Przemysław ZIELONY, Krzysztof PIOTROWSKI
IHP – Leibniz-Institut für innovative Mikroelektronik, Im Technologiepark 25
15236 Frankfurt (Oder), Germany

DISTRIBUTED DATA-CENTRIC MAINTENANCE AND MANAGEMENT LAYER FOR WSN

Managing wireless sensor networks (WSNs) requires effective node monitoring. This work discusses an approach based on the modular middleware tinyDSM. Through services and adapters, tinyDSM allows to adapt to various needs and control operations on nodes and in the network. By using and monitoring state machines at every level of the software architecture enables node monitoring, increasing system reliability and fault tolerance. This also contributes to scalability and adaptability to different environmental conditions, ensuring optimal resource utilization in various applications. Additionally, the ability to notify users and share data facilitates communication and troubleshooting, enables quick resolve to network errors.

Rozproszona, zorientowana na dane warstwa konserwacji i zarządzania WSN

Zarządzanie bezprzewodowymi sieciami sensorów (WSN) wymaga skutecznego monitorowania węzłów. Praca ta omawia podejście oparte o modułowy middleware tinyDSM. Poprzez serwisy i adaptery, tinyDSM umożliwia dostosowanie się do różnych potrzeb i kontrolę operacji na węzłach i w sieci. Wykorzystanie i monitorowanie maszyn stanów na każdym poziomie architektury oprogramowania umożliwia monitorowanie węzłów, zwiększając niezawodność systemu i tolerancję na awarie. To również przyczynia się do skalowalności i adaptacyjności do różnych warunków środowiskowych, zapewniając optymalne wykorzystanie zasobów w różnych aplikacjach. Ponadto, możliwość powiadamiania użytkowników i udostępniania danych ułatwia komunikację i rozwiązywanie problemów, umożliwiając szybką reakcję na błędy sieci.

1. INTRODUCTION

Monitoring of nodes in a Wireless Sensor Network (WSN) is crucial for various reasons. It enables the diagnosis of failures, whether within individual sensors or across the entire network, allowing for prompt troubleshooting and prevention of downtime. Additionally, monitoring node health facilitates efficient energy management by tracking energy consumption and battery levels, ensuring timely optimization or replacement. It also ensures data quality by checking and calibrating sensors regularly, thereby maintaining accuracy and consistency in collected data. Furthermore, it optimizes data routing by monitoring wireless connectivity and addressing transmission issues, leading to minimized delays and maximized throughput. Node monitoring enhances network security by detecting intrusion attempts or unauthorized access, enabling swift responses to potential threats and implementation of appropriate security measures. Lastly, it enables resource management by tracking node resources such as network bandwidth, cache memory, and processing power, optimizing their utilization for optimal network performance. Overall, node monitoring in WSN is essential for reliability, efficiency, and security.

2. RELATED WORK

Currently, there are few publications on WSN maintenance. One of the approaches focuses on the reliability of WSNs, defined as the ability of the network to fulfill set tasks in a given time and conditions [1]. Analytical data is aggregated from the entire network, not from individual nodes. Due to limitations such as battery life and cost, fault tolerance features cannot be easily added to sensor nodes [2].

WSN sustainability research examines the energy consumption of wireless sensors using algorithms such as Adaptive Sampling Algorithm (ASA), Compensation Adaptive Sampling Algorithm (CASA), and Resuscitation Adaptive Sampling Algorithm (RASA). These algorithms adjust the sampling rate to reduce power consumption and ensure that each sensor is self-sufficient [3]. Another approach to WSN maintenance involves predictive maintenance (PdM) using machine learning, specifically feedforward neural networks (FFNN). PdM collects data to estimate system operational performance, helping to assess condition, diagnose faults, and predict remaining system life. PdM optimizes the system life cycle, minimizes unplanned downtime and reduces maintenance costs [4].

The next approach refers to a network of urban wireless sensors where optimal route planning is crucial, which has a significant impact on the response time of the entire network in the event of a failure. The proposed heuristic algorithm, inspired by the biological computational model P_system, cooperates with various classical heuristics to accelerate fault localization and minimize network downtime [5].

3. PROPOSED APPROACH

In our approach, the operation of a wireless sensor network node is facilitated by system based on the tinyDSM [6] middleware. Through its modular structure, various services and adapters can be added to the middleware. The tinyDSM middleware allows to exchange data between software components using so called variables and supports the storage and exchange of these. It provides an API called Data Interface for Adapters and Services to write and read the shared variables.

Adapters are used to adapt external data sources to tinyDSM. This includes the basic microcontroller peripherals, external memory modules, communication protocols, radio modules, and can also be utilized to adapt sensor drivers to attach all kind of sensors (and actuators). Services are software system blocks that are meant to process data. They read variables, process their values and store the results back to the middleware.

The modular nature of the proposed software architecture enables the addition of services responsible for node management (energy, internal states of software modules, radio module control, including transmission power adjustment, etc.).

Efficient node operation management necessitates insight into node functionality and its individual modules and services. This could be addressed by introducing another service dedicated to collecting such information. However, to enable information collection, each program block would need to be monitored from the lowest to the highest layer. Such monitoring could be achieved by implementing state machines in every code block at all levels. Nonetheless, this approach would require collaboration with other system blocks. This defines the manner in which additional adapters and services need to extend basic functionality by adding monitoring functions during programming.

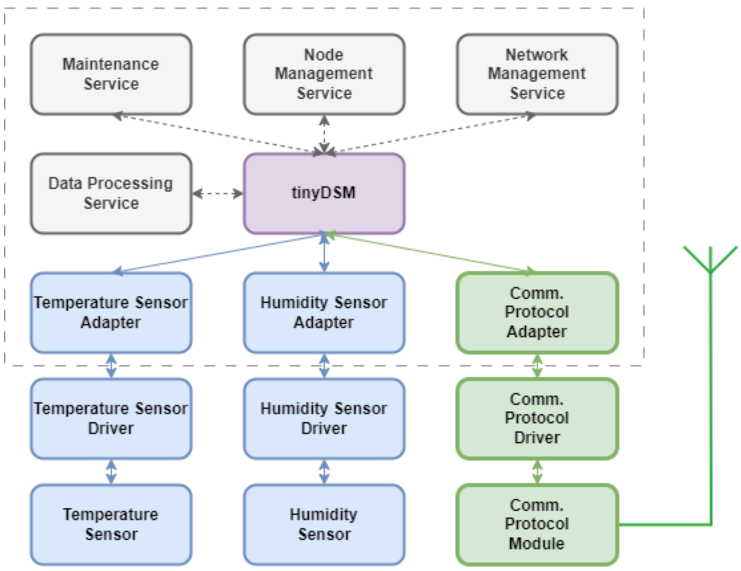

Fig. 1. An example system consisting of modules based on tinyDSM
Rys. 1. Przykład system zbudowanego z modułów bazujących na tinyDSM

Using the tinyDSM middleware, the data flow from the sensor to transmission over the wireless network would proceed as follows:
Data collection service from sensors -> *Sensor* -> *Sensor driver* -> *Sensor adapter* -> *tinyDSM* -> *Data processing service* -> *Data transmission service* -> *Wireless network controller adapter* -> *Wireless network module driver* -> *Data transmission to designated node.*

Each of these blocks would require implementation of a state machine, whose state is stored in a tinyDSM variable to be able to monitor and verify the correct operation in the chain of individual blocks.

The entire tinyDSM middleware is based on variables that may contain information about data from sensors, configuration of the internal modules and services, as well as the operation of tinyDSM itself. In the proposed approach, information about the states of individual modules would be saved as variables in the tinyDSM middleware, as well.

It is always desired to obtain the view on the state of the WSN at the operator desk. However, due to the large amount of data, it would be difficult to ensure that all states from all modules are sent to the WSN gateway. In the compile time configuration, the maintenance service would get parameters about how and what states are saved to memory and what information must be sent to the end user immediately. The maintenance services can also be notified about specific changes in the monitored parameters and can act properly.

State machine data can also be stored on a memory card in a structured manner using a service. This would enable the preservation of historical data, saving data to memory at intervals specified in the configuration. This approach would make it easier to identify which block of code is responsible for a given failure without overloading the microcontroller's internal memory.

However, it might still be difficult to constantly send maintenance data from all modules to the gateway. Therefore, in order not to significantly burden the wireless network, only information about the most important errors should reach the user directly. To make this possible, errors would need to be categorized. Such a solution would significantly help save memory, reduce wireless network congestion and facilitate error detection. This selection can be done on the node level as follows:

- Minor error - can be ignored if handled locally
- Moderate error - allows node operation, but requires user attention in a later time
- Major error - node cannot operate correctly and there is a need to notify the user immediately

Information regarding major errors need to be promptly sent to the gateway and provided to the network administrator. Information about moderate errors can be sent with lower priority, while information about minor errors should only be sent upon user request.

At a higher user-facing layer, network operation information can be presented as a WWW service. Such a tool will significantly ease communication and maintenance. An application with implemented maintenance and management services will greatly assist in the deployment of wireless networks. Wireless network nodes are often placed in challenging conditions, and the network layout always needs testing. Such tests could be conducted much faster.

This proposed approach can also be extended with high-level tools for analyzing the operation of WSNs. Thanks to such tools, it would be possible to perform a preliminary network analysis and identify network bottlenecks in various topologies.

Additionally, information about all faults can be received by the management service, which could autonomously resolve some errors on nodes and return them to a stable state.

An additional method for reading errors during network maintenance would be the possibility of using a second radio module, if available. After connecting on-site for example via smartphone, data from all state machines or only error codes occurring on a node could be displayed. Further, runtime configuring would be possible with using such a module.

Utilizing an energy management service increases network energy efficiency, allowing administrators and users to monitor energy consumption at individual nodes, facilitating identification of nodes with high energy consumption for optimization purposes. Additionally, the service contributes to increased network reliability by preemptively addressing battery depletion issues. Furthermore, it enhances the scalability and flexibility of the entire system/network, enabling users to adjust energy management settings to specific needs and environmental conditions, optimizing resource utilization in various applications. Effective energy management is crucial for achieving optimal system performance and continuous operation.

Through shared data provision, an energy management service running at the node level could synergistically interact with a service designed to manage the wireless network in the same framework. Using information about available battery power, such a service could intelligently adjust antenna gain, improving overall network performance or efficiently planning routes in wireless network.

The joint operation of energy and network management services allows for dynamic adaptation to changing environmental conditions and network requirements. For example, during periods of ample energy availability, the service could increase antenna gain to extend communication range or improve link quality. Conversely, when the battery level is low, it could reduce antenna gain to conserve energy without significantly compromising communication reliability.

CONCLUSIONS

The implementation of the maintenance and management layer based on tinyDSM offers a modular and flexible framework for managing wireless sensor network (WSN) nodes efficiently. By incorporating various adapters and services, tailored to adapt peripherals and control node operations, the middleware provides a robust foundation for WSN deployment and management. The utilization and monitoring of state machines at each level of the software architecture ensures proper monitoring and operation of individual blocks, enhancing system reliability and fault tolerance. This also contributes to scalability and adaptability to diverse environmental conditions, ensuring optimal resource utilization across different applications. Furthermore, the provision of user notifications and data sharing capabilities facilitates seamless communication and troubleshooting, enabling timely response to network anomalies and errors.

REFERENCES

1. Feng H., Dong J., Reliability analysis for WSN based on a modular k-out-of-n system, Journal of Systems Engineering and Electronics, 2017
2. Zhang Q., Dong Ch., Yu F., Maintenance of Large Scale Wireless Sensor Networks, 2022 IEEE 5th International Conference on Electronics Technology, 2022
3. Changmin Lee., Jaiyong Lee., "Harvesting and Energy aware Adaptive Sampling Algorithm for guaranteeing self-sustainability in Wireless Sensor Networks", International Conference on Information Networking, 2017
4. Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, “A Survey of Predictive Maintenance: Systems, Purposes and Approaches,” SignalProcessing, Dec. 2019
5. Ziqi Wei Mike MacGregor, A New Way to Model the Wireless Sensor Network Maintenance Job, Wireless Communications and Mobile Computing, 2022
6. Piotrowski K., Langendoerfer P., and Peter S.: tinyDSM: A highly reliable cooperative data storage for Wireless Sensor Networks, Collaborative Technologies and Systems, International Symposium, 2009.