electricity supply across Uganda.

   According to the manual [27], effective compliance is vital for the safe, reliable, and sustainable supply of electricity in Uganda. Compliance is not only about adhering to legal requirements but also about upholding the spirit of the regulations, which means acting in good faith and considering the broader impact of one's actions on all stakeholders. This approach helps build trust and ensures that the sector operates transparently and efficiently.

   ERA employs a tiered compliance approach, which includes legal compliance, risk-based compliance, and best practice compliance. Legal compliance involves ensuring that licensees fully adhere to the provisions of their licenses, the Electricity Act, and the associated regulations. This approach is fundamental to managing statutory non-compliance and maintaining the integrity of the regulatory framework. To facilitate this, ERA has developed compliance checklists that guide licensees and the compliance officers in ERA, simplifying the legal obligations and promoting voluntary compliance [27, p. 7].

The risk-based compliance approach focuses on identifying and mitigating compliance risks. ERA evaluates the potential damage presented by various risks and addresses these threats systematically. This approach incorporates best practices that are generally accepted as highly effective and cost-efficient. Best practices in the electricity sector, also known as Prudent Utility Practices, involve standard methods and procedures expected from skilled and experienced operators. These practices are not limited to optimal methods but rather encompass a spectrum of reasonable and prudent practices [27, p. 8].

ERA's enforcement strategy is both proactive and responsive. The authority conducts regular technical and commercial audits to ensure compliance with operational, safety, and customer service standards. Technical audits involve on-site inspections of licensees' plants and equipment, assessing compliance with maintenance, safety, and operational procedures. The environmental team plays a crucial role during these inspections, ensuring that licensees comply with environmental policies and safety regulations. The legal team focuses on corporate governance practices [27, p. 12].

Commercial audits are conducted annually and focus on customer-facing aspects such as billing, metering, and customer service. These audits aim to provide a comprehensive view of performance across the country, highlighting areas where particular licensees excel or underperform. The results of these audits are made publicly available, promoting transparency and accountability in the sector. ERA also benchmarks the performance of licensees against industry standards and uses this data to drive improvements and foster healthy competition among service providers [27, p. 13].

ERA's compliance and enforcement strategy also includes the development and implementation of an annual compliance program. This program focuses on key performance areas and prioritizes reporting on finance, operations, and safety. Licensees are required to submit detailed annual compliance statements, which are reviewed and evaluated by ERA. The findings from these evaluations, along with the results of commercial and technical audits, form the basis of ERA's annual compliance assessment, which is published for public scrutiny [27, p. 14].

ERA's enforcement actions are guided by a decision matrix that takes into account various factors, including safety consequences, the nature of the non-compliance, its impact on the regulatory process, and the licensee's history of compliance. This graded approach ensures that the regulatory response is proportionate to the risks posed by the non-compliance and aims to achieve the most appropriate outcomes, whether through corrective actions, sanctions, or other enforcement measures [27, p. 35].

4. **Policy Implementation:**

   ERA plays a pivotal role in implementing national energy policies [28], such as the Energy Policy 2023 [21], which aims to achieve universal access to electricity and increase the share of renewable energy. The authority supports the development of off-grid solutions and the integration of renewable energy sources into the national grid. This involves facilitating the development of large-scale renewable energy projects and encouraging smaller, community-based renewable energy initiatives. By fostering a diverse and sustainable energy mix, ERA contributes to the country's long-term energy security and sustainability. ERA also collaborates with other governmental and non-governmental organizations to promote energy efficiency and conservation.

The Energy Policy 2023 [21] focuses on expanding the electricity transmission and distribution grid networks. This is crucial for enhancing the reliability and reach of the electricity supply, particularly to underserved rural areas. For example, the policy outlines plans for significant investments in extending the national grid to rural areas, aiming to increase rural electrification rates from the current 10% to 30% by 2030 [21, p. 23]. Additionally, integrating renewable energy sources such as solar and wind is a key strategy. ERA's role includes regulatory oversight and facilitating investments necessary for these developments. The policy highlights specific projects like the development of solar mini-grids in remote areas to provide reliable electricity access [21, p. 24].

Increasing energy efficiency across various sectors is another critical focus area of the policy. ERA promotes energy efficiency measures by conducting energy audits and raising public awareness about energy conservation. For instance, the policy mandates the implementation of energy-efficient technologies in public buildings and industries, aiming to reduce overall energy consumption by 20% by 2030 [21, p. 13-14]. These measures are complemented by initiatives to promote the use of alternative energy sources, further reducing the environmental impact of energy consumption. Specific programs include the distribution of energy-efficient lighting and the establishment of standards for energy-consuming appliances.

ERA collaborates extensively with other governmental and non-governmental organizations to promote energy efficiency and conservation. This collaboration includes working with international partners to bring in technical expertise and funding for energy projects. For example, initiatives like the Sustainable Energy for All (SE4ALL) framework guide these collaborative efforts, aiming to double the global rate of improvement in energy efficiency by 2030 [21, p. 17]. By leveraging these partnerships, ERA aims to implement comprehensive and effective energy policies that meet the country's development objectives. Programs such as the partnership with the World Bank to fund renewable energy projects demonstrate ERA's commitment to these goals [21, p. 37].

5. **Consumer Protection:**

   ERA ensures that consumer interests are safeguarded through a structured process for handling complaints and enforcing consumer rights. The authority promotes transparency in service delivery and conducts public awareness campaigns to educate consumers about their rights and responsibilities. These campaigns also highlight the importance of energy conservation and the benefits of renewable energy. ERA works to ensure that electricity providers offer reliable and high-quality service to consumers, maintaining trust and satisfaction. The authority's efforts in consumer protection ensure a fair and accountable electricity sector, where consumers are well-informed and their grievances are addressed promptly [29].

6. **Regulatory Framework and Reforms:**

   The regulatory framework under ERA includes a variety of laws, regulations, guidelines, and standards that govern the electricity sector [30]. One of the key regulations is the Electricity (Installation Permits) Regulations [31], which governs the issuance of permits for electrical installations. This regulation ensures that only certified and competent personnel handle electrical works, thus maintaining safety and quality standards. The regulations specify the requirements for obtaining installation permits, procedures for inspecting installations, and penalties for non-compliance. These measures are critical for maintaining the integrity and safety of electrical installations across the country.

Another significant regulatory instrument is the Feed-in-Tariff (FiT). This tool aims to promote electricity generation from renewable sources by providing guaranteed tariffs, making investments financially viable. Initially for projects up to 20MW, it now covers up to 50MW. The FiT employs a linear tariff system based on actual installed capacity, ensuring fair compensation. For example, hydro, solar PV, and wind projects within the 5MW to 50MW range have specific tariffs reflecting their costs. The FiT supports various technologies like biomass, geothermal, and small hydropower, encouraging diversification and enhancing energy security [32].

Quality of service standards are an integral part of ERA's regulatory framework. These standards ensure that electricity providers meet minimum service quality requirements, including reliability, safety, and customer service. ERA monitors and enforces these standards through performance indicators, audits, and mandatory reporting by electricity providers. By maintaining high levels of service delivery, ERA ensures that consumers receive reliable and high-quality electricity supply. The standards cover various aspects of service delivery, from response times to outages to customer interactions, ensuring a comprehensive approach to quality assurance.

ERA has developed minimum quality of service standards that became effective on 1st March 2015 [33]. These standards are applicable to all electricity distribution companies and cover three main areas: access to electricity supply, customer service, and reliability of electricity supply.

(a) *Access to Electricity Supply*

Standards on access to electricity supply specify the maximum timeframes within which new connections should be completed. For example, the period taken to connect a customer single-phase without requiring pole service (QOS1) [33] should not exceed 10 working days from the time of payment for a service connection. Similarly, connections involving one or more poles (QOS2) [33] and three-phase (LIGHT) connections (QOS3) [33] should be completed within 15 working days. For heavier three-phase connections requiring significant network upgrades (QOS4) [33], the timeframe extends to 30 working days. These standards ensure timely access to electricity for new customers, enhancing the overall efficiency and responsiveness of the distribution network.

(b) *Customer Service*

Customer service standards focus on the interactions between the electricity provider and the customers, ensuring prompt and effective responses to their needs. For instance, the number of times the meter was read in three consecutive months (QOS6) [33] must be 100% of meters read at least once in a quarter. The period taken to reconnect a customer after payment for reconnection (QOS7) [33] should be within 48 hours. Standards also mandate that 70% of calls should be answered within 30 seconds (QOS8) [33], and 100% response to emergency calls within 30 minutes (QOS9) [33]. Additionally, non-technical complaints or queries (QOS11) [33] should be resolved within 30 working days, and technical complaints (QOS10) [33] within 7 working days. These standards ensure that customer interactions are handled efficiently, contributing to higher satisfaction and trust in the electricity provider.

(c) *Reliability of Electricity Supply*

Reliability standards ensure that electricity supply is consistent and outages are managed effectively. Notice of planned outages (QOS14) [33] should be given 48 hours in advance, with 100% compliance. Planned outages should not exceed 9 hours for all voltage levels and network types (QOS15) [33], and notice of unplanned outages (QOS16) [33] should be provided within 2 hours of occurrence. Standards also include the System Average Interruption Duration Index (SAIDI) (QOS17) [33] and the System Average Interruption Frequency Index (SAIFI) (QOS18) [33], which are measured annually to ensure ongoing reliability and service quality. These metrics help in assessing the overall performance of the distribution network and in identifying areas for improvement.

3.1.3 Electricity Supply and Demand

Uganda's electricity sector has witnessed remarkable growth in recent years, marked by significant increases in generation capacity and ongoing efforts to expand access to electricity. However, challenges such as balancing supply and demand, ensuring grid stability, and addressing the diverse needs of a largely rural population remain. This section provides an overview of the current state of electricity supply and demand in Uganda, explores consumption patterns, and outlines future projections and policy measures aimed at achieving sustainable and reliable electricity for all Ugandans.

Overview of the Electricity Supply

The country's electricity generating capacity increased from about 320 megawatts (MW) in 2002 to over 1,346 MW at the beginning of 2023 [21]. The commissioning of the Karuma hydroelectric power plant, which is expected to add a further 600 MW, will further boost this capacity. Approximately 92% of Uganda's generating capacity is renewable, with large hydro accounting for about 80%, sugar cane bagasse-fired plants 8%, and solar PV plants 4.5% [21].

Uganda's energy supply is predominantly renewable, primarily sourced from hydropower. However, the country also employs other renewable sources, including solar and biomass. The reliance on hydropower has significant implications for energy security, particularly due to potential impacts of climate change on water resources [21].

Table 3.1 summarizes the forms of energy supply in Uganda and their respective shares.

Table 3.1: Uganda's Generation Share Summary [21]

Energy Source	**Share in Total Generation**
Large Hydropower	80%
Sugar Cane Bagasse	8%
Solar PV	4.5%
Small Hydropower	2%
Thermal (HFO)	4%
Other Renewables	1.5%

In addition to domestic generation, Uganda also engages in energy imports and exports to manage supply and demand. Uganda imports electricity from neighboring countries such as Kenya, particularly during times of peak demand or when domestic generation is insufficient. Conversely, Uganda exports surplus electricity to countries like Rwanda, Tanzania, and South Sudan. These cross-border energy exchanges help stabilize the grid and ensure a more reliable supply of electricity across the region [34].

Uganda's interconnectedness with regional power pools, such as the East African Power Pool (EAPP), facilitates these energy exchanges. The EAPP aims to enhance regional integration, optimize energy resources, and improve the reliability and affordability of electricity supply in member countries [21].

Several key projects and developments are set to further enhance Uganda's electricity supply.

Key Projects and Developments

1. **Karuma Hydropower Plant:** Expected to add 600 MW to the national grid, significantly boosting Uganda's generating capacity and reducing reliance on thermal generation [34].

2. **Isimba Hydropower Plant:** Commissioned in 2019, adding 183 MW to the grid and improving the stability of electricity supply [21].

3. **Solar PV Projects:** Expansion of solar PV capacity, including projects like the 10 MW Tororo Solar Plant and the planned 20 MW Kabulasoke Solar Power Station, enhancing the renewable energy mix [21].

4. **Thermal Power Plants:** Utilization of Heavy Fuel Oil (HFO) plants to provide additional capacity during peak demand periods or when hydro generation is low due to drought conditions [21].

These efforts reflect Uganda's commitment to diversifying its energy mix and improving the resilience and sustainability of its electricity supply system.

Demand and Consumption Patterns

Electricity demand in Uganda has been growing steadily, driven by various factors including economic growth, urbanization, and efforts to expand electricity access to underserved areas [21]. In 2023, the maximum electricity demand increased from 843 MW in 2022 to 987.8 MW, reflecting a growth rate of 17% [5]. This is presented in figure 3.1.

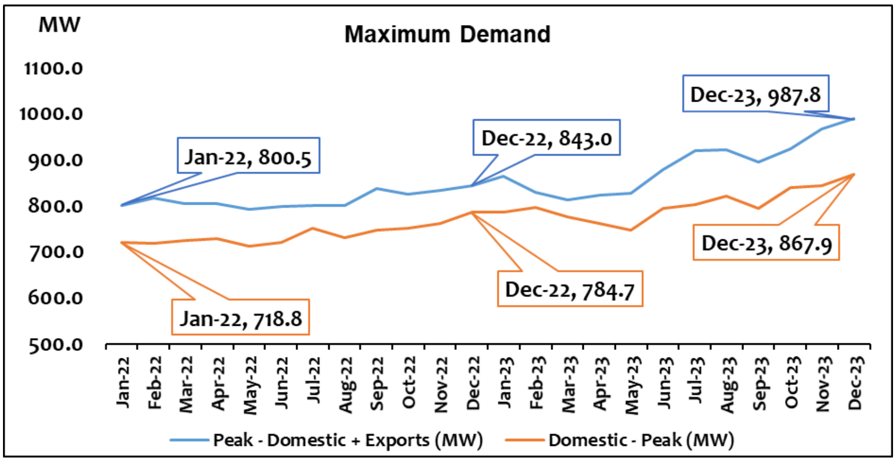

Figure 3.1: Yearly Maximum Demand in Uganda [5]

Figure 3.2, constructed from data provided by the World Bank [6], shows the rapidly growing access to electricity to Uganda's population.

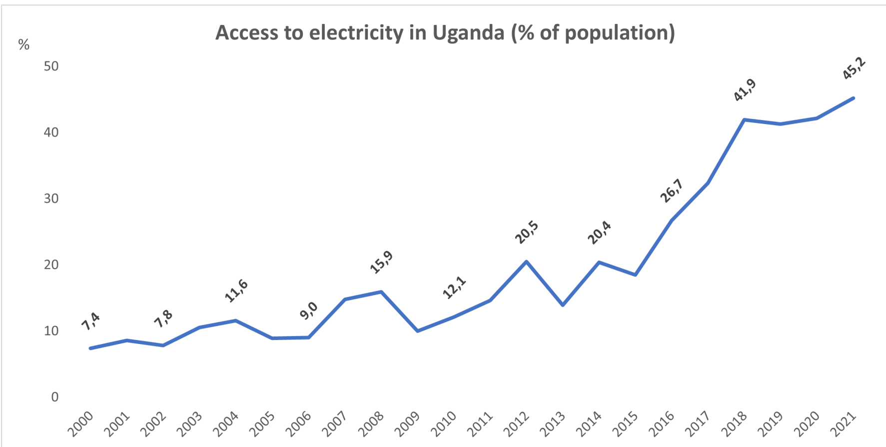

Figure 3.2: Access to Electricity in Uganda [6]

Key Drivers of Electricity Demand

1. **Economic Growth:** Uganda's GDP growth [35] has been a significant driver of increased electricity demand. As industries expand and new businesses emerge, the need for reliable electricity supply intensifies. The indus-

trial sector, in particular, has seen increased electricity consumption due to the establishment of manufacturing plants and other industrial facilities [21].

2. **Urbanization:** The rapid urbanization in Uganda has led to a surge in electricity demand in urban areas [36]. With more people moving to cities, the demand for residential electricity has risen, necessitating the expansion of urban electricity infrastructure. This urban growth has also spurred the development of commercial buildings, shopping centers, and office complexes, all of which require substantial electricity [21].

3. **Electrification Efforts:** The Ugandan government has been actively working to increase electricity access across the country [21]. Programs aimed at rural electrification have contributed to the rise in electricity consumption as more households and communities gain access to the national grid. Solar home systems and mini-grids have also played a role in extending electricity access to remote areas [37].

Consumption Patterns

1. **Residential Consumption:** Residential electricity consumption has been increasing [6], particularly in urban and peri-urban areas. The growth in household connections and the rising use of electrical appliances contribute to this trend. Programs to promote energy-efficient appliances are also being implemented to manage the growth in residential demand [21].

2. **Commercial and Industrial Consumption:** The commercial and industrial sectors are significant consumers of electricity in Uganda. Manufacturing industries, agro-processing plants, and mining operations are among the largest industrial users. The commercial sector, including shopping malls,

hotels, and office buildings, also contributes to substantial electricity consumption. Everything mentioned before drives energy consumption to sky-rocket in the upcoming years [38].

3. **Rural Consumption:** Despite efforts to increase rural electrification, rural areas still have lower electricity consumption compared to urban areas [39], 35.9% in rural areas in 2021 versus the 45.2% in general electricity access in the country. Many rural households rely on small-scale solar solutions for basic lighting and charging needs. However, as rural electrification programs expand, rural electricity consumption is expected to rise [21].

Challenges in Balancing Supply and Demand

1. **Low Power Demand:** During periods of low demand, particularly at night or during off-peak seasons, the electricity grid can become unstable. This instability is exacerbated by the high reliance on hydropower, which can fluctuate based on water availability.

2. **Grid Instability:** The scattered and sparse settlement patterns of Uganda's predominantly rural population [36] complicate electrification efforts. This makes it necessary to consider both on-grid and off-grid solutions to meet the demand efficiently.

3. **Investment in Infrastructure:** To meet the growing demand, substantial investment is needed in both generation and distribution infrastructure. This includes the development of new power plants like the projects that have been mentioned previously in 3.1.3, expansion of transmission lines, and upgrading of existing grid infrastructure.

Typical Demand Curves Typical demand curves in Uganda reflect variations in electricity consumption throughout the day. Peak demand usually occurs in the evening, between 6 PM and 10 PM, when residential usage is highest due to lighting, cooking, and other household activities. A secondary peak may occur in the early morning as industrial and commercial activities commence. The demand tends to drop significantly during the night and mid-day when residential and commercial activities are lower. These patterns are influenced by both the residential and commercial sectors' operating hours. There are no public real-time demand curves available as would happen in Spain with OMIE's website for example, however [7] presents a demand curve from the first of July 2014 extracted from the UETCL systems data, see figure 3.3.

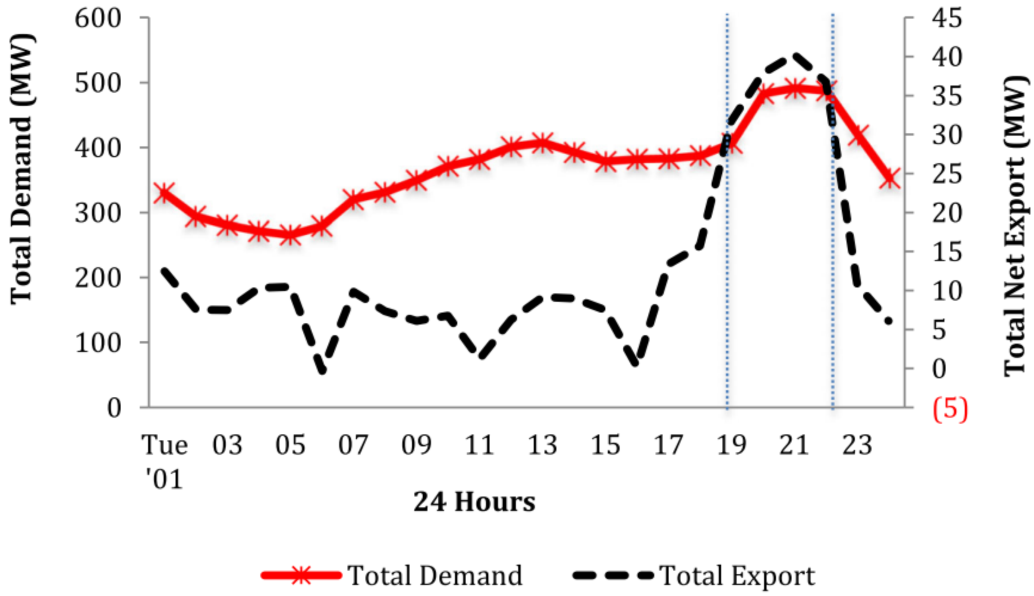

Figure 3.3: Load Profile of Total Demand and Net Export on 1st July 2014 [7]

3.1.4 Electricity Pricing and Tariffs

Uganda's electricity pricing and tariffs are crucial for ensuring the financial viability of the electricity sector while making electricity as affordable as possible for consumers. ERA is responsible for setting and periodically reviewing these tariffs to reflect the true cost of service delivery. Although this methodology has been presented in 2, there are a few extra considerations worthy of mention.

Tariff Structure The electricity tariff structure in Uganda is categorized based on different consumer segments, including domestic, commercial, medium industrial, large industrial, and street lighting. Each segment has specific tariffs designed to cover the costs of generation, transmission, and distribution, while also promoting efficient energy use [40].

1. **Domestic Consumers:** This category includes households. The tariff is typically structured with a lifeline rate for low-income consumers to ensure affordability for basic electricity needs. For example, the lifeline tariff is applicable for the first 15 kWh consumed in a month. After this threshold, the tariffs differ, up to USH 797.3 per kWh (around USD 2.13 as of June 20, 2024) [40].

2. **Commercial Consumers:** Small businesses and commercial establishments fall under this category. The tariffs are set to reflect the higher demand and usage patterns of these consumers. This tariff applies to three-phase voltage load not exceeding 100 amps. With the peak energy charge of USH 791.9 [40].

3. **Medium and Large Industrial Consumers:** These consumers have different tariff structures based on their substantial electricity usage. The tariffs for industrial consumers are designed to encourage efficient and high-volume

use of electricity, which is crucial for economic growth and industrial development. These customers are included in high voltage (11 kV or 33 kV), with maximum demand exceeding 500 kVA but up to 1,500 kVA with the highest energy charge in the peak block of USH 502.9 [40]. There is an extra segment for extra-large industries, for those that exceed the 1,500 kVA threshold [40].

4. **Street Lighting:** This category includes tariffs for public street lighting, which are managed by local governments and municipalities. The tariff pricing is fixed at USH 370.0 [40].

Tariff Adjustment Mechanism The ERA employs a comprehensive tariff adjustment mechanism to ensure that the tariffs remain fair and reflective of the actual costs incurred by the utility companies. This mechanism has already been mentioned in 2 and includes the Quarterly Tariff Adjustment (QTAM), the Annual Tariff Review, the Feed-in-Tariff (FiT) and the Performance Based Regulation .

3.2 UMEME's Overview and Responsibilities

UMEME is Uganda's leading electricity distribution company, playing a pivotal role in ensuring reliable electricity supply throughout the country. As the primary distributor, UMEME is responsible for the operation, maintenance, and expansion of the electricity distribution network [41].

UMEME has played a pivotal role in the nation's energy sector since the company's establishment. Operating under a 20-year concession that began in 2005, UMEME is responsible for distributing 97% of the electricity consumed in Uganda. This distribution network includes maintaining power lines, transformers, and substations, ensuring reliable electricity supply to residential, commercial, and industrial customers [42].

One of UMEME's primary responsibilities is to enhance the efficiency and reliability of electricity distribution. Over the years, the company has made significant investments to reduce technical and commercial energy losses, which have been lowered from 38% in 2005 to about 18% in recent years. These efforts have also seen the number of transformers increase and customer connections rise from 294,000 in 2005 to over 1.6 million by 2022 [42]. Additionally, UMEME has implemented advanced technologies like prepayment metering systems, commonly known as Yaka, to improve billing accuracy and revenue collection, covering 97% of their customer base [43].

The company also plays a vital role in customer service and engagement, striving to improve communication and address customer complaints efficiently. Despite these advancements, UMEME has faced challenges such as high power tariffs and accusations of service unreliability. These issues have sometimes led to protests and have been points of contention in discussions about extending the company's concession [43].

As UMEME's concession is set to expire in March 2025, the government has begun preparations for the Uganda Electricity Distribution Company Limited (UEDCL) to take over the distribution responsibilities. This transition aims to streamline operations by reducing third-party involvement in power distribution, potentially lowering tariffs for industrial users. The government is expected to hire over 3,000 workers to facilitate this transition, with priority given to the existing UMEME staff.

UMEME's legacy in Uganda's energy sector includes significant contributions to expanding the electricity distribution network and improving operational efficiencies, despite the challenges of high tariffs and service reliability issues [43] [42].

3.2.1 Challenges Faced by UMEME

UMEME Limited, faces a multitude of challenges that significantly impact its operations and the broader electricity sector. These challenges highlight the urgent need for technological interventions, such as the OneSite solution, designed to improve UMEME's control and efficiency over its meters and customer interactions. One of the most pressing issues is the high level of technical inefficiencies in the national power grid, which lead to substantial power losses during distribution. It is estimated that up to 20% of generated electricity is lost due to outdated infrastructure and transmission lines, which have not kept pace with the country's growing demand for power [34]. This inefficiency is partly due to an aging transmission infrastructure that is inadequate to support the needs of an increasing population. As Uganda continues to develop, the demand for electricity grows, yet the current infrastructure struggles to meet this demand efficiently, leading to frequent power outages and unreliable service [44] [45].

Additionally, there is a significant disparity in electricity access between urban and rural areas, posing a formidable challenge for UMEME. Urban centers like Kampala benefit from relatively stable power supplies, but many rural regions remain underserved, with limited or no access to electricity. Extending the power grid to these remote areas is costly and technically challenging, often resulting in a focus on more densely populated urban regions at the expense of rural electrification [44]. This disparity not only affects the quality of life in rural areas but also hinders economic development and efforts to achieve universal electrification. The logistical challenges of extending power lines over rugged and vast terrains contribute to the complexity and cost, making it a less attractive venture for investment [42].

Financial constraints further exacerbate these challenges. Despite significant investments in power generation, there has been insufficient funding for the expansion and maintenance of the transmission and distribution network. The high costs associated with extending the grid to remote areas result in delays and inadequate service coverage. Moreover, the financial framework within which UMEME operates is strained by high operational costs and the necessity to keep tariffs low for consumers, which limits the company's ability to invest in necessary infrastructure upgrades [45]. The electricity tariff structure in Uganda complicates the situation. High power tariffs, linked to UMEME's concession agreements, have sparked dissatisfaction among consumers and hindered the government's industrialization efforts [46]. This pricing structure, while attempting to recover costs, inadvertently affects the affordability and accessibility of electricity for many Ugandans, thus limiting economic opportunities and growth [46].

Policy inconsistencies and management inefficiencies also play a significant role in the challenges faced by UMEME. The power sector in Uganda has undergone numerous reforms and restructuring efforts, yet these have not resolved persistent issues. Bureaucratic red tape and corruption have sometimes hampered the effective deployment of resources and the completion of infrastructure projects, leading to delays and increased costs [45]. The complexities of navigating these bureaucratic challenges add another layer of difficulty for UMEME as it seeks to expand and improve its services [46].

Vandalism and electricity theft are additional issues that significantly impact UMEME's operations. The company incurs substantial costs in replacing vandalized lines and transformers, which not only affects revenue but also disrupts service delivery. Electricity theft, particularly in underserved areas, further exacerbates power losses and diminishes the company's financial resources [42]. The OneSite solution aims to address these multifaceted challenges by enhancing UMEME's ability to monitor and control their electricity meters and customer interactions.

By interconnecting meters through the IDIS (Interoperable Device Interface Specification) framework, UMEME can reduce technical losses, improve billing accuracy, and minimize electricity theft. This integration is crucial for modernizing Uganda's electricity distribution network, ensuring more reliable service, and contributing to the country's broader electrification goals. Implementing these solutions will be vital in transforming UMEME's operational capacity and ensuring that it can meet the growing demands of Uganda's energy sector effectively. The tests and simulations that will be conducted as part of this project are crucial in demonstrating the potential of the OneSite solution to overcome these operational challenges and support UMEME in providing more efficient and reliable electricity services.

3.3 OneSite Smart Metering Project

The Onesait Smart Metering project, led by Minsait, is a transformative initiative designed to modernize UMEME's electricity distribution and customer management systems. This project addresses UMEME's critical challenges, such as energy loss, inefficient billing, and limited customer engagement, through a comprehensive suite of advanced technologies and management solutions. The initiative is significant for UMEME, Uganda's primary electricity distribution company, as it seeks to enhance operational efficiency, reduce losses, and improve customer service. Developed by Minsait, the system is renowned for its advanced capabilities in smart metering, data management, and integration with existing utility systems. The Onesait Utilities Suite is a modular and integrated system that covers the entire Meter-to-Cash cycle, which is essential for UMEME's transition to a more efficient and automated grid system. The solution provides extensive functionalities that encompass customer care, billing, meter data management, and operational reporting. By implementing this system, UMEME aims to reduce energy losses, improve customer satisfaction, and streamline its operational processes, aligning with global best practices in the energy sector [47].

Built on a modular and scalable architecture, the system allows for seamless integration with existing utility systems. This flexibility is crucial for UMEME as it transitions from traditional metering systems to more advanced, automated solutions. The project is designed to support a wide range of functionalities, including meter data management, customer relationship management, billing, and collection processes. By providing a unified platform for these operations, the system enables UMEME to streamline its processes and enhance the accuracy and reliability of its services [47]. This integration is crucial for achieving real-time visibility and control over the electricity distribution network, enabling UMEME to respond quickly to issues and optimize resource allocation [47].

A significant technical advantage of the Onesait platform is its compatibility with multiple communication protocols, including DLMS/COSEM and IDIS, which ensures interoperability with a wide range of meter manufacturers and models [47]. This feature is essential for UMEME as it expands its smart metering infrastructure, allowing for flexible deployment and future scalability. The Onesait system supports bidirectional communication with smart meters and other field devices, facilitating real-time data exchange and analytics. This capability enables UMEME to monitor consumption patterns, detect anomalies, and prevent unauthorized usage [47].

One of the primary challenges faced by UMEME is the high level of electricity losses, often attributed to theft and technical inefficiencies. The Onesait platform addresses this issue by offering robust features for detecting and preventing electricity theft. The system's advanced analytics capabilities allow UMEME to monitor usage patterns and identify irregularities that may indicate unauthorized consumption. Additionally, the system supports real-time data collection and analysis, enabling UMEME to respond swiftly to potential issues and reduce non-technical losses [47].

Integrating smart meters through this platform is a critical component of UMEME's strategy to enhance customer service and engagement. Smart meters provide customers with detailed information about their energy consumption, enabling them to make informed decisions about their usage. This transparency not only empowers customers but also helps UMEME build trust and improve its relationship with consumers. Furthermore, the system's customer relationship management features enable UMEME to offer personalized services and more effectively address customer inquiries and concerns [47].

The deployment of the Onesait solution is expected to bring substantial operational benefits to UMEME. By automating routine tasks such as meter reading, billing, and collections, the system reduces the need for manual intervention, minimizing errors and improving efficiency. This automation extends to tasks such as meter reading, billing, and collections, which are critical for ensuring accurate and timely revenue generation. By automating these processes, UMEME can improve its operational efficiency and focus its resources on more strategic initiatives [page 13]. Furthermore, the system's robust analytics and reporting tools provide valuable insights into operational performance, enabling UMEME to make data-driven decisions that optimize efficiency and reduce costs. The ability to generate detailed reports and track key performance indicators is crucial for continuous improvement and strategic planning [47].

Another significant advantage of the system is its ability to support prepayment and post-payment models, catering to the diverse needs of UMEME's customer base. This flexibility is particularly important in regions where prepayment systems are prevalent, allowing customers to manage their electricity expenses more effectively. The system's prepayment capabilities include features for token generation and management, which are essential for UMEME's operations [47]. The Onesait project includes a sophisticated prepayment system that integrates seamlessly with UMEME's billing and revenue management processes. This system enables customers to purchase electricity in advance, helping them manage their consumption and expenses more effectively. The prepayment model is particularly beneficial in regions with limited financial resources, where customers prefer to control their spending on utilities [47]. By improving revenue collection and reducing delinquency rates, the prepayment system also strengthens UMEME's financial stability. The integration of token generation and management features ensures secure and efficient prepayment operations, supporting UMEME's revenue assurance strategies [47].

In addition to its operational and financial benefits, the Smart Metering project aligns with UMEME's commitment to sustainability and energy efficiency. The system's advanced data management capabilities allow UMEME to track and analyze energy usage patterns, identify inefficiencies, and implement measures to reduce waste. This focus on sustainability not only enhances UMEME's environmental stewardship but also positions the company as a leader in the transition towards greener energy solutions. The system's data management capabilities enable UMEME to track and analyze energy consumption patterns, identify areas for improvement, and implement measures to reduce waste and promote sustainable practices. This focus on sustainability is aligned with global trends toward greener energy solutions and positions UMEME as a forward-thinking utility provider [47].

The project's future readiness is further supported by its flexible and scalable design, which can accommodate emerging technologies and evolving market demands. This adaptability ensures that UMEME remains competitive and responsive to the needs of its stakeholders in a rapidly changing energy landscape. Overall, the smart metering project represents a significant advancement in UMEME's efforts to modernize its operations and improve service delivery. By leveraging the advanced capabilities of the system, UMEME can overcome its operational challenges, enhance customer satisfaction, and achieve its strategic objectives [47]. The successful implementation of this project is crucial for UMEME's transformation into a more efficient and customer-centric utility provider, capable of meeting the evolving needs of its stakeholders in a rapidly changing energy landscape [47].

Chapter 4

Prepayment Systems

Prepayment systems for electricity supply offer consumers the ability to pay for electricity before they use it, providing a flexible and manageable approach to energy consumption. These systems are especially beneficial in regions facing challenges such as high rates of energy theft and non-payment issues. By allowing consumers to control their electricity usage and payments, prepayment systems help improve cash flow for utilities and enhance overall customer satisfaction.

Prepayment systems also reduce the administrative burden on utilities, as they eliminate the need for meter reading and billing processes. These systems can significantly lower operational costs and improve revenue collection efficiency. Additionally, prepayment meters can provide real-time data on electricity usage, helping consumers to better manage their energy consumption and budget.

In Uganda, the adoption of prepayment systems has been a critical step towards modernizing the electricity sector and addressing the challenges of electricity theft and payment default. However, the current prepayment infrastructure faces several limitations, including the need for manual input and a lack of automation, which can lead to inefficiencies and potential scams.

Advanced technological solutions, such as the integration of DLMS/COSEM standards and IDIS certification, offer promising improvements to prepayment systems. These technologies ensure secure, reliable, and interoperable communication between meters and management systems, enhancing the overall functionality and user experience of prepayment systems.

4.1 Current Prepayment Systems in Uganda

Uganda's current prepayment system for electricity is primarily managed by UMEME. This system uses prepaid meters, commonly known as "Yaka meters," which require customers to purchase electricity tokens in advance. These tokens are then entered into the meters to access electricity.

The Yaka prepayment system allows consumers to buy electricity much like they would purchase airtime for their mobile phones. Upon connecting a household or business to the electricity grid, customers are provided with a unique prepaid meter number. To buy electricity, customers can purchase tokens from various vendors, including mobile money platforms, banks, and authorized retail outlets. They receive a 20-digit token which is entered into the meter to load the purchased kWh [48].

The current prepayment system offers several benefits. One of the primary advantages is the convenience it provides to customers, who can purchase electricity tokens at any time from a variety of locations, including using mobile money services. This system also allows consumers to monitor their electricity usage more closely and manage their spending effectively, giving them greater control over their electricity costs. Additionally, the prepayment model eliminates the need for monthly bills, reducing administrative burdens for both consumers and the utility company and minimizing the risk of billing errors [48].

Despite these benefits, the prepayment system also has significant limitations

[48]. The reliance on manual input requires customers to physically enter tokens into their meters, a process that can be cumbersome and prone to human error. This manual nature of the system can also expose it to potential scams and fraud, undermining the security of the transactions. Moreover, the existing Yaka meters require upgrades to meet new global standards, such as the Standard Transfer Specification (STS) for prepayment systems. The government has announced that all Yaka meters must be upgraded by November 2024 to comply with these standards [49].

UMEME has initiated an upgrade of the Yaka meters to enhance security and efficiency. This upgrade includes transitioning to the Standard Transfer Specification (STS)2 Token Identifier (TID) standards, which ensure better token recognition and interoperability between different system components. The upgrade process is being conducted on an area-by-area basis to minimize service interruptions, with customers receiving SMS notifications about the upgrade schedule and instructions [50].

The introduction of automated and advanced prepayment solutions is expected to address many of the current system's limitations. Technologies such as DLMS/-COSEM and IDIS standards can enhance the functionality, security, and interoperability of prepayment systems, paving the way for more efficient and user-friendly electricity management in Uganda.

4.2 Advanced Prepayment Solutions

The evolution of prepayment systems for electricity has introduced a range of advanced solutions designed to improve efficiency, security, and user convenience. These modern systems leverage technology to automate processes, enhance data management, and provide real-time control over electricity usage.

4.2.1 Automated Prepayment Systems

Automated prepayment systems eliminate the need for manual entry of tokens and allow for seamless transactions through various digital platforms. These systems can be integrated with mobile money services, enabling customers to purchase electricity credits using their smartphones. This convenience not only simplifies the process for consumers but also reduces the risk of errors associated with manual entry.

For instance, advanced prepayment solutions like Itron's Smart Pay system [51] offer real-time billing where customers are billed from their prepayment accounts as consumption occurs. This system includes features such as remote disconnection and reconnection based on account balances, which helps in maintaining revenue assurance for utilities while providing customers with greater control over their electricity usage.

4.2.2 Mobile and Digital Payment Solutions

Mobile and digital payment solutions are increasingly becoming popular in the prepayment landscape. These solutions allow users to top-up their electricity meters using mobile banking applications or through online platforms. The integration of these payment methods with prepayment systems enhances accessibility, especially in regions with high mobile penetration but limited banking infrastructure.

These digital platforms provide consumers with real-time feedback on their electricity consumption and remaining balance, allowing for better budgeting and energy management. Additionally, utilities benefit from reduced operational costs as the need for physical vending stations and manual meter readings is minimized.

4.2.3 Minsait's Onesait Prepayment Solution

Minsait, offers a comprehensive prepayment solution through its Onesait Utilities platform [52]. Onesait Utilities is designed to accelerate the digital transformation of utility companies, supporting customer engagement and meter-to-cash processes. This solution manages both prepaid and postpaid contracts, providing value-added services and powerful customer engagement tools.

The Onesait platform integrates advanced metering infrastructure with mobile and digital payment solutions, allowing customers to manage their electricity usage conveniently through mobile apps and online portals. This integration not only improves customer satisfaction but also increases the efficiency of the meter-to-cash cycle by automating processes such as reading, billing, and invoicing.

Furthermore, Onesait enhances the security of transactions and reduces operational costs by offering real-time monitoring and data analytics capabilities. These features enable utilities to detect and address issues promptly, improving overall service reliability and reducing the risk of fraud.

The Onesait solution connected meters have advanced data management capabilities, allowing them to read and process various critical parameters that enhance the functionality and efficiency of prepayment systems. These meters can record and display the unit price of electricity for each quarter, enabling consumers to understand their billing rates and manage their usage accordingly. Additionally, they offer an emergency credit feature, which provides a buffer for consumers who exhaust their prepaid energy. This emergency credit prevents disconnection until the next recharge, thus avoiding sudden power outages and giving consumers time to top up their credit.

The balance credit display shows the remaining prepaid credit available for consumption, helping consumers keep track of their energy usage and budget their expenses. The meters also ensure all recorded data is accurately time-stamped with the current date and time, which is crucial for monitoring and analysis. They record the date and time of the last recharge, providing transparency and accountability in transactions, and keep track of the last amount recharged, which helps both the consumer and the utility company monitor payment patterns.

The meters display the amount of credit consumed in the current month, aiding in monthly budgeting and usage analysis, and show the previous month's consumption, allowing consumers to compare their usage and adjust their habits accordingly. Additionally, the active tariff rate is indicated, helping consumers understand their current billing rate. The meters also handle additional parameters that enhance data granularity and system efficiency.

Having access to this comprehensive set of data helps the distribution company detect cases of fraud and possible energy theft. Irregularities in consumption patterns or discrepancies between recorded and actual usage can signal tampering or unauthorized connections. This data-driven approach enables utilities to take proactive measures to secure the grid and ensure accurate billing. For consumers, these features provide greater transparency and control over their electricity usage. By knowing their consumption patterns and having detailed billing information, customers can manage their energy use more effectively, avoid unexpected disconnections, and budget more accurately. For the utility company, this detailed data collection enhances operational efficiency, reduces the risk of revenue loss due to fraud, and improves overall service reliability.

It is exactly this solution that will be applied in Uganda through UMEME, thus the need for interoperability and the importance of standards such as DLSM/-COSEM and IDIS.

4.3 Role of DLMS/COSEM in Prepayment

DLMS/COSEM standards are critical in modernizing prepayment systems for electricity by ensuring interoperability, security, and efficiency. As explained in previous sections, these standards provide a structured approach for data exchange between smart meters and utility management systems, which is essential for accurate and secure communication. One of the key features of DLMS/COSEM in prepayment systems is interoperability. These standards facilitate seamless communication between different devices and systems, ensuring that various smart meters, irrespective of the manufacturer, can work together within the same infrastructure. This capability is vital for utilities that aim to integrate prepayment systems with existing and future technologies.

Moreover, DLMS/COSEM standards provide robust security mechanisms to protect data integrity and confidentiality. In prepayment systems, securing transaction data is crucial to prevent fraud and ensure accurate billing. The standard includes features like encryption and authentication to safeguard communication between meters and the utility's central system. By standardizing communication protocols, DLMS/COSEM also supports the scalability of prepayment systems. Utilities can expand their infrastructure without worrying about compatibility issues, enabling them to roll out prepayment solutions across large and diverse customer bases efficiently.

4.4 IDIS in Enhancing Prepayment Systems

The Interoperable Device Interface Specification certification plays a crucial role in enhancing prepayment systems by ensuring interoperability, reliability, and security across different devices and systems. This certification is built on the International Electrotechnical Commission (IEC) 62056 DLMS/COSEM standards, providing a robust framework for smart meter communication and data exchange.

IDIS-certified devices are designed to support various communication protocols [4], including G3-PLC (Power Line Communication), Ethernet, and GPRS. This flexibility ensures that prepayment systems can integrate with existing and future network infrastructures, providing utilities with a cost-effective and scalable solution [4]. For instance, IDIS 2-certified meters support IP-based networks, enabling deeper integration into diverse networks with devices and software from different manufacturers [4]. This level of interoperability is essential for creating cohesive and efficient prepayment systems that can adapt to the evolving needs of the utility sector.

One of the significant advantages of IDIS certification is the enhanced security it brings to prepayment systems. IDIS standards include robust security features that protect against tampering and unauthorized access [4], ensuring that transactions and customer data remain secure. This is crucial for maintaining consumer trust and safeguarding the financial interests of utilities. By adhering to high-security standards, IDIS-certified devices help prevent fraud and ensure the accuracy of prepayment transactions.

Moreover, IDIS certification facilitates the implementation of advanced metering infrastructure (AMI) solutions [4]. These solutions enable real-time data collection and analysis, providing utilities with valuable insights into electricity consumption patterns and network performance. For example, the integration of IDIS-certified meters in smart prepayment systems allows for the remote management of meters, including real-time monitoring, remote disconnection, and reconnection [4]. This capability enhances the efficiency and reliability of prepayment systems, reducing operational costs and improving customer service.

IDIS-certified solutions have been successfully implemented in various regions, demonstrating their effectiveness in enhancing prepayment systems. For instance, in South Africa [53], IDIS-certified prepayment systems have significantly reduced non-payment issues and improved revenue collection for utilities. These implementations highlight the practical benefits of combining IDIS certification with advanced prepayment solutions, paving the way for more efficient and secure electricity management.

Chapter 5

IDIS Use Cases Tests

This chapter presents the application of various IDIS use cases within smart metering systems. These use cases are designed to ensure that metering devices adhere to the stringent standards outlined in the IDIS Package 3 [54]. Each use case targets a specific function within the Advanced Metering Infrastructure (AMI), confirming that meters operate as intended and integrate seamlessly into the broader network.

The simulations discussed in this chapter are implemented to replicate real-world scenarios, enabling the evaluation of meter performance and reliability. Each section will detail the method used to simulate the specific use case and will provide an analysis of the test outcomes. While the coding details are critical to these tests, they are presented separately in [appendix A], allowing this chapter to focus on the conceptual framework and the results obtained from the simulations.

5.1 UC 1: Meter Registration

The first use case, Meter Registration, is essential for the operation of a smart metering system. This process ensures that a newly installed meter is correctly integrated into the AMI network, enabling communication with the Head-End

System (HES). During meter registration, the meter identifies itself to the HES, establishing its presence within the network by transmitting key information, including its unique device ID, IP address, and system title.

In the simulation of this use case, the registration process is replicated to evaluate how effectively the meter integrates into the AMI network. The simulation begins with the initialization of critical parameters that the meter typically uses during registration. The meter then sends a registration message to the HES, containing all necessary details. Upon receipt, the HES processes the registration request, verifies the details, and sends an acknowledgment back to the meter, confirming its successful registration [appendix A].

The expected outcome of this test is the successful recognition of the meter by the HES, followed by a confirmation message indicating that the registration has been completed. If the registration process fails due to missing or incorrect information, the HES would reject the registration, and this failure would be documented. This use case is foundational, as it ensures that the meter is properly integrated into the network and prepared to perform subsequent operations.

5.2 UC 2: Remote Tariff Programming

The Remote Tariff Programming use case addresses the ability of the Head-End System to remotely update the tariff structure within a meter [54]. This functionality is critical in dynamic pricing environments, where electricity rates may change based on time-of-use, demand, or other factors. The ability to remotely adjust tariffs ensures that the metering system can accurately reflect the current pricing model without requiring physical intervention at the meter site.

In this use case, the meter is first initialized with a predefined consumer category and associated tariff rates. These rates are representative of different consumer categories, such as Domestic, Commercial, or Industrial, each with its respective pricing structure. The meter is then tested to ensure it can receive and correctly implement a new tariff sent by the HES.

The simulation [appendix A] begins by transmitting the current tariff information from the meter to the HES. The HES then processes this information and sends back a new tariff structure, which the meter is expected to apply. This involves verifying that the meter can handle different types of tariff updates, including changes to the base rate and, in the case of domestic consumers, adjustments to the lifeline rate.

The expected outcome for this test is that the meter accurately updates its internal tariff configuration based on the instructions received from the HES. The meter should reflect the new rates immediately or within a short delay, depending on the system's design. If the meter fails to update its tariff or applies the incorrect rate, the test would be marked as a failure, indicating an issue with the remote programming functionality.

This use case is essential for maintaining flexibility and accuracy in billing within a smart metering system, allowing utilities to adjust pricing in response to market conditions or regulatory requirements. The successful execution of this test confirms that the meter can adapt to changes in tariff structures as directed by the HES, ensuring consistent and accurate billing for the end consumer.

5.3 UC 3: On-Demand Reading

This use case focuses on the ability of the Head-End System to request and receive real-time data from the meter at any given moment. This capability is crucial for providing accurate and timely information about electricity consumption, which can be used for various purposes, including billing, monitoring energy usage patterns, and detecting potential issues with the metering system [54].

The meter is initialized with several key energy consumption parameters, including active and reactive energy import and export values [54]. These parameters represent the total energy consumed and generated by the household over a certain period. The simulation involves the HES sending a request to the meter for an immediate reading of these values.

Upon receiving the request from the HES, the meter responds by transmitting the current values of its energy registers [appendix A]. This data includes the amount of active energy imported and exported, as well as the reactive energy. The accuracy and timeliness of this response are critical, as they ensure that the HES receives up-to-date information that can be used for immediate analysis or billing purposes.

The expected outcome of this test is that the meter successfully retrieves and sends the requested data without any significant delay. The values returned should match the internal state of the meter's registers at the time of the request. Any discrepancies between the requested data and the returned values, or any delay in response, would indicate a failure in the on-demand reading functionality.

5.4 UC 4: Periodic Reading

The Periodic Reading use case ensures that the meter can automatically send energy consumption data to the Head-End System at regular intervals. This functionality is essential for maintaining a consistent flow of information from the meter to the HES, which is crucial for accurate billing, energy management, and detecting irregular consumption patterns [54].

In this scenario, the meter is initialized with cumulative and rate-specific energy registers [54]. These registers track the total energy consumed over time and the energy consumed during specific periods (such as different tariff periods). The simulation involves setting the meter to send these readings to the HES at predetermined intervals.

During the test, the meter periodically transmits its cumulative and rate-specific readings to the HES [appendix A]. The HES processes these readings, storing them for future use in billing and analysis. The accuracy of these readings and the regularity of their transmission are key factors in determining the success of this use case.

The expected outcome is that the meter sends accurate and timely readings at the specified intervals, without any missed transmissions or errors in the data. A failure in this test would be indicated by irregular transmission intervals, incorrect readings, or missed data packets, which could lead to inaccurate billing and compromised energy management.

This use case is critical for ensuring that the meter consistently provides the HES with the data needed to monitor and manage energy consumption effectively.

5.5 UC 5: Connection and Disconnection

The Connection/Disconnection test pertains to the ability of the HES to remotely control the connection status of the meter, effectively allowing or disallowing the flow of electricity to the premises [54]. This capability is essential for managing situations such as non-payment, safety concerns, or demand management.

In this use case, the meter is initialized with a connection state, which can be either connected, disconnected, or ready for reconnection [54]. Additionally, the meter is configured to monitor the instant current to assess whether it exceeds a predefined threshold, which could trigger a disconnection event.

The simulation [appendix A] involves the HES sending a command to either connect or disconnect the meter based on the current state or the instant current reading. The meter must then execute this command and update its internal state accordingly. The test also includes an automatic disconnection feature that triggers if the instant current exceeds the predefined safety threshold.

The expected outcome is that the meter accurately responds to the connection or disconnection commands from the HES and autonomously disconnects if the current exceeds the safe limit. Any failure to change the connection state as commanded, or failure to disconnect when necessary, would indicate a failure in this functionality.

This test is crucial for ensuring that the utility has control over the delivery of electricity to the consumer, particularly in scenarios that require immediate action for safety or financial reasons.

5.6 UC 6: Clock Synchronization

Ensuring that the meter's internal clock is synchronized with the HES is vital for the accurate functioning of various time-dependent processes, such as time-of-use billing and event logging [54]. Even minor discrepancies in the meter's clock can lead to significant issues, such as incorrect billing or difficulty in tracking energy usage over time.

For this use case, the simulation starts with the meter initialized to a local time that may not be perfectly aligned with the actual time. The HES then sends a command to the meter, instructing it to adjust its internal clock to match the correct time provided by the system. The meter must then update its clock accordingly [appendix A].

The primary goal here is to observe whether the meter accurately adjusts its internal clock without delay, aligning itself with the time sent by the HES. Any failure to do so would be problematic, as it could result in errors in time-sensitive operations. The success of this use case is marked by the meter's ability to eliminate any time deviation and to operate in perfect synchronization with the HES.

This functionality is foundational for the accurate execution of all operations within the meter that rely on precise timing.

5.7 UC 7: Quality of Supply Reporting

Monitoring the quality of the electricity supply is another critical function of smart meters. The Quality of Supply use case is concerned with the meter's ability to track and report on key electrical parameters, such as voltage levels, current, and the occurrence of power disturbances like sags, swells, and outages [54]. These parameters are essential for assessing the reliability and stability of the electricity supply.

In the simulation [appendix A] of this use case, the meter is set up to monitor several quality indicators, including instantaneous voltage and current, total active power, and the frequency of voltage sags, swells, and power failures [54]. The meter continuously tracks these parameters and logs any events that indicate a deviation from normal supply conditions.

The meter's ability to detect and report abnormalities in the power supply, such as a sudden drop in voltage or a power outage, is the focus of this test. These events are crucial for diagnosing potential problems in the supply network. The expected result is that the meter will accurately identify and record any disruptions or irregularities in the power supply, promptly reporting these to the HES. Failures in this area would manifest as missed events or incorrect data, which could undermine the reliability of the power system.

This use case is vital for maintaining the integrity of the electrical supply and ensuring that any issues are detected and addressed in a timely manner.

5.8 UC 8: Load Management by Relay

Load management is a very important aspect of modern energy systems, where the HES can remotely control the load on the network through the use of relays [54]. The Load Management by Relay use case involves the meter's ability to respond to commands from the HES to either connect or disconnect the load based on current demand or predefined conditions. This functionality is particularly important for managing peak loads, ensuring grid stability, and preventing overloads.

In this scenario, the meter is pre-configured with a relay status that determines whether the connected load is active or inactive. The meter also monitors the active demand import, a key parameter in deciding whether a load should remain connected or be disconnected [54]. The HES can send commands to change the relay status, thereby controlling the flow of electricity to the consumer's premises.

The simulation [appendix A] involves testing the meter's responsiveness to these commands and its ability to manage loads effectively. For instance, if the active demand exceeds a certain threshold, the meter should disconnect the load to prevent overload. Conversely, the HES may issue a command to reconnect the load when the demand is within acceptable limits.

The success of this use case is determined by the meter's ability to correctly interpret and act on the HES commands, ensuring that the load is managed according to the specified parameters. Any failure to execute these commands accurately, such as not disconnecting an overloaded circuit or failing to reconnect when safe, would indicate a problem in the load management functionality.

5.9 UC 9: Firmware Update

The firmware upgrade use case is centered on the ability of the meter to receive and apply updates to its firmware remotely [54]. Firmware upgrades are essential for maintaining the meter's functionality, introducing new features, fixing bugs, and enhancing security. The ability to perform these upgrades remotely, without the need for physical access to the meter, is a significant advantage in smart metering systems.

For this use case, the meter starts with an initial firmware version, and the HES initiates the upgrade process by sending the new firmware package to the meter. The simulation tests the meter's capacity to receive, validate, and apply this new firmware version without disruption to its ongoing operations [appendix A].

During the upgrade process, the meter must ensure that the new firmware is correctly received and that it is compatible with the meter's hardware and software environment [54]. The upgrade should be applied in such a way that minimizes downtime, ensuring that the meter remains operational or returns to operation quickly after the upgrade.

The expected outcome of this use case is the successful installation of the new firmware, with the meter restarting and continuing normal operations under the updated software. A failure in this test could manifest as an inability to apply the firmware, corruption of the firmware data, or the meter becoming unresponsive post-upgrade.

5.10 UC 10: Meter Supervision

Meter Supervision is a use case that involves monitoring the meter's internal and external conditions to ensure its reliable operation. This use case encompasses a wide range of functions, including the detection of tampering, tracking power outages, identifying error codes, and monitoring unexpected resets [54]. The ability to supervise these aspects is essential for maintaining the integrity and security of the metering system.

In this use case, the meter is equipped with various supervision parameters that allow it to detect and respond to potential issues [54]. For instance, tampering detection is crucial for preventing unauthorized access or manipulation of the meter, which could lead to inaccurate readings or loss of revenue. Similarly, tracking power outages and resets ensures that the meter can log any disruptions in service and help diagnose potential problems.

The simulation [appendix A] involves testing the meter's ability to accurately detect and report these conditions to the Head-End System (HES). For example, if tampering is detected, the meter should immediately log the event and notify the HES. Likewise, if an unexpected reset occurs, the meter should record this event and transmit the information to the HES for further analysis.

Success in this use case is determined by the meter's ability to consistently and accurately monitor and report all relevant supervision parameters. Any failure to detect or correctly log these events would be a significant issue, potentially compromising the meter's security and reliability.

5.11 UC 11: Consumer Information

The Consumer Information use case focuses on the meter's ability to store and manage data related to the end consumer. This includes essential details such as the consumer's name, address, account number, and tariff category [54]. Additionally, the meter may store information on the consumer's energy usage patterns, billing cycles, and energy efficiency ratings. The accurate storage and retrieval of this information are critical for effective customer management and billing.

In this scenario, the meter is initialized with various consumer-related data, simulating a typical setup where the meter is linked to a specific customer. The simulation tests the meter's capacity to store, manage, and retrieve this data as needed by the HES or during interactions with the consumer [54].

During the test, the HES may request specific consumer information, and the meter must respond by accurately providing the requested data [appendix A]. The integrity of this data is crucial, as it directly affects billing accuracy, customer service, and the overall customer experience.

The expected outcome for this use case is that the meter successfully manages and transmits accurate consumer information upon request. Any discrepancies in the data, or the inability to retrieve or update consumer information, would indicate a failure in this functionality.

5.12 UC 12: Communication Supervision

Communication Supervision is a key use case that focuses on monitoring the communication link between the meter and the Head-End System [54]. Reliable communication is fundamental to the operation of a smart metering system, as it enables the transmission of consumption data, remote commands, and firmware updates. Any disruptions or errors in communication can significantly impact the system's overall performance.

In this use case, the meter is equipped with parameters that monitor the strength of the communication signal, the rate of data errors, and the uptime of the connection [54]. These parameters allow the meter to assess the quality of its communication link and to report any issues to the HES.

The simulation [appendix A] involves the meter continuously monitoring its communication status and logging any events that indicate a problem, such as a weak signal or a high data error rate. The meter must also track the duration of its connection uptime, ensuring that it remains connected to the HES as required.

Success in this use case is determined by the meter's ability to detect and report any communication issues promptly. The meter should provide accurate and timely updates on its communication status, enabling the HES to take corrective action if necessary. A failure in this test would be indicated by missed communication events, incorrect status reporting, or an inability to maintain a stable connection.

This use case is crucial for ensuring the reliability of the communication link between the meter and the HES, which underpins the entire smart metering system.

5.13 UC 13: Outage Supervision

Outage Supervision deals with the meter's ability to detect and report power outages to the HES. In a smart metering environment, timely and accurate reporting of outages is crucial for both the utility and the consumer [54]. It helps in quickly identifying and addressing issues in the power distribution network, minimizing downtime, and improving overall service reliability.

In this use case, the meter is configured to detect power outages and store relevant data, such as the time and duration of the outage [54]. Additionally, the meter is set to send notifications to a predefined destination, typically the HES, whenever an outage is detected. The simulation [appendix A] tests the meter's capability to accurately log outage events and promptly communicate these events to the HES.

The meter's ability to send notifications immediately upon detecting an outage is essential for the success of this use case. The expected outcome is that the meter will reliably detect any loss of power, log the event, and notify the HES without delay. Failure to detect outages, or delays in notification, would indicate a critical issue in the meter's outage supervision functionality.

5.14 UC 14: Remote Parameter Configuration

Remote Parameter Configuration allows utilities to update or modify various settings on the meter remotely, without the need for physical access to the device. This capability is vital for adapting to changing operational requirements, regulatory updates, or customer-specific needs. Parameters that can be configured remotely include display settings, alarm filters, and consumer messages [54].

In this scenario, the meter is pre-loaded with default configuration settings, including display modes, data formats, and alarm filters [54]. The Head-End System can then send commands to update these settings based on new requirements. The simulation assesses the meter's ability to receive, apply, and confirm these remote configuration changes.

During the test, the HES sends new configuration commands to the meter, which must then update its internal settings accordingly. The meter is expected to acknowledge the changes and apply them without error, ensuring that the new parameters are in effect immediately [appendix A].

The expected outcome is that the meter will successfully receive and apply the new configuration settings as instructed by the HES. Any failure to correctly update these settings, or to confirm the changes, would highlight a problem in the remote parameter configuration functionality.

5.15 UC 15: Warning Message Management

Warning Message Management involves the meter's ability to generate and send warning messages to the Head-End System or directly to the consumer when certain thresholds or conditions are met [54]. These warnings are critical for alerting the utility or consumer to potential issues, such as excessive load, which could lead to circuit overloads or other problems.

In this use case, the meter is set up with predefined thresholds, such as a maximum allowable load limit [54]. When the current load exceeds this limit, the meter generates a warning message that is sent to the specified destination, which could be the utility's HES or directly to the consumer.

The simulation [appendix A] evaluates the meter's capability to detect when these thresholds are exceeded and to promptly send an accurate warning message. The meter must also correctly identify the destination for the warning, whether it be the utility or the consumer, and ensure that the message is delivered without delay.

The success of this use case is determined by the meter's ability to consistently monitor the relevant parameters and generate timely warnings when necessary. Any failure to detect threshold breaches, generate warnings, or deliver messages to the correct destination would indicate a significant issue with the warning message management functionality.

5.16 Test Results and Analysis

The simulation of the IDIS use case tests [54] is conducted using a series of MATLAB scripts [appendix A] designed to mimic the interaction between a smart meter and a Head-End System. To ensure proper communication between the meter (client) and the HES (server), it is necessary to run the scripts in separate MATLAB sessions. This separation allows for the effective use of TCP/IP communication, enabling real-time data exchange between the simulated meter and the HES. The scripts are structured to automatically execute each use case in sequence, testing various functionalities such as meter registration, remote tariff programming, and quality of supply monitoring.

Each meter undergoes a series of tests where it is assessed on its ability to perform specific functions defined by the IDIS use cases. The results of these tests are then plotted, providing a visual representation of the meter's performance, where each use case is marked as either passed or failed. The purpose of these simulations is to validate the meter's compliance with the required standards and to identify any areas where the meter's performance may need improvement.

5.16.1 Meter 1 Simulation

The first test was conducted on a meter that failed to pass two critical use cases: Remote Tariff Programming (UC2) and Connection/Disconnection (UC5). As illustrated in the results below (figure 5.1), this meter successfully completed the majority of the tests, demonstrating its capability in areas such as meter registration, on-demand reading, and firmware upgrades. However, its failure in the remote tariff programming and connection/disconnection use cases indicates significant issues in these functionalities.

The failure in UC2, Remote Tariff Programming, suggests that the meter was

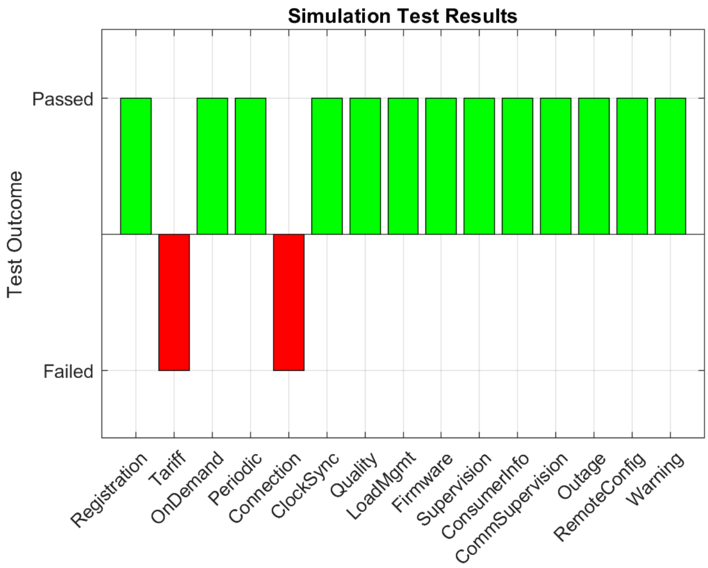

Figure 5.1: Meter 1 - Tests Results

unable to properly receive or implement the new tariff structure sent by the HES. This could be due to a variety of reasons, such as issues with the communication link, problems in handling the tariff data, or bugs in the firmware that prevent the correct application of the new rates.

The failure in UC5, Connection/Disconnection, is particularly concerning, as this function is critical for the safe operation and management of the electrical supply to the consumer. The inability to reliably connect or disconnect the meter could lead to situations where the meter remains in an incorrect state, potentially causing overloading, safety hazards, or issues with energy management.

These failures highlight the importance of robust testing and the need for further investigation into the meter's design and firmware to address these deficiencies. The successful execution of the remaining use cases suggests that while the meter has strong foundational capabilities, there are specific areas that require significant attention to ensure compliance with the full range of IDIS requirements. Simulation outputs are included in appendix A.

5.16.2 Meter 2 Simulation

The second test was conducted on a meter that demonstrated significant issues in three specific use cases: On-Demand Reading (UC3), Consumer Information (UC11), and Communication Supervision (UC12). While this meter passed the majority of the use cases, including key functionalities such as registration, periodic readings, and firmware upgrades, its failures highlight some critical weaknesses that could hinder its performance in practical applications. Results are showcased in figure 5.2.

The failure in UC3, On-Demand Reading, points to problems in the meter's ability to accurately and reliably respond to real-time data requests from the Head-End System. This function is vital for scenarios that require immediate access to consumption data, such as troubleshooting, real-time billing adjustments, or monitoring. A failure in this area suggests that the meter might not be able to provide instantaneous consumption readings or that there could be delays in responding to the HES request, compromising the utility's ability to manage energy usage effectively.

The failure in UC11, Consumer Information, indicates a significant issue with the meter's handling and management of consumer data. This use case is important for ensuring that all relevant consumer details, such as account numbers, addresses, and billing information, are correctly stored and retrieved. A failure here suggests potential inaccuracies in consumer data storage or transmission, which could lead to billing errors or difficulties in customer service interactions. Such a failure could also expose the utility to compliance risks, as accurate consumer data management is often subject to regulatory scrutiny.

The failure in UC12, Communication Supervision, is particularly concerning. This use case focuses on the meter's ability to monitor and maintain the quality of its communication link with the HES. Reliable communication is essential for the entire smart metering infrastructure, as it enables the flow of consumption data, remote commands, and firmware updates. A failure in communication supervision could result in unreported connection drops, high data error rates, or an overall unstable connection, making it difficult for the utility to ensure consistent and accurate operation of the meter.

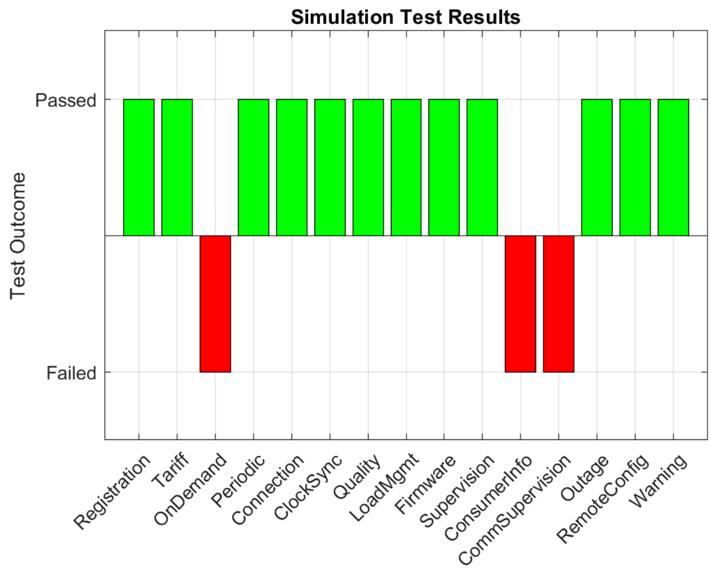

Figure 5.2: Meter 2 - Tests Results

The overall performance of this meter suggests that while it excels in many core functionalities, it suffers from serious deficiencies in communication and data management. These are critical areas that must be addressed to ensure the meter's full compliance with IDIS standards and its effective operation in real-world scenarios. Simulation outputs are also included in appendix A.

5.16.3 Meter 3 Simulation

The third and final meter tested stands out as it successfully passed all the IDIS use case tests [54]. This meter demonstrated full compliance with the IDIS standards, showing that it is fully capable of operating within the specified Advanced Metering Infrastructure (AMI) environment. The successful execution of all the use cases highlights this meter's robustness and readiness for practical deployment.

By passing all tests, the meter shows that it can be effectively managed remotely, provide accurate billing and consumption data, and handle firmware upgrades, all while ensuring proper communication and security measures are in place. The ability to monitor and report outages, manage load via relay, and generate warning messages further demonstrates its compliance with operational and safety standards. The results are shown in figure 5.3.

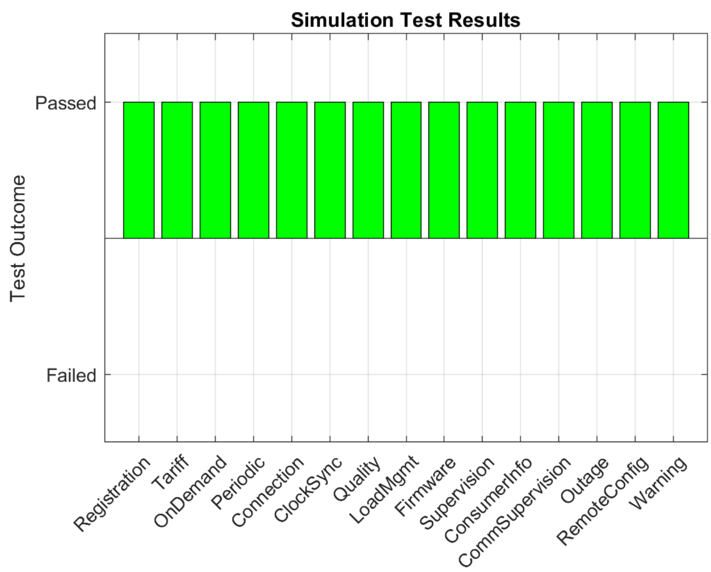

Figure 5.3: Meter 3 - Tests Results

This full compliance makes the meter ready to use in the UMEME project. Given that it meets the rigorous specifications of IDIS, it can be confidently deployed in UMEME's network, ensuring efficiency, reliability, and customer satisfaction. The simulation outputs are properly included in appendix A.

Chapter 6

Cybersecurity in Smart Metering

This chapter critically examines the cybersecurity landscape in the context of smart metering systems, with a special focus on UMEME's current measures and the implications for the energy sector in Uganda. Given the widespread adoption of smart grid technologies, it is crucial to understand and enhance the cybersecurity frameworks protecting these systems. This discussion evaluates existing vulnerabilities and proposes actionable enhancements to strengthen security, align with international standards, and mitigate potential cyber threats.

6.1 Current State of Cybersecurity in Utility Data Transfer

UMEME's smart metering infrastructure, akin to many in developing regions, operates with notably minimal cybersecurity protections, heavily reliant on DLMS/COSEM protocols. These foundational smart meter communications protocols present significant security vulnerabilities susceptible to various cyber threats. A primary concern is the inherent weaknesses within DLMS/COSEM, such as insufficient encryption measures and authentication mechanisms. This lack of robust security can lead to unauthorized data interception and manipulation, exposing consumer data and potentially disrupting utility operations [1].

The physical security of the meters, coupled with the architecture of the communication networks they use, further compounds these security challenges. Smart meters are often installed in accessible locations, rendering them vulnerable to physical tampering. Moreover, the communication uplinks—critical for transmitting data to utility providers—lack robust encryption or effective authentication mechanisms, leaving the entire system susceptible to cyber-attacks. This vulnerability could lead to scenarios where malicious entities intercept or manipulate data, impacting service reliability and customer trust [3].

Additionally, the security implementations within UMEME's metering systems do not sufficiently leverage advanced security features that could mitigate these risks. The basic security measures currently in place fail to protect against sophisticated cyber threats, such as session hijacking and data spoofing. Such vulnerabilities expose users to significant privacy breaches and provide an entry point for further attacks on the network, potentially leading to widespread service disruptions and financial fraud [55].

Communication between smart meters and utility providers, essential for real-time data exchange necessary for dynamic power adjustments and billing, emerges as a potential failure point under cyber threat. This vulnerability is underscored by the prevalent use of insecure firmware and software configurations that do not conform to international security standards, thereby increasing the risk of unauthorized access and data breaches [56].

Another risk scenario involves the potential exploitation of the metering infrastructure for Distributed Denial of Service (DDoS) attacks. Attackers could harness the connectivity of smart meters to launch DDoS attacks aimed at overloading the energy grid's network, causing widespread power outages. This type of attack not only disrupts the electricity supply but can also lead to significant economic losses and compromise the safety of citizens [1].

Furthermore, the implementation of DLMS/COSEM often lacks measures to secure firmware updates, leaving the system vulnerable to firmware tampering. Attackers could modify firmware to inject malicious code, allowing them continuous and undetected access to the network. Such breaches could facilitate the manipulation of meter readings and grid operations, further emphasizing the need for secure update mechanisms and stringent access controls [56].

Given these vulnerabilities, there is an urgent need for a comprehensive overhaul of cybersecurity protocols within UMEME's smart metering systems. Addressing these issues is imperative to protect against the current spectrum of threats and to prepare for future challenges as cyber threat landscapes continue to evolve. The existing security measures are clearly insufficient and require some enhancements to safeguard the integrity and reliability of the utility's operations and to protect consumer data.

6.2 Proposed Security Enhancements

To enhance the cybersecurity framework of UMEME's smart metering infrastructure effectively, a targeted approach that incorporates modern security protocols and regional capabilities is essential. This strategy involves adopting a multi-layered security framework that integrates advanced encryption, authentication, and continuous monitoring solutions.

Firstly, it is imperative to implement robust encryption protocols across all communication channels. Currently, DLMS/COSEM protocols are widely used but suffer from vulnerabilities, including weak authentication and encryption methods susceptible to downgrade attacks [57]. Upgrading these systems to use Advanced

Encryption Standard (AES) with Galois/Counter Mode (GCM) provides authenticated encryption that protects data integrity and confidentiality. This encryption standard is recommended for its efficiency and strong security guarantees, which can significantly reduce the risk of data interception and manipulation [57].

Authentication is another critical component. Implementing High-Level Security (HLS) protocols for mutual authentication between smart meters and utility providers ensures that only authorized entities can access the network [55]. This approach involves a challenge-response mechanism, enhancing protection against unauthorized access. Incorporating Public Key Infrastructure (PKI) allows for secure key exchange and management, which is essential for maintaining secure communications [57]. By ensuring that both client and server are authenticated, the risk of man-in-the-middle attacks is substantially reduced.

Additionally, adopting a comprehensive key management strategy is crucial. Utilizing asymmetric cryptographic methods like Elliptic Curve Diffie-Hellman (ECDH) for key exchange offers a robust mechanism for establishing secure communications channels [3]. This strategy mitigates risks associated with symmetric key distribution, where compromise of a single key can lead to widespread security breaches. Regular key rotation and management practices are recommended to maintain system integrity and security over time [3].

Moreover, securing firmware updates is essential to prevent exploitation through outdated or vulnerable software. All firmware updates should be digitally signed and verified using asymmetric cryptography before installation, ensuring that only authentic and approved updates are applied to devices [56]. This process is crucial for maintaining system security and preventing unauthorized modifications to device firmware, which could otherwise lead to potential exploitation.

Network segmentation further enhances security by isolating different parts of the infrastructure, thus limiting the impact of a breach. Implementing Virtual

Private Networks (VPNs) for communication between metering devices and the utility's head-end system provides an additional layer of security by encrypting data in transit [1]. Additionally, deploying firewalls and intrusion detection systems (IDS) can help monitor and prevent unauthorized access attempts, ensuring that any suspicious activity is promptly identified and addressed [1].

Incorporating tamper detection mechanisms in smart meters is another effective security measure. These systems can send alerts if physical tampering is detected, allowing for immediate response to potential security threats [3]. This capability is vital for maintaining the physical security of meters and ensuring that any unauthorized attempts to access or modify the device are quickly mitigated.

Implementing Role-Based Access Control (RBAC) ensures that access to the network is restricted based on the roles and responsibilities of users [56]. By limiting access rights and privileges, the risk of insider threats and unauthorized access is minimized. This approach should be complemented by regular security audits and training programs for personnel, ensuring that all users are aware of best practices and potential threats.

For UMEME, deploying the Interoperable Device Interface Specification (IDIS) can enhance the overall security framework. IDIS provides a standardized approach to implementing advanced security features, including encryption, authentication, and key management [55]. By aligning with international standards, UMEME can ensure its systems are resilient against evolving cyber threats, thereby enhancing operational reliability and consumer trust.

The implementation of these security measures requires a phased approach. Initial steps should include a comprehensive assessment of the current infrastructure to identify vulnerabilities and areas for improvement. Engaging with technology partners and experts in cybersecurity can provide valuable insights into best practices and innovative solutions. Pilot testing in a controlled environment can help identify potential issues and refine the security strategy before full deployment [57].

Continuous monitoring and updating of security measures are critical to adapting to new threats as they emerge. Establishing a robust incident response plan ensures that UMEME can respond swiftly to security breaches, minimizing their impact and restoring normal operations [1]. This proactive approach not only protects the integrity of UMEME's smart metering system but also aligns with global standards, thereby reinforcing consumer confidence in the utility's commitment to security and reliability [1].

In conclusion, by adopting a multi-layered security strategy that incorporates advanced encryption, robust authentication, secure key management, and continuous monitoring, UMEME can significantly enhance the cybersecurity posture of its smart metering system. This approach ensures that UMEME is well-prepared to face current and future cyber threats, safeguarding its operations and maintaining the trust of its consumers and stakeholders.

6.3 Implementation Strategy

Enhancing the cybersecurity framework of UMEME's smart metering infrastructure requires a detailed and systematic approach. This plan outlines a comprehensive strategy to integrate modern security protocols tailored to the specific needs of UMEME and the Onesait project.

6.3.1 Step 1: Conduct a Technical Assessment and Engage Stakeholders

The implementation process begins with a thorough technical assessment to identify vulnerabilities and determine the compatibility of the existing infrastructure with proposed security enhancements. This assessment should evaluate the DLMS/COSEM protocols in use, current encryption standards, and overall network architecture. Engaging key stakeholders—including technology partners, cybersecurity experts, regulatory bodies, and manufacturers is crucial for aligning implementation with best practices and compliance requirements. Collaboration with these partners can provide valuable insights and support, facilitating the deployment of advanced security solutions.

In this phase, it is essential to gather input from all involved parties to ensure that the proposed enhancements meet both technical and operational requirements. For instance, consulting with smart meter manufacturers on the integration of tamper detection mechanisms can help tailor solutions to UMEME's specific needs. Engaging with regulatory bodies ensures that the implementation aligns with local and international cybersecurity standards, fostering compliance and stakeholder confidence.

6.3.2 Step 2: Upgrade Encryption, Authentication, and Key Management

A crucial part of the security upgrade involves implementing robust encryption protocols. Transitioning to AES with GCM provides authenticated encryption, enhancing data integrity and confidentiality across all communication channels. AES-GCM is favored for its efficiency and strong security guarantees, making it suitable for the high-volume data transmissions typical in smart metering systems. This upgrade significantly reduces the risk of data interception and manipulation [57].

Implementing HLS protocols for mutual authentication ensures that only authorized entities can access the network. This involves a challenge-response mechanism, enhancing protection against unauthorized access. Incorporating Public

Key Infrastructure (PKI) allows for secure key exchange and management, essential for maintaining secure communications. PKI systems use digital certificates to authenticate entities, providing a secure framework for verifying identities within the network. This approach reduces the risk of man-in-the-middle attacks, as both the client and server are authenticated before data exchange [55].

Adopting a comprehensive key management strategy is equally important. Utilizing asymmetric cryptographic methods, such as ECDH for key exchange, provides a robust mechanism for establishing secure communication channels. This strategy mitigates risks associated with symmetric key distribution, where the compromise of a single key could lead to widespread security breaches. Regular key rotation and management practices are recommended to maintain system integrity and security over time. For example, implementing a policy that mandates key changes every 90 days can reduce the likelihood of keys being compromised [3].

6.3.3 Step 3: Implement Secure Firmware Management and Network Segmentation

Securing firmware updates is critical to preventing exploitation through outdated or vulnerable software. All firmware updates should be digitally signed and verified using asymmetric cryptography before installation, ensuring that only authentic and approved updates are applied to devices. This step is crucial for maintaining system security and preventing unauthorized modifications to device firmware, which could otherwise lead to potential exploitation. The use of secure boot processes can further enhance this by ensuring that only trusted firmware is executed [56].

Enhancing network security through segmentation involves isolating different parts of the infrastructure to limit the impact of potential breaches. Implementing VPNs for secure communication between metering devices and the utility's head-end system provides an additional layer of security by encrypting data in transit. VPNs create encrypted tunnels that protect data as it travels across public networks, reducing the risk of interception. Additionally, deploying firewalls and intrusion detection systems (IDS) can help monitor and prevent unauthorized access attempts, ensuring that any suspicious activity is promptly identified and addressed [1].

An example of effective network segmentation is using VLANs (Virtual Local Area Networks) to separate different types of traffic, such as separating meter data from control data. This reduces the attack surface and limits the potential impact of a breach on the entire network.

6.3.4 Step 4: Integrate Tamper Detection and Access Controls

Incorporating tamper detection mechanisms in smart meters can significantly enhance physical security. These systems can send alerts if physical tampering is detected, allowing for immediate response to potential security threats. For instance, implementing sensor-based tamper detection that triggers an alarm if a meter cover is removed without authorization can help deter unauthorized access and protect the device from physical attacks [3].

Implementing Role-Based Access Control (RBAC) ensures that access to the network is restricted based on users' roles and responsibilities. By limiting access rights and privileges, the risk of insider threats and unauthorized access is minimized. This approach should be complemented by regular security audits and training programs for personnel, ensuring that all users are aware of best practices and potential threats. For example, conducting quarterly security awareness workshops can keep staff informed about the latest cybersecurity threats and mitigation strategies [56].

Step 5: Pilot Testing, Training, and Continuous Monitoring

Conducting pilot tests in controlled environments is essential to address potential issues and refine the security strategy before full-scale deployment. This phase involves evaluating the effectiveness of implemented measures, identifying any areas for improvement, and ensuring compatibility with existing infrastructure. Pilot testing allows UMEME to gather valuable feedback and make necessary adjustments, ensuring a smooth transition to the new security framework [47].

Providing comprehensive training for technical staff and conducting awareness programs to educate users about new security features and protocols are critical components of this strategy. Ensuring that all personnel are informed about best practices and potential threats is vital for maintaining a secure and resilient system. Implementing ongoing training programs and certification requirements for cybersecurity personnel can enhance the overall security posture of the organization [57].

Establishing continuous monitoring and incident response plans ensures that UMEME can adapt to emerging threats and respond swiftly to security breaches. Implementing advanced monitoring tools and analytics can help detect anomalies in real-time, allowing for proactive threat mitigation. A robust incident response plan should outline specific steps for identifying, containing, and eradicating threats, as well as recovery procedures to restore normal operations. This proactive approach not only protects the integrity of UMEME's smart metering system but also aligns with global standards, reinforcing consumer confidence in the utility's commitment to security and reliability [1].

By following this detailed implementation plan, UMEME can effectively enhance its cybersecurity posture, ensuring the protection and reliability of its smart metering infrastructure. This strategy not only addresses current vulnerabilities but also prepares UMEME for future challenges, safeguarding its operations and maintaining consumer trust in the rapidly evolving digital landscape [3].

Chapter 7

Conclusion

The completion of this thesis marks a significant milestone in the research and implementation of smart metering systems within the context of Uganda's electricity distribution network, particularly UMEME's OneSait Smart Metering Project. The main objectives of the thesis—analyzing the DLMS/COSEM and IDIS standards, designing and simulating IDIS-compliant test cases, and exploring cybersecurity measures for smart metering infrastructures—have been successfully addressed and achieved. These outcomes provide valuable contributions to both the academic understanding of smart metering technology and the practical implementation of this technology in real-world scenarios.

In Chapter 4, the focus was placed on UMEME's smart metering project, which aims to modernize Uganda's energy infrastructure by deploying advanced smart meters that are compliant with international standards. The challenges faced in implementing such a system, including ensuring interoperability between devices from different manufacturers, are critical to the project's success. By utilizing the DLMS/COSEM communication protocol and ensuring adherence to IDIS standards, UMEME can create a smart metering network that not only supports efficient energy distribution but also provides the foundation for more advanced services such as real-time monitoring, remote configuration, and dynamic billing. The thesis has contributed to this project by identifying key use cases essential for IDIS compliance, which serve as benchmarks for evaluating the functionality of smart meters in this network.

The importance of prepayment and billing in Uganda's electricity sector was also highlighted in Chapter 4, particularly with the integration of Minsait's Onesait Prepayment Solution. This system allows for more accurate billing and revenue assurance by enabling consumers to pay for electricity in advance, based on real-time consumption data gathered from smart meters. By ensuring IDIS compliance, the system becomes more robust, providing UMEME with the flexibility to adjust tariffs dynamically while improving customer satisfaction and reducing losses due to inaccuracies or theft. This integration of prepayment systems into the broader smart metering infrastructure demonstrates the practical relevance of the test cases developed in this thesis, ensuring that UMEME's smart meters meet the requirements needed to function effectively within this framework.

Chapter 5 focused on the design and implementation of the use case tests, a critical component of the thesis that ensured UMEME's smart meters adhered to IDIS standards. The successful development of these test cases was one of the key accomplishments of this research, directly addressing the objective of implementing IDIS standards into UMEME's smart metering system. Each use case was carefully constructed to simulate real-world scenarios, ensuring that the meters could handle essential functionalities such as remote tariff programming, on-demand and periodic readings, connection and disconnection, load management, and firmware upgrades.

The simulation environment played a pivotal role in the testing process, allowing the smart meters to be evaluated under controlled conditions. By simulating different types of meters and testing their performance across multiple use cases, the thesis was able to validate the meters' compliance with IDIS standards and their suitability for deployment in Uganda's electricity grid. Moreover, the simulation provided a practical demonstration of how these meters would behave in the field, offering valuable insights into their strengths and weaknesses.

For instance, in the testing of various meters, it became evident that while many of the meters successfully passed most use cases, there were still some that faced challenges in specific areas such as tariff programming and connection management. These findings underscore the importance of rigorous testing before deployment, ensuring that meters can operate effectively under the diverse conditions they will encounter in the real world. Furthermore, the results from the simulation tests highlighted the effectiveness of the designed test cases in identifying potential issues with meter compliance, allowing for targeted improvements to be made prior to full-scale deployment.

Chapter 6 took a deeper dive into the cybersecurity aspect of smart metering systems—an area of growing concern as energy infrastructures become increasingly digitized and interconnected. The research within this chapter revealed significant vulnerabilities in smart meters that, if left unaddressed, could expose UMEME's infrastructure to a variety of cyber threats. These vulnerabilities include weak encryption protocols, inadequate authentication mechanisms, and insufficient tamper detection systems, all of which could be exploited by malicious actors to disrupt service, steal sensitive data, or manipulate billing records.

In response to these findings, Chapter 6 proposed a range of cybersecurity measures aimed at strengthening UMEME's smart metering system. One of the primary recommendations was the implementation of more advanced encryption techniques, such as AES-256, to secure the communication between meters and the Head-End System. This ensures that even if communication is intercepted, the data remains protected and unusable to unauthorized parties. Additionally, the introduction of stronger authentication protocols, including multifactor authentication and public key infrastructure, was recommended to prevent unauthorized access to the smart meters and the HES. These measures would significantly reduce the risk of cyberattacks by ensuring that only authorized users and devices can interact with the system.

Another critical recommendation was the enhancement of tamper detection mechanisms within the meters. Smart meters are often deployed in accessible locations, making them vulnerable to physical tampering, which can lead to inaccurate readings or service disruptions. By implementing more sophisticated tamper detection algorithms and hardware features, the meters can more effectively detect and respond to unauthorized physical access, ensuring the integrity of the data they collect and transmit.

In addressing the objectives of the thesis, the research has demonstrated that by adopting these cybersecurity measures, UMEME can significantly mitigate the risks associated with deploying smart meters in its network. The proposed solutions are not only practical but also scalable, allowing UMEME to enhance the security of its infrastructure as the network continues to grow.

The combination of comprehensive testing for IDIS compliance and enhanced cybersecurity measures positions UMEME's smart metering project as a robust and future-proof solution for Uganda's electricity distribution network. The research presented in this thesis has shown that it is possible to design and implement a smart metering system that is not only compliant with international standards but also secure against emerging threats. This dual focus on interoperability and security is essential for ensuring the long-term success of the OneSait Smart Metering Project, enabling UMEME to provide reliable, efficient, and secure electricity services to its customers.

In conclusion, this thesis has achieved its primary objectives by successfully designing and implementing use case tests for IDIS compliance and by addressing the cybersecurity challenges inherent in smart metering systems. The work has laid a strong foundation for UMEME to move forward with its smart metering project, confident in the knowledge that its meters are capable of operating within the IDIS framework while being protected from potential cyber threats. The results of this research will not only benefit UMEME but also serve as a valuable resource for other utilities and organizations looking to implement smart metering systems in similar contexts.

As UMEME continues to expand its smart metering network, the contributions of this thesis will play a key role in ensuring that the infrastructure remains reliable, secure, and compliant with international standards. By embracing the recommendations outlined in this thesis, UMEME can position itself as a leader in smart metering technology, setting an example for other utilities in Africa and around the world. The combination of rigorous testing, adherence to global standards, and a proactive approach to cybersecurity will ensure the success of UMEME's smart metering project for years to come.

Appendix A

IDIS Tests Matlab Code

A.1 Main Code

```
clc
clear all

% Meter and HES parameters

meter = initializeMeter();
hesIP = '127.0.0.1'; % Localhost for simulation
hesPort = 502; % Example port

%simulateMeter(meter, hesIP, hesPort);
simulateMeterAllTests(meter, hesIP, hesPort);
```

A.2 Initialize Meter Code

```
function meter = initializeMeter()

```

```matlab
% UC1 - Initialize the meter structure with default values
meter.deviceID = '1541874';  % Example device ID
meter.ipAddress = '192.168.1.100'; % Example IP address
meter.systemTitle = 'MeterSystemTitle'; % Example system title

% UC2 - Initialize tariff-related variables
%     meter.tariffs = struct(...
%     'Domestic', struct('lifelineRate', '0.05', 'rate', '797.3'), ...
%     'Commercial', struct('rate', '791.9'), ...
%     'Medium_Industrial', struct('rate', '502.9'), ...
%     'Large_Industrial', struct('rate', '502.9'), ...
%     'Street_Lighting', struct('rate', '370.0') ...
% );
meter.consumerCategory = 'Domestic';  % Default consumer category

% UC3  - Initialize energy registers for on-demand reading with random realistic values
% Typical household energy consumption values
meter.activeEnergyImport = round(500 + rand() * 500, 2);  % Active energy import in kWh (500-1000)
meter.activeEnergyExport = round(rand() * 100, 2);        % Active energy export in kWh (0-100)
meter.reactiveEnergyImport = round(rand() * 50, 2);       % Reactive energy import in kVARh (0-50)
meter.reactiveEnergyExport = round(rand() * 30, 2);       % Reactive energy export in kVARh (0-30)

% UC4 - Initialize cumulative registers for periodic reading
meter.cumulativeActiveImport = round(300 + rand() * 900, 2); % 300 to 1200 kWh
```

```
    meter.cumulativeActiveExport = round(rand() * 200, 2);        % 0 to 200 kWh
    % Initialize rate registers for periodic reading
    meter.rateActiveImportT1 = round(100 + rand() * 500, 2);      % 100 to 600 kWh
    meter.rateActiveExportT1 = round(rand() * 50, 2);             % 0 to 50 kWh

    % UC5 - Initialize connection status
    % Possible states: 0 = Disconnected, 1 = Connected, 2 = Ready for Reconnection
    meter.connectionState = 1; % Assume initially connected
    % Initialize instant current
    meter.instantCurrent = round(10 + rand() * 20, 2);  % Instant current in Amperes, range 10 to 30 A

    % UC6 - Initialize local time with some deviation (e.g., within 5 minutes)
    initialTimeOffset = minutes(rand() * 10 - 5);  % Random offset between -5 to +5 minutes
    meter.localTime = datetime('now', 'TimeZone', 'local') + initialTimeOffset;
    meter.localTime.Format = 'yyyy-MM-dd HH:mm:ss';

    % UC7 - Initialize quality of supply parameters
    meter.instVoltageL1 = round(220 + randn() * 10, 2); % Instantaneous Voltage L1 (V)
    meter.instCurrentL1 = round(5 + randn() * 2, 2);    % Instantaneous Current L1 (A)
    meter.instTotalActivePower = round(1100 + randn() * 200, 2); % Total Active Power (W)
    meter.voltageSagL1 = randi([0, 10]);                % Voltage Sag Count L1
```

```
    meter.voltageSwellL1 = randi([0, 5]);              % Voltage Swell Count L1
    meter.powerFailureCount = randi([0, 3]);           % Power Failure Count

    % UC8 - Initialize load management parameters
    meter.activeDemandImport = round(15 + rand() * 10, 2); % Active demand in kW
    meter.relayStatus = randi([0, 1]); % Randomly start with relay ON or OFF

    % UC9 - Initialize firmware parameters
    meter.currentFirmwareVersion = 1.0; % Starting firmware version

    % UC10 - Initialize meter supervision parameters
    meter.tamperStatus = randi([0, 1]);         % 0 = No Tamper, 1 = Tamper Detected
    meter.powerOutageCount = randi([0, 5]);     % Number of power outages detected
    meter.errorCode = randi([0, 3]);            % 0 = No Error, 1-3 = Various Error Codes
    meter.unexpectedResetDetected = randi([0, 1]); % 0 = No Reset, 1 = Unexpected Reset Detected

    % UC11 - Initialize consumer information
    meter.consumerName = 'Mazinger';                     % Example consumer name
    meter.consumerAddress = '420 High Road, Kampala';  % Example consumer address
    meter.tariffCategory = 'Residential';              % Tariff category
```

```
meter.accountNumber = 'ACC123456789';              % Consumer account number
meter.emailAddress = 'mazinger.z@example.com';        % Consumer email address
meter.phoneNumber = '+257701234567';               % Consumer phone number
meter.contractType = 'Standard Residential';        % Contract type
% Initialize additional electricity-related variables
meter.averageMonthlyConsumption = round(400 + rand() * 200, 2); % Average monthly consumption in kWh
meter.peakDemand = round(2 + rand() * 3, 2);                      % Peak demand in kW
meter.billingCycle = 'Monthly';                                   % Billing cycle
meter.energyEfficiencyRating = 'A';                               % Energy efficiency rating

% UC 12 - Initialize communication supervision parameters
meter.signalStrength = round(-70 + randn() * 5); % Signal strength in dBm
meter.dataErrorRate = round(rand() * 5, 2);      % Data error rate in percentage
meter.connectionUptime = round(rand() * 5, 2);   % Connection uptime in hours
% Initialize communication events
meter.communicationEvents = {
    'Start remote communication';
    'End remote communication';
    'Start local communication';
    'End local communication';
};

```

```
    % UC 13 - Initialize outage supervision variables
    meter.outageNotificationEnabled = true; % Enable or disable outage notifications
    meter.predefinedDestinationAddress = 'HES_MAIN'; % Destination for outage notifications

    % UC 14 - Initialize display configuration variables
    meter.scrollDisplayMode = 1; % 1 to 240 seconds, 255 to deactivate
    meter.displayDataFormat = struct('energy', 1, 'demand', 1, 'voltage', 1, 'current', 1); % Example format types
    meter.alarmDisplayFilter = 0; % Alarm filter status
    meter.consumerMessage = 'Welcome'; % Default consumer message

    % UC 15 - Initialize warning message parameters
    meter.warningDestinationAddress = 'Utility'; % Could be 'Utility' or 'Customer'
    meter.loadLimit = 5000; % Maximum allowable load in watts before a warning is sent
    meter.currentLoad = randi([1000, 6000]); % Random initial load in watts

end
```

A.3 Meter Simulation Code

```
function simulateMeter(meter, hesIP, hesPort)
    while true
        % Display options to the user
        fprintf('\nSelect a use case to test:\n');
        fprintf('1. Meter Registration\n');
        fprintf('2. Remote Tariff Programming\n');
```

```
        fprintf('3. On-Demand Reading\n');
        fprintf('4. Periodic Reading\n');
        fprintf('5. Connection/Disconnection\n');
        fprintf('6. Clock Synchronization\n');
        fprintf('7. Quality of Supply\n');
        fprintf('8. Load Management by Relay\n');
        fprintf('9. Firmware Upgrade\n');
        fprintf('10. Meter Supervision\n');
        fprintf('11. Consumer Information\n');
        fprintf('12. Communication Supervision\n');
        fprintf('13. Outage Supervision\n');
        fprintf('14. Remote Parameter Configuration\n');
        fprintf('15. Warning Message Management\n');
        fprintf('16. Exit\n');
        choice = input('Enter your choice: ');

        % Exit the loop if the choice is 16
        if choice == 16
            fprintf('Exiting simulation.\n');
            break;
        end

        % Create a TCP client to connect to the HES
        try
            t = tcpclient(hesIP, hesPort, 'Timeout', 10);
        catch ME
            fprintf('Failed to connect to HES: %s\n', ME.message);
            continue;
        end

        % Execute the selected use case
        try
            switch choice
```

```
            case 1
                sendRegistrationNotification(t, meter);
            case 2
                performRemoteTariffProgramming(t, meter);
            case 3
                requestOnDemandReading(t, meter);
            case 4
                sendPeriodicReadings(t, meter);
            case 5
                performConnectionDisconnection(t, meter);
            case 6
                performClockSynchronization(t, meter);
            case 7
                performQualityOfSupplyReporting(t, meter);
            case 8
                performLoadManagementByRelay(t, meter);
            case 9
                performFirmwareUpgrade(t, meter);
            case 10
                performMeterSupervision(t, meter);
            case 11
                performConsumerInformationUpdate(t, meter);
            case 12
                performCommunicationSupervision(t, meter);
            case 13
                performOutageSupervision(t, meter);
            case 14
                performRemoteParameterConfiguration(t, meter);
            case 15
                performWarningMessageManagement(t, meter);
            otherwise
                fprintf('Invalid choice. Please select a valid option.\n');
```

```
            end
        catch ME
            fprintf('An error occurred during use case execution: %s\n', ME.message);
        end

        % Close the connection after processing
        clear t;
    end
end

% Function to send registration notification
function sendRegistrationNotification(t, meter)
    registrationMessage = sprintf('REGISTER|MeterID:%s|IP:%s|SystemTitle:%s', ...
                                  meter.deviceID, meter.ipAddress, meter.systemTitle);
    write(t, registrationMessage, "char");
    fprintf('Sent registration message: %s\n', registrationMessage);

    % Wait for and process response
    pause(1);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
        fprintf('No response from HES.\n');
    end
end

function performRemoteTariffProgramming(t, meter)
```

```
    % Define tariffs based on consumer category with valid field names
    tariffs = struct(...
        'Domestic', struct('lifelineRate', '0.05', 'rate', '797.3'), ...
        'Commercial', struct('rate', '791.9'), ...
        'Medium_Industrial', struct('rate', '502.9'), ...
        'Large_Industrial', struct('rate', '502.9'), ...
        'Street_Lighting', struct('rate', '370.0') ...
    );

    % Determine current tariff based on the consumer category
    currentCategory = strrep(meter.consumerCategory, ' ', '_'); % Replace spaces with underscores
    currentTariff = tariffs.(currentCategory);

    % Display available tariff options
    fprintf('Current tariff for %s is %s.\n', currentCategory, currentTariff.rate);
    fprintf('Select a new tariff rate from the options below:\n');
    availableRates = fieldnames(tariffs);

    for i = 1:length(availableRates)
        rate = tariffs.(availableRates{i}).rate;
        fprintf('%d. %s Rate: %s\n', i, availableRates{i}, rate);
    end

    % Get user selection
    choice = input('Enter the number corresponding to your choice: ');
    if choice < 1 || choice > length(availableRates)
        fprintf('Invalid choice. Using current rate.\n');
        newRate = currentTariff.rate;
```

```
    else
        newRate = tariffs.(availableRates{choice}).rate;
    end

    % Create message with current and new tariff information
    message = sprintf('TARIFF|MeterID:%s|Category:%s|CurrentRate:%s|NewRate:%s', ...
                      meter.deviceID, currentCategory, currentTariff.rate, newRate);

    % Only include LifelineRate for Domestic tariffs
    if strcmp(currentCategory, 'Domestic')
        message = sprintf('%s|LifelineRate:%s', message, currentTariff.lifelineRate);
    end

    write(t, message, "char");
    fprintf('Sent tariff change request: %s\n', message);

    % Allow some time for the server to process and respond
    pause(2); % Adjust this pause duration if necessary

    % Continuously check for response from HES
    totalWaitTime = 0;
    while totalWaitTime < 20 % 20 seconds timeout, adjust as needed
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);
            break;
        else
            pause(0.5); % Wait a little before checking again
            totalWaitTime = totalWaitTime + 0.5;
```

```
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function requestOnDemandReading(t, meter)
    % Create the on-demand reading request message including meter parameters
    message = sprintf('ON_DEMAND_READ|MeterID:%s|Request:OnDemand|ActiveImport:%.2f|ActiveExport:%.2f|ReactiveImport:%.2f|ReactiveExport:%.2f', ...
        meter.deviceID, meter.activeEnergyImport, meter.activeEnergyExport, ...
        meter.reactiveEnergyImport, meter.reactiveEnergyExport);

    % Send the request to the HES
    write(t, message, "char");
    fprintf('Sent on-demand reading request: %s\n', message);

    % Wait for the response from HES
    pause(2);
    totalWaitTime = 0;
    while totalWaitTime < 20 % Timeout after 20 seconds
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received on-demand reading response: %s\n', response);
            break;
        else
            pause(0.5);
```

```
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function sendPeriodicReadings(t, meter)
    % Generate periodic reading data
    cumulativeImport = meter.cumulativeActiveImport;  % Active energy import cumulative
    cumulativeExport = meter.cumulativeActiveExport;  % Active energy export cumulative
    rateImportT1 = meter.rateActiveImportT1;          % Active energy import for rate T1
    rateExportT1 = meter.rateActiveExportT1;          % Active energy export for rate T1

    % Create message with periodic reading information
    message = sprintf('PERIODIC_READ|MeterID:%s|CumulativeImport:%.2f|CumulativeExport:%.2f|RateImportT1:%.2f|RateExportT1:%.2f', ...
                      meter.deviceID, cumulativeImport, cumulativeExport, rateImportT1, rateExportT1);

    % Send the message to the HES
    write(t, message, "char");
    fprintf('Sent periodic reading data: %s\n', message);

    % Wait for acknowledgment from HES
```

```
    pause(2);
    totalWaitTime = 0;
    while totalWaitTime < 20 % Timeout after 20 seconds
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received acknowledgment: %s\n', response);
            break;
        else
            pause(0.5);
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function performConnectionDisconnection(t, meter)
    % Define possible states
    states = struct('Disconnected', 0, 'Connected', 1, 'ReadyForReconnection', 2);

    % Read the initial meter state from the meter structure
    fprintf('Current meter state: %d\n', meter.connectionState);

    % Create message with current state and instant current information
    message = sprintf('CONNECTION|MeterID:%s|State:%d|InstantCurrent:%.2f', ...
                      meter.deviceID, meter.connectionState, meter.instantCurrent);
    write(t, message, "char");
```

```
    fprintf('Sent connection state and instant current info: %s\n', message);

    % Allow some time for the server to process and respond
    pause(2); % Adjust this pause duration if necessary

    % Continuously check for response from HES
    totalWaitTime = 0;
    while totalWaitTime < 20 % 20 seconds timeout, adjust as needed
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);

            % Parse and update meter state based on HES response
            if contains(response, 'ACK|State:')
                newState = extractBetween(response, 'ACK|State:', '|');
                meter.connectionState = str2double(newState{1});
                fprintf('Updated meter state to: %d\n', meter.connectionState);
            end
            break;
        else
            pause(0.5); % Wait a little before checking again
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end
```

```
function performClockSynchronization(t, meter)
    % Use the pre-initialized local time from the meter structure
    localTime = char(meter.localTime);  % Convert to string

    % Create message with current local time
    message = sprintf('CLOCK_SYNC|MeterID:%s|LocalTime:%s', meter.deviceID, localTime);
    write(t, message, "char");
    fprintf('Sent clock synchronization request: %s\n', message);

    % Allow some time for the server to process and respond
    pause(2);

    % Continuously check for response from HES
    totalWaitTime = 0;
    while totalWaitTime < 20  % 20 seconds timeout
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);

            % Parse and update meter time based on HES response
            if contains(response, 'ACK|NewTime:')
                newTime = extractAfter(response, 'ACK|NewTime:');
                meter.localTime = datetime(newTime, 'InputFormat', 'yyyy-MM-dd HH:mm:ss', 'TimeZone', 'local');
                fprintf('Updated meter time to: %s\n', newTime);
            end
            break;
        else
            pause(0.5);  % Wait before checking again
            totalWaitTime = totalWaitTime + 0.5;
        end
```

```
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function performQualityOfSupplyReporting(t, meter)
    % Create a message with quality of supply data
    message = sprintf('QUALITY_REPORT|MeterID:%s|InstVoltageL1:%.2f|InstCurrentL1:%.2f|InstPower:%.2f|VoltageSagL1:%d|VoltageSwellL1:%d|PowerFailureCount:%d', ...
        meter.deviceID, meter.instVoltageL1, meter.instCurrentL1, meter.instTotalActivePower, meter.voltageSagL1, meter.voltageSwellL1, meter.powerFailureCount);
    write(t, message, "char");
    fprintf('Sent quality of supply report: %s\n', message);

    % Wait for acknowledgment from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
        fprintf('No response from HES.\n');
    end
end

function performLoadManagementByRelay(t, meter)
    % Monitor active demand and relay status
    currentDemand = meter.activeDemandImport;

    % Create a message to send current relay status and demand
```

```
    message = sprintf('LOAD_MANAGEMENT|MeterID:%s|CurrentDemand
       :%.2f|RelayStatus:%d', ...
        meter.deviceID, currentDemand, meter.relayStatus);
    write(t, message, "char");
    fprintf('Sent load management report: %s\n', message);

    % Wait for a response or command from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);

        % Process HES command
        if contains(response, 'COMMAND|Relay:ON')
            meter.relayStatus = 1; % Turn relay ON
            fprintf('Relay turned ON.\n');
        elseif contains(response, 'COMMAND|Relay:OFF')
            meter.relayStatus = 0; % Turn relay OFF
            fprintf('Relay turned OFF.\n');
        end
    else
        fprintf('No response from HES.\n');
    end
end

function performFirmwareUpgrade(t, meter)
    % Simulate downloading firmware
    fprintf('Starting firmware download...\n');
    pause(2); % Simulate download time
    firmwareVersion = sprintf('v%.1f', meter.
       currentFirmwareVersion + 0.1);

```

```
    % Create message to send firmware download completion and version
    message = sprintf('FIRMWARE_UPGRADE|MeterID:%s|NewVersion:%s|Status:Downloaded', ...
        meter.deviceID, firmwareVersion);
    write(t, message, "char");
    fprintf('Sent firmware upgrade status: %s\n', message);

    % Wait for verification response from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);

        % Check if the firmware is verified
        if contains(response, 'VERIFIED')
            % Activate new firmware
            meter.currentFirmwareVersion = str2double(extractAfter(firmwareVersion, 'v'));
            fprintf('Firmware version updated to %s\n', firmwareVersion);
        else
            fprintf('Firmware verification failed.\n');
        end
    else
        fprintf('No response from HES.\n');
    end
end

function performMeterSupervision(t, meter)
    % Simulate event detection
    tamperDetected = meter.tamperStatus;
    powerOutages = meter.powerOutageCount;
```

```
    errorCode = meter.errorCode;
    unexpectedReset = meter.unexpectedResetDetected;

    % Create message with meter supervision status
    message = sprintf('SUPERVISION_REPORT|MeterID:%s|Tamper:%d|PowerOutages:%d|ErrorCode:%d|UnexpectedReset:%d', ...
        meter.deviceID, tamperDetected, powerOutages, errorCode, unexpectedReset);
    write(t, message, "char");
    fprintf('Sent meter supervision report: %s\n', message);

    % Wait for acknowledgment from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);

        % Reset unexpected reset flag after reporting
        meter.unexpectedResetDetected = 0;
    else
        fprintf('No response from HES.\n');
    end
end

function performConsumerInformationUpdate(t, meter)
    % Create a message with consumer information
    message = sprintf('CONSUMER_INFO|MeterID:%s|ConsumerName:%s|ConsumerAddress:%s|TariffCategory:%s|AccountNumber:%s|EmailAddress:%s|PhoneNumber:%s|ContractType:%s|AvgMonthlyConsumption:%.2f|PeakDemand:%.2f|BillingCycle:%s|EnergyEfficiencyRating:%s', ...
        meter.deviceID, meter.consumerName, meter.consumerAddress, meter.tariffCategory, meter.accountNumber, meter.
```

```
            emailAddress, meter.phoneNumber, meter.contractType,
            meter.averageMonthlyConsumption, meter.peakDemand,
            meter.billingCycle, meter.energyEfficiencyRating);
    write(t, message, "char");
    fprintf('Sent consumer information: %s\n', message);

    % Wait for acknowledgment from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
        fprintf('No response from HES.\n');
    end
end

function performCommunicationSupervision(t, meter)
    % Select a random event from the initialized communication
       events
    eventId = randi([1, length(meter.communicationEvents)]);
    eventName = meter.communicationEvents{eventId};
    timestamp = datestr(now, 'yyyy-mm-dd HH:MM:SS');

    % Create a message with event information
    message = sprintf('COMM_EVENT|MeterID:%s|EventID:%d|EventName
       :%s|Timestamp:%s', ...
                      meter.deviceID, eventId, eventName,
                         timestamp);

    % Send message to HES
    write(t, message, "char");
    fprintf('Sent communication event: %s\n', message);

```

```
    % Wait for acknowledgment from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
        fprintf('No response from HES.\n');
    end
end

function performOutageSupervision(t, meter)
    % Check if outage notifications are enabled
    if meter.outageNotificationEnabled
        % Simulate an outage event
        outageEvent = 'Total Power Outage';
        timestamp = datestr(now, 'yyyy-mm-dd HH:MM:SS');

        % Create message for outage notification
        message = sprintf('OUTAGE|MeterID:%s|Event:%s|Timestamp:%s|Destination:%s', ...
                          meter.deviceID, outageEvent, timestamp, meter.predefinedDestinationAddress);

        % Send outage notification to HES
        write(t, message, "char");
        fprintf('Sent outage notification: %s\n', message);

        % Wait for acknowledgment from HES
        pause(2);
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);
        else
```

```
            fprintf('No response from HES.\n');
        end
    else
        fprintf('Outage notification is disabled.\n');
    end
end

function performRemoteParameterConfiguration(t, meter)
    % Create a message with the current configuration
    message = sprintf('CONFIG|MeterID:%s|ScrollMode:%d|EnergyFormat:%d|DemandFormat:%d|VoltageFormat:%d|CurrentFormat:%d|AlarmFilter:%d|Message:%s', ...
        meter.deviceID, meter.scrollDisplayMode, meter.displayDataFormat.energy, ...
        meter.displayDataFormat.demand, meter.displayDataFormat.voltage, meter.displayDataFormat.current, ...
        meter.alarmDisplayFilter, meter.consumerMessage);

    % Send current configuration to HES
    write(t, message, "char");
    fprintf('Sent current configuration: %s\n', message);

    % Wait for configuration update from HES
    totalWaitTime = 0;
    while totalWaitTime < 20 % 20 seconds timeout, adjust as needed
        if t.NumBytesAvailable > 0
            data = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received configuration update: %s\n', data);

            % Parse configuration update
            parsedData = split(data, '|');

```

```matlab
        if length(parsedData) < 9
            fprintf('Error: Incomplete message received.\n');
            writeline(t, 'ERROR|Incomplete Configuration Message');
            return;
        end

        % Apply configuration update
        meter.scrollDisplayMode = str2double(extractAfter(parsedData{3}, 'ScrollMode:'));
        meter.displayDataFormat.energy = str2double(extractAfter(parsedData{4}, 'EnergyFormat:'));
        meter.displayDataFormat.demand = str2double(extractAfter(parsedData{5}, 'DemandFormat:'));
        meter.displayDataFormat.voltage = str2double(extractAfter(parsedData{6}, 'VoltageFormat:'));
        meter.displayDataFormat.current = str2double(extractAfter(parsedData{7}, 'CurrentFormat:'));
        meter.alarmDisplayFilter = str2double(extractAfter(parsedData{8}, 'AlarmFilter:'));
        meter.consumerMessage = extractAfter(parsedData{9}, 'Message:');

        fprintf('Updated Meter Configuration:\n');
        fprintf('Scroll Display Mode: %d seconds\n', meter.scrollDisplayMode);
        fprintf('Display Data Format: Energy=%d, Demand=%d, Voltage=%d, Current=%d\n', ...
            meter.displayDataFormat.energy, meter.displayDataFormat.demand, meter.displayDataFormat.voltage, meter.displayDataFormat.current);
```

```
            fprintf('Alarm Display Filter: %d\n', meter.alarmDisplayFilter);
            fprintf('Consumer Message: %s\n', meter.consumerMessage);

            % Send acknowledgment back to the HES
            ackMessage = sprintf('ACK|MeterID:%s|ConfigUpdateSuccess', meter.deviceID);
            writeline(t, ackMessage);
            fprintf('Response sent: %s\n', ackMessage);
            break;
        else
            pause(0.5); % Wait a little before checking again
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No configuration update received from HES after waiting.\n');
    end
end


function performWarningMessageManagement(t, meter)
    % Check current load against load limit
    if meter.currentLoad > meter.loadLimit
        % Create warning message
        warningMessage = sprintf('WARNING|MeterID:%s|CurrentLoad:%d|LoadLimit:%d|Destination:%s', ...
            meter.deviceID, meter.currentLoad, meter.loadLimit, meter.warningDestinationAddress);

```

```
        % Send warning message
        write(t, warningMessage, "char");
        fprintf('Sent warning message: %s\n', warningMessage);

        % Wait for acknowledgment from HES
        pause(2);
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);
        else
            fprintf('No acknowledgment received from HES.\n');
        end
    else
        fprintf('Load within limits. No warning sent.\n');
    end
end
```

A.4 Head End System Simulation Code

```
function simulateHES(ip, port)
    % Create a TCP/IP server
    t = tcpserver(ip, port, 'Timeout', 30);
    fprintf('Simulated HES running on %s:%d\n', ip, port);

    while true
        if t.Connected
            % Check if there is data available
            if t.NumBytesAvailable > 0
                data = read(t, t.NumBytesAvailable, "char");
                fprintf('Received message: %s\n', data);

                if contains(data, 'REGISTER|')
```

```
                processRegistration(t, data);
            elseif contains(data, 'TARIFF|')
                processTariffProgramming(t, data);
            elseif contains(data, 'ON_DEMAND_READ|')
                processOnDemandReading(t, data);
            elseif contains(data, 'PERIODIC_READ|')
                processPeriodicReadings(t, data);
            elseif contains(data, 'CONNECTION|')
                processConnectionDisconnection(t, data);
            elseif contains(data, 'CLOCK_SYNC|')
                processClockSynchronization(t, data);
            elseif contains(data, 'QUALITY_REPORT|')
                processQualityOfSupply(t, data);
            elseif contains(data, 'LOAD_MANAGEMENT|')
                processLoadManagement(t, data);
            elseif contains(data, 'FIRMWARE_UPGRADE|')
                processFirmwareUpgrade(t, data);
            elseif contains(data, 'SUPERVISION_REPORT|')
                processMeterSupervision(t, data);
            elseif contains(data, 'CONSUMER_INFO|')
                processConsumerInformation(t, data);
            elseif contains(data, 'COMM_EVENT|')
                processCommunicationEvent(t, data);
            elseif contains(data, 'OUTAGE|')
                processOutageNotification(t, data);
            elseif contains(data, 'CONFIG|')
                processRemoteParameterConfiguration(t, data);
            elseif contains(data, 'WARNING|')
                processWarningMessage(t, data);
            else
                writeline(t, 'ERROR|Invalid Message');
                fprintf('Response sent: ERROR|Invalid Message\
                   n');
```

```
                end
            end
        else
            pause(0.1); % Brief pause to avoid busy-waiting
        end
    end
end

function processRegistration(t, data)
    % Process registration
    writeline(t, 'ACK|Registration Successful');
    fprintf('Response sent: ACK|Registration Successful\n');
end

function processTariffProgramming(t, data)
    % Parse current and new tariff info from the meter
    fprintf('Processing tariff programming...\n');
    parsedData = split(data, '|');

    % Debugging: Display the parsed data
    fprintf('Parsed data:\n');
    disp(parsedData);

    % Check if the parsed data has the expected length
    if length(parsedData) < 5
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    % Extract values using regular expressions for robustness
    category = regexp(parsedData{3}, 'Category:(.*)', 'tokens', 'once');
```

```
    currentRate = regexp(parsedData{4}, 'CurrentRate:(.*)', 'tokens', 'once');
    newRate = regexp(parsedData{5}, 'NewRate:(.*)', 'tokens', 'once');

    % Handle the extraction when values are missing
    if isempty(category) || isempty(currentRate) || isempty(newRate)
        fprintf('Error: Unable to extract values from the message.\n');
        writeline(t, 'ERROR|Invalid Message Format');
        return;
    end

    category = category{1}; % Extract the token from the cell array
    currentRate = currentRate{1};
    newRate = newRate{1};

    % Check if the message includes a LifelineRate (specific for Domestic)
    if strcmp(category, 'Domestic') && length(parsedData) >= 6
        lifelineRate = regexp(parsedData{6}, 'LifelineRate:(.*)', 'tokens', 'once');
        if ~isempty(lifelineRate)
            lifelineRate = lifelineRate{1};
            fprintf('Category: %s, Current Rate: %s, New Rate: %s, Lifeline Rate: %s\n', ...
                    category, currentRate, newRate, lifelineRate);
        end
    else
        fprintf('Category: %s, Current Rate: %s, New Rate: %s\n', ...
```

```matlab
                    category, currentRate, newRate);
    end

    % Simulate tariff update logic
    fprintf('Tariff for category %s updated from %s to %s.\n', category, currentRate, newRate);

    % Send acknowledgment back to the meter with the applied new rate
    ackMessage = sprintf('ACK|Category:%s|AppliedNewRate:%s', category, newRate);
    writeline(t, ackMessage);
    fprintf('Response sent: %s\n', ackMessage);
end

function processOnDemandReading(t, data)
    % Process on-demand reading request
    fprintf('Processing on-demand reading...\n');
    parsedData = split(data, '|');

    % Extract MeterID and energy values from the message
    meterID = regexp(parsedData{2}, 'MeterID:(\w+)', 'tokens', 'once');
    activeImport = regexp(parsedData{4}, 'ActiveImport:(\d+\.\d+)', 'tokens', 'once');
    activeExport = regexp(parsedData{5}, 'ActiveExport:(\d+\.\d+)', 'tokens', 'once');
    reactiveImport = regexp(parsedData{6}, 'ReactiveImport:(\d+\.\d+)', 'tokens', 'once');
    reactiveExport = regexp(parsedData{7}, 'ReactiveExport:(\d+\.\d+)', 'tokens', 'once');

    % Handle cases where extraction fails
```

```
    if isempty(meterID) || isempty(activeImport) || isempty(activeExport) || isempty(reactiveImport) || isempty(reactiveExport)
        fprintf('Error: Unable to extract values from the message.\n');
        writeline(t, 'ERROR|Invalid Message Format');
        return;
    end

    % Convert extracted values from strings
    meterID = meterID{1};
    activeImport = str2double(activeImport{1});
    activeExport = str2double(activeExport{1});
    reactiveImport = str2double(reactiveImport{1});
    reactiveExport = str2double(reactiveExport{1});

    % Print the extracted values
    fprintf('MeterID: %s\n', meterID);
    fprintf('Active Import: %.2f kWh\n', activeImport);
    fprintf('Active Export: %.2f kWh\n', activeExport);
    fprintf('Reactive Import: %.2f kVARh\n', reactiveImport);
    fprintf('Reactive Export: %.2f kVARh\n', reactiveExport);

    % Prepare a response message acknowledging the request
    responseMessage = sprintf('ACK|MeterID:%s|ReadComplete|ActiveImport:%.2f|ActiveExport:%.2f|ReactiveImport:%.2f|ReactiveExport:%.2f', ...
        meterID, activeImport, activeExport, reactiveImport, reactiveExport);

    % Send the response back to the meter
    writeline(t, responseMessage);
    fprintf('Response sent: %s\n', responseMessage);
```

```matlab
end

function processPeriodicReadings(t, data)
    % Process periodic reading data
    fprintf('Processing periodic reading data...\n');
    parsedData = split(data, '|');

    % Extract MeterID and reading values from the message
    meterID = regexp(parsedData{2}, 'MeterID:(\w+)', 'tokens', 'once');
    cumulativeImport = regexp(parsedData{3}, 'CumulativeImport:(\d+\.\d+)', 'tokens', 'once');
    cumulativeExport = regexp(parsedData{4}, 'CumulativeExport:(\d+\.\d+)', 'tokens', 'once');
    rateImportT1 = regexp(parsedData{5}, 'RateImportT1:(\d+\.\d+)', 'tokens', 'once');
    rateExportT1 = regexp(parsedData{6}, 'RateExportT1:(\d+\.\d+)', 'tokens', 'once');

    % Handle cases where extraction fails
    if isempty(meterID) || isempty(cumulativeImport) || isempty(cumulativeExport) || isempty(rateImportT1) || isempty(rateExportT1)
        fprintf('Error: Unable to extract values from the message.\n');
        writeline(t, 'ERROR|Invalid Message Format');
        return;
    end

    % Convert extracted values from strings
    meterID = meterID{1};
    cumulativeImport = str2double(cumulativeImport{1});
    cumulativeExport = str2double(cumulativeExport{1});
```

```
    rateImportT1 = str2double(rateImportT1{1});
    rateExportT1 = str2double(rateExportT1{1});

    % Print the extracted values
    fprintf('MeterID: %s\n', meterID);
    fprintf('Cumulative Import: %.2f kWh\n', cumulativeImport);
    fprintf('Cumulative Export: %.2f kWh\n', cumulativeExport);
    fprintf('Rate Import T1: %.2f kWh\n', rateImportT1);
    fprintf('Rate Export T1: %.2f kWh\n', rateExportT1);

    % Simulate processing and logging the data
    fprintf('Periodic reading data processed and logged for billing.\n');

    % Send acknowledgment back to the meter
    ackMessage = sprintf('ACK|MeterID:%s|ReadProcessed', meterID);
    writeline(t, ackMessage);
    fprintf('Response sent: %s\n', ackMessage);
end

function processConnectionDisconnection(t, data)
    % Process connection/disconnection with instant current check
    fprintf('Processing connection/disconnection...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 4
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    % Extract the current state and instant current from the message
```

```
    currentState = str2double(extractAfter(parsedData{3}, 'State:'));
    instantCurrent = str2double(extractAfter(parsedData{4}, 'InstantCurrent:'));

    % Define possible states
    states = struct('Disconnected', 0, 'Connected', 1, 'ReadyForReconnection', 2);

    fprintf('Current State: %d\n', currentState);
    fprintf('Instant Current: %.2f A\n', instantCurrent);

    % Define current threshold for disconnection
    currentThreshold = 25.0; % Threshold in Amperes

    % User menu for action selection
    fprintf('Select an action:\n');
    fprintf('1. Connect the disconnected meter\n');
    fprintf('2. Disconnect the connected meter\n');
    fprintf('3. Instantly disconnect if exceeds threshold\n');
    choice = input('Enter your choice (1-3): ');

    % Decide on the action based on user choice
    switch choice
        case 1
            if currentState == states.Disconnected
                fprintf('Connecting meter...\n');
                newState = states.Connected;
            else
                fprintf('Meter is already connected.\n');
                newState = currentState;
            end
        case 2
```

```
            if currentState == states.Connected
                fprintf('Disconnecting meter...\n');
                newState = states.Disconnected;
            else
                fprintf('Meter is already disconnected.\n');
                newState = currentState;
            end
        case 3
            if currentState == states.Connected && instantCurrent
                > currentThreshold
                fprintf('Instant current exceeds threshold.
                    Disconnecting meter...\n');
                newState = states.Disconnected;
            else
                fprintf('No change required.\n');
                newState = currentState;
            end
        otherwise
            fprintf('Invalid choice. No action taken.\n');
            newState = currentState;
    end

    % Send acknowledgment back to the meter with the new state
    ackMessage = sprintf('ACK|State:%d', newState);
    writeline(t, ackMessage);
    fprintf('Response sent: %s\n', ackMessage);
end

function processClockSynchronization(t, data)
    % Process clock synchronization
    fprintf('Processing clock synchronization...\n');
    parsedData = split(data, '|');

```

```
    if length(parsedData) < 3
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    meterID = extractAfter(parsedData{2}, 'MeterID:');
    localTime = extractAfter(parsedData{3}, 'LocalTime:');

    fprintf('Meter ID: %s, Local Time: %s\n', meterID, localTime);

    % Get current accurate time
    accurateTime = datetime('now', 'TimeZone', 'UTC+2', 'Format',
        'yyyy-MM-dd HH:mm:ss');
    newTime = char(accurateTime);  % Convert to string

    fprintf('Setting new time to: %s\n', newTime);

    % Send acknowledgment back to the meter with the new time
    ackMessage = sprintf('ACK|NewTime:%s', newTime);
    writeline(t, ackMessage);
    fprintf('Response sent: %s\n', ackMessage);
end

function processQualityOfSupply(t, data)
    % Process quality of supply report
    fprintf('Processing quality of supply report...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 8
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
```

```
    end

    % Extract values from the message
    meterID = extractAfter(parsedData{2}, 'MeterID:');
    instVoltageL1 = str2double(extractAfter(parsedData{3}, 'InstVoltageL1:'));
    instCurrentL1 = str2double(extractAfter(parsedData{4}, 'InstCurrentL1:'));
    instPower = str2double(extractAfter(parsedData{5}, 'InstPower:'));
    voltageSagL1 = str2double(extractAfter(parsedData{6}, 'VoltageSagL1:'));
    voltageSwellL1 = str2double(extractAfter(parsedData{7}, 'VoltageSwellL1:'));
    powerFailureCount = str2double(extractAfter(parsedData{8}, 'PowerFailureCount:'));

    fprintf('Meter ID: %s\n', meterID);
    fprintf('Instantaneous Voltage L1: %.2f V\n', instVoltageL1);
    fprintf('Instantaneous Current L1: %.2f A\n', instCurrentL1);
    fprintf('Instantaneous Power: %.2f W\n', instPower);
    fprintf('Voltage Sag Count L1: %d\n', voltageSagL1);
    fprintf('Voltage Swell Count L1: %d\n', voltageSwellL1);
    fprintf('Power Failure Count: %d\n', powerFailureCount);

    % Example logic: Check for abnormal conditions
    if instVoltageL1 < 210 || instVoltageL1 > 250
        fprintf('Warning: Voltage L1 out of normal range!\n');
    end

    if voltageSagL1 > 5
        fprintf('Alert: High number of voltage sags detected!\n');
    end
```

```
    % Send acknowledgment back to the meter
    writeline(t, 'ACK|QualityReportReceived');
    fprintf('Response sent: ACK|QualityReportReceived\n');
end

function processLoadManagement(t, data)
    % Process load management report
    fprintf('Processing load management report...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 4
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    % Extract values from the message
    meterID = extractAfter(parsedData{2}, 'MeterID:');
    currentDemand = str2double(extractAfter(parsedData{3}, '
        CurrentDemand:'));
    relayStatus = str2double(extractAfter(parsedData{4}, '
        RelayStatus:'));

    fprintf('Meter ID: %s\n', meterID);
    fprintf('Current Demand: %.2f kW\n', currentDemand);
    fprintf('Relay Status: %d\n', relayStatus);

    % Example logic: Command to manage relay
    if currentDemand > 20  % Example threshold
        fprintf('Demand is high. Sending command to turn off relay
            .\n');
        writeline(t, 'COMMAND|Relay:OFF');
```

```
    elseif relayStatus == 0
        fprintf('Demand is normal. Sending command to turn on
           relay.\n');
        writeline(t, 'COMMAND|Relay:ON');
    else
        writeline(t, 'ACK|No Action Required');
        fprintf('No action required.\n');
    end
end

function processFirmwareUpgrade(t, data)
    % Process firmware upgrade status
    fprintf('Processing firmware upgrade...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 4
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    % Extract values from the message
    meterID = extractAfter(parsedData{2}, 'MeterID:');
    newVersion = extractAfter(parsedData{3}, 'NewVersion:');
    status = extractAfter(parsedData{4}, 'Status:');

    fprintf('Meter ID: %s\n', meterID);
    fprintf('New Firmware Version: %s\n', newVersion);
    fprintf('Status: %s\n', status);

    % Simulate firmware verification process
    if strcmp(status, 'Downloaded')
        fprintf('Verifying firmware...\n');
```

```
        pause(1); % Simulate verification time
        verified = rand() > 0.2; % Randomly verify firmware with 80% success rate

        if verified
            fprintf('Firmware verified successfully.\n');
            writeline(t, 'VERIFIED|Firmware upgrade verified.');
        else
            fprintf('Firmware verification failed.\n');
            writeline(t, 'FAILED|Firmware verification failed.');
        end
    end
end

function processMeterSupervision(t, data)
    % Process meter supervision report
    fprintf('Processing meter supervision report...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 6
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    % Extract values from the message
    meterID = extractAfter(parsedData{2}, 'MeterID:');
    tamperDetected = str2double(extractAfter(parsedData{3}, 'Tamper:'));
    powerOutages = str2double(extractAfter(parsedData{4}, 'PowerOutages:'));
    errorCode = str2double(extractAfter(parsedData{5}, 'ErrorCode:'));
```

```
    unexpectedReset = str2double(extractAfter(parsedData{6}, '
       UnexpectedReset:'));

    fprintf('Meter ID: %s\n', meterID);
    fprintf('Tamper Detected: %d\n', tamperDetected);
    fprintf('Power Outages: %d\n', powerOutages);
    fprintf('Error Code: %d\n', errorCode);
    fprintf('Unexpected Reset: %d\n', unexpectedReset);

    % Example logic: Alert if any issues are detected
    if tamperDetected
        fprintf('Alert: Tamper detected on meter %s!\n', meterID);
    end

    if powerOutages > 0
        fprintf('Alert: %d power outages reported by meter %s!\n',
            powerOutages, meterID);
    end

    % Handle different error codes
    switch errorCode
        case 1
            fprintf('Error: Low battery warning on meter %s.\n',
               meterID);
        case 2
            fprintf('Error: Communication failure with meter %s.\n
               ', meterID);
        case 3
            fprintf('Error: Internal sensor malfunction on meter %
               s.\n', meterID);
        otherwise
            if errorCode > 0
```

```
                fprintf('Error: Unknown error code %d reported by meter %s.\n', errorCode, meterID);
            end
    end

    if unexpectedReset
        fprintf('Alert: Unexpected reset detected on meter %s!\n', meterID);
    end

    % Send acknowledgment back to the meter
    writeline(t, 'ACK|SupervisionReportReceived');
    fprintf('Response sent: ACK|SupervisionReportReceived\n');
end

function processConsumerInformation(t, data)
    % Process consumer information
    fprintf('Processing consumer information...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 13
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    % Extract values from the message
    meterID = extractAfter(parsedData{2}, 'MeterID:');
    consumerName = extractAfter(parsedData{3}, 'ConsumerName:');
    consumerAddress = extractAfter(parsedData{4}, 'ConsumerAddress:');
    tariffCategory = extractAfter(parsedData{5}, 'TariffCategory:');
```

```
    accountNumber = extractAfter(parsedData{6}, 'AccountNumber:');
    emailAddress = extractAfter(parsedData{7}, 'EmailAddress:');
    phoneNumber = extractAfter(parsedData{8}, 'PhoneNumber:');
    contractType = extractAfter(parsedData{9}, 'ContractType:');
    avgMonthlyConsumption = str2double(extractAfter(parsedData
       {10}, 'AvgMonthlyConsumption:'));
    peakDemand = str2double(extractAfter(parsedData{11}, '
       PeakDemand:'));
    billingCycle = extractAfter(parsedData{12}, 'BillingCycle:');
    energyEfficiencyRating = extractAfter(parsedData{13}, '
       EnergyEfficiencyRating:');

    fprintf('Meter ID: %s\n', meterID);
    fprintf('Consumer Name: %s\n', consumerName);
    fprintf('Consumer Address: %s\n', consumerAddress);
    fprintf('Tariff Category: %s\n', tariffCategory);
    fprintf('Account Number: %s\n', accountNumber);
    fprintf('Email Address: %s\n', emailAddress);
    fprintf('Phone Number: %s\n', phoneNumber);
    fprintf('Contract Type: %s\n', contractType);
    fprintf('Average Monthly Consumption: %.2f kWh\n',
       avgMonthlyConsumption);
    fprintf('Peak Demand: %.2f kW\n', peakDemand);
    fprintf('Billing Cycle: %s\n', billingCycle);
    fprintf('Energy Efficiency Rating: %s\n',
       energyEfficiencyRating);

    % Simulate additional processing if necessary
    % For example, update consumer records or prepare billing
       information

    % Send acknowledgment back to the meter
    writeline(t, 'ACK|ConsumerInfoReceived');
```

```
    fprintf('Response sent: ACK|ConsumerInfoReceived\n');
end

function processCommunicationEvent(t, data)
    % Parse communication event from the meter
    fprintf('Processing communication event...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 5
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    eventId = extractAfter(parsedData{3}, 'EventID:');
    eventName = extractAfter(parsedData{4}, 'EventName:');
    timestamp = extractAfter(parsedData{5}, 'Timestamp:');

    fprintf('Event ID: %s, Event Name: %s, Timestamp: %s\n',
       eventId, eventName, timestamp);

    % Log the event (this could involve appending to a file or
       database)
    fprintf('Logging event: ID=%s, Name=%s, Timestamp=%s\n',
       eventId, eventName, timestamp);

    % Send acknowledgment back to the meter
    ackMessage = sprintf('ACK|EventID:%s|Logged', eventId);
    writeline(t, ackMessage);
    fprintf('Response sent: %s\n', ackMessage);
end

function processOutageNotification(t, data)
```

```
    % Parse outage notification data
    fprintf('Processing outage notification...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 5
        fprintf('Error: Incomplete message received.\n');
        writeline(t, 'ERROR|Incomplete Message');
        return;
    end

    eventId = parsedData{2};
    event = extractAfter(parsedData{3}, 'Event:');
    timestamp = extractAfter(parsedData{4}, 'Timestamp:');
    destination = extractAfter(parsedData{5}, 'Destination:');

    fprintf('Outage Event: %s, Timestamp: %s, Destination: %s\n',
       event, timestamp, destination);

    % Log the outage event (for record-keeping)
    fprintf('Logging outage event: Event=%s, Timestamp=%s,
       Destination=%s\n', event, timestamp, destination);

    % Send acknowledgment back to the meter
    ackMessage = sprintf('ACK|Event:%s|Logged', event);
    writeline(t, ackMessage);
    fprintf('Response sent: %s\n', ackMessage);
end

function processRemoteParameterConfiguration(t, data)
    % Assume data is already received and passed to the function
    fprintf('Processing configuration message from meter...\n');

    % Parse configuration message from meter
```

```
    parsedData = split(data, '|');

    if length(parsedData) < 9
        fprintf('Error: Incomplete configuration message received.\n');
        writeline(t, 'ERROR|Incomplete Configuration Message');
        return;
    end

    meterID = extractAfter(parsedData{2}, 'MeterID:');
    scrollMode = extractAfter(parsedData{3}, 'ScrollMode:');
    energyFormat = extractAfter(parsedData{4}, 'EnergyFormat:');
    demandFormat = extractAfter(parsedData{5}, 'DemandFormat:');
    voltageFormat = extractAfter(parsedData{6}, 'VoltageFormat:');
    currentFormat = extractAfter(parsedData{7}, 'CurrentFormat:');
    alarmFilter = extractAfter(parsedData{8}, 'AlarmFilter:');
    consumerMessage = extractAfter(parsedData{9}, 'Message:');

    % Log the current configuration
    fprintf('Current Meter Configuration:\n');
    fprintf('MeterID: %s\n', meterID);
    fprintf('Scroll Mode: %s seconds\n', scrollMode);
    fprintf('Display Data Formats - Energy: %s, Demand: %s, Voltage: %s, Current: %s\n', ...
        energyFormat, demandFormat, voltageFormat, currentFormat);
    fprintf('Alarm Display Filter: %s\n', alarmFilter);
    fprintf('Consumer Message: %s\n', consumerMessage);

    % Ask the user to update each parameter
    fprintf('\nEnter new configuration settings for the meter.\n');

    % Update Scroll Display Mode
```

```
    fprintf('1. Scroll Display Mode Options:\n');
    fprintf('   1. 10 seconds\n');
    fprintf('   2. 30 seconds\n');
    fprintf('   3. 60 seconds\n');
    fprintf('   4. 240 seconds\n');
    fprintf('   5. Deactivate (255)\n');
    scrollChoice = input('Select a new scroll display mode (1-5): ');
    switch scrollChoice
        case 1
            scrollMode = '10';
        case 2
            scrollMode = '30';
        case 3
            scrollMode = '60';
        case 4
            scrollMode = '240';
        case 5
            scrollMode = '255';
        otherwise
            fprintf('Invalid choice. Keeping current setting.\n');
    end

    % Update Display Data Formats
    fprintf('2. Display Data Format Options:\n');
    fprintf('   1. Format Type 1\n');
    fprintf('   2. Format Type 2\n');
    fprintf('   3. Format Type 3\n');
    fprintf('   4. Format Type 4\n');

    energyChoice = input('Select a new energy display format (1-4): ');
```

```
    demandChoice = input('Select a new demand display format (1-4): ');
    voltageChoice = input('Select a new voltage display format (1-4): ');
    currentChoice = input('Select a new current display format (1-4): ');

    energyFormat = num2str(energyChoice);
    demandFormat = num2str(demandChoice);
    voltageFormat = num2str(voltageChoice);
    currentFormat = num2str(currentChoice);

    % Update Alarm Display Filter
    fprintf('3. Alarm Display Filter:\n');
    fprintf('   0. Disabled\n');
    fprintf('   1. Enabled\n');
    alarmChoice = input('Select alarm display filter status (0 or 1): ');
    if alarmChoice == 0 || alarmChoice == 1
        alarmFilter = num2str(alarmChoice);
    else
        fprintf('Invalid choice. Keeping current setting.\n');
    end

    % Update Consumer Message
    consumerMessage = input('Enter new consumer message: ', 's');

    % Create update message
    updateMessage = sprintf('CONFIG_UPDATE|MeterID:%s|ScrollMode:%s|EnergyFormat:%s|DemandFormat:%s|VoltageFormat:%s|CurrentFormat:%s|AlarmFilter:%s|Message:%s', ...
                            meterID, scrollMode, energyFormat, demandFormat, voltageFormat,
```

```
                                currentFormat, alarmFilter,
                                consumerMessage);
    writeline(t, updateMessage);
    fprintf('Sent updated configuration: %s\n', updateMessage);

    % Wait for acknowledgment from the meter
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
        fprintf('No response from meter.\n');
    end
end


function processWarningMessage(t, data)
    % Parse warning message
    fprintf('Processing warning message...\n');
    parsedData = split(data, '|');

    if length(parsedData) < 5
        fprintf('Error: Incomplete warning message received.\n');
        writeline(t, 'ERROR|Incomplete Warning Message');
        return;
    end

    meterID = extractAfter(parsedData{2}, 'MeterID:');
    currentLoad = str2double(extractAfter(parsedData{3}, '
       CurrentLoad:'));
    loadLimit = str2double(extractAfter(parsedData{4}, 'LoadLimit:
       '));
    destination = extractAfter(parsedData{5}, 'Destination:');
```

```

    % Log the warning
    fprintf('Warning from MeterID: %s\n', meterID);
    fprintf('Current Load: %d, Load Limit: %d\n', currentLoad, loadLimit);
    fprintf('Message sent to: %s\n', destination);

    % Acknowledge the warning message
    ackMessage = sprintf('ACK|Warning received from MeterID:%s', meterID);
    writeline(t, ackMessage);
    fprintf('Response sent: %s\n', ackMessage);
end
```

A.5 Head End System Simulation Code - Perform All Tests and Plot Variant

```
function simulateMeterAllTests(meter, hesIP, hesPort)
    % Create TCP/IP client
    t = tcpclient(hesIP, hesPort, 'Timeout', 30);

    % Initialize test results
    testResults = struct('Registration', false, 'Tariff', false, 'OnDemand', false, ...
                         'Periodic', false, 'Connection', false, 'ClockSync', false, ...
                         'Quality', false, 'LoadMgmt', false, 'Firmware', false, ...
                         'Supervision', false, 'ConsumerInfo', false, 'CommSupervision', false, ...
```

```
                                'Outage', false, 'RemoteConfig', false, 'Warning', false);

    % UC1 - Meter Registration Test
    if all(isfield(meter, {'deviceID', 'ipAddress', 'systemTitle'}))
        sendRegistrationNotification(t, meter);
        testResults.Registration = true;
    end

    % UC2 - Remote Tariff Programming Test
    if isfield(meter, 'consumerCategory')
        performRemoteTariffProgramming(t, meter);
        testResults.Tariff = true;
    end

    % UC3 - On-Demand Reading Test
    if all(isfield(meter, {'activeEnergyImport', 'activeEnergyExport', 'reactiveEnergyImport', 'reactiveEnergyExport'}))
        requestOnDemandReading(t, meter);
        testResults.OnDemand = true;
    end

    % UC4 - Periodic Reading Test
    if all(isfield(meter, {'cumulativeActiveImport', 'cumulativeActiveExport', 'rateActiveImportT1', 'rateActiveExportT1'}))
        sendPeriodicReadings(t, meter);
        testResults.Periodic = true;
    end

    % UC5 - Connection/Disconnection Test
```

```
    if all(isfield(meter, {'connectionState', 'instantCurrent'}))
        performConnectionDisconnection(t, meter);
        testResults.Connection = true;
    end

    % UC6 - Clock Synchronization Test
    if isfield(meter, 'localTime')
        performClockSynchronization(t, meter);
        testResults.ClockSync = true;
    end

    % UC7 - Quality of Supply Test
    if all(isfield(meter, {'instVoltageL1', 'instCurrentL1', 'instTotalActivePower', 'voltageSagL1', 'voltageSwellL1', 'powerFailureCount'}))
        performQualityOfSupplyReporting(t, meter);
        testResults.Quality = true;
    end

    % UC8 - Load Management by Relay Test
    if all(isfield(meter, {'activeDemandImport', 'relayStatus'}))
        performLoadManagementByRelay(t, meter);
        testResults.LoadMgmt = true;
    end

    % UC9 - Firmware Upgrade Test
    if isfield(meter, 'currentFirmwareVersion')
        performFirmwareUpgrade(t, meter);
        testResults.Firmware = true;
    end

    % UC10 - Meter Supervision Test
```

```
    if all(isfield(meter, {'tamperStatus', 'powerOutageCount', 'errorCode', 'unexpectedResetDetected'}))
        performMeterSupervision(t, meter);
        testResults.Supervision = true;
    end

    % UC11 - Consumer Information Test
    if all(isfield(meter, {'consumerName', 'consumerAddress', 'tariffCategory', 'accountNumber', 'emailAddress', 'phoneNumber', 'contractType', 'averageMonthlyConsumption', 'peakDemand', 'billingCycle', 'energyEfficiencyRating'}))
        performConsumerInformationUpdate(t, meter);
        testResults.ConsumerInfo = true;
    end

    % UC12 - Communication Supervision Test
    if all(isfield(meter, {'signalStrength', 'dataErrorRate', 'connectionUptime', 'communicationEvents'}))
        performCommunicationSupervision(t, meter);
        testResults.CommSupervision = true;
    end

    % UC13 - Outage Supervision Test
    if all(isfield(meter, {'outageNotificationEnabled', 'predefinedDestinationAddress'}))
        performOutageSupervision(t, meter);
        testResults.Outage = true;
    end

    % UC14 - Remote Parameter Configuration Test
    if all(isfield(meter, {'scrollDisplayMode', 'displayDataFormat', 'alarmDisplayFilter', 'consumerMessage'}))
        performRemoteParameterConfiguration(t, meter);
```

```
        testResults.RemoteConfig = true;
    end

    % UC15 - Warning Message Management Test
    if all(isfield(meter, {'warningDestinationAddress', 'loadLimit', 'currentLoad'}))
        performWarningMessageManagement(t, meter);
        testResults.Warning = true;
    end

    % Cancel TCP connection
    clear t;

    % Plot results
    plotTestResults(testResults);
    %plotTestResultsHeatmap(testResults);
    %plotTestResultsRadar(testResults);
    %plotTestResultsTable(testResults);
end

function plotTestResults(testResults)
    % Convert structure to a table for easy plotting
    resultsTable = struct2table(testResults);
    testNames = resultsTable.Properties.VariableNames;
    testOutcomes = double(table2array(resultsTable));  % Convert logical to double

    % Assign -1 to failed tests
    testOutcomes(testOutcomes == 0) = -1;

    % Create categorical array with the specified order
    testNamesCategorical = categorical(testNames, testNames, 'Ordinal', true);
```

```matlab

    % Create a figure for the results
    figure;
    b = bar(testNamesCategorical, testOutcomes, 'FaceColor', 'flat');

    % Set colors: green for passed (1), red for failed (-1)
    for i = 1:length(testOutcomes)
        if testOutcomes(i) == 1
            b.CData(i, :) = [0 1 0];  % Green
        elseif testOutcomes(i) == -1
            b.CData(i, :) = [1 0 0];  % Red
        end
    end

    % Adjust the y-axis to only show -1, 0, and 1
    ylim([-1.5, 1.5]);
    yticks([-1, 0, 1]);
    yticklabels({'Failed', '', 'Passed'});

    ylabel('Test Outcome');
    title('Simulation Test Results');
    set(gca, 'XTickLabelRotation', 45);
    grid on;
end



function plotTestResultsHeatmap(testResults)
    % Convert structure to a table for easy plotting
    resultsTable = struct2table(testResults);
    testNames = resultsTable.Properties.VariableNames;
```

```
    testOutcomes = double(table2array(resultsTable));  % Convert logical to double (1 or 0)

    % Prepare data for heatmap
    dataMatrix = testOutcomes;  % Use test outcomes directly as data matrix

    % Create vectors for X and Y axis labels
    xLabels = categorical(testNames);
    yLabels = {'Test Status'};  % Single row indicating status

    % Ensure dataMatrix is 2D for heatmap
    dataMatrix = reshape(dataMatrix, [1, length(dataMatrix)]);

    % Create a heatmap with the appropriate data
    figure;
    heatmap(xLabels, yLabels, dataMatrix, ...
        'Colormap', parula, 'ColorbarVisible', 'on', 'CellLabelColor','none');
    title('Heatmap of Test Results');
    xlabel('Test Cases');
    ylabel('Status');
end




function plotTestResultsRadar(testResults)
    % Convert structure to a table for easy plotting
    resultsTable = struct2table(testResults);
    testNames = resultsTable.Properties.VariableNames;
    testOutcomes = table2array(resultsTable);

```

```
    % Create a radar chart
    figure;
    polarplot(linspace(0, 2*pi, length(testOutcomes)+1), [testOutcomes, testOutcomes(1)], '-o');
    ax = gca;
    ax.ThetaTick = linspace(0, 360, length(testOutcomes)+1);
    ax.ThetaTickLabel = [testNames, testNames(1)];
    ax.RLim = [0, 1];
    title('Radar Chart of Test Results');
end

function plotTestResultsTable(testResults)
    % Convert structure to a table for easy display
    resultsTable = struct2table(testResults);

    % Display the results table
    disp('Test Results Summary:');
    disp(resultsTable);
end

% Function to send registration notification
function sendRegistrationNotification(t, meter)
    registrationMessage = sprintf('REGISTER|MeterID:%s|IP:%s|SystemTitle:%s', ...
                                  meter.deviceID, meter.ipAddress, meter.systemTitle);
    write(t, registrationMessage, "char");
    fprintf('Sent registration message: %s\n', registrationMessage);

    % Wait for and process response
    pause(1);
    if t.NumBytesAvailable > 0
```

```
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
        fprintf('No response from HES.\n');
    end
end

function performRemoteTariffProgramming(t, meter)
    % Define tariffs based on consumer category with valid field names
    tariffs = struct(...
        'Domestic', struct('lifelineRate', '0.05', 'rate', '797.3'), ...
        'Commercial', struct('rate', '791.9'), ...
        'Medium_Industrial', struct('rate', '502.9'), ...
        'Large_Industrial', struct('rate', '502.9'), ...
        'Street_Lighting', struct('rate', '370.0') ...
    );

    % Determine current tariff based on the consumer category
    currentCategory = strrep(meter.consumerCategory, ' ', '_'); % Replace spaces with underscores
    currentTariff = tariffs.(currentCategory);

    % Display available tariff options
    fprintf('Current tariff for %s is %s.\n', currentCategory, currentTariff.rate);
    fprintf('Select a new tariff rate from the options below:\n');
    availableRates = fieldnames(tariffs);

    for i = 1:length(availableRates)
        rate = tariffs.(availableRates{i}).rate;
        fprintf('%d. %s Rate: %s\n', i, availableRates{i}, rate);
```

```
    end

    % Get user selection
    choice = input('Enter the number corresponding to your choice: ');
    if choice < 1 || choice > length(availableRates)
        fprintf('Invalid choice. Using current rate.\n');
        newRate = currentTariff.rate;
    else
        newRate = tariffs.(availableRates{choice}).rate;
    end

    % Create message with current and new tariff information
    message = sprintf('TARIFF|MeterID:%s|Category:%s|CurrentRate:%s|NewRate:%s', ...
                      meter.deviceID, currentCategory, currentTariff.rate, newRate);

    % Only include LifelineRate for Domestic tariffs
    if strcmp(currentCategory, 'Domestic')
        message = sprintf('%s|LifelineRate:%s', message, currentTariff.lifelineRate);
    end

    write(t, message, "char");
    fprintf('Sent tariff change request: %s\n', message);

    % Allow some time for the server to process and respond
    pause(2); % Adjust this pause duration if necessary

    % Continuously check for response from HES
    totalWaitTime = 0;
```

```matlab
    while totalWaitTime < 20 % 20 seconds timeout, adjust as needed
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);
            break;
        else
            pause(0.5); % Wait a little before checking again
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function requestOnDemandReading(t, meter)
    % Create the on-demand reading request message including meter parameters
    message = sprintf('ON_DEMAND_READ|MeterID:%s|Request:OnDemand|ActiveImport:%.2f|ActiveExport:%.2f|ReactiveImport:%.2f|ReactiveExport:%.2f', ...
        meter.deviceID, meter.activeEnergyImport, meter.activeEnergyExport, ...
        meter.reactiveEnergyImport, meter.reactiveEnergyExport);

    % Send the request to the HES
    write(t, message, "char");
    fprintf('Sent on-demand reading request: %s\n', message);

    % Wait for the response from HES
    pause(2);
```

```matlab
    totalWaitTime = 0;
    while totalWaitTime < 20 % Timeout after 20 seconds
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received on-demand reading response: %s\n', response);
            break;
        else
            pause(0.5);
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function sendPeriodicReadings(t, meter)
    % Generate periodic reading data
    cumulativeImport = meter.cumulativeActiveImport;  % Active energy import cumulative
    cumulativeExport = meter.cumulativeActiveExport;  % Active energy export cumulative
    rateImportT1 = meter.rateActiveImportT1;          % Active energy import for rate T1
    rateExportT1 = meter.rateActiveExportT1;          % Active energy export for rate T1

    % Create message with periodic reading information
    message = sprintf('PERIODIC_READ|MeterID:%s|CumulativeImport:%.2f|CumulativeExport:%.2f|RateImportT1:%.2f|RateExportT1:%.2f', ...
```

```
                      meter.deviceID, cumulativeImport, cumulativeExport, rateImportT1, rateExportT1);

    % Send the message to the HES
    write(t, message, "char");
    fprintf('Sent periodic reading data: %s\n', message);

    % Wait for acknowledgment from HES
    pause(2);
    totalWaitTime = 0;
    while totalWaitTime < 20 % Timeout after 20 seconds
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received acknowledgment: %s\n', response);
            break;
        else
            pause(0.5);
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function performConnectionDisconnection(t, meter)
    % Define possible states
    states = struct('Disconnected', 0, 'Connected', 1, 'ReadyForReconnection', 2);

    % Read the initial meter state from the meter structure
```

```
    fprintf('Current meter state: %d\n', meter.connectionState);

    % Create message with current state and instant current information
    message = sprintf('CONNECTION|MeterID:%s|State:%d|InstantCurrent:%.2f', ...
                      meter.deviceID, meter.connectionState, meter.instantCurrent);
    write(t, message, "char");
    fprintf('Sent connection state and instant current info: %s\n', message);

    % Allow some time for the server to process and respond
    pause(2); % Adjust this pause duration if necessary

    % Continuously check for response from HES
    totalWaitTime = 0;
    while totalWaitTime < 20 % 20 seconds timeout, adjust as needed
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);

            % Parse and update meter state based on HES response
            if contains(response, 'ACK|State:')
                newState = extractBetween(response, 'ACK|State:', '|');
                meter.connectionState = str2double(newState);
                fprintf('Updated meter state to: %d\n', meter.connectionState);
            end
            break;
        else
```

```
            pause(0.5); % Wait a little before checking again
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function performClockSynchronization(t, meter)
    % Use the pre-initialized local time from the meter structure
    localTime = char(meter.localTime);  % Convert to string

    % Create message with current local time
    message = sprintf('CLOCK_SYNC|MeterID:%s|LocalTime:%s', meter.
       deviceID, localTime);
    write(t, message, "char");
    fprintf('Sent clock synchronization request: %s\n', message);

    % Allow some time for the server to process and respond
    pause(2);

    % Continuously check for response from HES
    totalWaitTime = 0;
    while totalWaitTime < 20  % 20 seconds timeout
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);

            % Parse and update meter time based on HES response
            if contains(response, 'ACK|NewTime:')
                newTime = extractAfter(response, 'ACK|NewTime:');
```

```
                meter.localTime = datetime(newTime, 'InputFormat', 'yyyy-MM-dd HH:mm:ss', 'TimeZone', 'local');
                fprintf('Updated meter time to: %s\n', newTime);
            end
            break;
        else
            pause(0.5);  % Wait before checking again
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No response from HES after waiting.\n');
    end
end

function performQualityOfSupplyReporting(t, meter)
    % Create a message with quality of supply data
    message = sprintf('QUALITY_REPORT|MeterID:%s|InstVoltageL1:%.2f|InstCurrentL1:%.2f|InstPower:%.2f|VoltageSagL1:%d|VoltageSwellL1:%d|PowerFailureCount:%d', ...
        meter.deviceID, meter.instVoltageL1, meter.instCurrentL1, meter.instTotalActivePower, meter.voltageSagL1, meter.voltageSwellL1, meter.powerFailureCount);
    write(t, message, "char");
    fprintf('Sent quality of supply report: %s\n', message);

    % Wait for acknowledgment from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
```

```
        fprintf('No response from HES.\n');
    end
end

function performLoadManagementByRelay(t, meter)
    % Monitor active demand and relay status
    currentDemand = meter.activeDemandImport;

    % Create a message to send current relay status and demand
    message = sprintf('LOAD_MANAGEMENT|MeterID:%s|CurrentDemand:%.2f|RelayStatus:%d', ...
        meter.deviceID, currentDemand, meter.relayStatus);
    write(t, message, "char");
    fprintf('Sent load management report: %s\n', message);

    % Wait for a response or command from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);

        % Process HES command
        if contains(response, 'COMMAND|Relay:ON')
            meter.relayStatus = 1; % Turn relay ON
            fprintf('Relay turned ON.\n');
        elseif contains(response, 'COMMAND|Relay:OFF')
            meter.relayStatus = 0; % Turn relay OFF
            fprintf('Relay turned OFF.\n');
        end
    else
        fprintf('No response from HES.\n');
    end
end
```

```
function performFirmwareUpgrade(t, meter)
    % Simulate downloading firmware
    fprintf('Starting firmware download...\n');
    pause(2); % Simulate download time
    firmwareVersion = sprintf('v%.1f', meter.currentFirmwareVersion + 0.1);

    % Create message to send firmware download completion and version
    message = sprintf('FIRMWARE_UPGRADE|MeterID:%s|NewVersion:%s|Status:Downloaded', ...
        meter.deviceID, firmwareVersion);
    write(t, message, "char");
    fprintf('Sent firmware upgrade status: %s\n', message);

    % Wait for verification response from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);

        % Check if the firmware is verified
        if contains(response, 'VERIFIED')
            % Activate new firmware
            meter.currentFirmwareVersion = str2double(extractAfter(firmwareVersion, 'v'));
            fprintf('Firmware version updated to %s\n', firmwareVersion);
        else
            fprintf('Firmware verification failed.\n');
        end
    else
```

```
        fprintf('No response from HES.\n');
    end
end

function performMeterSupervision(t, meter)
    % Simulate event detection
    tamperDetected = meter.tamperStatus;
    powerOutages = meter.powerOutageCount;
    errorCode = meter.errorCode;
    unexpectedReset = meter.unexpectedResetDetected;

    % Create message with meter supervision status
    message = sprintf('SUPERVISION_REPORT|MeterID:%s|Tamper:%d|
       PowerOutages:%d|ErrorCode:%d|UnexpectedReset:%d', ...
        meter.deviceID, tamperDetected, powerOutages, errorCode,
            unexpectedReset);
    write(t, message, "char");
    fprintf('Sent meter supervision report: %s\n', message);

    % Wait for acknowledgment from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);

        % Reset unexpected reset flag after reporting
        meter.unexpectedResetDetected = 0;
    else
        fprintf('No response from HES.\n');
    end
end

function performConsumerInformationUpdate(t, meter)
```

```
    % Create a message with consumer information
    message = sprintf('CONSUMER_INFO|MeterID:%s|ConsumerName:%s|ConsumerAddress:%s|TariffCategory:%s|AccountNumber:%s|EmailAddress:%s|PhoneNumber:%s|ContractType:%s|AvgMonthlyConsumption:%.2f|PeakDemand:%.2f|BillingCycle:%s|EnergyEfficiencyRating:%s', ...
        meter.deviceID, meter.consumerName, meter.consumerAddress, meter.tariffCategory, meter.accountNumber, meter.emailAddress, meter.phoneNumber, meter.contractType, meter.averageMonthlyConsumption, meter.peakDemand, meter.billingCycle, meter.energyEfficiencyRating);
    write(t, message, "char");
    fprintf('Sent consumer information: %s\n', message);

    % Wait for acknowledgment from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
        fprintf('No response from HES.\n');
    end
end

function performCommunicationSupervision(t, meter)
    % Select a random event from the initialized communication events
    eventId = randi([1, length(meter.communicationEvents)]);
    eventName = meter.communicationEvents{eventId};
    timestamp = datestr(now, 'yyyy-mm-dd HH:MM:SS');

    % Create a message with event information
```

```
    message = sprintf('COMM_EVENT|MeterID:%s|EventID:%d|EventName
       :%s|Timestamp:%s', ...
                      meter.deviceID, eventId, eventName,
                         timestamp);

    % Send message to HES
    write(t, message, "char");
    fprintf('Sent communication event: %s\n', message);

    % Wait for acknowledgment from HES
    pause(2);
    if t.NumBytesAvailable > 0
        response = char(read(t, t.NumBytesAvailable, "char"));
        fprintf('Received response: %s\n', response);
    else
        fprintf('No response from HES.\n');
    end
end

function performOutageSupervision(t, meter)
    % Check if outage notifications are enabled
    if meter.outageNotificationEnabled
        % Simulate an outage event
        outageEvent = 'Total Power Outage';
        timestamp = datestr(now, 'yyyy-mm-dd HH:MM:SS');

        % Create message for outage notification
        message = sprintf('OUTAGE|MeterID:%s|Event:%s|Timestamp:%s
           |Destination:%s', ...
                          meter.deviceID, outageEvent, timestamp,
                             meter.predefinedDestinationAddress);

        % Send outage notification to HES
```

```
        write(t, message, "char");
        fprintf('Sent outage notification: %s\n', message);

        % Wait for acknowledgment from HES
        pause(2);
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);
        else
            fprintf('No response from HES.\n');
        end
    else
        fprintf('Outage notification is disabled.\n');
    end
end

function performRemoteParameterConfiguration(t, meter)
    % Create a message with the current configuration
    message = sprintf('CONFIG|MeterID:%s|ScrollMode:%d|EnergyFormat:%d|DemandFormat:%d|VoltageFormat:%d|CurrentFormat:%d|AlarmFilter:%d|Message:%s', ...
        meter.deviceID, meter.scrollDisplayMode, meter.displayDataFormat.energy, ...
        meter.displayDataFormat.demand, meter.displayDataFormat.voltage, meter.displayDataFormat.current, ...
        meter.alarmDisplayFilter, meter.consumerMessage);

    % Send current configuration to HES
    write(t, message, "char");
    fprintf('Sent current configuration: %s\n', message);

    % Wait for configuration update from HES
    totalWaitTime = 0;
```

```matlab
    while totalWaitTime < 20 % 20 seconds timeout, adjust as needed
        if t.NumBytesAvailable > 0
            data = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received configuration update: %s\n', data);

            % Parse configuration update
            parsedData = split(data, '|');

            if length(parsedData) < 9
                fprintf('Error: Incomplete message received.\n');
                writeline(t, 'ERROR|Incomplete Configuration Message');
                return;
            end

            % Apply configuration update
            meter.scrollDisplayMode = str2double(extractAfter(parsedData{3}, 'ScrollMode:'));
            meter.displayDataFormat.energy = str2double(extractAfter(parsedData{4}, 'EnergyFormat:'));
            meter.displayDataFormat.demand = str2double(extractAfter(parsedData{5}, 'DemandFormat:'));
            meter.displayDataFormat.voltage = str2double(extractAfter(parsedData{6}, 'VoltageFormat:'));
            meter.displayDataFormat.current = str2double(extractAfter(parsedData{7}, 'CurrentFormat:'));
            meter.alarmDisplayFilter = str2double(extractAfter(parsedData{8}, 'AlarmFilter:'));
            meter.consumerMessage = extractAfter(parsedData{9}, 'Message:');

            fprintf('Updated Meter Configuration:\n');
```

```
            fprintf('Scroll Display Mode: %d seconds\n', meter.scrollDisplayMode);
            fprintf('Display Data Format: Energy=%d, Demand=%d, Voltage=%d, Current=%d\n', ...
                meter.displayDataFormat.energy, meter.displayDataFormat.demand, meter.displayDataFormat.voltage, meter.displayDataFormat.current);
            fprintf('Alarm Display Filter: %d\n', meter.alarmDisplayFilter);
            fprintf('Consumer Message: %s\n', meter.consumerMessage);

            % Send acknowledgment back to the HES
            ackMessage = sprintf('ACK|MeterID:%s|ConfigUpdateSuccess', meter.deviceID);
            writeline(t, ackMessage);
            fprintf('Response sent: %s\n', ackMessage);
            break;
        else
            pause(0.5); % Wait a little before checking again
            totalWaitTime = totalWaitTime + 0.5;
        end
    end

    if totalWaitTime >= 20
        fprintf('No configuration update received from HES after waiting.\n');
    end
end


function performWarningMessageManagement(t, meter)
```

```
    % Check current load against load limit
    if meter.currentLoad > meter.loadLimit
        % Create warning message
        warningMessage = sprintf('WARNING|MeterID:%s|CurrentLoad:%d|LoadLimit:%d|Destination:%s', ...
            meter.deviceID, meter.currentLoad, meter.loadLimit, meter.warningDestinationAddress);

        % Send warning message
        write(t, warningMessage, "char");
        fprintf('Sent warning message: %s\n', warningMessage);

        % Wait for acknowledgment from HES
        pause(2);
        if t.NumBytesAvailable > 0
            response = char(read(t, t.NumBytesAvailable, "char"));
            fprintf('Received response: %s\n', response);
        else
            fprintf('No acknowledgment received from HES.\n');
        end
    else
        fprintf('Load within limits. No warning sent.\n');
    end
end
```

A.6 HES Simulation Output - Meter 1

```
>> simulateHES('127.0.0.1', 502)
Simulated HES running on 127.0.0.1:502
Received message: REGISTER|MeterID:1234567890|IP:192.168.1.100|SystemTitle:MeterSystemTitle
Response sent: ACK|Registration Successful
```

```
Received message: ON_DEMAND_READ|MeterID:1234567890|Request:
    OnDemand|ActiveImport:952.36|ActiveExport:60.99|ReactiveImport
    :30.88|ReactiveExport:25.78
Processing on-demand reading...
MeterID: 1234567890
Active Import: 952.36 kWh
Active Export: 60.99 kWh
Reactive Import: 30.88 kVARh
Reactive Export: 25.78 kVARh
Response sent: ACK|MeterID:1234567890|ReadComplete|ActiveImport
    :952.36|ActiveExport:60.99|ReactiveImport:30.88|ReactiveExport
    :25.78
Received message: PERIODIC_READ|MeterID:1234567890|
    CumulativeImport:1024.94|CumulativeExport:115.34|RateImportT1
    :191.46|RateExportT1:12.00
Processing periodic reading data...
MeterID: 1234567890
Cumulative Import: 1024.94 kWh
Cumulative Export: 115.34 kWh
Rate Import T1: 191.46 kWh
Rate Export T1: 12.00 kWh
Periodic reading data processed and logged for billing.
Response sent: ACK|MeterID:1234567890|ReadProcessed
Received message: CLOCK_SYNC|MeterID:1234567890|LocalTime
    :2024-08-14 20:43:42
Processing clock synchronization...
Meter ID: 1234567890, Local Time: 2024-08-14 20:43:42
Setting new time to: 2024-08-14 20:39:59
Response sent: ACK|NewTime:2024-08-14 20:39:59
Received message: QUALITY_REPORT|MeterID:1234567890|InstVoltageL1
    :203.88|InstCurrentL1:4.95|InstPower:710.23|VoltageSagL1:10|
    VoltageSwellL1:4|PowerFailureCount:2
Processing quality of supply report...
```

```
Meter ID: 1234567890
Instantaneous Voltage L1: 203.88 V
Instantaneous Current L1: 4.95 A
Instantaneous Power: 710.23 W
Voltage Sag Count L1: 10
Voltage Swell Count L1: 4
Power Failure Count: 2
Warning: Voltage L1 out of normal range!
Alert: High number of voltage sags detected!
Response sent: ACK|QualityReportReceived
Received message: LOAD_MANAGEMENT|MeterID:1234567890|CurrentDemand
    :19.71|RelayStatus:0
Processing load management report...
Meter ID: 1234567890
Current Demand: 19.71 kW
Relay Status: 0
Demand is normal. Sending command to turn on relay.
Received message: FIRMWARE_UPGRADE|MeterID:1234567890|NewVersion:
    v1.1|Status:Downloaded
Processing firmware upgrade...
Meter ID: 1234567890
New Firmware Version: v1.1
Status: Downloaded
Verifying firmware...
Firmware verified successfully.
Received message: SUPERVISION_REPORT|MeterID:1234567890|Tamper:1|
    PowerOutages:0|ErrorCode:0|UnexpectedReset:1
Processing meter supervision report...
Meter ID: 1234567890
Tamper Detected: 1
Power Outages: 0
Error Code: 0
Unexpected Reset: 1
```

```
Alert: Tamper detected on meter 1234567890!
Alert: Unexpected reset detected on meter 1234567890!
Response sent: ACK|SupervisionReportReceived
Received message: CONSUMER_INFO|MeterID:1234567890|ConsumerName:John Kato|ConsumerAddress:456 Kampala Rd, Kampala|TariffCategory:Residential|AccountNumber:ACC123456789|EmailAddress:john.kato@example.com|PhoneNumber:+256701234567|ContractType:Standard Residential|AvgMonthlyConsumption:419.35|PeakDemand:4.45|BillingCycle:Monthly|EnergyEfficiencyRating:A
Processing consumer information...
Meter ID: 1234567890
Consumer Name: John Kato
Consumer Address: 456 Kampala Rd, Kampala
Tariff Category: Residential
Account Number: ACC123456789
Email Address: john.kato@example.com
Phone Number: +256701234567
Contract Type: Standard Residential
Average Monthly Consumption: 419.35 kWh
Peak Demand: 4.45 kW
Billing Cycle: Monthly
Energy Efficiency Rating: A
Response sent: ACK|ConsumerInfoReceived
Received message: COMM_EVENT|MeterID:1234567890|EventID:3|EventName:Start local communication|Timestamp:2024-08-14 20:40:13
Processing communication event...
Event ID: 3, Event Name: Start local communication, Timestamp: 2024-08-14 20:40:13
Logging event: ID=3, Name=Start local communication, Timestamp=2024-08-14 20:40:13
Response sent: ACK|EventID:3|Logged
```

```
Received message: OUTAGE|MeterID:1234567890|Event:Total Power
    Outage|Timestamp:2024-08-14 20:40:15|Destination:HES_MAIN
Processing outage notification...
Outage Event: Total Power Outage, Timestamp: 2024-08-14 20:40:15,
    Destination: HES_MAIN
Logging outage event: Event=Total Power Outage, Timestamp
    =2024-08-14 20:40:15, Destination=HES_MAIN
Response sent: ACK|Event:Total Power Outage|Logged
Received message: CONFIG|MeterID:1234567890|ScrollMode:1|
    EnergyFormat:1|DemandFormat:1|VoltageFormat:1|CurrentFormat:1|
    AlarmFilter:0|Message:Welcome
Processing configuration message from meter...
Current Meter Configuration:
MeterID: 1234567890
Scroll Mode: 1 seconds
Display Data Formats - Energy: 1, Demand: 1, Voltage: 1, Current:
    1
Alarm Display Filter: 0
Consumer Message: Welcome

Enter new configuration settings for the meter.
1. Scroll Display Mode Options:
   1. 10 seconds
   2. 30 seconds
   3. 60 seconds
   4. 240 seconds
   5. Deactivate (255)
Select a new scroll display mode (1-5): 1
2. Display Data Format Options:
   1. Format Type 1
   2. Format Type 2
   3. Format Type 3
   4. Format Type 4
```

```
Select a new energy display format (1-4): 2
Select a new demand display format (1-4): 3
Select a new voltage display format (1-4): 1
Select a new current display format (1-4): 2
3. Alarm Display Filter:
   0. Disabled
   1. Enabled
Select alarm display filter status (0 or 1): 1
Enter new consumer message: Hello
Sent updated configuration: CONFIG_UPDATE|MeterID:1234567890|
    ScrollMode:10|EnergyFormat:2|DemandFormat:3|VoltageFormat:1|
    CurrentFormat:2|AlarmFilter:1|Message:Hello
Received response: ACK|MeterID:1234567890|ConfigUpdateSuccess
```

A.7 Meter Simulation Output - Meter 1

```
Sent registration message: REGISTER|MeterID:1234567890|IP
    :192.168.1.100|SystemTitle:MeterSystemTitle
Received response: ACK|Registration Successful

Sent on-demand reading request: ON_DEMAND_READ|MeterID:1234567890|
    Request:OnDemand|ActiveImport:952.36|ActiveExport:60.99|
    ReactiveImport:30.88|ReactiveExport:25.78
Received on-demand reading response: ACK|MeterID:1234567890|
    ReadComplete|ActiveImport:952.36|ActiveExport:60.99|
    ReactiveImport:30.88|ReactiveExport:25.78

Sent periodic reading data: PERIODIC_READ|MeterID:1234567890|
    CumulativeImport:1024.94|CumulativeExport:115.34|RateImportT1
    :191.46|RateExportT1:12.00
Received acknowledgment: ACK|MeterID:1234567890|ReadProcessed

```

```
Sent clock synchronization request: CLOCK_SYNC|MeterID:1234567890|LocalTime:2024-08-14 20:43:42
Received response: ACK|NewTime:2024-08-14 20:39:59

Updated meter time to: 2024-08-14 20:39:59

Sent quality of supply report: QUALITY_REPORT|MeterID:1234567890|InstVoltageL1:203.88|InstCurrentL1:4.95|InstPower:710.23|VoltageSagL1:10|VoltageSwellL1:4|PowerFailureCount:2
Received response: ACK|QualityReportReceived

Sent load management report: LOAD_MANAGEMENT|MeterID:1234567890|CurrentDemand:19.71|RelayStatus:0
Received response: COMMAND|Relay:ON

Relay turned ON.
Starting firmware download...
Sent firmware upgrade status: FIRMWARE_UPGRADE|MeterID:1234567890|NewVersion:v1.1|Status:Downloaded
Received response: VERIFIED|Firmware upgrade verified.

Firmware version updated to v1.1
Sent meter supervision report: SUPERVISION_REPORT|MeterID:1234567890|Tamper:1|PowerOutages:0|ErrorCode:0|UnexpectedReset:1
Received response: ACK|SupervisionReportReceived

Sent consumer information: CONSUMER_INFO|MeterID:1234567890|ConsumerName:John Kato|ConsumerAddress:456 Kampala Rd, Kampala|TariffCategory:Residential|AccountNumber:ACC123456789|EmailAddress:john.kato@example.com|PhoneNumber:+256701234567|ContractType:Standard Residential|AvgMonthlyConsumption:419.35|PeakDemand:4.45|BillingCycle:Monthly|EnergyEfficiencyRating:A
```

```
Received response: ACK|ConsumerInfoReceived

Sent communication event: COMM_EVENT|MeterID:1234567890|EventID:3|EventName:Start local communication|Timestamp:2024-08-14 20:40:13
Received response: ACK|EventID:3|Logged

Sent outage notification: OUTAGE|MeterID:1234567890|Event:Total Power Outage|Timestamp:2024-08-14 20:40:15|Destination:HES_MAIN
Received response: ACK|Event:Total Power Outage|Logged

Sent current configuration: CONFIG|MeterID:1234567890|ScrollMode:1|EnergyFormat:1|DemandFormat:1|VoltageFormat:1|CurrentFormat:1|AlarmFilter:0|Message:Welcome
Received configuration update: CONFIG_UPDATE|MeterID:1234567890|ScrollMode:10|EnergyFormat:2|DemandFormat:3|VoltageFormat:1|CurrentFormat:2|AlarmFilter:1|Message:Hello

Updated Meter Configuration:
Scroll Display Mode: 10 seconds
Display Data Format: Energy=2, Demand=3, Voltage=1, Current=2
Alarm Display Filter: 1
Consumer Message: Hello

Response sent: ACK|MeterID:1234567890|ConfigUpdateSuccess
Load within limits. No warning sent.
```

A.8 HES Simulation Output - Meter 2

```
>> simulateHES('127.0.0.1', 502)
Simulated HES running on 127.0.0.1:502
```

```
Received message: REGISTER|MeterID:21564789|IP:192.168.1.100|
    SystemTitle:MeterSystemTitle
Response sent: ACK|Registration Successful
Received message: TARIFF|MeterID:21564789|Category:Domestic|
    CurrentRate:797.3|NewRate:502.9|LifelineRate:0.05
Processing tariff programming...
Parsed data:
     {'TARIFF'           }
     {'MeterID:21564789' }
     {'Category:Domestic'}
     {'CurrentRate:797.3'}
     {'NewRate:502.9'    }
     {'LifelineRate:0.05'}

Category: Domestic, Current Rate: 797.3, New Rate: 502.9, Lifeline
     Rate: 0.05
Tariff for category Domestic updated from 797.3 to 502.9.
Response sent: ACK|Category:Domestic|AppliedNewRate:502.9
Received message: PERIODIC_READ|MeterID:21564789|CumulativeImport
    :1128.30|CumulativeExport:10.54|RateImportT1:468.93|
    RateExportT1:13.46
Processing periodic reading data...
MeterID: 21564789
Cumulative Import: 1128.30 kWh
Cumulative Export: 10.54 kWh
Rate Import T1: 468.93 kWh
Rate Export T1: 13.46 kWh
Periodic reading data processed and logged for billing.
Response sent: ACK|MeterID:21564789|ReadProcessed
Received message: CONNECTION|MeterID:21564789|State:1|
    InstantCurrent:18.46
Processing connection/disconnection...
Current State: 1
```

```
Instant Current: 18.46 A
Select an action:
1. Connect the disconnected meter
2. Disconnect the connected meter
3. Instantly disconnect if exceeds threshold
Enter your choice (1-3): 1
Meter is already connected.
Response sent: ACK|State:1
Received message: CLOCK_SYNC|MeterID:21564789|LocalTime:2024-08-14
    21:23:04
Processing clock synchronization...
Meter ID: 21564789, Local Time: 2024-08-14 21:23:04
Setting new time to: 2024-08-14 21:22:51
Response sent: ACK|NewTime:2024-08-14 21:22:51
Received message: QUALITY_REPORT|MeterID:21564789|InstVoltageL1
   :227.48|InstCurrentL1:4.45|InstPower:1415.26|VoltageSagL1:3|
   VoltageSwellL1:4|PowerFailureCount:2
Processing quality of supply report...
Meter ID: 21564789
Instantaneous Voltage L1: 227.48 V
Instantaneous Current L1: 4.45 A
Instantaneous Power: 1415.26 W
Voltage Sag Count L1: 3
Voltage Swell Count L1: 4
Power Failure Count: 2
Response sent: ACK|QualityReportReceived
Received message: LOAD_MANAGEMENT|MeterID:21564789|CurrentDemand
   :20.39|RelayStatus:1
Processing load management report...
Meter ID: 21564789
Current Demand: 20.39 kW
Relay Status: 1
Demand is high. Sending command to turn off relay.
```

```
Received message: FIRMWARE_UPGRADE|MeterID:21564789|NewVersion:v1
    .1|Status:Downloaded
Processing firmware upgrade...
Meter ID: 21564789
New Firmware Version: v1.1
Status: Downloaded
Verifying firmware...
Firmware verification failed.
Received message: SUPERVISION_REPORT|MeterID:21564789|Tamper:1|
    PowerOutages:1|ErrorCode:0|UnexpectedReset:1
Processing meter supervision report...
Meter ID: 21564789
Tamper Detected: 1
Power Outages: 1
Error Code: 0
Unexpected Reset: 1
Alert: Tamper detected on meter 21564789!
Alert: 1 power outages reported by meter 21564789!
Alert: Unexpected reset detected on meter 21564789!
Response sent: ACK|SupervisionReportReceived
Received message: OUTAGE|MeterID:21564789|Event:Total Power Outage
    |Timestamp:2024-08-14 21:23:03|Destination:HES_MAIN
Processing outage notification...
Outage Event: Total Power Outage, Timestamp: 2024-08-14 21:23:03,
    Destination: HES_MAIN
Logging outage event: Event=Total Power Outage, Timestamp
    =2024-08-14 21:23:03, Destination=HES_MAIN
Response sent: ACK|Event:Total Power Outage|Logged
Received message: CONFIG|MeterID:21564789|ScrollMode:1|
    EnergyFormat:1|DemandFormat:1|VoltageFormat:1|CurrentFormat:1|
    AlarmFilter:0|Message:Welcome
Processing configuration message from meter...
Current Meter Configuration:
```

```
MeterID: 21564789
Scroll Mode: 1 seconds
Display Data Formats - Energy: 1, Demand: 1, Voltage: 1, Current: 1
Alarm Display Filter: 0
Consumer Message: Welcome

Enter new configuration settings for the meter.
1. Scroll Display Mode Options:
   1. 10 seconds
   2. 30 seconds
   3. 60 seconds
   4. 240 seconds
   5. Deactivate (255)
Select a new scroll display mode (1-5): 2
2. Display Data Format Options:
   1. Format Type 1
   2. Format Type 2
   3. Format Type 3
   4. Format Type 4
Select a new energy display format (1-4): 1
Select a new demand display format (1-4): 2
Select a new voltage display format (1-4): 3
Select a new current display format (1-4): 4
3. Alarm Display Filter:
   0. Disabled
   1. Enabled
Select alarm display filter status (0 or 1): 1
Enter new consumer message: Hello Jack
Sent updated configuration: CONFIG_UPDATE|MeterID:21564789|ScrollMode:30|EnergyFormat:1|DemandFormat:2|VoltageFormat:3|CurrentFormat:4|AlarmFilter:1|Message:Hello Jack
Received response: ACK|MeterID:21564789|ConfigUpdateSuccess
```

A.9 Meter Simulation Output - Meter 2

```
Sent registration message: REGISTER|MeterID:21564789|IP
    :192.168.1.100|SystemTitle:MeterSystemTitle
Received response: ACK|Registration Successful

Current tariff for Domestic is 797.3.
Select a new tariff rate from the options below:
1. Domestic Rate: 797.3
2. Commercial Rate: 791.9
3. Medium_Industrial Rate: 502.9
4. Large_Industrial Rate: 502.9
5. Street_Lighting Rate: 370.0
Enter the number corresponding to your choice: 3
Sent tariff change request: TARIFF|MeterID:21564789|Category:
    Domestic|CurrentRate:797.3|NewRate:502.9|LifelineRate:0.05
Received response: ACK|Category:Domestic|AppliedNewRate:502.9

Sent periodic reading data: PERIODIC_READ|MeterID:21564789|
    CumulativeImport:1128.30|CumulativeExport:10.54|RateImportT1
    :468.93|RateExportT1:13.46
Received acknowledgment: ACK|MeterID:21564789|ReadProcessed

Current meter state: 1
Sent connection state and instant current info: CONNECTION|MeterID
    :21564789|State:1|InstantCurrent:18.46
Received response: ACK|State:1

Updated meter state to:
```

```
Sent clock synchronization request: CLOCK_SYNC|MeterID:21564789|LocalTime:2024-08-14 21:23:04
Received response: ACK|NewTime:2024-08-14 21:22:51

Updated meter time to: 2024-08-14 21:22:51

Sent quality of supply report: QUALITY_REPORT|MeterID:21564789|InstVoltageL1:227.48|InstCurrentL1:4.45|InstPower:1415.26|VoltageSagL1:3|VoltageSwellL1:4|PowerFailureCount:2
Received response: ACK|QualityReportReceived

Sent load management report: LOAD_MANAGEMENT|MeterID:21564789|CurrentDemand:20.39|RelayStatus:1
Received response: COMMAND|Relay:OFF

Relay turned OFF.
Starting firmware download...
Sent firmware upgrade status: FIRMWARE_UPGRADE|MeterID:21564789|NewVersion:v1.1|Status:Downloaded
Received response: FAILED|Firmware verification failed.

Firmware verification failed.
Sent meter supervision report: SUPERVISION_REPORT|MeterID:21564789|Tamper:1|PowerOutages:1|ErrorCode:0|UnexpectedReset:1
Received response: ACK|SupervisionReportReceived

Sent outage notification: OUTAGE|MeterID:21564789|Event:Total Power Outage|Timestamp:2024-08-14 21:23:03|Destination:HES_MAIN
Received response: ACK|Event:Total Power Outage|Logged

Sent current configuration: CONFIG|MeterID:21564789|ScrollMode:1|EnergyFormat:1|DemandFormat:1|VoltageFormat:1|CurrentFormat:1|AlarmFilter:0|Message:Welcome
```

```
Received configuration update: CONFIG_UPDATE|MeterID:21564789|
    ScrollMode:30|EnergyFormat:1|DemandFormat:2|VoltageFormat:3|
    CurrentFormat:4|AlarmFilter:1|Message:Hello Jack

Updated Meter Configuration:
Scroll Display Mode: 30 seconds
Display Data Format: Energy=1, Demand=2, Voltage=3, Current=4
Alarm Display Filter: 1
Consumer Message: Hello Jack

Response sent: ACK|MeterID:21564789|ConfigUpdateSuccess
Load within limits. No warning sent.
```

A.10 HES Simulation Output - Meter 3

```
>> simulateHES('127.0.0.1', 502)
Simulated HES running on 127.0.0.1:502
Received message: REGISTER|MeterID:1541874|IP:192.168.1.100|
    SystemTitle:MeterSystemTitle
Response sent: ACK|Registration Successful
Received message: TARIFF|MeterID:1541874|Category:Domestic|
    CurrentRate:797.3|NewRate:370.0|LifelineRate:0.05
Processing tariff programming...
Parsed data:
    {'TARIFF'           }
    {'MeterID:1541874'  }
    {'Category:Domestic'}
    {'CurrentRate:797.3'}
    {'NewRate:370.0'    }
    {'LifelineRate:0.05'}

```

```
Category: Domestic, Current Rate: 797.3, New Rate: 370.0, Lifeline
     Rate: 0.05
Tariff for category Domestic updated from 797.3 to 370.0.
Response sent: ACK|Category:Domestic|AppliedNewRate:370.0
Received message: ON_DEMAND_READ|MeterID:1541874|Request:OnDemand|
    ActiveImport:545.41|ActiveExport:26.65|ReactiveImport:7.68|
    ReactiveExport:8.43
Processing on-demand reading...
MeterID: 1541874
Active Import: 545.41 kWh
Active Export: 26.65 kWh
Reactive Import: 7.68 kVARh
Reactive Export: 8.43 kVARh
Response sent: ACK|MeterID:1541874|ReadComplete|ActiveImport
    :545.41|ActiveExport:26.65|ReactiveImport:7.68|ReactiveExport
    :8.43
Received message: PERIODIC_READ|MeterID:1541874|CumulativeImport
    :696.08|CumulativeExport:105.43|RateImportT1:328.71|
    RateExportT1:43.77
Processing periodic reading data...
MeterID: 1541874
Cumulative Import: 696.08 kWh
Cumulative Export: 105.43 kWh
Rate Import T1: 328.71 kWh
Rate Export T1: 43.77 kWh
Periodic reading data processed and logged for billing.
Response sent: ACK|MeterID:1541874|ReadProcessed
Received message: CONNECTION|MeterID:1541874|State:1|
    InstantCurrent:20.36
Processing connection/disconnection...
Current State: 1
Instant Current: 20.36 A
Select an action:
```

```
1. Connect the disconnected meter
2. Disconnect the connected meter
3. Instantly disconnect if exceeds threshold
Enter your choice (1-3): 2
Disconnecting meter...
Response sent: ACK|State:0
Received message: CLOCK_SYNC|MeterID:1541874|LocalTime:2024-08-15
    00:56:05
Processing clock synchronization...
Meter ID: 1541874, Local Time: 2024-08-15 00:56:05
Setting new time to: 2024-08-15 00:51:52
Response sent: ACK|NewTime:2024-08-15 00:51:52
Received message: QUALITY_REPORT|MeterID:1541874|InstVoltageL1
    :223.12|InstCurrentL1:8.61|InstPower:955.38|VoltageSagL1:7|
    VoltageSwellL1:1|PowerFailureCount:2
Processing quality of supply report...
Meter ID: 1541874
Instantaneous Voltage L1: 223.12 V
Instantaneous Current L1: 8.61 A
Instantaneous Power: 955.38 W
Voltage Sag Count L1: 7
Voltage Swell Count L1: 1
Power Failure Count: 2
Alert: High number of voltage sags detected!
Response sent: ACK|QualityReportReceived
Received message: LOAD_MANAGEMENT|MeterID:1541874|CurrentDemand
    :21.95|RelayStatus:0
Processing load management report...
Meter ID: 1541874
Current Demand: 21.95 kW
Relay Status: 0
Demand is high. Sending command to turn off relay.
```

```
Received message: FIRMWARE_UPGRADE|MeterID:1541874|NewVersion:v1
    .1|Status:Downloaded
Processing firmware upgrade...
Meter ID: 1541874
New Firmware Version: v1.1
Status: Downloaded
Verifying firmware...
Firmware verified successfully.
Received message: SUPERVISION_REPORT|MeterID:1541874|Tamper:0|
    PowerOutages:1|ErrorCode:2|UnexpectedReset:1
Processing meter supervision report...
Meter ID: 1541874
Tamper Detected: 0
Power Outages: 1
Error Code: 2
Unexpected Reset: 1
Alert: 1 power outages reported by meter 1541874!
Error: Communication failure with meter 1541874.
Alert: Unexpected reset detected on meter 1541874!
Response sent: ACK|SupervisionReportReceived
Received message: CONSUMER_INFO|MeterID:1541874|ConsumerName:
    Mazinger|ConsumerAddress:420 High Road, Kampala|TariffCategory:
    Residential|AccountNumber:ACC123456789|EmailAddress:mazinger.z@
    example.com|PhoneNumber:+257701234567|ContractType:Standard
    Residential|AvgMonthlyConsumption:468.89|PeakDemand:4.34|
    BillingCycle:Monthly|EnergyEfficiencyRating:A
Processing consumer information...
Meter ID: 1541874
Consumer Name: Mazinger
Consumer Address: 420 High Road, Kampala
Tariff Category: Residential
Account Number: ACC123456789
Email Address: mazinger.z@example.com
```

```
Phone Number: +257701234567
Contract Type: Standard Residential
Average Monthly Consumption: 468.89 kWh
Peak Demand: 4.34 kW
Billing Cycle: Monthly
Energy Efficiency Rating: A
Response sent: ACK|ConsumerInfoReceived
Received message: COMM_EVENT|MeterID:1541874|EventID:4|EventName:
    End local communication|Timestamp:2024-08-15 00:52:06
Processing communication event...
Event ID: 4, Event Name: End local communication, Timestamp:
    2024-08-15 00:52:06
Logging event: ID=4, Name=End local communication, Timestamp
    =2024-08-15 00:52:06
Response sent: ACK|EventID:4|Logged
Received message: OUTAGE|MeterID:1541874|Event:Total Power Outage|
    Timestamp:2024-08-15 00:52:08|Destination:HES_MAIN
Processing outage notification...
Outage Event: Total Power Outage, Timestamp: 2024-08-15 00:52:08,
    Destination: HES_MAIN
Logging outage event: Event=Total Power Outage, Timestamp
    =2024-08-15 00:52:08, Destination=HES_MAIN
Response sent: ACK|Event:Total Power Outage|Logged
Received message: CONFIG|MeterID:1541874|ScrollMode:1|EnergyFormat
    :1|DemandFormat:1|VoltageFormat:1|CurrentFormat:1|AlarmFilter
    :0|Message:Welcome
Processing configuration message from meter...
Current Meter Configuration:
MeterID: 1541874
Scroll Mode: 1 seconds
Display Data Formats - Energy: 1, Demand: 1, Voltage: 1, Current:
    1
Alarm Display Filter: 0
```

```
Consumer Message: Welcome

Enter new configuration settings for the meter.
1. Scroll Display Mode Options:
   1. 10 seconds
   2. 30 seconds
   3. 60 seconds
   4. 240 seconds
   5. Deactivate (255)
Select a new scroll display mode (1-5): 4
2. Display Data Format Options:
   1. Format Type 1
   2. Format Type 2
   3. Format Type 3
   4. Format Type 4
Select a new energy display format (1-4): 4
Select a new demand display format (1-4): 3
Select a new voltage display format (1-4): 2
Select a new current display format (1-4): 1
3. Alarm Display Filter:
   0. Disabled
   1. Enabled
Select alarm display filter status (0 or 1): 1
Enter new consumer message: Arms out
Sent updated configuration: CONFIG_UPDATE|MeterID:1541874|
    ScrollMode:240|EnergyFormat:4|DemandFormat:3|VoltageFormat:2|
    CurrentFormat:1|AlarmFilter:1|Message:Arms out
Received response: ACK|MeterID:1541874|ConfigUpdateSuccess
```

A.11 Meter Simulation Output - Meter 3

```
Sent registration message: REGISTER|MeterID:1541874|IP:192.168.1.100|SystemTitle:MeterSystemTitle
Received response: ACK|Registration Successful

Current tariff for Domestic is 797.3.
Select a new tariff rate from the options below:
1. Domestic Rate: 797.3
2. Commercial Rate: 791.9
3. Medium_Industrial Rate: 502.9
4. Large_Industrial Rate: 502.9
5. Street_Lighting Rate: 370.0
Enter the number corresponding to your choice: 5
Sent tariff change request: TARIFF|MeterID:1541874|Category:Domestic|CurrentRate:797.3|NewRate:370.0|LifelineRate:0.05
Received response: ACK|Category:Domestic|AppliedNewRate:370.0

Sent on-demand reading request: ON_DEMAND_READ|MeterID:1541874|Request:OnDemand|ActiveImport:545.41|ActiveExport:26.65|ReactiveImport:7.68|ReactiveExport:8.43
Received on-demand reading response: ACK|MeterID:1541874|ReadComplete|ActiveImport:545.41|ActiveExport:26.65|ReactiveImport:7.68|ReactiveExport:8.43

Sent periodic reading data: PERIODIC_READ|MeterID:1541874|CumulativeImport:696.08|CumulativeExport:105.43|RateImportT1:328.71|RateExportT1:43.77
Received acknowledgment: ACK|MeterID:1541874|ReadProcessed

Current meter state: 1
Sent connection state and instant current info: CONNECTION|MeterID:1541874|State:1|InstantCurrent:20.36
Received response: ACK|State:0
```

```

Updated meter state to:
Sent clock synchronization request: CLOCK_SYNC|MeterID:1541874|LocalTime:2024-08-15 00:56:05
Received response: ACK|NewTime:2024-08-15 00:51:52

Updated meter time to: 2024-08-15 00:51:52

Sent quality of supply report: QUALITY_REPORT|MeterID:1541874|InstVoltageL1:223.12|InstCurrentL1:8.61|InstPower:955.38|VoltageSagL1:7|VoltageSwellL1:1|PowerFailureCount:2
Received response: ACK|QualityReportReceived

Sent load management report: LOAD_MANAGEMENT|MeterID:1541874|CurrentDemand:21.95|RelayStatus:0
Received response: COMMAND|Relay:OFF

Relay turned OFF.
Starting firmware download...
Sent firmware upgrade status: FIRMWARE_UPGRADE|MeterID:1541874|NewVersion:v1.1|Status:Downloaded
Received response: VERIFIED|Firmware upgrade verified.

Firmware version updated to v1.1
Sent meter supervision report: SUPERVISION_REPORT|MeterID:1541874|Tamper:0|PowerOutages:1|ErrorCode:2|UnexpectedReset:1
Received response: ACK|SupervisionReportReceived

Sent consumer information: CONSUMER_INFO|MeterID:1541874|ConsumerName:Mazinger|ConsumerAddress:420 High Road, Kampala|TariffCategory:Residential|AccountNumber:ACC123456789|EmailAddress:mazinger.z@example.com|PhoneNumber:+257701234567|ContractType:Standard Residential|AvgMonthlyConsumption:468.89|
```

```
PeakDemand:4.34|BillingCycle:Monthly|EnergyEfficiencyRating:A
Received response: ACK|ConsumerInfoReceived

Sent communication event: COMM_EVENT|MeterID:1541874|EventID:4|EventName:End local communication|Timestamp:2024-08-15 00:52:06
Received response: ACK|EventID:4|Logged

Sent outage notification: OUTAGE|MeterID:1541874|Event:Total Power Outage|Timestamp:2024-08-15 00:52:08|Destination:HES_MAIN
Received response: ACK|Event:Total Power Outage|Logged

Sent current configuration: CONFIG|MeterID:1541874|ScrollMode:1|EnergyFormat:1|DemandFormat:1|VoltageFormat:1|CurrentFormat:1|AlarmFilter:0|Message:Welcome
Received configuration update: CONFIG_UPDATE|MeterID:1541874|ScrollMode:240|EnergyFormat:4|DemandFormat:3|VoltageFormat:2|CurrentFormat:1|AlarmFilter:1|Message:Arms out

Updated Meter Configuration:
Scroll Display Mode: 240 seconds
Display Data Format: Energy=4, Demand=3, Voltage=2, Current=1
Alarm Display Filter: 1
Consumer Message: Arms out

Response sent: ACK|MeterID:1541874|ConfigUpdateSuccess
Load within limits. No warning sent.
```

Appendix B

Sustainable Development Goals

The implementation of smart metering technology in Uganda, particularly through the UMEME OneSait project, aligns with global efforts to promote sustainability, innovation, and development. This project contributes significantly to the realization of the United Nations Sustainable Development Goals (SDGs) [58]. Specifically, it plays a vital role in advancing two of the SDGs: Goal 7, "Affordable and Clean Energy," and Goal 9, "Industry, Innovation, and Infrastructure."

B.1 Goal 7: Affordable and Clean Energy

Goal 7 aims to "ensure access to affordable, reliable, sustainable, and modern energy for all [58]." In many developing nations, access to electricity remains limited, particularly in rural areas. Uganda, like many other African countries, has struggled with providing widespread access to electricity. The OneSait smart metering project aligns closely with this goal by modernizing the country's electricity distribution infrastructure, contributing to the increased availability and reliability of energy.

Smart metering enables more efficient monitoring and management of energy resources, which is crucial for both utility companies and consumers. By offering real-time data on energy consumption, smart meters help to optimize the distribution of electricity, reducing losses and ensuring that power is supplied where it is needed most. This contributes to the affordability of electricity by minimizing waste and allowing for more precise billing, ensuring consumers only pay for what they use. It also facilitates demand-side management, which helps in load balancing and minimizing power outages, particularly in areas with underdeveloped infrastructure.

Additionally, the smart meters deployed in this project provide a pathway for integrating renewable energy sources, such as solar power, into the grid. Uganda's energy mix, heavily reliant on hydropower, is increasingly incorporating other renewable energy sources like solar, which contributes to the country's sustainability efforts. Smart meters can help track the contribution of these renewable sources and provide data that can be used to further optimize the generation and consumption of clean energy. In doing so, the project aligns with the objective of achieving universal access to affordable and clean energy, as outlined in SDG 7.

B.2 Goal 9: Industry, Innovation, and Infrastructure

Goal 9 emphasizes the importance of building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation [58]. The One-Sait project is a prime example of how innovative technology can improve infrastructure, particularly in the utility sector. The transition from traditional meters to smart meters represents a significant innovation in how energy consumption is measured, billed, and managed.

By implementing IDIS-compliant smart meters, the project enhances the resilience of Uganda's electricity infrastructure. One key benefit of smart metering is its ability to detect outages and provide real-time data to utility companies, allowing for faster response times and more efficient repairs. This is particularly important in developing nations, where infrastructure is often vulnerable to disruptions. The smart meters also support the implementation of more sophisticated grid management techniques, such as time-of-use pricing, which encourages consumers to shift their energy consumption to off-peak times, thereby reducing strain on the grid during peak hours.

Moreover, the project fosters sustainable industrialization by improving the efficiency of electricity distribution and consumption. Industries rely on a stable and reliable power supply to function effectively, and smart meters contribute to this by ensuring that electricity is used efficiently and that outages are minimized. The project also opens up opportunities for further innovation within the energy sector, including the integration of more advanced technologies such as artificial intelligence and machine learning, which could be used to optimize grid management and predict maintenance needs.

Finally, this project exemplifies how partnerships between public utilities, private companies, and international organizations can drive innovation and infrastructure development in developing nations. By collaborating with IDIS and adhering to international standards, UMEME and its partners are ensuring that Uganda's electricity infrastructure is built to a high standard, capable of supporting future growth and development in the energy sector. This directly contributes to the achievement of Goal 9 by promoting the development of resilient and sustainable infrastructure that can support long-term industrialization and innovation.

Bibliography

[1] C. L. Wang, J. A. Shih, I. E. Liao, and C. T. Chien, “An Evaluation of Cybersecurity Risks of DLMS/COSEM Smart Meter Using Fuzzing Testing.” Institute of Electrical and Electronics Engineers Inc., 2022.

[2] D. U. Association, “EXCERPT FROM Companion Specification for Energy Metering DLMS/COSEM Architecture and Protocols DLMS User Association,” 2020.

[3] H. Mendes, I. Medeiros, and N. Neves, “Validating and Securing DLMS/COSEM Implementations with the ValiDLMS Framework.” Institute of Electrical and Electronics Engineers Inc., 7 2018, pp. 179–184.

[4] IDIS, “Package 3 IP Profile X (extended functionality) Edition 2.0,” 2023. [Online]. Available: https://www.idis-association.com/downloads

[5] "Uganda's Electricity Maximum Demand". Electricity Regulatory Authority (ERA). (accessed 2024-06-19). [Online]. Available: https://www.era.go.ug/index.php/stats/transmission-stats/maximum-demand

[6] "Access to Electricity in Uganda". World Bank Group. (accessed 2024-06-19). [Online]. Available: https://data.worldbank.org/indicator/EG.ELC.ACCS.ZS?end=2021&locations=UG&start=2000&view=chart

[7] G. Okoboi and J. Mawejje, “Electricity peak demand in Uganda: insights and foresight,” *Energy, Sustainability and Society*, vol. 6, 12 2016.

[8] T. J. Ngcobo and F. Ghayoor, "An overview of DLMS/COSEM and g3-plc for smart metering applications," *International Journal on Smart Sensing and Intelligent Systems*, vol. 15, 1 2022.

[9] G. Štruklec and J. Marši, *Implementing DLMS/COSEM in Smart Meters.* IEEE, 2011.

[10] A. Sahu and A. Goulart, *Implementation of a C-UNB Module for NS-3 and Validation for DLMS-COSEM Application Layer Protocol*, 2019.

[11] N. Rodríguez-Pérez, J. M. Domingo, and G. L. López, "ICT Scalability and Replicability Analysis for Smart Grids: Methodology and Application," *Energies*, vol. 17, 2 2024.

[12] A. Mohammadali, M. H. Tadayon, and M. Asadian, *A New Key management for AMI Systems Based on DLMS/COSEM Standard*, 2014.

[13] T. Lieskovan, J. Hajny, and P. Cika, *Smart Grid Security: Survey and Challenges*, 2019.

[14] A. Kheaksong and W. Lee, *Packet Transfer of DLMS/COSEM Standards for Smart Grid*, 2014.

[15] C. A. Wulfing, F. G. Reck, F. G. Carloto, C. H. Barriquello, P. R. Marin, and E. Nascimento, "Evaluation of dlms/cosem data processing setups applied to smart metering." Institute of Electrical and Electronics Engineers Inc., 2022.

[16] S. Feuerhahn, M. Zillgith, C. Wittwer, and C. Wietfeld, *Comparison of the Communication Protocols DLMS/COSEM, SML and IEC 61850 for Smart Metering Applications.* IEEE, 2011.

[17] C. T. Chien, C. L. Wang, I. E. Liao, and S. J. Wang, "Implementing OIML R46 Communication Unit for DLMS/COSEM Security Suite 1 and Passing CTT V3.1 Test." Institute of Electrical and Electronics Engineers Inc., 2023.

[18] IDIS, “IDIS 2018 Leaflet,” 2018. [Online]. Available: https://www.idis-association.com/downloads

[19] "The IDIS Standard". IDIS Association. (accessed 2024-05-27). [Online]. Available: https://www.idis-association.com/about-1

[20] "Qualitative Testing of DLMS/COSEM ICS compliant Energy Meters". Electrical India. (accessed 2024-05-14). [Online]. Available: https://www.electricalindia.in/qualitative-testing-of-dlms-cosem-ics-compliant-energy-meters/

[21] International Energy Agency (IEA), “Energy Policy Review Uganda 2023,” 2023. [Online]. Available: www.iea.org

[22] "Uganda’s Electricity Sector Overview". Electricity Regulatory Authority (ERA). (accessed 2024-06-06). [Online]. Available: https://www.era.go.ug/index.php/sector-overview/uganda-electricity-sector

[23] "Uganda’s Licesing Procedure". Electricity Regulatory Authority (ERA). (accessed 2024-06-11). [Online]. Available: https://www.era.go.ug/index.php/licensing/licensing/licensing-procedure

[24] Electricity Regulatory Authority (ERA), “Manual for the ERA Tariff Model,” 2012.

[25] ——, “Quarterly Tariff Adjustment Methodology,” 2021.

[26] "Consumer Price Index Excel Tables". Uganda Bureau of Statistics. (accessed 2024-06-11). [Online]. Available: https://www.ubos.org/explore-statistics/0/

[27] Electricity Regulatory Authority (ERA), “The ERA Compliance and Enforcement Manual,” 2019. [Online]. Available: https://www.era.go.ug/index.php/resource-centre/regulatory-instruments/guidelines-and-standards

[28] "Uganda’s Electricity Policies". Electricity Regulatory Authority (ERA). (accessed 2024-06-12). [Online]. Available: https://www.era.go.ug/index.php/resource-centre/regulatory-instruments/policies

[29] "Uganda's Electricity Consumers Protection and Complaints Handling". Electricity Regulatory Authority (ERA). (accessed 2024-06-12). [Online]. Available: https://www.era.go.ug/index.php/consumers/complaints-handling

[30] "Uganda's Electricity Regulations and Codes". Electricity Regulatory Authority (ERA). (accessed 2024-06-12). [Online]. Available: https://www.era.go.ug/index.php/resource-centre/regulatory-instruments/regulations-codes

[31] Electricity Regulatory Authority (ERA), "Electricity (installation permits) regulations 2018," 2018.

[32] "Uganda's Electricity Feed-In-Tariff". Electricity Regulatory Authority (ERA). (accessed 2024-06-12). [Online]. Available: https://www.era.go.ug/index.php/tariffs/generation-tariffs/feed-in-tariff

[33] Electricity Regulatory Authority (ERA), "Quality service standards," 2017.

[34] "Uganda's Executive Summary 2023". International Energy Agency (IEA). (accessed 2024-06-12). [Online]. Available: https://www.iea.org/reports/uganda-2023/executive-summary

[35] "Uganda's GDP Annual Growth". World Bank Group. (accessed 2024-06-19). [Online]. Available: https://data.worldbank.org/indicator/NY.GDP.MKTP.KD.ZG?locations=UG

[36] "Uganda's Urban Population". World Bank Group. (accessed 2024-06-19). [Online]. Available: https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS?locations=UG

[37] GET.transform, "Success in Rural Electrification Regulatory Case Studies UGANDA A Bundled Approach to Mini-Grid Tendering," 2020. [Online]. Available: www.get-transform.eu

[38] "Uganda Electricity Demand to Triple by 2023". Africa Uganda Business Travel Guide. (accessed 2024-06-19). [Online]. Available: https://www.africa-uganda-business-travel-guide.com/uganda-electricity-demand-to-triple-by-2023.html

[39] "Access to Electricity in Uganda (Rural)". World Bank Group. (accessed 2024-06-19). [Online]. Available: https://data.worldbank.org/indicator/EG.ELC.ACCS.RU.ZS?locations=UG

[40] Electricity Regulatory Authority (ERA), "SCHEDULE OF BASE END-USER TARIFFS APPLICABLE FOR THE SUPPLY OF ELECTRICITY BY UMEME LIMITED FOR THE YEAR 2024," 2024. [Online]. Available: www.era.go.ug

[41] "Our Profile - UMEME". UMEME. (accessed 2024-06-20). [Online]. Available: https://www.umeme.co.ug/about

[42] "What does UMEME exit mean for electricity consumers in Uganda?". EPRC. (accessed 2024-07-12). [Online]. Available: https://eprcug.org/eprc-highlights/what-does-umeme-exit-mean-for-electricity-consumers-in-uganda/

[43] "Daily Express on UMEME". Daily Express. (accessed 2024-07-21). [Online]. Available: https://dailyexpress.co.ug/2024/07/21/umeme-exit-will-need-govt-to-hire-over-3000-workers-for-electricity-distribution-uedcl-boss/

[44] "Power Struggles: Uganda's Uneven Electricity Distribution". NilePost. (accessed 2024-07-21). [Online]. Available: https://nilepost.co.ug/uganda/202469/power-struggles-ugandas-uneven-electricity-distribution

[45] "Achievements and Challenges of Uganda's Power Sector". RMI. (accessed 2024-07-21). [Online]. Available: https://rmi.org/achievements-and-challenges-of-ugandas-power-sector/

[46] "Powering Uganda to 52,000MW: The challenges and prospects". Monitor. (accessed 2024-07-22). [Online]. Available: https://www.monitor.co.ug/uganda/oped/commentary/powering-uganda-to-52-000mw-the-challenges-and-prospects-4471808

[47] Minsait, “Utility Customer Information System - Technical Proposal for UMEME,” 2024.

[48] "Prepayment Metering". UEDCL. (accessed 2024-07-11). [Online]. Available: https://www.uedcl.co.ug/prepayment-metering/

[49] "Umeme starts upgrade of Yaka meters". UG Standard. (accessed 2024-07-11). [Online]. Available: https://www.ugstandard.com/umeme-starts-upgrade-of-yak a-meters/

[50] "Umeme embarks on upgrade of Yaka meters". KIKUBO LANE. (accessed 2024-07-11). [Online]. Available: https://kikubolane.com/2024/01/08/umeme-emb arks-on-upgrade-of-yaka-meters/

[51] "Prepayment". Itron. (accessed 2024-07-11). [Online]. Available: https://es.itron.c om/es/categories/prepayment

[52] M. Onesait, “A World Class Customer Management Solution Supporting the future of utilities.” [Online]. Available: www.acspower.com

[53] H. Hayes, “The new metering requirements for South Africa,” 2019.

[54] IDIS, “IDIS Package 3 Specification Summary,” 2023.

[55] S. G. Hoffmann, R. Massink, and G. Bumiller, “New Security Features in DLMS/COSEM-a Comparison to the Smart Meter Gateway.”

[56] D. Kohout, T. Lieskovan, and P. Mlynek, “Smart Metering Cybersecurity—Requirements, Methodology, and Testing,” *Sensors*, vol. 23, 4 2023.

[57] I. Rigoev and A. Sikora, *Security Aspects of Smart Meter Infrastructures.* Springer Science and Business Media Deutschland GmbH, 2023, vol. 97, pp. 77–154.

[58] "The 17 Sustainable Development Goals". United Nations. (accessed 2024-05-14). [Online]. Available: https://sdgs.un.org/goals