SPARK: A Kripke-Based Formal Analysis of the SN-MPR Routing Protocol in Wireless Sensor Networks

Souad Marir*, Hind Laouici*, Walid Guenounou*, Saadi Boudjit†
*MOVEP Laboratory, USTHB, Algiers, Algeria.
†LITIS Lab, University of Rouen Normandy, France.
{smarir@usthb.dz, hind.laouici@etu.usthb.dz, walid.guenounou@etu.usthb.dz, saadi.boudjit@univ-rouen.fr}

***Abstract*—Wireless Sensor Networks (WSNs) are a promising technology with numerous applications, particularly in constrained environments such as smart agriculture and environmental monitoring. However, their design remains challenging due to severe resource constraints, especially in terms of energy consumption. SN-MPR is an energy-efficient routing protocol designed to address these limitations, but it has not yet undergone formal verification. In this paper, we propose a formal model of the SN-MPR protocol using Kripke structures and rewriting logic. This model offers a precise and rigorous description of the protocol's behaviour, providing a solid foundation for formal analysis. We implement and execute the model using the Maude system, enabling the formal validation of key protocol properties. To demonstrate our approach, we present a case study in a smart agriculture context within a WSN, where we verify two critical properties: data portability and energy efficiency. The results highlight the feasibility and effectiveness of applying formal methods to the verification of energy-aware routing protocols, specifically SN-MPR in WSNs.**

***Index Terms*—WSNs, Routing, SN-MPR, Formal modelling, Maude, Kripke, LTL, Portability**

I. Introduction

Recent advances in networking, microfabrication, and processor integration have enabled the rise of Wireless Sensor Networks (WSNs), widely deployed in resource-constrained environments such as smart agriculture. These networks consist of fixed and mobile autonomous sensor nodes that are interconnected but operate under strict energy and storage limitations. To address these constraints, the SN-MPR (Sensor Network – MultiPoint Relay) [1] routing algorithm was proposed to minimise energy consumption while maintaining reliable connectivity with a mobile sink node. Despite promising simulation results, this protocol has not yet been formally verified.

Verifying a WSN using SN-MPR requires addressing the system's heterogeneity, dynamic behaviour (e.g., node mobility and failure), and constant message exchanges. Traditional approaches, such as Domain-Specific Modelling Languages (DSMLs) or ad hoc formalisations, often rely on simulation alone and lack exhaustive behavioural guarantees. In contrast, formal methods offer a rigorous framework to specify, simulate, and verify critical properties of such systems.

In this paper, we present a formal modelling approach for the SN-MPR protocol, based on rewriting logic and Kripke structures, and implemented using the Maude system. Our contributions consist in proposing a formal specification of a WSN using SN-MPR through the SPARK model (Sensor Protocol Analysis with Kripke Reasoning), based on rewriting and sorting logics to model behavioural aspects of a WSN and categorise network elements. We execute the encoding of the model in Maude language, including structural and behavioural aspects. We formally verify critical LTL properties such as deadlock-freedom, data portability, and energy efficiency using Maude's model checker. Finally, a case study in smart agriculture featuring scenarios for fire detection and crop monitoring is proposed to validate our approach. This approach is illustrated in Figure 1.

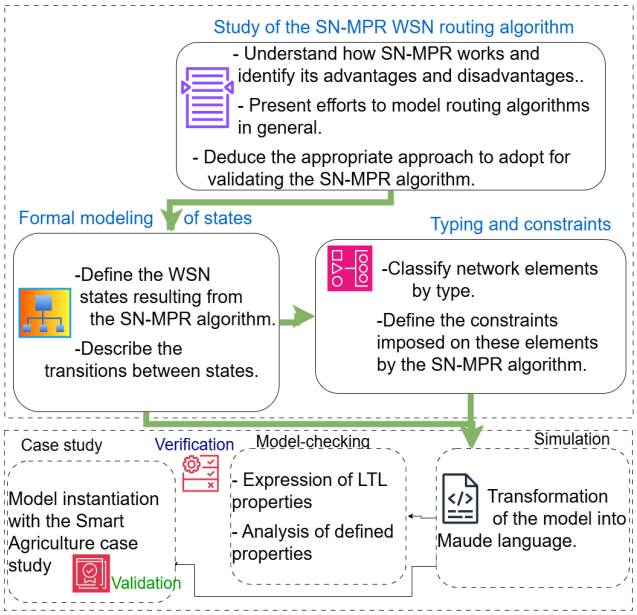

Fig. 1. SPARK approach.

II. SN-MPR Presentation

The SN-MPR algorithm [1] is an energy-efficient routing algorithm in WSN (Wireless Sensor Networks) designed to

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

minimise communication costs while maintaining reliable connectivity between sensor nodes and the sink that collects the data captured and transmitted by the nodes.

This is achieved through two key mechanisms: the use of multi-point relays (MPR) and the implementation of a local repair mechanism.

The specific objectives of the SN-MPR algorithm are as follows:

- Reduce the control traffic generated by the sink location update (SLU) messages.
- Extend the lifetime of the sensors.
- Maintain an efficient reverse routing tree with reliable data forwarding.
- Reduce data loss by introducing a data buffering mechanism.

The SN-MPR algorithm manages route updates based on SLU (Sink Location Update) messages. It optimises transmissions by avoiding unnecessary dissemination of updates, thus limiting energy consumption.

It includes SLU message management, the formation of a reverse routing tree, and local path repair [1]. It comprises seven main phases.

1) Neighbour discovery
2) Sensor-to-sensor neighbour discovery
3) Mobile sink neighbour discovery
4) SLU message management
5) Formation of the reverse routing tree
6) Local path repair
7) Storage of data in the preventive buffer

The SN-MPR algorithm has produced good results in terms of simulation. However, it is difficult to guarantee that the wireless sensor network will function as intended, with respect to the inherent properties of WSNs. It is therefore useful to formally analyse this protocol to ensure that it meets the criteria defined during its specification phase. This need for analysis has been identified, in particular, by Olveczky [2]. In this paper, this analysis is performed by the combination of simulation and formal verification. Simulation consists of verifying that a system is running by observing its behaviour on a subset of all possible scenarios. This is the case for SN-MPR, whose performance has been evaluated by simulations using NS-2 [1], with results showing that this algorithm works well in terms of coverage and node activation. This paper aims to validate the SN-MPR algorithm by verifying certain properties of safety and liveness, that an energy-efficient routing algorithm for a WSN should respect. Formal verification examines the system's state space in detail, providing assurance to the system designer that it respects the property.

III. Related Work

In this section, we give an overview of existing formal WSN modelling approaches, and then position our contributions accordingly. Overall, the currently published work highlights the routing in WSN by simulating or formally modelling it. The different available contributions go towards evaluating WSN routing protocols. The authors of [3] present VeriSensor, a DSML dedicated to the modelling and verification of wireless sensor networks (WSN). VeriSensor relies on temporal Petri nets to model the behaviour of WSN components using Instantiable Transition Systems (ITS). Model verification was performed by a model checker based on ITS, the latter exploiting decision diagram libraries to reduce state explosion. This approach was applied to a biomedical wireless sensor network (BAN - Body Area Network). In [4], the authors present formal modelling and verification in sensor networks, their contribution being the quantification of energy consumption on the Petri net model. They propose the EgPN (Energy Petri Net) formalism which integrates and explicitly models the energy factor. This approach was applied to the LEACH (Low-Energy Adaptive Clustering Hierarchy) protocol using a new variant of Coloured Petri Nets (CPN) called EgPN (Energy Petri Net) to verify properties concerning the evaluation of energy dissipated in the network and the impact of clustering on energy management. The article [5] presents formal modelling and validation work of an energy-efficient clustering protocol for WSNs based on coalition game theory and formalized as an SOS-like transition system with the aim of arranging sensor nodes to improve energy efficiency. This protocol is modelled as an SOS (Structural Operational Semantics) transition system and implemented in Maude, transforming the protocol model into rewriting logic. The analysis was performed on a six-node model, including two group leaders, with energy impact scenarios. They verified the protocol by standard simulation and model-checking in Maude. The authors of [6] propose the EAR-SDMWSN (Energy-Aware Routing for SDWSNs) routing algorithm, which is a routing protocol with reduced energy and regulated overhead based on the SDN (centralized controller) model. They proceed with this modelling through the use of ND and checksum packets as well as data clustering and DFS (Depth First Search) control overhead reduction to build routes and simulate dynamic network reconfiguration. Using a Mininet-based simulator interconnected to a Floodlight controller, they conclude good overall performance, improved longevity and low packet loss. In [7], the DORA (Destination Oriented Routing Algorithm) algorithm intended for PEGASIS-based multi-chain routing (Based on optimal direction and distance to sink and hierarchical cluster formation) is presented. To evaluate it, the authors followed a methodology based first on the choice of the optimal transmission node. They then defined a mathematical model for transmission energy before analysing orientation, cluster size, and hops before moving to comparative simulation with PEGASIS and RPC. Several other interesting routing algorithms have been proposed such as [8], [9], [10], [11], [12] and [13] but have not been evaluated or verified. Thus, our approach is closest to that proposed in [5], where the behaviour of an energy-efficient routing protocol is modelled using structural operational semantics (SOS). However, unlike this approach, we did not need to transform the network into a transition system based on SOS semantics. Indeed, the use of the Maude language and rewriting logic allows us to

model both the structure and behaviour of the network in a unified and executable manner. Moreover, SOS-based models can, unless explicitly enriched, abstract certain aspects related to component heterogeneity, which our approach natively supports through the use of typed modules.

IV. Maude based approach for modelling SN-MPR Behaviour

Formally specifying and verifying Wireless Sensor Networks (WSNs) in an executable and analyzable manner remains a challenging task, especially when addressing critical properties such as energy efficiency and data portability. To tackle this issue, our methodology begins with a detailed study of the SN-MPR algorithm, in order to capture its operational principles and the sequence of actions it performs. Based on this understanding, we construct a behavioural model of a WSN running SN-MPR using Kripke structures, where the different system states and their transitions are explicitly represented.

We then rely on sorting logic to classify the system's components and constrain their interactions. This sorting discipline ensures that each element can only perform the actions it is allowed to, thus preventing unintended behaviours and maintaining the semantic consistency of the model.

The proposed model is subsequently implemented in the Maude system, where the routing algorithm is encoded as an executable specification. To evaluate our approach, we apply it to a representative case study in the field of smart agriculture. The system's behaviour is formally validated using Maude's built-in model checker, verifying the set of properties specified during the modelling stage.

The modelling and verification process of SN-MPR is structured into five coherent phases: (i) defining the structural configuration of the WSN, including sensor types, exchanged messages, and their interrelations, thus providing a precise abstraction of the environment laying the foundation for subsequent analysis and verification; (ii) introducing predicates to observe and diagnose system states, such as the existence and symmetry of communication links, the proper election of MPR nodes, and the validity of routing paths. These predicates enable formal reasoning over system states and are critical to ensuring consistency before adaptation or deployment; (iii) to support system evolution, specifying generic adaptation rules that describe how the configuration evolves under internal or external events, while preserving structural and semantic consistency; (iv) implementing the operational behaviour of sensors through conditional rewrite rules, allowing the simulation of dynamic evolution and testing responsiveness to environmental changes; and finally, (v) verifying temporal correctness of the system against its design objectives, notably energy efficiency and data portability, using temporal logic specifications and model-checking techniques.

These phases are supported by the Maude system, a high-level formal environment based on rewriting logic, which provides both expressive modelling facilities and integrated verification tools. Maude addresses the requirements of each phase by: supporting expressive structural modelling of WSNs (nodes, links, messages), offering first-order and boolean predicates for runtime monitoring, enabling structural reconfiguration with correctness-by-construction, encoding adaptive behaviour via conditional rewrite rules, and incorporating an LTL model checker that symbolically represents Kripke structures to reason over temporal properties.A Maude specification is typically structured into two complementary modules:

- A functional module, specifying the static part of the system in membership equational logic (types, operators, equations, attributes).
- A system module, describing the dynamic behaviour via rewrite rules over the equational theory.

In this paper, we define The module `sme`, which contains the configuration of the WSN, the module `sm-predicats`, which encodes the Kripke structure necessary for LTL-based model checking of key system properties, the module `sm-topology` which describes a WSN topology and `sm` module includes conditional and non conditional adaptation actions. This modular and executable approach allows for a clear separation of concerns between structure, behaviour, and verification, while promoting flexibility, extensibility, and reusability throughout the modelling process.

The Figure 2 describes the SPARK modelling, implementation and verification approach for SN-MPR.

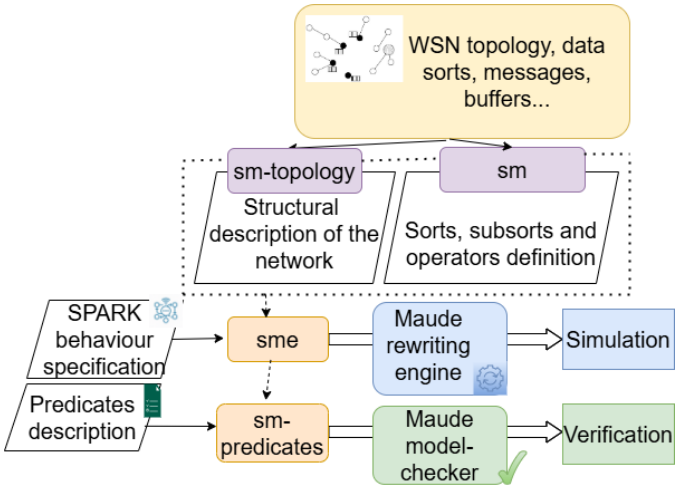

Fig. 2. Maude modules for SPARK-model

The Kripke structure associated to SPARK contains mainly a set of states S_{SM}, and a set of rewriting rules that allow the transition from a state to another as shown in Figure 3.

S_{SM} is the set of states as $S_{SM} = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}\}$

- s_0 : The initial state where the network isn't configured yet and nodes are not linked
- s_1 : Sensors made up symetrical links with their neighbours and built their neighbourhoud tables
- s_2 : Networks consisting of a sink with connected sensor nodes.

- s_3 : The network is configured, and each node and the sink have a neighbourhood table.
- s_4 : Receiving the SLU message generated by the sink, containing the sink identifier, request, TTL, sequence number, sender address and sink position.
- s_5 : The SLU message is rejected.
- s_6 : The SLU is stored in the temporary buffer.
- s_7 : The distance to the sink is updated without retransmission (the reverse routing tree is not affected by sink mobility).
- s_8: Distance update (dist=ttl) with retransmission (the routing tree is updated).
- s_9: Update of the distance between n_i and the sink and of the reverse routing tree.
- s_{10}: The SLU is not transmitted.
- s_{11}: The SLU is transmitted.
- s_{12}: The reverse routing tree is generated.
- s_{13} : Data reception and delay by sink neighbours.
- s_{14}: Data received and collected by the sink.

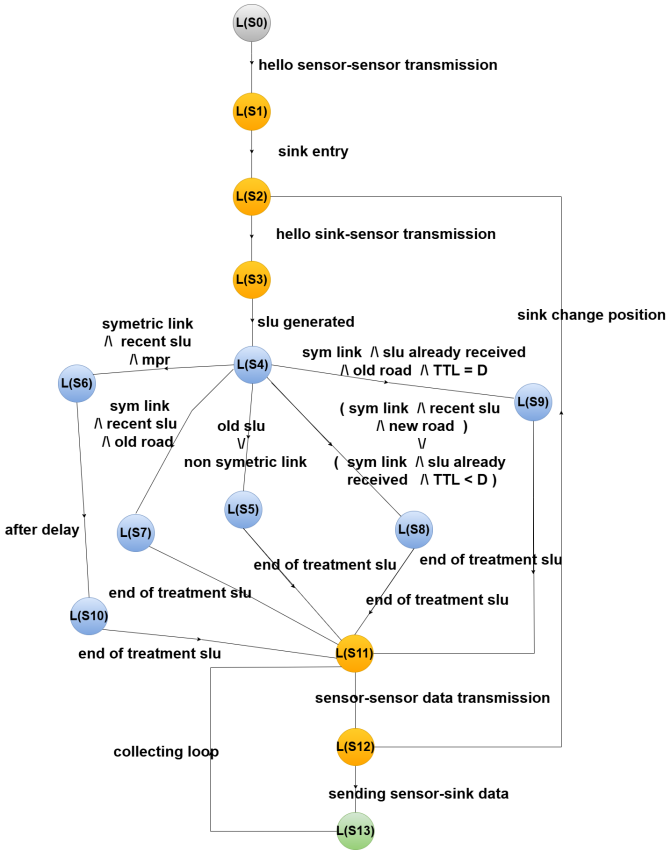

Fig. 3. SPARK's schematic view

V. Case study : Smart agriculture WSN

The SN-MPR algorithm is designed to optimize energy consumption in Wireless Sensor Networks (WSNs). To evaluate its effectiveness, we consider a case study in the domain of smart agriculture, which provides a representative environment for WSN deployment. This application domain relies on autonomous sensors for environmental monitoring under strict energy constraints, making it an ideal setting for testing energy-aware routing protocols. In our case study, the SN-MPR protocol is deployed in a smart farming scenario, where a set of static sensor nodes is distributed across an agricultural field. A mobile drone serves as the sink node, periodically collecting data from the ground sensors. Each sensor node is equipped with sensing, processing, and wireless transmission capabilities to monitor environmental parameters such as soil moisture, temperature, and light intensity. Due to limited energy resources, optimizing communication becomes crucial, and the use of the SN-MPR protocol is justified. As the drone moves over the monitored area, it periodically broadcasts its position to the network, triggering the SN-MPR relay re-evaluation mechanism. This dynamic adjustment ensures efficient data forwarding while minimizing energy consumption across the network.

In this context, we present two main scenarios : fire detection and crop growth monitoring.

A. Fire detection

In this execution scenario, we simulate a case of fire detection on a smart farm. The network consists of static sensors distributed across agricultural land. They collect several parameters and data that may indicate the start of a fire. These parameters include temperature, humidity and the concentration of flammable gases such as CO. The sensors begin by exchanging "hello" messages to identify their neighbours. When a drone enters the area, it also sends "hello" messages each time it changes position to discover nearby sensors. Data is collected and transmitted according to the SN-MPR protocol. At each stopping point, the drone retrieves data from the sensors via the reverse routing tree. A temperature rise above 45°C, accompanied by a drop in humidity below 20% and an increase in CO concentrations (above 10 ppm), indicates the start of a fire. In this case, an alert is immediately generated, allowing for a response. This scenario is modelled by a Kripke structure and encoded into Maude then executed. Figure 4 illustrates the final state reached after executing this case. We note that sensor number 2 plays a key role: it detects critical values `data(85, 10, 20)` corresponding to a high temperature (85°C), very low humidity (10%) and a high CO concentration (20%). These values exceed the threshold set to activate a fire alarm.

- The data message `dmsg(1, 2, 1, 2, data(85, 10, 20))` generated by sensor 2: `sensor(2, 1, 1, 2, 1, true(MPR), buffer slu())`, transmitted following the reverse routing tree established via SLU messages, and relayed by MPR sensors to `sink(1, loc(0, 0), 2, buffer dmsg(), buffer slu())`.
- Other sensors captured standard environmental data `data(30, 25, 5)` indicating a normal situation (a localised alert situation in a stable network in the rest of the field).
- The sink buffers contain all the transmitted data.

```
Maude> rew initial .
rewrite in smtop : initial .
rewrites: 200965 in 174ms cpu (177ms real) (1154971 rewrites/second)
result State: < sink(1, loc(0, 0), 2, dmsg(1, 2, 1, 2, data(85, 10, 20)) ; dmsg(
    1, 2, 1, 4, data(30, 25, 5)) ; dmsg(1, 2, 1, 5, data(30, 25, 5)) ; dmsg(1,
    2, 1, 6, data(30, 25, 5)) ; dmsg(1, 2, 1, 7, data(30, 25, 5)) ; dmsg(1, 3,
    1, 3, data(30, 25, 5))) slu(1, 1, 1, 8, loc(0, 0)) slu(1, 2, 2, 7, loc(0,
    0)) slu(1, 3, 5, 7, loc(0, 0)) slu(2, 1, 1, 8, loc(0, 0)) sensor(2, 1, 1, 2,
    1, true, slu(1, 1, 1, 9, loc(0, 0)) ; slu(2, 1, 1, 9, loc(0, 0)), dmsg(1, 2,
    1, 2, data(85, 10, 20)) ; dmsg(1, 4, 2, 4, data(30, 25, 5)) ; dmsg(1, 5, 2,
    5, data(30, 25, 5)) ; dmsg(1, 5, 2, 6, data(30, 25, 5)) ; dmsg(1, 5, 2, 7,
    data(30, 25, 5))) sensor(3, 1, 1, 2, 1, false, empty-buffer, dmsg(1, 3, 1,
    3, data(30, 25, 5))) sensor(4, 2, 2, 1, 1, false, empty-buffer, dmsg(1, 4,
    2, 4, data(30, 25, 5))) sensor(5, 2, 2, 1, 1, true, slu(1, 2, 2, 7, loc(0,
    0)), dmsg(1, 5, 2, 5, data(30, 25, 5)) ; dmsg(1, 6, 5, 6, data(30, 25, 5)) ;
    dmsg(1, 7, 5, 7, data(30, 25, 5))) sensor(6, 5, 3, 1, 1, false,
    empty-buffer, dmsg(1, 6, 5, 6, data(30, 25, 5))) sensor(7, 5, 3, 1, 1,
    false, empty-buffer, dmsg(1, 7, 5, 7, data(30, 25, 5))) >
Maude>
```

Fig. 4. Execution scenario of fire detection

B. Crop growth monitoring

In this scenario, we assess crop growth by analysing the parameters that affect agricultural progress: humidity, height and temperature. Crop maturity is indicated by a stabilisation in plant size, a decrease in soil moisture and a specific temperature, which will be collected by sensors and transmitted to the data well. In our case, we monitor corn, strawberries, and tomatoes. For corn, maturity is signalled when the height stabilises at around 2.5 metres and soil moisture decreases below 20% with a temperature threshold above 26°C. Strawberries reach maturity when the height of the plants remains constant at around 30 cm, soil moisture falls below 25%, and the temperature exceeds 22°C, indicating that the fruit is ripe. For tomatoes, the end of growth is detected when the height stabilises at around 1.5 metres, with decreasing soil moisture ($< 25\%$) and a constant ambient temperature ($>= 28C$). In this case, the protocol follows a stable collection behaviour, with the sensors collecting the data and transmitting it to the UAV according to the routing tree. The environmental parameters are normal data(11, 30, 10), indicating neither plant maturity nor an abnormal situation. Temperature and humidity are within expected limits, and plant height has not yet reached harvest level. The sink buffers contain all the data transmitted.

- The data message `dmsg(1, 2, 1, 2, data(27, 18, 250))` generated by sensor 2 `sensor(2, 1, 1, 2, 1, true(MPR), buffer slu())`, indicates that the corn is ripe. It is transmitted according to the reverse routing tree established via SLU messages and relayed by the MPR sensors to `sink(1, loc(0, 0), 2, buffer dmsg(), buffer slu())`. The sink buffers contain all the data transmitted according to the routing tree.
- Sensor 3 generates the message `data(23, 20, 30)` indicating strawberry ripeness, and sensor 4 generates the message `data(28, 20, 150)` indicating tomato ripeness.
- Other sensors (4 -¿20) captured standard environmental data `data(11, 30, 10)` indicating a normal situation (indicating neither plant ripeness nor an abnormal situation).

Figure 5 illustrates the Maude execution of the SPARK model in this case.

```
Maude> rew initial .
rewrite in sm-ltl : initial .
rewrites: 65327 in 43ms cpu (45ms real) (1519232 rewrites/second)
result State: < sink(1, loc(0, 0), 2, dmsg(1, 2, 1, 2, data(11, 30, 10)) ;
    dmsg(1, 2, 1, 4, data(11, 30, 10)) ; dmsg(1, 2, 1, 5, data(11, 30, 10)) ;
    dmsg(1, 2, 1, 6, data(11, 30, 10)) ; dmsg(1, 3, 1, 3, data(11, 30, 10)))
    slu(1, 1, 1, 8, loc(0, 0)) slu(1, 2, 2, 7, loc(0, 0)) slu(1, 3, 5, 8, loc(
    0, 0)) slu(2, 1, 1, 8, loc(0, 0)) sensor(2, 1, 1, 2, 1, true, slu(1, 1, 1,
    9, loc(0, 0)) ; slu(2, 1, 1, 9, loc(0, 0)), dmsg(1, 2, 1, 2, data(11, 30,
    10)) ; dmsg(1, 4, 2, 4, data(11, 30, 10)) ; dmsg(1, 5, 2, 5, data(11, 30,
    10)) ; dmsg(1, 5, 2, 6, data(11, 30, 10))) sensor(3, 1, 1, 2, 1, false,
    empty-buffer, dmsg(1, 3, 1, 3, data(11, 30, 10))) sensor(4, 2, 2, 1, 1,
    false, empty-buffer, dmsg(1, 4, 2, 4, data(11, 30, 10))) sensor(5, 2, 2, 1,
    1, true, slu(1, 2, 2, 7, loc(0, 0)), dmsg(1, 5, 2, 5, data(11, 30, 10)) ;
    dmsg(1, 6, 5, 6, data(11, 30, 10))) sensor(6, 5, 3, 1, 1, false,
    empty-buffer, dmsg(1, 6, 5, 6, data(11, 30, 10))) >
Maude>
```

Fig. 5. Execution scenario of abnormal crop growth

VI. Implementing and Verifying Portability and Energy Efficiency

Predicates encoded into Maude

Before any verification, any system must be executed avoiding any deadlock. A deadlock in a packet-switched network is a state in which one or more messages have not yet reached their destinations but cannot progress any further, and no element should remain inactive indefinitely. [14]. This means that every SLU packet generated must be transmitted to all nodes in the network, and no SLU packet remains blocked at a node; it is systematically relayed until it is received by all nodes. This property is formulated using LTL logic by: `eq deadlock(S) = [](slu-generated(S) -> <> processed-by-all(S)) .`

```
Maude> red modelCheck(initial, deadlock(1)).
reduce in sm-ltl : modelCheck(initial, deadlock(1)) .
rewrites: 546136582 in 332823ms cpu (345921ms real) (1640921 rewrites/second)
result Bool: true
Maude>
```

Fig. 6. Model checking of no deadlock.

Energy optimisation: Energy efficiency in wireless sensor networks (WSNs) can be defined as the proportion of successfully transmitted data to total energy consumed [15]. In the SPARK model, this is equivalent to the fact that aach sensor uses an optimal path to the sink to transmit its data, minimising unnecessary transmissions. Energy efficiency is formulated using LTL logic by: `eq energy-efficiency = [] optimal-path .`

```
Maude> red modelCheck(initial, energy-efficiency(1)).
reduce in sm-ltl : modelCheck(initial, energy-efficiency(1)) .
rewrites: 546317138 in 327432ms cpu (338226ms real) (1668490 rewrites/second)
result Bool: true
Maude>
```

Fig. 7. Energy efficiency verification

Portability is the ability of a protocol to function consistently across heterogeneous network configurations and hardware platforms while ensuring proper operation and performance. We verify the, that the SLUs are always processed, and that optimal paths

are established, regardless of the network or its size. the associated LTL formula is as follows: `eq portability(S) = [](valid-topology -> <> (processed-by-all(S) /\optimal-path(S))) .`

```
Maude> red modelCheck(initial, portability(1)).
reduce in sm-ltl : modelCheck(initial, portability(1)) .
rewrites: 546435647 in 316586ms cpu (326032ms real) (1726025 rewrites/second)
result Bool: true
Maude>
```

Fig. 8. Data portability verification.

It is important to emphasise that our verification approach is qualitative rather than quantitative. By relying on rewriting logic and Maude's model-checking capabilities, the analysis explores all possible execution states of the system, independently of the number of nodes involved. Consequently, if a temporal property such as energy efficiency, data portability, or deadlock freedom holds in our model with a small number of sensors, it will also hold for larger deployments (e.g., thousands of nodes), since the verification concerns the logical correctness of the protocol's mechanisms rather than specific numerical performance metrics. In this sense, the case study with a reduced network size serves as a representative abstraction, without compromising the generality of the results.

VII. Conclusion

In this paper, we proposed a formal modelling and verification approach for the SN-MPR energy-efficient routing protocol in Wireless Sensor Networks (WSNs). Our method leverages the expressiveness and executability of the Maude system to capture both the structural and behavioural aspects of the protocol using rewriting logic and Kripke structures. We demonstrated how this formalisation supports rigorous reasoning about key properties such as energy efficiency, data portability, and deadlock freedom through LTL-based model checking. To validate our model, we conducted a case study in the context of smart agriculture, simulating realistic scenarios such as fire detection and crop growth monitoring. The results confirm that the SN-MPR protocol satisfies the verified properties, underscoring its relevance for energy-constrained and dynamic WSN environments. This work illustrates the practical value of formal methods in rigorously validating the behaviour of routing protocols prior to deployment. As future work, we aim to extend our framework by incorporating Maude's strategy language to support the specification and verification of multiple routing strategies beyond SN-MPR, thereby enhancing the applicability and flexibility of our approach to a broader class of WSN protocols. In addition, we plan to perform a quantitative analysis of energy savings by integrating queueing Theory verification. This will allow us to complement qualitative property verification with a complementary formal evaluation of the energy performance of SN-MPR, providing a more comprehensive validation of its effectiveness in large-scale WSNs.

References

[1] Yasir Faheem and Saadi Boudjit. Sn-mpr: A multi-point relay based routing protocol for wireless sensor networks. In *2010 IEEE/ACM Int'l Conference on Green Computing and Communications and Int'l Conference on Cyber, Physical and Social Computing*, pages 761–767, 2010.

[2] Peter Csaba Ölveczky and Stian Thorvaldsen. Formal modeling and analysis of the ogdc wireless sensor network algorithm in real-time maude. In *International conference on Formal methods for open object-based distributed systems*, pages 122–140. Springer, 2007.

[3] Yann Ben Maissa, Fabrice Kordon, Salma Mouline, and Yann Thierry-Mieg. Modeling and analyzing wireless sensor networks with verisensor: An integrated workflow. *Trans. Petri Nets Other Model. Concurr.*, 8:24–47, 2013.

[4] Amel Berrachedi, Malika Ioualalen, and Ahmed Hammad. Towards the formal modeling methodology of wsn through the transformation of sysml into dspns. In *SIMULTECH*, pages 83–91, 2021.

[5] Fatemeh Kazemeyni, Einar Broch Johnsen, Olaf Owe, and Ilangko Balasingham. Formal modeling and validation of a power-efficient grouping protocol for wsns. *The journal of logic and algebraic programming*, 81(3):284–297, 2012.

[6] F Fernando Jurado-Lasso, Ken Clarke, Andres Navarro Cadavid, and Ampalavanapillai Nirmalathas. Energy-aware routing for software-defined multihop wireless sensor networks. *IEEE Sensors Journal*, 21(8):10174–10182, 2021.

[7] Kun Wang, Chih-Min Yu, and Li-Chun Wang. Dora: A destination-oriented routing algorithm for energy-balanced wireless sensor networks. *IEEE Internet of Things Journal*, 8(3):2080–2081, 2020.

[8] Dung Nguyen Quoc, Niansheng Liu, and Donghui Guo. A hybrid fault-tolerant routing based on gaussian network for wireless sensor network. *Journal of Communications and Networks*, 24(1):37–46, 2021.

[9] Syed Sherjeel A Gilani, Amir Qayyum, Rao Naveed Bin Rais, and Mukhtiar Bano. Sdnmesh: An sdn based routing architecture for wireless mesh networks. *IEEE Access*, 8:136769–136781, 2020.

[10] Zhendong Wang, Liwei Shao, Shuxin Yang, and Junling Wang. Lemh: Low-energy-first electoral multipath alternating multihop routing algorithm for wireless sensor networks. *IEEE Sensors Journal*, 22(16):16687–16704, 2022.

[11] Xiuwen Fu, Yongsheng Yang, and Octavian Postolache. Sustainable multipath routing protocol for multi-sink wireless sensor networks in harsh environments. *IEEE Transactions on Sustainable Computing*, 6(1):168–181, 2020.

[12] Jiazu Xie, Baoju Zhang, and Cuiping Zhang. A novel relay node placement and energy efficient routing method for heterogeneous wireless sensor networks. *IEEE Access*, 8:202439–202444, 2020.

[13] Ziaur Rahman, Fazirulhisyam Hashim, Mohd Fadlee A Rasid, Mohamed Othman, and Kamal Ali Alezabi. Normalized advancement based totally opportunistic routing algorithm with void detection and avoiding mechanism for underwater wireless sensor network. *IEEE access*, 8:67484–67500, 2020.

[14] Anna Stramaglia, Jeroen J. A. Keiren, and Hans Zantema. Deadlock in packet switching networks. In Ali Movaghar, Farnaz Moradi, and Marjan Sirjani, editors, *Fundamentals of Software Engineering: 9th International Conference, FSEN 2021, Virtual Event, May 19–21, 2021, Revised Selected Papers*, volume 13057 of *Lecture Notes in Computer Science*, pages 127–141. Springer International Publishing, 2021.

[15] I. Surenther, K.P. Sridhar, and Michaelraj Kingston Roberts. Maximizing energy efficiency in wireless sensor networks for data transmission: A deep learning-based grouping model approach. *Alexandria Engineering Journal*, 83:53–65, 2023.