LATINOWARE 2024
21º Congresso Latino-americano de Software Livre e Tecnologias Abertas
Latin.Science 2024

WIMA: An Open-Source API for efficient Image Transmission on Wireless Sensor Networks

Janislley Oliveira De Sousa, Ricardo Nogueira Santos, João Danilo P. Júnior, and Moysés M. Lima
Sidia Science and Technology Institute, Manaus, Brazil
janislley.sousa@sidia.com, ricardo.santos@sidia.com, joao.junior@sidia.com, moyses.mendes@sidia.com

***Abstract*—This paper explores the integration of image sensors into Wireless Sensor Networks (WSN) for Internet of Things (IoT) applications, an area currently limited by technical constraints. It introduces WSN Image API (WIMA), an open-source and adaptable API designed to unify and simulate the stages of image transmission. Developed and designed on the Contiki-NG/Cooja platform, WIMA has demonstrated its viability through simulations of use cases, validating its efficacy in supporting image transmission in WSN environments. The API offers a standardized approach for incorporating image data into WSN, addressing a significant gap in current methodologies to simulate this scenario for IoT applications. Future work will focus on deploying WIMA in real-world scenarios to evaluate and improve its image transmission capabilities.**

***Keywords*—Image transmission; Wireless sensor network; Internet of things; Simulation; Open-source.**

I. Introduction

The recent advancements in Internet of Things technologies have spurred the development of various applications. The IoT represents a system of interconnected smart devices, including sensors and actuators, facilitating communication between objects and other devices. WSNs play a crucial role in the implementation of IoT applications. Comprising a network of sensors that communicate wirelessly, WSNs are key for computational systems to interact with the environment [1]. As more sensors connect to the internet, researchers envision a wide array of applications for IoT, and WSNs are equipped with various sensors to support these diverse IoT applications [2]. However, implementing visual applications in WSNs remains a challenge due to technical constraints that make image transmission difficult. While there are scenarios where using WSNs for visual purposes is advantageous, such as remote areas without wired or Wi-Fi connections, transmitting large amounts of data poses significant challenges due to performance and energy limitations. Additionally, the limited battery life, processing power, and lifespan of sensors in WSNs can hinder their effectiveness in image processing and transmission [3], [4].

Image sensors have not yet become a fully integrated class of devices in WSNs for IoT applications [5]. Traditional WSNs handle simple scalar sensors, but handling images requires more complex solutions [6]. Implementing image compression algorithms in typical WSN hardware is challenging due to limited resources. In this sense, efficient image transmission in WSNs is hindered by factors like interference and limited bandwidth, leading to increased network traffic [7].

This work presents WSN Image API (WIMA), a standardized framework for efficient image transmission in WSN and IoT applications. By applying design patterns and software engineering standards, WIMA unifies all stages of the image transmission process, offering flexibility and extensibility in its application. To achieve this, our approach seeks to develop a design for an API that supports techniques for image transmission, while addressing the constraints and limitations inherent to WSNs. A prototype was implemented on the Contiki-NG/Cooja platform[1], validating the design's efficacy through simulations of use cases, including image transmission in an IEEE 802.15.4 network[2] and 6LoWPAN adaptation layer in Contiki.

The remainder of this paper is organized as follows: Section II reviews the literature on techniques for image transmission in WSNs and identifies current challenges in transmitting images over WSNs for IoT applications. Section III describes the development process of WIMA. The prototyping and analysis conducted are detailed in Section IV. Section V presents the conclusions reached through this research.

II. Related works

WSN image transmission research has explored various techniques to address challenges, including compression algorithms, change detection, transmission protocols, and prioritization methods [6]. This section describes the requisites involved in this process.

Corresponding author: J. Oliveira (janislley.sousa@sidia.com)
Ricardo Nogueira Santos and João Danilo P. Júnior are no longer part of the Sidia Science and Technology Institute. They can be reached at their personal email addresses: *ricardns@gmail.com* and *joaodanilo1992@gmail.com*

[1]https://github.com/contiki-ng/contiki-ng
[2]https://standards.ieee.org/ieee/802.15.4/7029/

Realização: itaipu parquetec | ITAIPU BINACIONAL MAIS QUE ENERGIA | GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO
27 a 29 de novembro de 2024
Foz do Iguaçu | Paraná | Brasil

A. Image Compression

Compression algorithms minimize data size by eliminating redundancy, with researchers refining traditional methods like DCT and DWT. A summary of key attributes for each technique is presented in Table I based on [8], highlighting memory usage, compression ratio, energy consumption, Peak Signal-to-Noise Ratio (PSNR), and time.

Table I: Techniques details for image compression on WSN.

Parameter	LC-DCT	Fast Zonal DCT	Entropy based	IC-DWT	HIC using DWT and DCT	HIC using DWC/DWT
Base	DCT	DCT	DWT	DWT	DWC/DWT	DWC/DWT
Memory	Low	Low	High	High	High	Low
Compress	Low	Low	High	Low	High	High
Energy	Low	Low	High	High	High	Low
PSNR	Low	Low	Medium	High	High	Medium
Time	Low	Low	High	High	High	Low
Energy	High	High	Low	Low	Medium	Medium
Complexity	High	High	Low	Low	Medium	Medium

B. Change Detection Technique

This technique capitalizes on change detection to transmit only modified regions within an image, minimizing data transmission and maximizing network efficiency. It significantly reduces network burden by transmitting only altered information, enabling more efficient communication. According to [9], various change detection methods are examined, including pixel-based, feature-based, and hybrid techniques, highlighting their strengths and limitations. Our strategy updates localized areas only, decreasing network traffic and energy consumption.

C. Transmission Protocol

The study in [10] reviews various transmission protocols developed for Wireless Sensor Networks (WSNs), emphasizing the significance of considering factors such as data type, bandwidth, and security when selecting a protocol. For instance, in [7] the authors prototyped a canola cultivation monitoring system using LoRa, which periodically sends an image to a remote location. This system employs a protocol that compresses the image before sending, with acknowledgments for received packets handled in groups rather than individually.

D. Prioritization Technique

The approach [11] reviewed various techniques for prioritizing components in WSNs, aiming to optimize network performance by assigning different levels of importance to different components, such as prioritizing certain data or traffic types. In [12], the authors proposed a method where the sender transmits only the most significant image components, focusing on key information pixels, enabling receivers to achieve higher PSNR in less time compared to traditional methods. This approach is beneficial in scenarios where obtaining at least a basic image quickly is crucial [13]–[19].

III. WSN IMAGE API (WIMA)

Given the general problem of modeling the transmission of an image in a WSN, this section provides details on the architecture of the open-source API developed and integrated into the Contiki-NG system through the Cooja simulator.

A. Modeling Design Approach

A design model for image transmission over WSN should include algorithms and methods to reduce, fragment, and transmit data. These components are necessary to transmit the image over the WSN. Figure 1 summarizes the challenges identified in the literature review and implemented by WIMA.

To design an adaptable API, we employed a modular approach based on software engineering principles. Our design methodology comprised several key steps:

Componentization: We identified key techniques from the literature on image transmission in WSNs and IoT systems, such as compression, encryption, and error correction.

Modular Architecture: API designed with a modular architecture that allows for easy integration of new components or replacement of existing ones.

Interface Definition: A standardized interface for each component to ensure seamless communication between them.

Component Interoperability: Rules and guidelines implemented to ensure compatibility among components.

B. WIMA Architecture

This research defines an API as a fundamental characteristic, the ease of adaptation and extension for use in IoT applications, allowing the experimentation of various identified scenarios: change detection techniques, low-processing image compression algorithms, and transmission techniques that minimize data traffic. Table 2 presents a summary of the approaches and challenges found in the image transmission literature, comparing them according to their implementations by WIMA.

The evaluation of solutions for a WSN involves diverse scenarios, as techniques are often implemented with variations and arrangements. To facilitate modeling of image transmission in a WSN, the following key components were conceptualized:

- **Slicer** (***Slicer***): A network has a maximum size of data that can be transmitted at once. If the size of the transmitted data block is greater than a frame of the network, the fragmentation into smaller parts is required.
- **Image** (***tImage***): Represents the image data information to be transmitted over the network.

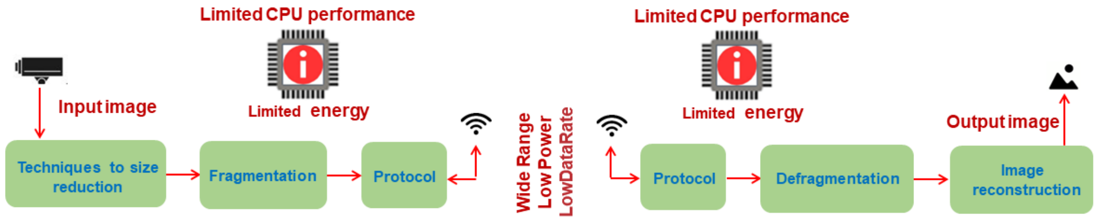

Figure 1: Challenges in image transmission over WSNs designed for the WIMA.

Table II: Comparative analysis of literature studies and the proposed API.

Studies	Priorization Technique (Slicer)	Change Detection	Image Compression	Transmission (AppProtocol)
Lecuire et al., 2012 [13]	✓		✓	
Mozammel et al., 2012 [14]			✓	
Kouadria et al., 2013 [15]	✓		✓	
Bharath et al., 2013 [16]	✓		✓	
Kishk et al., 2014 [17]	✓		✓	
Pham C., 2014 [20]	✓	✓	✓	
Wu et al., 2016 [18]	✓		✓	
Deepthi et al., 2017 [19]	✓		✓	
Chen et al., 2019 [7]	✓		✓	✓
Felemban et al., 2020 [12]	✓			✓
WIMA	✓	✓	✓	✓

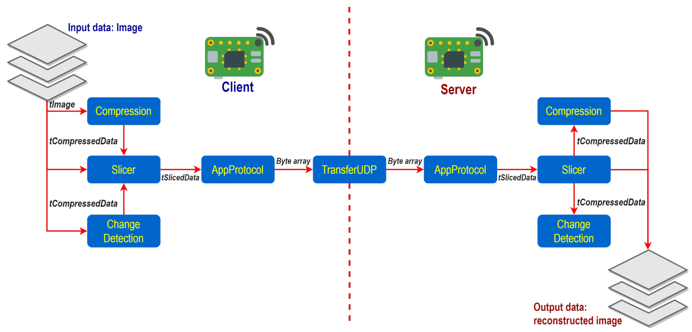

Figure 2: WIMA model diagram communication flow during image transmission on both the client and server side.

- **Image database (*ImageDb*)**: To take advantage of the simulator environment, the component represents a group of ready-to-use images to automate the simulation.
- **Application Protocol (*AppProtocol*)**: It is used to coordinate data transmission between transmitter and receiver.
- **Communication (*Transfer*)**: An entity to interface the transport layer, in which the data will be transmitted.
- **Change Detector (*ChangeDetector*)**: Component to implement the change detection algorithm.
- **Image Compressor (*Compression*)**: This component is used to represent the image compression algorithm.
- **Data Compressed (*CompressedData*)**: When a compressor exists, it is used to represent the compressed data.

C. WIMA Model

The communication protocol within a WSN involves two primary elements: a transmitter (client) that sends data and a receiver (server) that receives it. Figure 2 depicts the detailed communication process between the API components, which encompasses their attributes, methods, and interactions.

IV. API Prototyping and Design

To validate the WIMA API architecture and implementation prototype, simulations are conducted in Cooja, utilizing use case scenarios that define transmitter (client) and receiver (server) entities. The WIMA prototype was developed in C for Contiki-NG, with each API component crafted as a data structure incorporating function pointers. The API's source code and documentation are publicly available under the Apache 2.0 license on WIMA's GitHub repository[3]. The detailed user cases and API documentation can be found in GitHub.

Given the main problem of modeling image transmission in a WSN, during the simulation, the compression steps are executed through the image compression algorithm implemented by WIMA. If an image was already compressed, it is necessary to consider a component that represents the compressed data. Also, The change detection component is used to implement the change detection algorithm. The integration of compression, change detection, and fragmentation elements within the API framework enables efficient data transmission in a WSN.

Using compression techniques during the simulation phase is crucial for reducing the size of image data sets and optimizing network traffic. By implementing compression algorithms, the API can minimize the amount of data to be transmitted, thereby conserving network resources and energy consumption. In parallel, the incorporation of the change detection component facilitates the implementation of change detection algorithms, which further enhance the efficiency of data transmission. By

[3]https://github.com/RicardoNogueiraSantos/WIMA

identifying unchanged regions in a sequence of images, unnecessary data transmission can be avoided, leading to significant reductions in data traffic and energy usage within the network. Moreover, the fragmentation component plays a pivotal role in handling image data within the constraints of WSNs, where low transmission rates and limited packet sizes are prevalent. By segmenting large data packets into smaller fragments, the API guarantees smooth and optimized data transmission throughout the network. Also, UDP was used as the primary protocol for WIMA. By utilizing the 6LoWPAN adaptation layer in Contiki, developers can effortlessly integrate UDP for reliable and efficient transmission of IPv6 packets within the network [20].

The integration of the compression, change detection, fragmentation and transmission components is showcased in the use case (more details in Github API documentation), highlighting how these elements work together to optimize data transmission in a WSN environment. The use case diagrams provide a comprehensive overview of the API's operation, emphasizing the seamless flow of data and the efficient communication between the client and server nodes. Given the API's structure, modifying its functionality can be straightforward by implementing a new function with a different behavior and initializing the component with a pointer to this function. If a component needs new functionality, like encoding a unique message for a different application, this can be accomplished by incorporating a pointer to the new function into the component. This approach also underscores the API's ease of extensibility.

V. Conclusion

This work presents a novel open-source API for image transmission in WSNs, enabling standardization and experimentation with various image transmission techniques. A prototype was designed and implemented on Contiki-NG and simulated using Cooja. WIMA encapsulates key techniques from the literature, providing software components for image manipulation in WSNs and IoT systems. Additionally, this work identifies a comprehensive range of techniques and WSN types, optimizing complexity reduction, computational power, energy consumption, and data traffic. The WIMA framework improves the standardization of experimentation and simulation systems, enabling the efficient treatment and transmission of images in Wireless Sensor Networks and IoT applications. For future work, a performance evaluation (level of compression, latency, package loss, and power consumption) will be executed to compare the techniques used for each API component.

Acknowledgment

This article is the result of the R&D&I project *SEICO*, carried out by Sidia Science and Technology Institute in partnership with Samsung Eletrônica da Amazônia Ltd., using resources from Brazilian Federal Law 8,387/1991, and its disclosure and publicity are in accordance with the provisions of article 39 of Brazilian Decree 10,521/2020.

References

[1] M. Altayeb, S. Sharif, and S. Abdella, "The Internet-of-Things and Integration with Wireless Sensor Network Comprehensive Survey and System Implementation," *2018 ICCCEEE*, pp. 1–6, 2018.
[2] M. Majid and *et al.*, "Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review," *Sensors*, vol. 22, no. 6, p. 2087, 2022.
[3] M. Javaid, A. Haleem, S. Rab, R. Pratap Singh, and R. Suman, "Sensors for daily life: A review," *Sensors International*, vol. 2, p. 100121, 2021.
[4] M. Mishra, G. Sen Gupta, and X. Gui, "Investigation of energy cost of data compression algorithms in wsn for iot applications," *Sensors*, vol. 22, no. 19, p. 7685, 2022.
[5] R. Krishnamurthi and *et al.*, "An overview of iot sensor data processing, fusion, and analysis techniques," *Sensors*, vol. 20, no. 21, p. 6076, 2020.
[6] B. A. Lungisani, C. K. Lebekwe, A. M. Zungeru, and A. Yahya, "Image compression techniques in wireless sensor networks: A survey and comparison," *IEEE Access*, vol. 10, pp. 82 511–82 530, 2022.
[7] T. Chen, D. Eager, and D. Makaroff, "Efficient image transmission using LoRa technology in agricultural monitoring IoT systems," in *2019 iThings*, Jul. 2019, pp. 937–944.
[8] N. Patel and J. Chaudhary, "Energy efficient wmsn using image compression: A survey," in *2017 ICIMIA*. IEEE, 2017, pp. 124–128.
[9] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, "Change detection from remotely sensed images: From pixel-based to object-based approaches," *ISPRS Journal*, vol. 80, pp. 91–106, 2013.
[10] S. K. Singh, M. Singh, D. K. Singh *et al.*, "Routing protocols in wireless sensor networks–a survey," *IJCSES*, vol. 1, no. 2, pp. 63–83, 2010.
[11] R. Sharma and D. Mohapatra, "A survey of component prioritization techniques in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1450–1462, 2011.
[12] E. Felemban, A. Naseer, and A. Amjad, "Priority-based routing framework for image transmission in visual sensor networks: Experimental analysis," *IJASCA*, vol. 11, no. 1, pp. 668–677, 2020.
[13] V. Lecuire, L. Makkaoui, and J.-M. Moureaux, "Fast zonal dct for energy conservation in wireless image sensor networks," *Electronics Letters*, vol. 48, pp. 125–127, 01 2012.
[14] M. Mozammel, H. Chowdhury, and A. Khatun, "Image compression using discrete wavelet transform," *International Journal of Computer Science Issues*, vol. 9, 07 2012.
[15] N. Kouadria, N. Doghmane, D. Messadeg, and S. Harize, "Low complexity dct for image compression in wireless visual sensor networks," *Electronics Letters*, vol. 49, pp. 1531–1532, 11 2013.
[16] K. Bharath and G. Padmajadevi, "Compression Using DWT-DCT and Huffman Encoding Techniques for Biomedical Image and Video Applications," *IJCSMC*, vol. 2, no. 5, pp. 255–261, 2013.
[17] A. Kishk, N. Messiha, N. El-Fishawy, A. Alkafs, and A. Madian, "Hybrid compression algorithm for wireless sensor network," *Journal of Advances in Computer Networks*, vol. 2, pp. 147–150, 06 2014.
[18] A. Mittal, C. Kundu, R. Bose, and R. K. Shevgaonkar, "Entropy based image segmentation with wavelet compression for energy efficient lte systems," in *2016 ICT*, 2016, pp. 1–6.
[19] S. A. Deepthi, E. S. Rao, and M. G. Prasad, "Image compression techniques in wireless sensor networks," in *2017 IEEE ICSTM*. IEEE, 2017, pp. 286–289.
[20] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The contiki-ng open source operating system for next generation iot devices," *SoftwareX*, vol. 18, p. 101089, 2022.