

DOI 10.54596/2958-0048-2026-3-296-308

UDC 004.7

IRSTI 50.39.15

A SIMULATION-DRIVEN FRAMEWORK FOR NETWORK-AWARE PERFORMANCE EVALUATION OF IOT SYSTEMS

A.S. Bayraktar¹, C. Ceken^{2*}

¹*Sakarya University, Department of Software Engineering, Sakarya, Türkiye*

^{2*}*Sakarya University, Department of Computer Engineering, Sakarya, Türkiye*

**Corresponding author: celalceken@sakarya.edu.tr*

Abstract

Selecting and deploying an appropriate wireless sensor network (WSN) is an important challenge in IoT system design. Evaluating different WSN configurations through physical deployment can be costly and time-consuming. This study proposes a simulation-driven framework that integrates a simulated WSN with an MQTT-based IoT communication infrastructure to enable network-aware performance evaluation prior to physical deployment. An SDN-WISE-based WSN was implemented using the Cooja simulator, and the generated sensor traffic was forwarded to a Mosquitto-based MQTT cluster via a socket-based integration mechanism. Unlike conventional approaches that use independent traffic generators, the MQTT workload in the proposed framework originates directly from the simulated WSN. Experiments with 24-, 36-, and 48-node scenarios showed that average end-to-end delay increased from 37.51 ms to 138.01 ms, while total application throughput increased from 142.51 bps to 281.12 bps. Packet delivery remained effectively near 100% across all scenarios. The proposed approach provides a controlled, repeatable environment for evaluating various WSN configurations and can reduce the need for multiple physical deployments in the early stages of IoT system design.

Keywords: Internet of Things, wireless sensor networks, Cooja, MQTT, network simulation, IoT system design.

IoT ЖҮЙЕЛЕРІНІҢ ЖЕЛІГЕ БЕЙІМДЕЛГЕН ӨНІМДІЛІГІН БАҒАЛАУҒА АРНАЛҒАН МОДЕЛЬДЕУГЕ НЕГІЗДЕЛГЕН ҚҰРЫЛЫМ

Bayraktar A.S.¹, Ceken C.^{2*}

¹*Сакарія университеті, Бағдарламалық қамтамасыз ету инженериясы кафедрасы,
Сакарія, Түркия*

^{2*}*Сакарія университеті, Компьютерлік инженерия кафедрасы, Сакарія, Түркия*

**Хат-хабар үшін автор: celalceken@sakarya.edu.tr*

Аңдатпа

IoT жүйесін жобалауда тиісті сымсыз сенсорлық желіні (WSN) таңдау және оны орналастыру маңызды мәселелердің бірі болып табылады. Өртүрлі WSN конфигурацияларын физикалық түрде орналастыру арқылы бағалау көп уақыт пен шығынды қажет етуі мүмкін. Осы зерттеуде физикалық орналастыруға дейін желіге бейімделген өнімділікті бағалауға мүмкіндік беретін, модельдеу негізіндегі құрылым ұсынылады. Ұсынылған құрылым модельденген WSN-ді MQTT негізіндегі IoT коммуникациялық инфрақұрылымымен біріктіреді. SDN-WISE негізіндегі WSN Cooja симуляторы арқылы іске асырылып, қалыптастырылған сенсорлық трафик сокет арқылы біріктіру механизмі көмегімен Mosquitto негізіндегі MQTT кластеріне жіберілді. Тәуелсіз трафик генераторларын пайдаланатын дәстүрлі тәсілдерден айырмашылығы, ұсынылған құрылымдағы MQTT жүктемесі тікелей модельденген WSN арқылы қалыптастырылады. 24, 36 және 48 түйіннен тұратын сценарийлерде жүргізілген эксперименттер нәтижесінде орташа соңғы нүктелер арасындағы кідіріс 37,51 мс-тен 138,01 мс-ке дейін артқаны, ал қолданбалық деңгейдегі жалпы өткізу қабілеті 142,51 бит/с-тен 281,12 бит/с-ке дейін өскені анықталды. Барлық сценарийлерде пакеттерді жеткізу деңгейі 100%-ға жуық сақталды. Ұсынылған тәсіл өртүрлі WSN конфигурацияларын бағалау үшін басқарылатын және қайталанатын ортаны қамтамасыз етеді және IoT

жүйесін жобалаудың бастапқы кезеңдерінде бірнеше физикалық орналастыру қажеттілігін азайтуға мүмкіндік береді.

Кілт сөздер: Заттар интернеті, сымсыз сенсорлық желілер, Cooja, MQTT, желіні модельдеу, IoT жүйелерін жобалау.

ИМИТАЦИОННАЯ МОДЕЛЬ ДЛЯ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ IoT-СИСТЕМ С УЧЕТОМ СЕТЕВЫХ ХАРАКТЕРИСТИК

Bayraktar A.S.¹, Ceken C.^{2*}

¹Университет Сакарья, кафедра программной инженерии, Сакарья, Турция

^{2*}Университет Сакарья, кафедра компьютерной инженерии, Сакарья, Турция

*Автор для корреспонденции: celalceken@sakarya.edu.tr

Аннотация

Выбор и развертывание подходящей беспроводной сенсорной сети (WSN) является важной задачей при проектировании IoT-систем. Оценка различных конфигураций WSN посредством физического развертывания может требовать значительных затрат времени и ресурсов. В данной работе предлагается структура, основанная на моделировании, которая интегрирует моделируемую WSN с коммуникационной инфраструктурой IoT на базе MQTT для оценки производительности системы с учетом характеристик сети до физического развертывания. WSN на основе SDN-WISE была реализована с использованием симулятора Cooja, а генерируемый сенсорный трафик передавался в MQTT-кластер на базе Mosquitto посредством механизма интеграции через сокетное соединение. В отличие от традиционных подходов, использующих независимые генераторы трафика, MQTT-нагрузка в предлагаемой структуре формируется непосредственно моделируемой WSN. Эксперименты со сценариями, содержащими 24, 36 и 48 узлов, показали, что средняя сквозная задержка увеличилась с 37,51 мс до 138,01 мс, тогда как общая пропускная способность на уровне приложений увеличилась с 142,51 бит/с до 281,12 бит/с. Во всех сценариях коэффициент доставки пакетов сохранялся практически на уровне 100%. Предлагаемый подход обеспечивает контролируемую и воспроизводимую среду для оценки различных конфигураций WSN и позволяет сократить необходимость многократного физического развертывания на ранних этапах проектирования IoT-систем.

Ключевые слова: Интернет вещей, беспроводные сенсорные сети, Cooja, MQTT, моделирование сетей, проектирование IoT-систем.

Introduction

The Internet of Things (IoT) enables distributed devices and sensors to collect and exchange data for diverse application domains such as environmental monitoring, smart agriculture, industrial automation, smart cities, healthcare, and intelligent infrastructure. IoT systems include communication networks to transfer sensor data from distributed devices to cloud infrastructure. Various wireless technologies and network architectures can be employed in the vertical domain of IoT systems, including ZigBee, Bluetooth Low Energy (BLE), Wi-Fi, SDN-based WSNs, and NB-IoT. These technologies differ in terms of communication range, data rate, latency, reliability, energy consumption, and network capacity. Therefore, selecting an appropriate WSN technology and configuration is an important design decision for a target IoT system. However, designing, deploying, and testing physical WSNs can be expensive, time-consuming, and difficult, especially when different technologies, protocols, topologies, and configurations need to be compared.

Simulation provides an effective alternative for evaluating WSN designs before physical deployment. A WSN simulator allows researchers to design various network configurations and

investigate their behavior under controlled, repeatable conditions. Network parameters can be changed without replacing physical devices or modifying a deployed testbed. This makes simulation useful for exploring alternative network designs and identifying potentially suitable configurations before investing in physical infrastructure.

Several studies have investigated simulation-based approaches for evaluating WSN and IoT performance. Simulation is particularly useful for IoT networks because physical testbeds can introduce limitations related to scalability, hardware compatibility, deployment space, and cost [1, 2]. Various simulation environments have therefore been developed and used to evaluate IoT and WSN communication under different network configurations and conditions [2, 3, 4]. Among these environments, Contiki-NG provides an open-source operating system for resource-constrained IoT devices and supports the rapid prototyping and evaluation of IoT systems, while Cooja enables the simulation of networks composed of such devices [3]. Previous studies using Contiki-NG and Cooja have evaluated different network topologies and communication scenarios using performance measures such as throughput, packet delivery ratio, end-to-end delay, energy consumption, and communication overhead [5, 6].

Software-defined networking has also been investigated as an approach for improving the flexibility and programmability of WSNs. SDN-WISE provides a software-defined architecture for WSNs and enables programmable control of network behavior and communication functions [7, 8]. The SD-WISE architecture further extends this approach by introducing software abstractions for sensor-node resources and supporting flexible network management and control [9]. More recent work has evaluated SDN-WISE under different network conditions, including static and mobile WSN scenarios, demonstrating the influence of network configuration and node mobility on performance [10]. These studies show that software-defined WSNs provide a flexible basis for investigating alternative network configurations and communication behaviors.

At the IoT communication layer, MQTT is widely used because of its lightweight publish/subscribe communication model and suitability for resource-constrained devices [11, 12]. Comparative studies have evaluated MQTT together with other messaging protocols, including CoAP, AMQP, and HTTP, while other studies have examined its performance under different network conditions, security configurations, workloads, and resource constraints [12, 13]. Recent studies have also compared MQTT with alternative communication approaches and investigated the effects of network size, transmission conditions, and traffic characteristics on communication performance [14, 15]. These results indicate that both the underlying WSN configuration and the IoT communication layer can influence the performance observed by an IoT application.

Although existing research has extensively investigated WSN simulation, software-defined WSNs, and MQTT-based IoT communication, these components are commonly evaluated independently. In particular, application traffic is often generated separately from the simulated WSN rather than being produced by the WSN itself. This study addresses this limitation by integrating the WSN simulation with an MQTT-based IoT pipeline. The main idea is that the IoT workload is generated directly by the simulated WSN rather than by a conventional traffic generator. Sensor nodes operating in the simulation produce sensor data and network traffic, which are transferred to the MQTT-based IoT system for further processing. Thus, the simulated WSN serves as an active and controllable workload source for the IoT platform, providing a basis for evaluating how changes in WSN configuration affect the traffic delivered to the IoT infrastructure.

To test the proposed framework and evaluate system performance, we implemented an SDN-WISE-based WSN using the Cooja simulator. The simulated WSN generates sensor traffic, which is integrated with the MQTT-based IoT pipeline for further processing and performance evaluation. The proposed approach provides a controlled and cost-effective environment for examining how WSN behavior and network conditions affect the overall performance of an IoT system.

The main contribution of this study is the development of a simulation-driven framework that integrates a simulated WSN with an MQTT-based IoT data pipeline, where sensor traffic processed by the IoT system is generated directly by the simulated WSN rather than by an independent artificial traffic generator. This integration provides a controlled and repeatable environment for evaluating different WSN configurations and, where suitable simulation models are available, comparing alternative communication technologies before physical deployment. The proposed approach can therefore support network-aware IoT system design and help identify suitable WSN configurations, while reducing the cost and effort of developing and testing multiple physical WSN deployments.

The remainder of this paper is organized as follows. Section 2 presents the proposed system architecture and describes the integration between the WSN simulation environment and the MQTT-based IoT pipeline. Section 3 introduces the experimental assumptions and setup and presents the performance evaluation and discussion of the results. Finally, the paper concludes with the last section, which provides final remarks and potential directions for future work.

The Proposed System Architecture

The proposed architecture presents a simulation-driven approach for integrating a wireless sensor network (WSN) with an IoT communication infrastructure. As illustrated in Figure 1, the architecture follows the general structure of a traditional IoT system, comprising a vertical domain and a horizontal domain. The vertical domain represents the sensing and communication infrastructure, whereas the horizontal domain represents the higher-level IoT services that process and use sensor data.

In the present study, the experimental implementation focuses on integrating an SDN-WISE-based WSN simulated in Cooja with an MQTT cluster. The cloud-side data pipeline and services, such as data storage, real-time monitoring, and data analytics, are included in the overall architectural view to demonstrate how the proposed framework can be integrated into a complete IoT system. However, these components are not implemented as part of the current experimental study.

System Overview

Figure 1 illustrates the big-picture architecture of the proposed system. In a traditional IoT architecture, the vertical domain represents the path by which data is collected from the physical environment and transferred via the communication infrastructure to higher-level services. This domain typically includes sensing devices, wireless communication networks, and gateway or border components.

In the proposed architecture, the WSN in the vertical domain is implemented as an SDN-WISE-based network executed in the Cooja simulator. The simulated sensor nodes generate sensor traffic according to the defined network topology and communication behavior. This simulated traffic is then transferred from the WSN environment to an external MQTT cluster.

The integration of the simulated WSN with MQTT is the primary component implemented in the proposed framework. Rather than generating MQTT traffic independently using a conventional traffic generator, the MQTT messages originate from the simulated WSN.

Therefore, the simulated network acts as the source of the traffic delivered to the IoT communication infrastructure. This allows the characteristics of the simulated WSN, including its topology, node behavior, communication patterns, and network conditions, to influence the generated workload.

The horizontal domain represents the broader cloud-side IoT architecture. As shown in Figure 1, this domain may include a data pipeline and services for data storage, real-time monitoring, stream processing, data analytics, visualization, and other application-specific operations. These components illustrate the potential integration of the proposed framework into a complete IoT system. They are not implemented or evaluated in the present study.

The architectural separation between the vertical and horizontal domains is useful because the proposed WSN–MQTT integration can serve as an interface between network simulation and higher-level IoT services. Once the simulated WSN traffic is available via the MQTT infrastructure, various downstream components can subscribe to and process the same data stream. This enables the framework to be extended in future work to evaluate end-to-end IoT services, including storage, real-time processing, and data analytics.

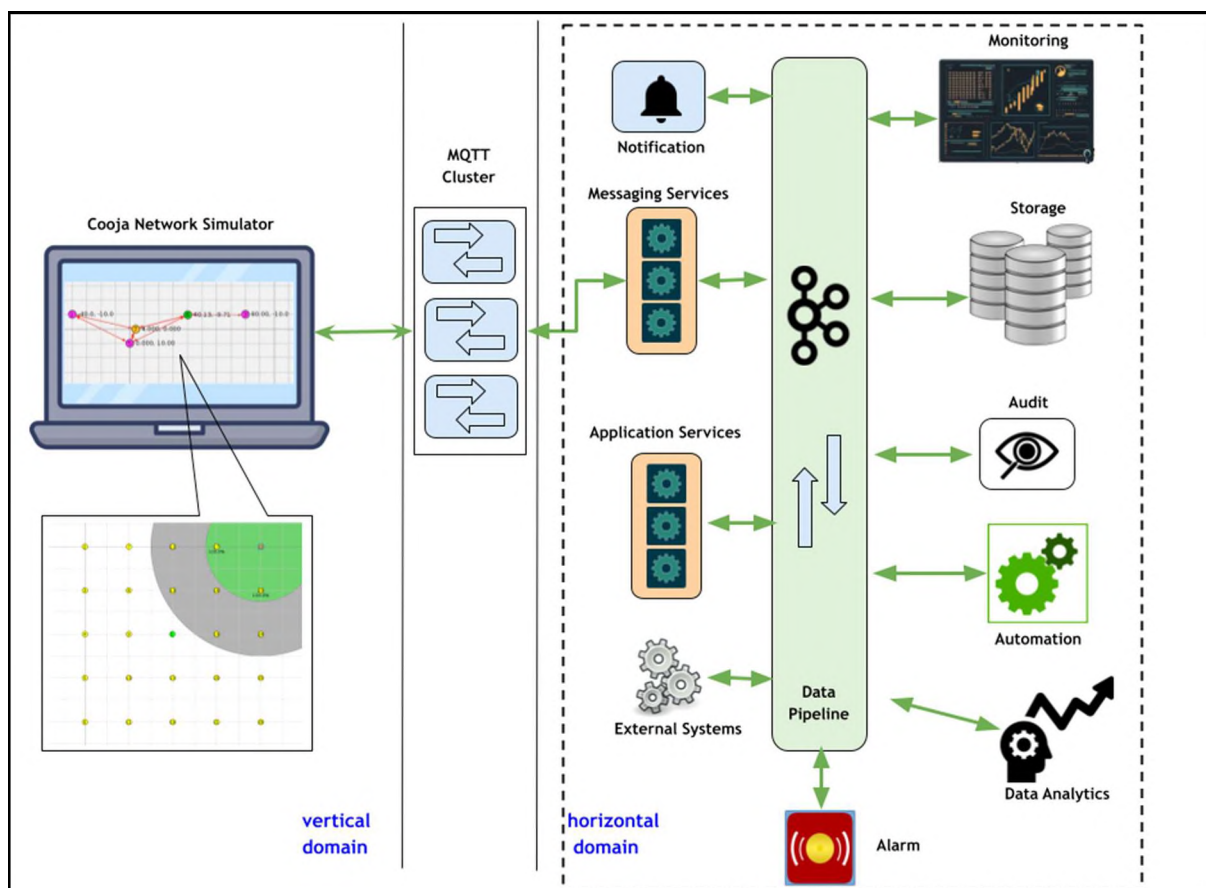


Figure 1. Overall architecture of the proposed simulation-driven IoT framework, with the WSN simulation and MQTT cluster representing the implemented components

Wireless Sensor Network Simulation Model

The experimental WSN topology used in this study is illustrated in Figure 2. The network is implemented as a grid-based topology in the Cooja simulator and consists of SDN-WISE

nodes, including one controller. The simulated nodes are developed using the Contiki-NG [3] operating system and the SDN-WISE framework [8]. Contiki-NG is an open-source operating system designed for low-power and memory-constrained networked devices used in wireless IoT applications. It provides a lightweight environment for developing embedded network applications and is widely used in WSN and IoT research. SDN-WISE provides the software-defined architecture for the experimental WSN and offers a programmable and modular structure for managing network behavior. This allows the communication and forwarding behavior of the network to be configured and modified for different experimental scenarios.

The Cooja simulator is used to execute the Contiki-NG-based SDN-WISE nodes. Cooja provides a simulation and emulation environment for wireless sensor networks and supports the execution of Contiki-NG applications in a simulated network. This enables investigation of the behavior of the experimental WSN without requiring physical deployment of sensor nodes.

Together with its core SDN control functions, the controller node also acts as a bridge between the simulated WSN and the external MQTT cluster. A C-based socket connection is used to transfer data from the WSN to a Python-based integration script. The script receives the data through the socket and converts it into MQTT messages using an MQTT client. These messages are then published to the Mosquitto-based MQTT cluster, making the WSN-generated data available to external and cloud-based services. This WSN-MQTT integration constitutes the experimental implementation evaluated in this study and provides a controlled and repeatable basis for examining how different simulated network conditions affect the traffic delivered to the IoT infrastructure.

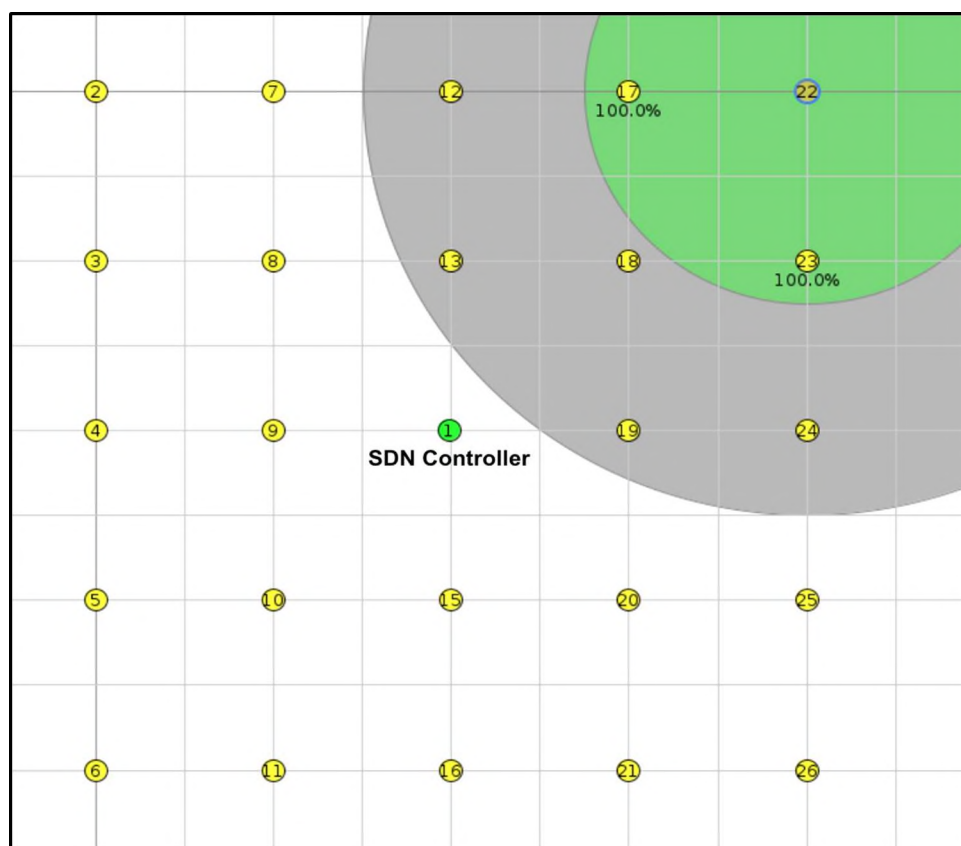


Figure 2. Grid-Based WSN Topology implemented in the Cooja simulator

Experimental Results and Discussions

This section presents the experimental assumptions and setup used to evaluate the proposed simulation-driven framework, followed by the obtained results and their discussion.

Experimental Assumptions and Setup

The simulations were conducted using the Cooja simulator. The main experimental parameters are summarized in Table 1. All of these parameters were kept constant in all simulation scenarios. The interarrival time was set to 8 s, meaning that each sensing node generated a sensor message at eight-seconds intervals. The message size was fixed at 6 bytes to ensure that the payload size remained consistent across all experiments. The TX/RX success ratio was set to 100% to establish an idealized and controlled channel condition, thereby isolating the effect of network size and WSN configuration without introducing packet losses caused by unreliable wireless links.

These fixed parameters enabled the evaluation of the effect of the number of sensing nodes on the generated network and MQTT traffic under comparable conditions.

Table 1. Simulation Parameters

Parameter	Value/Model
Radio Medium	Unit Disk Graph Medium
Radio	CC2650
MAC driver	ContikiMAC
TX/RX success ratio	100%
Transmission range	50 m
Interference range	100 m
Distance between adjacent nodes	20 m
Interarrival Time	8 s
Message size	6 Bytes

The simulations were performed using three different network sizes: 24, 36, and 48 sensing nodes. These scenarios were used to examine the behavior of the proposed WSN–MQTT integration as the number of traffic-generating nodes increased. The remaining experimental conditions were kept unchanged across the scenarios.

Figure 3 shows the experimental environment for the 24-node scenario. The right side of the figure shows the Cooja simulation environment, which contains 24 sensing nodes and one SDN controller node, labeled Node 1 in green. The sensing nodes generate SDN-WISE data packets, while the controller performs SDN control functions and serves as the communication bridge between the simulated WSN and the external MQTT infrastructure.

The controller node establishes a client socket connection with a server socket created by the Python-based gateway script, `mitm.py`, shown in the lower-left part of Figure 3. Through this socket connection, UART data generated by the controller node is transferred from the Cooja simulation environment to the gateway script. The related Python script receives SDN-WISE packets, extracts the sender information and the message payload, and converts the data into MQTT messages. The messages are then published to the `test/topic` topic in the Mosquitto-based MQTT cluster.

The upper-left part of Figure 3 shows a `mosquitto_sub` client subscribed to the `test/topic` MQTT topic. This subscription was used to verify that messages generated by the simulated sensing nodes were successfully transferred through the controller and gateway script and

published to the MQTT infrastructure. Therefore, the experimental setup provides an end-to-end path from the simulated WSN to the external MQTT cluster:

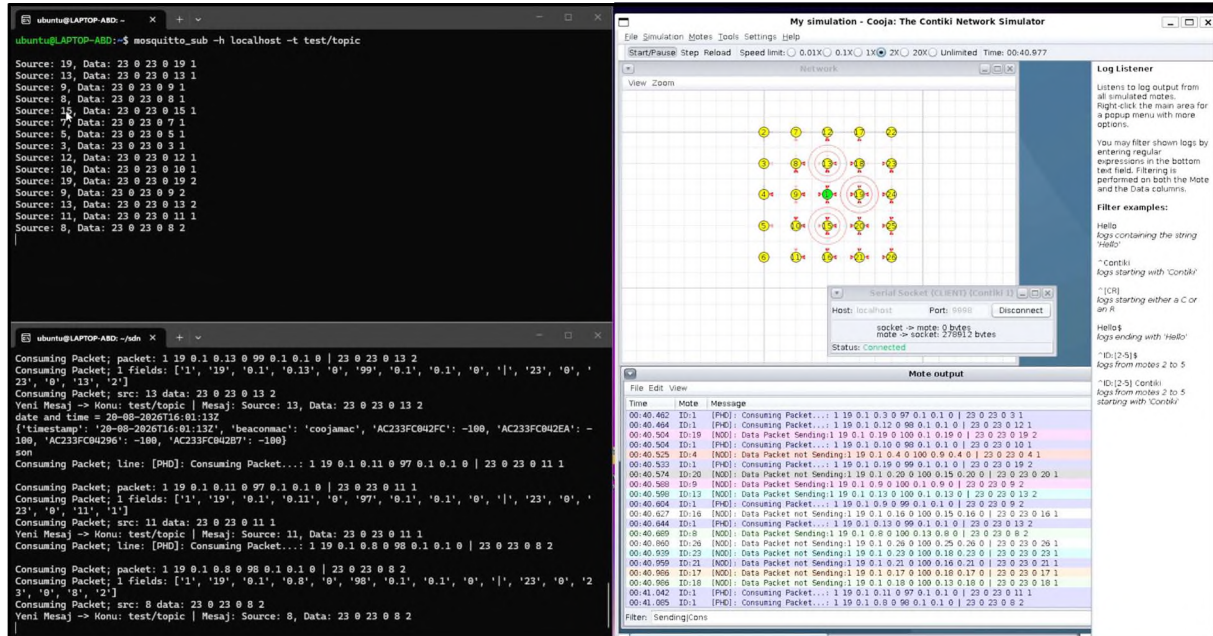


Figure 3. Experimental setup of the Cooja-based WSN and MQTT integration for the 24-sensing-node scenario

Performance Evaluation and Discussions

In this subsection, the simulation results for the defined network scenarios are presented and analyzed. The performance of the proposed WSN–MQTT integration is evaluated under different network sizes, followed by a comparison and discussion of the obtained results.

Figure 4 presents the average end-to-end delay measured for each sensing node in the 24-node network scenario. The results show that the delay varies among nodes, ranging approximately from 26.79 ms to 77.99 ms. Most nodes have an average delay of 27–40 ms, while a small number experience higher delays. These differences are primarily due to variations in communication paths and network conditions within the simulated WSN. The results demonstrate that the proposed framework can capture node-level differences in end-to-end communication performance, rather than only providing an overall network average.

Figure 5 presents the average end-to-end delay for the three network scenarios. The results show that the average delay increased with the number of sensing nodes. The average delay was 37.51 ms, 100.30 ms, and 138.01 ms for the 24-, 36-, and 48-node scenarios, respectively. This increase is associated with the higher volume of network traffic and communication load generated as more sensing nodes are added to the simulated WSN. The results demonstrate that changes in the WSN configuration are reflected in the end-to-end performance of the proposed WSN–MQTT integration.

Figure 6 illustrates the Total Application Throughput (bps) obtained under the three network configurations. An upward trend in throughput can be observed as the number of sensing nodes increases. The measured throughput values are 142.51 bps, 212.78 bps, and 281.12 bps for the 24-, 36-, and 48-node scenarios, respectively. This behavior indicates that the proposed WSN–MQTT integration is able to process and deliver a greater amount of

application-level data as the network scale expands. The observed throughput variation therefore reflects the effect of network size on the data delivery capacity of the overall system.

The Packet Delivery Ratio (PDR) results for the three network scenarios indicate a highly reliable data delivery performance of the proposed WSN–MQTT integration. The PDR was 100.00000% for the 24-node scenario, while it remained at 99.99996% and 99.99988% for the 36- and 48-node scenarios, respectively. Although a slight decrease in PDR can be observed as the number of sensing nodes increases, the change is negligible and the delivery ratio remains effectively close to 100% in all scenarios. This result indicates that the increase in network size has a minimal impact on packet delivery reliability within the simulated conditions. Overall, the results demonstrate that the proposed framework maintains highly reliable communication even as the number of sensing nodes increases.

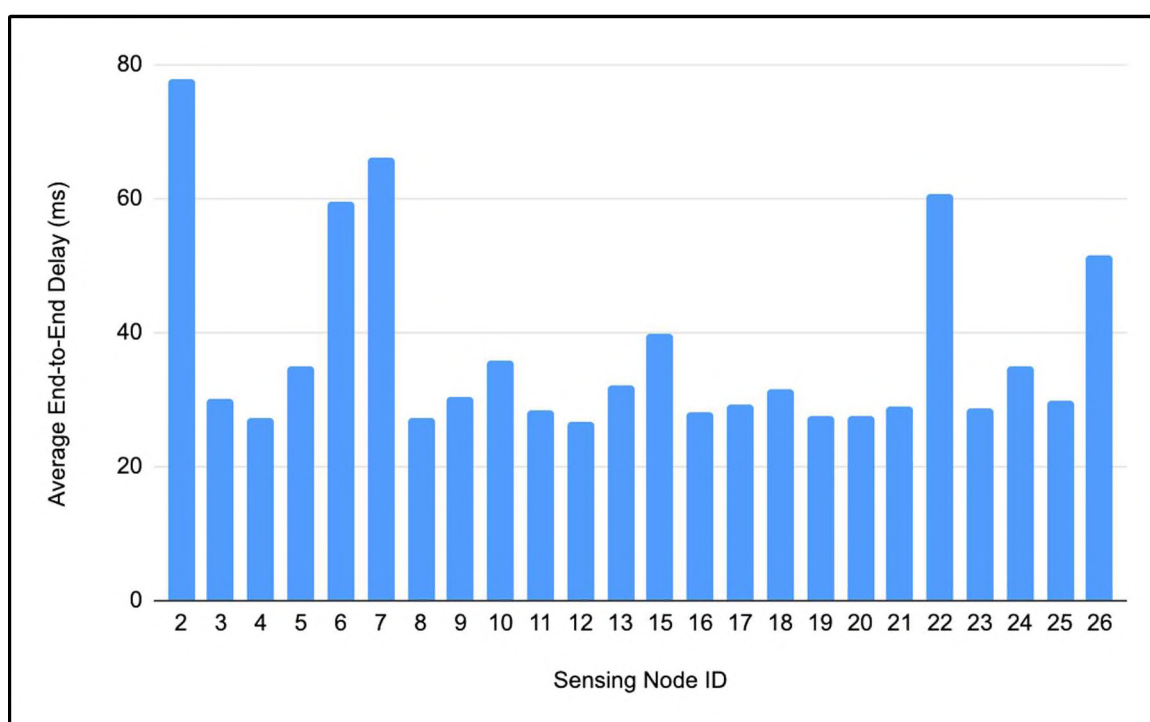


Figure 4. Average End-to-End Delay per Sensing Node in the 24-Node network scenario

To further evaluate the communication overhead of the proposed WSN–MQTT integration, the total network traffic was compared between the baseline condition without application-level data transmission and the corresponding condition with application communication enabled. As shown in Table 2, the total traffic increased from 2,369,624 to 3,046,445 bytes in the 24-node scenario, corresponding to a 28.56% increase. For the 36- and 48-node scenarios, the traffic increase was 33.63% and 38.43%, respectively. The increasing trend indicates that the additional communication load introduced by application-level data transmission becomes more pronounced as the network size increases. These results provide an additional perspective on the communication cost and scalability of the proposed WSN–MQTT integration, complementing the latency, throughput, and PDR analyses.

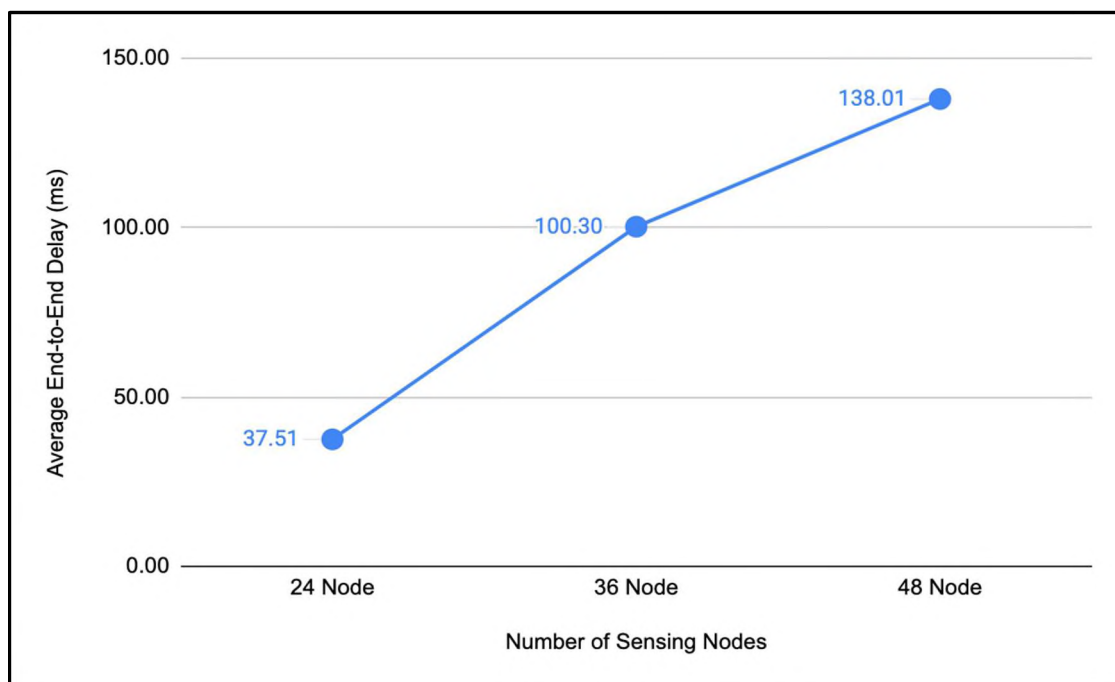


Figure 5. Average End-to-End Delay under different network scenarios

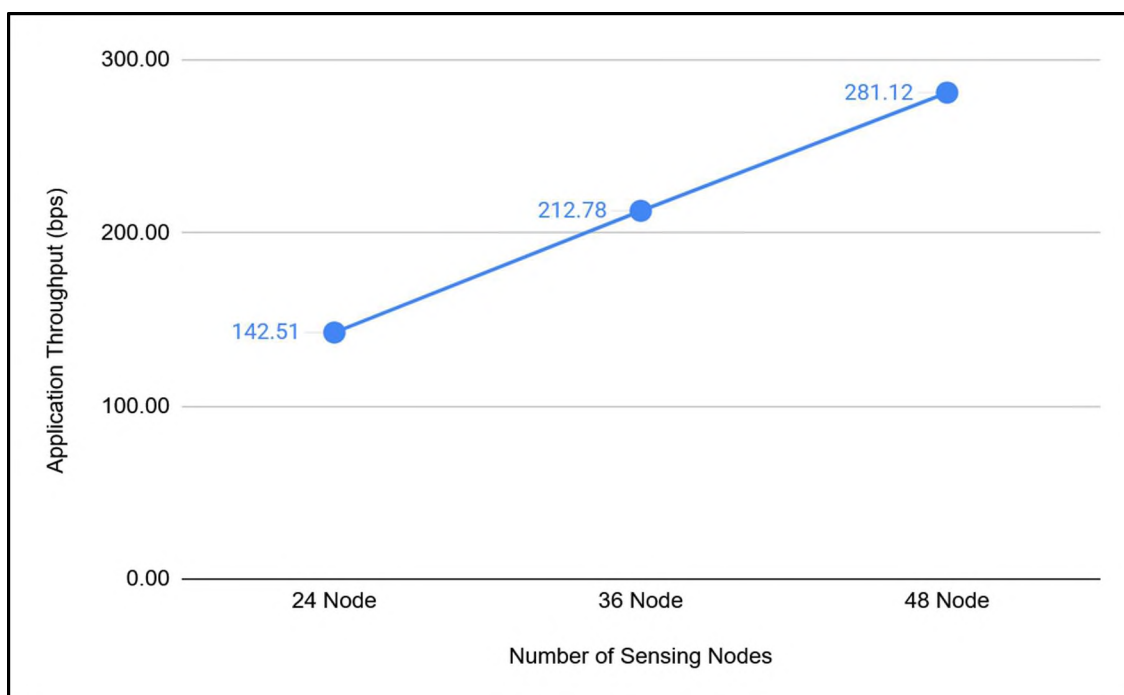


Figure 6. Total Application Throughput (bps) under different network scenarios

Table 2. Comparison of baseline and application-enabled communication traffic across network scenarios

Network Scenario	Baseline Traffic (Byte)	Traffic with Application (Byte)	Additional Traffic (Byte)	Traffic Increase (%)
24 Node	2,369,624	3,046,445	676,821	28.56%
36 Node	5,547,224	7,412,941	1,865,717	33.63%
48 Node	8,050,967	11,144,798	3,093,831	38.43%

As tabulated in Table 3, the computational time required to execute the simulation increased as the network size increased. Although all scenarios represent the same 3600 s of simulated time, the execution time increased from 527.875 s for the 24-node scenario to 697.668 s and 963.927 s for the 36- and 48-node scenarios, respectively. This increase indicates that larger network configurations require greater computational effort due to the increased number of network entities and communication events.

Table 3. Simulation execution time for different network scenarios

Network Scenario	Simulated Time (s)	Simulation Runtime (s)
24 Node	3600	527.875
36 Node	3600	697.668
48 Node	3600	963.927

The results also demonstrate the flexibility of the proposed framework in evaluating different WSN configurations. By modifying the simulated WSN while keeping the MQTT and external IoT infrastructure unchanged, alternative network sizes and configurations can be tested without requiring a new physical WSN deployment. This simplifies comparative evaluation and supports efficient pre-deployment network design.

Conclusions

The framework presented in this study integrates an SDN-WISE-based WSN implemented in the Cooja simulator with an MQTT-based communication infrastructure. Unlike conventional approaches that generate test traffic independently, the MQTT workload in the proposed framework originates directly from the simulated WSN. The experimental results from the 24-, 36-, and 48-node scenarios showed that the average end-to-end delay increased with the number of sensing nodes, yielding delays of 37.51 ms, 100.30 ms, and 138.01 ms, respectively. At the same time, total application throughput increased from 142.51 bps to 212.78 bps and 281.12 bps as the network size increased, reflecting the greater volume of application data generated by larger WSN configurations. Packet delivery remained highly reliable, with PDR values of 100.00000%, 99.99996%, and 99.99988%, respectively. The additional network traffic associated with application communication also increased from 28.56% for the 24-node scenario to 33.63% and 38.43% for the 36- and 48-node scenarios.

These results demonstrate that the framework can capture the effects of WSN configuration on latency, data delivery, and communication load in the integrated IoT infrastructure.

Overall, the results demonstrate the feasibility of using a simulated WSN as a traffic source for evaluating IoT performance. The proposed framework provides a practical link between WSN simulation and IoT infrastructure, supporting network-aware design and pre-deployment evaluation of IoT systems. Future work will focus on evaluating additional network configurations and communication technologies, and on extending the experimental setup to integrate higher-level IoT services such as data processing, storage, and analytics.

Acknowledgment

Artificial Intelligence Statement: Artificial intelligence tools were used to assist with the preparation and language editing of this manuscript. The authors determined the scientific content, methodology, experimental design, analysis, and interpretation of the study. AI-generated suggestions were critically evaluated, filtered, and revised by the authors before inclusion. The authors take full responsibility for the final content, accuracy, originality, and integrity of the manuscript.

References:

1. Gallegos Ramonet, A., Pecorella, T., Picano, B., Kinoshita, K. "Perspectives on IoT-oriented Network Simulation Systems", *Computer Networks*. 2024, Vol. 253. Art. No. 110749. DOI: <https://doi.org/10.1016/j.comnet.2024.110749>.
2. Idris, S., Karunathilake, T., Förster, A. "Survey and Comparative Study of LoRa-Enabled Simulators for Internet of Things and Wireless Sensor Networks", *Sensors*. 2022, Vol. 22, No. 15. Art. No. 5546. DOI: <https://doi.org/10.3390/s22155546>.
3. Oikonomou, G., Duquennoy, S., Elsts, A., Eriksson, J., Tanaka, Y., Tsiftes, N. "The Contiki-NG Open Source Operating System for Next Generation IoT Devices", *SoftwareX*. 2022, Vol. 18. Art. No. 101089. DOI: <https://doi.org/10.1016/j.softx.2022.101089>.
4. Oğur, N.B., Al-Hubaishi, M., Çeken, C. "IoT Data Analytics Architecture for Smart Healthcare Using RFID and WSN", *ETRI Journal*. 2022, Vol. 44, No. 1, P. 135–146. DOI: 10.4218/etrij.2020-0036.
5. Prajapati, A., Banerjee, B. "Simulating Distributed Wireless Sensor Networks for Edge-AI", 2022 5th Conference on Cloud and Internet of Things (CIoT). 2022. P. 75–82. DOI: <https://doi.org/10.1109/CIoT53061.2022.9766520>.
6. Bekele, B.E., Tokarz, K., Gebeyehu, N.Y., Pochopień, B., Mrozek, D. "Performance Evaluation of UDP-Based Data Transmission with Acknowledgment for Various Network Topologies in IoT Environments", *Electronics*. 2024, Vol. 13, No. 18. Art. No. 3697. DOI: <https://doi.org/10.3390/electronics13183697>.
7. Tanganelli, G., Vallati, C., Mingozzi, E. "SDN-WISE: A Software-Defined Network for the Internet of Things", *IEEE Internet of Things Journal*. 2016, Vol. 3, No. 5, P. 771–784. DOI: <https://doi.org/10.1109/JIOT.2016.2515573>.
8. Galluccio, L., Milardo, S., Morabito, G., Palazzo, S. "SDN-WISE: Design, Prototyping and Experimentation of a Stateful SDN Solution for Wireless Sensor Networks", *Proceedings of IEEE INFOCOM*. 2015, Vol. 26, P. 513–521. DOI: <https://doi.org/10.1109/INFOCOM.2015.7218418>.
9. Anadiotis, A.-C.G., Galluccio, L., Milardo, S., Morabito, G., Palazzo, S. "SD-WISE: A Software-Defined Wireless Sensor Network", *Computer Networks*. 2019, Vol. 159. P. 84–95. DOI: <https://doi.org/10.1016/j.comnet.2019.04.029>.
10. Tcherak, A., Khaoua, M.O., Loucif, S. "Performance Evaluation of SDN-WISE in Mobile Wireless Sensors Networks", 2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAEECCS). 2023, P. 1–6. DOI: <https://doi.org/10.1109/ICAEECCS56710.2023.10104806>.
11. Banks, A., Gupta, R. "MQTT Version 3.1.1", OASIS Standard. 2014.
12. Naik, N. "Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP", 2017 IEEE International Systems Engineering Symposium (ISSE). 2017. P. 1–7. DOI: <https://doi.org/10.1109/SysEng.2017.8088251>.

13. Seoane, V., García-Rubio, C., Almenares, F., Campo, C. “Performance Evaluation of CoAP and MQTT with Security Support for IoT Environments”, Computer Networks. 2021, Vol. 197. Art. No. 108338. DOI: <https://doi.org/10.1016/j.comnet.2021.108338>.
14. Putra, F.P.E., Ubaidi, Holipah, Mahmud, M.A., Paradina, R. “Comparing the Performance of LoRaWAN and MQTT Protocols for IoT Sensor Networks”, Jurnal Informasi dan Teknologi. 2024, Vol. 6, No. 2, P. 221-228. DOI: <https://doi.org/10.60083/jidt.v6i2.565>.
15. Timalisina, A., Khatiwoda, N.R., Dawadi, B.R., Bohara, A., Joshi, S.R., Calafate, C.T., Manzoni, P. “Assessing the Adequacy of MQTT and ZeroMQ for 5G-Enabled V2X Networks”, Electronics. 2025, Vol. 14, No. 22. Art. No. 4509. DOI: <https://doi.org/10.3390/electronics14224509>.

Information about the authors

A.S. Bayraktar – PhD Candidate, Department of Software Engineering, Sakarya, Türkiye; e-mail: asbayraktar@sakarya.edu.tr; ORCID: 0000-0002-7985-8696;

C. Çeken – corresponding author, PhD, Sakarya University, Department of Computer Engineering, Sakarya, Türkiye; e-mail: celalceken@sakarya.edu.tr; ORCID: 0000-0002-6912-0057.