Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

Research Article

A Comparative Study of Wi-Fi Technologies in Wireless Sensor Networks

Mui D. Nguyen[1], Lexter D.A. Tizon[1], Ngoan T. Le[2], Dung T. Nguyen[3], Thang C. Vu[3], Tao V. Nguyen[3], Long Q. Dinh[3], Quang A. Tran[4], Minh T. Nguyen[1*]

[1] Department of Electrical Engineering, Thai Nguyen University of Technology, Thai Nguyen, Vietnam
[2] Department of Economic Policy, Graduate School of International Cooperation Studies, Kobe University, Japan
[3] Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam
[4] Thai Nguyen University, Thai Nguyen, Vietnam
E-mail: nguyentuanminh@tnut.edu.vn

Received: 13 November 2024; **Revised:** 5 February 2025; **Accepted:** 13 February 2025

Abstract: Wireless Sensor Networks (WSNs) have become a fundamental part in various Internet of Things (IoT) applications, such as smart cities, industrial automation, and environmental monitoring. This paper focuses on utilizing Wi-Fi technology within WSNs due to its high data rates and widespread infrastructure, which are essential for real-time monitoring and control applications. We conduct a comparative analysis of popular wireless communication technologies, including ZigBee, Bluetooth, and LoRa, and identify Wi-Fi as a suitable option for scenarios requiring extensive data transmission. The system design involves using ESP32 microcontrollers as sensor nodes to collect and transmit data wirelessly to a central gateway. Performance evaluation demonstrates the reliability and efficiency of the Wi-Fi-based WSN, with notable improvements in data transmission reliability, reduced power consumption using Wi-Fi 6's Target Wake Time (TWT) feature, and low-latency performance suitable for real-time applications. Despite the challenges posed by high power consumption and susceptibility to interference, hybrid solutions combining Wi-Fi with other low-power technologies like ZigBee or LoRa are suggested to enhance energy efficiency and coverage. This study highlights Wi-Fi's potential in WSNs and its applicability to a wide range of IoT implementations.

***Keywords*:** Wireless Sensor Networks, Wi-Fi, IoT, IoT applications, data transmission, Wi-Fi 6, hybrid solutions

1. Introduction

Wireless Sensor Networks (WSNs) are composed of spatially distributed sensors that collect and monitor environmental and physical parameters like temperature, humidity, and motion [1]. These networks are crucial for many IoT applications, including smart cities, healthcare, industrial automation, and environmental monitoring in [2]. By wirelessly transmitting data to a central location or server for analysis, WSNs enable real-time monitoring without the need for extensive cabling or any infrastructure of communications [3].

When gathering data from nodes and sending it to a base station (BS), as shown in Figure 1 as one example of s clustered WSN, one technique is data aggregation that includes a wireless communication technology, such as Wi-Fi, Bluetooth, LoRa, etc. [4]. The method is not only for routing sensing data in such networks but also for reducing energy consumption and getting rid of network redundancy. Data aggregation is the process of sending aggregated information to the base station by combining data from numerous sensors to minimize needless data transmission. There exist many

DOI: https://doi.org/10.37256/cnc.3120256070

routing methods to support the networks to aggregate sensing data, such as tree-based, random walk based, etc [5]. At the source node, data from multiple sensors is combined, and the combined data is then sent to the BS. In order to provide the BS with useful energy-accumulated data information, data aggregation is a technique that combines data from several sensors and minimizes redundant data broadcasts. Data aggregation removes less expensive data transmission and improves network sustainability over the long run. Wireless technologies support data aggregation with those tasks [6].

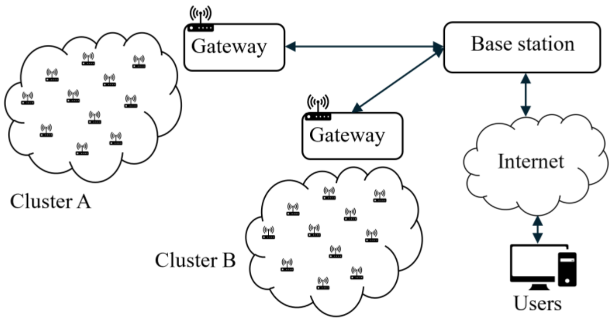

Figure 1. An architecture of Wireless Sensor Networks

There are many wireless communication technologies, as shown in Table 1, deploying in WSNs, such as ZigBee, Bluetooth, Wi-Fi, … Each one offers distinct advantages and trade-offs differently [7]. First, Wi-Fi, based on the IEEE 802.11 standards, has become popular due to its high data rates and wide coverage, making it suitable for applications requiring substantial data transmission [8]. As IoT expands, the need for efficient communication technologies in WSNs increases. Wi-Fi's ability to provide high throughput makes it ideal for applications such as video surveillance, real-time analytics, and industrial control [9]. However, Wi-Fi's high-power consumption and susceptibility to interference pose challenges, particularly in energy-constrained environments where sensor nodes rely on batteries or energy-harvesting technologies [10]. Next, Bluetooth technology, particularly Bluetooth Low Energy (BLE), has become a popular choice in Wireless Sensor Networks due to its low power consumption, cost-effectiveness, and compatibility with a wide range of consumer devices. Operating in the 2.4 GHz ISM band, BLE supports efficient short-range communication, making it ideal for applications like healthcare monitoring, smart home systems, and environmental sensing. Also, ZigBee operates with much lower power consumption and is optimized for low-data-rate applications, making it ideal for remote monitoring and home automation in [11, 12]. Finally, LoRa, on the other hand, is designed for long-range, low-power communication and, while offering lower data rates than Wi-Fi, can cover distances over 10 km, making it suitable for rural or remote WSN deployments in [13]. LoRa (Long Range) is a prominent wireless communication technology in Wireless Sensor Networks (WSNs) known for its long-range capabilities and low power consumption. Operating in unlicensed sub-GHz frequency bands (e.g., 433 MHz, 868 MHz, and 915 MHz) in [14, 15], LoRa can transmit data over several kilometers, making it ideal for applications in smart agriculture, environmental monitoring, and smart cities. In [16], when there are many devices from different manufacturers, which can cause compatibility issues, ZigBee does not work with devices made by other brands. In contrast, Wi-Fi and Bluetooth are easier to use. Wi-Fi makes it simple to connect most IoT devices, and Bluetooth is commonly available in many electronic devices, such as mobile phones.

Table 1. A comparison of various wireless communication technologies

Technology	Frequency band	Data rate	Range	Power consumption	Application example
Wi-Fi	2.4/5 GHz	Up to 9.6 Gbps	30–200 m	High	Video surveillance, Smart homes
ZigBee	2.4 GHz	250 kbps	10–100 m	Low	Home automation, Smart lighting
Bluetooth	2.4 GHz	Up to 3 Mbps	10–100 m	Moderate	Wearables, Personal devices
LoRa	Sub-GHz	<50 kbps	>10 km	Very low	Remote sensing, Rural monitoring

This paper focuses on analysis of Wi-Fi technologies with related applications for monitoring objects with videos surveillance, transmitting data collected from sensors, displaying real-time information, etc. The technology provide high data rates for the covered areas and also transmit control signal to operate actuators. We analyze the technology and compare with others based on some criterions as transmission rates, power consumption, latency, in wireless communication fields. As Wi-Fi technology is chosen for design and implementation in different applications, this work also enables possibilities to connect to IoT networks with potential approaches.

The remainder of this paper is organized as follows. Section 2 provides background information on Wi-Fi and other wireless communication technologies commonly used in WSNs, along with a comparative analysis of their features. Section 3 addresses the system design and implementation of a Wi-Fi-based WSN, focusing on general real projects. Section 4 presents a performance evaluation of the system, analyzing data reliability, power consumption, and latency under different conditions. Conclusions and future study with advanced wireless technologies for the WSNs are provided in Section 5.

2. Background of Wi-Fi technology in WSNs

Wi-Fi operates on both the 2.4 GHz and 5 GHz frequency bands, supporting data transfer rates of up to several gigabits per second with modern standards like 802.11ac and 802.11ax (Wi-Fi 6) seeing more in Table 2 in [17]. Its high data rate and broad range make it a compelling choice for applications requiring rapid data transmission. Moreover, Wi-Fi's established infrastructure in urban areas reduces deployment costs for WSNs, as existing networks can be utilized in [12].

Table 2. Wi-Fi characteristics

Characteristic	Specification
Frequency bands	2.4 GHz, 5 GHz
Data rates	Up to 9.6 Gbps (Wi-Fi 6)
Typical range	30–100 m indoors
Power consumption	High
Standards	IEEE 802.11a/b/g/n/ac/ax

Table 3 presents a comparison of the advantages and disadvantages of Wi-Fi across three key aspects: Data Rate, Coverage, and Compatibility. In terms of data rate, Wi-Fi provides high throughput, making it efficient for handling large volumes of data. However, this advantage is offset by its high-power consumption during continuous operation. For coverage, Wi-Fi benefits from extensive infrastructure availability, which facilitates widespread access. Nonetheless, it is prone to interference, especially in crowded environments, which can degrade performance. Regarding compatibility, Wi-Fi supports a wide range of devices, making it highly versatile. However, in densely populated areas with many connected devices, efficiency can decrease, leading to potential performance issues [18, 19, 20].

Table 3. Advantages and disadvantages of Wi-Fi

Aspect	Advantages	Disadvantages
Data rate	High throughput for large data volumes	High power usage in continuous operation
Coverage	Extensive infrastructure availability	Susceptible to interference in crowded areas
Compatibility	Wide range of compatible devices	Reduced efficiency in dense deployments

In [21, 22], Table 4 highlights three commonly used Wi-Fi-enabled devices: ESP8266, ESP32, and Raspberry Pi, comparing their processor types, operating frequencies, power consumption, and typical use cases. The ESP8266 features a single-core processor operating at 2.4 GHz, with moderate power consumption, making it suitable for simple IoT projects. The ESP32 has a dual-core processor, also at 2.4 GHz, but stands out with its low power consumption, making it an ideal choice for applications like Wireless Sensor Networks (WSNs) and smart sensors. Lastly, the Raspberry Pi, equipped with a quad-core processor and supporting both 2.4 GHz and 5 GHz frequencies, consumes more power, making it better suited for complex processing tasks.

Table 4. Common devices using with Wi-Fi

Device	Processor	Frequency	Power consumption	Use case
ESP8266	Single-core	2.4 GHz	Moderate	Simple IoT projects
ESP32	Dual-core	2.4 GHz	Low	WSNs, Smart sensors
Raspberry Pi	Quad-core	2.4/5 GHz	High	Complex processing tasks

Among these options, the ESP32 is a widely used device, especially in WSN applications, due to its low power consumption and dual-core capabilities. Its efficiency in handling wireless communication and sensor data processing while maintaining minimal energy use makes it an optimal choice for battery-powered or low-energy environments, such as smart homes and industrial IoT applications. Additionally, the ESP32's integrated Wi-Fi and Bluetooth support provide enhanced connectivity options, further solidifying its role in WSN implementations seeing more in Table 5. The ESP32 microcontroller used in this study supports IEEE 802.11ax (Wi-Fi 6) in addition to earlier standards (b/g/n). Wi-Fi 6 features, such as Target Wake Time (TWT), were utilized in this evaluation to optimize energy efficiency in Section 4.2.

Table 5. Specifications of Wi-Fi-enabled sensor node (ESP32)

Component	Specification
Processor	Tensilica LX6 Dual-Core
Clock speed	240 MHz
Memory	512 KB SRAM, 4 MB Flash
Wireless LAN	IEEE 802.11 b/g/n/ax (Wi-Fi 6)
Bluetooth	Classic/LE
Power	3.3 V (5 V via USB)
Dimensions	48 × 26 × 11.5 mm
Weight	10 g

3. Wi-Fi-based system design and implementation

The Wi-Fi-based WSN system is designed for adaptability, scalability, security, and real-time operation to implement the requests presented at the end of Section 1. The system architecture in Figure 2 consists of multiple sensor nodes forming a wireless sensor network. These nodes are designed to collect and transmit data wirelessly. The gathered data is sent to a central Gateway, which acts as an intermediary between the sensor network and external networks. The gateway forwards the data to the Internet, enabling remote access and monitoring. The data can then be accessed by various end-user devices, such as laptops and smartphones, through the Internet, allowing for real-time analysis and visualization.

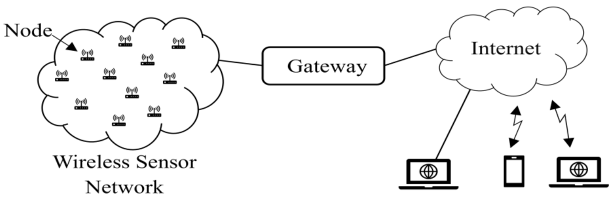

Figure 2. A system architecture of a Wi-Fi-based WSN

The block diagram in Figure 3 depicts a node using an ESP32 microcontroller. It comprises a power supply, a CPU, a Wi-Fi module, sensors, and an actuator. The power supply provides power to the entire system. The CPU acts as the brain, processing data from the sensors and controlling the actuator. The Wi-Fi module enables communication with the network, while the sensors gather environmental data. The actuator performs actions based on the received commands. This architecture forms the foundation of IoT applications, allowing devices to interact with their surroundings and communicate data wirelessly.

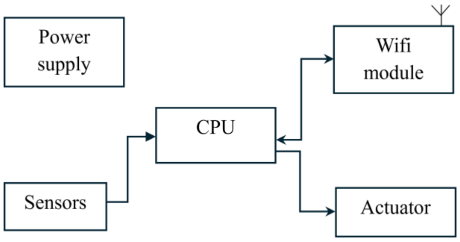

Figure 3. Block diagram of a node using ESP32 microcontroller

A node operates in Figure 4 as a central unit for controlling actuators and collecting data via Wi-Fi in this system. Upon startup, the ESP32 initializes its settings and checks for a Wi-Fi connection. If connected, it listens for control signals from a remote interface. When a control signal is received, the ESP32 activates the connected actuator accordingly. If no control signal is detected, the microcontroller proceeds to gather data from sensors, such as temperature or motion detectors. This collected data is then transmitted over Wi-Fi to a remote server or monitoring application for real-time observation. The process continuously loops, ensuring that the system maintains connectivity, controls actuators as needed, and provides up-to-date sensor data for monitoring purposes.

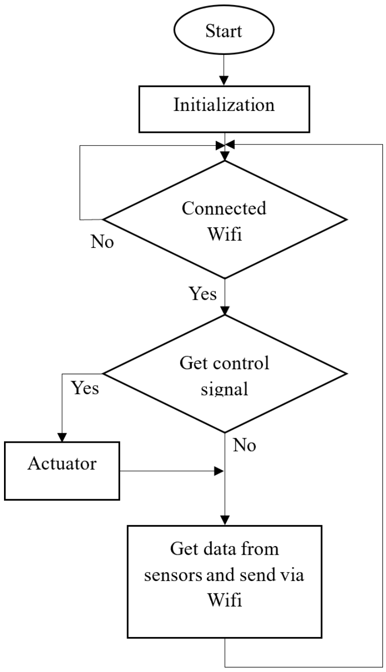

Figure 4. The flowchart illustrates the operation of a node using an ESP32 microcontroller

The system utilizes ESP32-based sensor nodes equipped with temperature sensors, with a total of 10 nodes deployed in a grid layout, spaced 5 meters apart. Each sensor collects data at a polling frequency of 1 Hz, chosen to balance granularity with energy efficiency because the temperature changes very slowly over time. The nodes are positioned 1 m above the ground to optimize signal transmission and minimize ground-level interference. The Wi-Fi network operates on the 2.4 GHz band using the IEEE 802.11ax standard, and signal and interference levels are measured with tools such as a Fluke Networks analyzer or Wireshark to capture RSSI and SINR values. Channel interference is assessed through spectral scans across the deployment area. The environmental model for rural areas assumes an open-field setting with minimal interference, using a path loss exponent of 2, while urban areas are modeled with dense building layouts and high device density, resulting in a path loss exponent of 3.5 and significant signal attenuation due to obstacles.

One issue that arises is that the Forward Error Correction (FEC) algorithm cannot be programmed directly on the ESP32 and cannot be directly tuned to evaluate Data Transmission Reliability, but in Wi-Fi 6, FEC is integrated at the physical layer (PHY) through Modulation and Coding Models (MCS), Specifically, MCS6 to MCS9 levels use Low-Density Parity-Check, a more advanced error correction method than Binary Convolutional Codes used at lower MCS levels. Depending on the specific network conditions, the MCS level is selected appropriately to enhance reliability. In addition, the ESP32 can combine FEC with Hybrid Automatic Repeat Request (HARQ) at the MAC layer. If the error cannot be corrected by FEC, HARQ will trigger a retransmission mechanism for damaged packets, ensuring data integrity.

4. Performance evaluation based on existing studies

The system was evaluated under various conditions to measure key performance indicators, including data transmission reliability, power consumption, and latency. With two primary types of traffic which are periodic sensor data and real-time video streams. Sensor data traffic, originating from temperature sensors, was transmitted at 1 Hz, representing a low-bandwidth scenario typical of environmental monitoring. Real-time video traffic was introduced to simulate high-bandwidth, latency-sensitive applications like video surveillance. The performance of sensor data, including Packet Delivery Rate (PDR) and power consumption, was prioritized, while for real-time video, latency was the key metric. Latency was measured under different network conditions to evaluate its suitability for applications that require minimal delay.

4.1 *Data transmission reliability*

The Data Transmission Reliability in wireless sensor networks, particularly using Wi-Fi, is heavily influenced by environmental factors such as interference, congestion, and network conditions. The performance of Wi-Fi in different environments, from rural areas to urban settings, can vary significantly. This section quantifies and models these impacts and explores techniques, such as Forward Error Correction (FEC) and channel hopping, which can improve transmission reliability.

Packet Delivery Rate (PDR): The Packet Delivery Rate (PDR) calculated by Equation (1) is a crucial metric for evaluating data transmission reliability. It is defined as the ratio of the number of successfully received packets to the total number of transmitted packets:

$$PDR = \frac{N_{delivered}}{N_{transmitted}} \tag{1}$$

where, $N_{delivered}$ is the number of packets successfully received at the destination. $N_{transmitted}$ is the total number of packets sent.

Impact of Interference on Packet Delivery Rate: Interference from competing devices and environmental noise can degrade the packet delivery rate as a function of SINR (PDR_{SINR}). The relationship between the Signal-to-Interference-plus-Noise Ratio (SINR) and PDR is calculated by Equation (2):

$$PDR_{SINR} = \frac{1}{1+\left(\frac{1}{SINR}\right)^{\gamma}} \tag{2}$$

where, $SINK = \frac{S}{I+N}$ is the signal-to-interference-plus-noise ratio, where S is the signal power, I is the interference power, and N is the noise power. γ is constant based on the modulation and channel conditions.

Forward Error Correction (FEC): To combat packet loss due to interference and channel errors, Forward Error Correction (FEC) is used. FEC adds redundancy to transmitted data, allowing the receiver to detect and correct errors without needing retransmissions. The FEC coding rate R_{FEC} is calculated by Equation (3):

$$R_{FEC} = \frac{N_{data}}{N_{total}} \tag{3}$$

where, N_{data} is the number of actual data bits. N_{total} is the total number of bits after redundancy is added (including error-correcting bits).

The reliability improvement from FEC can be modeled by an improvement factor applied to the original PDR:

$$PDR_{FEC} = PDR_{original}\left(1+\alpha\frac{R_{FEC}}{R_{FEC}+\beta}\right) \tag{4}$$

where, $PDR_{original}$ is the packet delivery rate without FEC. α and β are constants based on the environment and the specific FEC scheme used. R_{FEC} is from substituting Equation (3) into Equation (4).

Channel Hopping: Channel Hopping calculated by Equation (5) improves reliability by switching between different frequency channels, helping to avoid interference from congested or noisy channels. The effectiveness of channel hopping can be quantified by a channel diversity factor $D_{channel}$, which enhances the PDR:

$$PDR_{hop} = PDR_{original}\left(1+D_{channel}\right) \tag{5}$$

where, $D_{channel}$ is the channel diversity factor, which represents the improvement in PDR due to channel hopping. Typically, this factor can range from 0 to 1, where higher values indicate better improvement.

The improvement in Packet Delivery Rate (PDR) using Forward Error Correction (FEC) and channel hopping was analyzed based on specific parameter values. The FEC coding rate (R_{FEC}) was set to 0.75, which indicates that 75% of the transmission contains actual data and 25% is redundancy bits for error correction. The parameters $\alpha$and βwere assigned values of 0.4 and 0.2, respectively, based on channel conditions and observed error rates in experimental scenarios. Additionally, the channel diversity factor ($D_{channel}$) was set to 0.8, reflecting an 80% improvement in reliability due to channel hopping in high-interference environments.

Wi-Fi's performance was tested in different environments, from uncongested rural areas to heavily trafficked urban areas. In rural settings, the system maintained a 95% packet delivery rate, reflecting minimal interference due to fewer competing signals. This high PDR can be modeled using the equation $PDR_{rural} = 0.95$ while in urban environments, it dropped to 80% ($PDR_{urban} = 0.80$) primarily due to high levels of signal interference and network congestion [23]. Techniques such as Forward Error Correction (FEC) and channel hopping improved transmission reliability by up to 10% resulting in a final PDR of 90% ($PDR_{urban,\ FEC\&hop} = 0.80 \times 1.125 = 0.90$) [7, 21]. Figure 5 displays a graph comparing packet delivery rates in rural versus urban settings, before and after implementing techniques like FEC and channel hopping. The improvement is visually represented in Figure 5, where the optimized urban scenario demonstrates a significant rise in reliability, highlighting the effectiveness of FEC and channel hopping in mitigating interference issues and enhancing Wi-Fi performance.

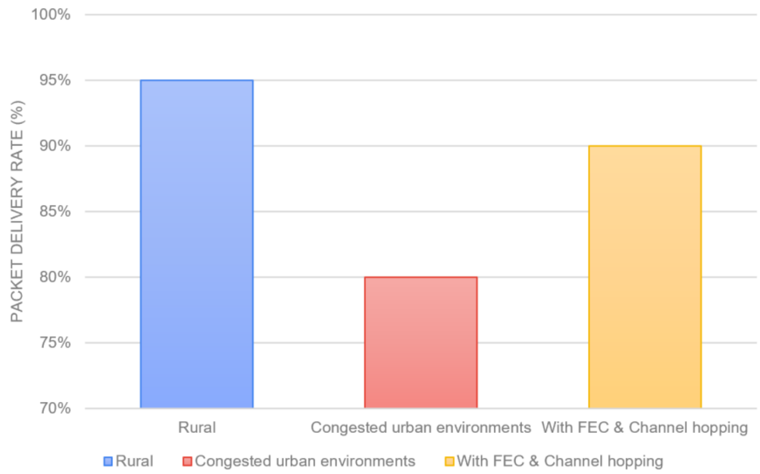

Figure 5. Data transmission reliability in different environments

4.2 *Power consumption analysis*

Using Wi-Fi 6's TWT feature to optimize the wake-up schedule of devices, significantly reducing power usage by minimizing idle listening time and data compression techniques to reduces the volume of transmitted data, decreasing the transmission duration and consequently lowering power consumption under the similar conditions. Channel switching on the ESP32 also relies on TWT to schedule channel scans at pre-determined intervals, ensuring minimal impact on overall power consumption. The settings were chosen to balance energy efficiency with acceptable latency for typical IoT applications, such as environmental monitoring. The 500 ms interval and 10% duty cycle significantly reduced idle listening time, contributing to the observed 40% reduction in power consumption compared to traditional Wi-Fi modes.

The total power consumption of a wireless sensor node (P_{total}) can be expressed as the sum of all components calculated by Equation (6):

$$P_{total} = P_{sense} + P_{proc} + P_{comm} + P_{idle} \tag{6}$$

Here, Sensing Power Consumption (P_{sense}) includes the sensor and the analog-to-digital converter (ADC). Processing Power Consumption (P_{proc}), often a microcontroller, is responsible for data processing and decision-making. Communication Power Consumption (P_{comm}), typically a radio transceiver, is one of the most energy-intensive components. Its power consumption is calculated based on the transmission (P_{tx}) and reception (P_{rx}) modes $P_{comm} = P_{tx} + P_{rx}$. Idle Power Consumption (P_{idle}) consumes power even when the sensor node is in idle mode.

The power consumption of traditional Wi-Fi ($P_{traditional}$) is equal to the initial baseline power consumption P_0 which represents the baseline power consumption of the ESP32 microcontroller operating in traditional Wi-Fi mode (IEEE 802.11 b/g/n) without any energy-saving optimizations such as TWT or data compression ($P_{traditional} = P_0$). Calculation of power reduction ($R_{reduction}$) is calculated by Equation (7):

$$R_{reduction} = \frac{P_{traditional} - P_{wifi\ 6}}{P_{traditional}} \times 100\% \tag{7}$$

Power Consumption with Wi-Fi 6 TWT and Data Compression ($P_{wifi\ 6}$) is calculated by Equation (8):

$$P_{wifi\ 6} = P_0 \times (1 - R_{reduction}) \tag{8}$$

The implementation of Wi-Fi 6's Target Wake Time (TWT) feature, combined with data compression techniques, has proven to significantly reduce power consumption in wireless sensor networks. Traditional Wi-Fi configurations typically consume a fixed amount of power due to continuous data transmission and idle listening. However, by scheduling precise wake-up times with TWT and optimizing data transmission through compression, Wi-Fi 6 achieves substantial energy savings. As illustrated in Figure 6, the power consumption reduction reaches up to 40% ($\left(\frac{100-60}{100}\right) \times 100\%$) compared to traditional setups. This is evident from the decrease in power usage from a baseline of 100 units to just 60 units $(100 \times (1-0.4))$ when using Wi-Fi 6's enhancements. This reduction not only prolongs the battery life of sensor nodes but also improves the overall efficiency of wireless networks in both low-power IoT applications and larger-scale deployments. The demonstrated efficiency highlights the importance of incorporating advanced features like TWT and compression in modern Wi-Fi standards to address the growing need for energy-efficient wireless communication [24, 25, 26].

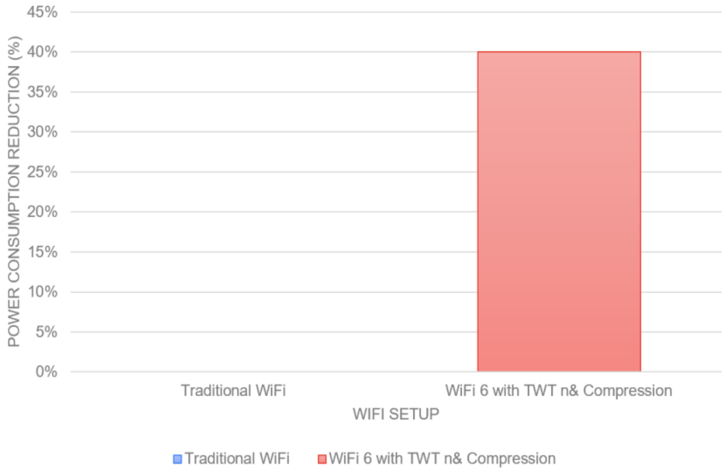

Figure 6. Power consumption reduction without and with Wi-Fi 6 TWT & compression

4.3 *Latency*

Latency is a crucial factor that impacts the performance in delay-sensitive Wireless Sensor Networks (WSNs). It can be defined as the time interval from when a packet is sent from the source node until it reaches its final destination. Real-time video traffic and Emergency alert traffic were chosen for evaluation because Real-time video traffic represents a demanding use case where consistent and low-latency transmission is critical for applications such as surveillance. On the other hand, emergency alert traffic was selected to simulate scenarios where timely delivery of critical messages is paramount.

Real-time video traffic in Table 6 is characterized by a high bandwidth of 5 Mbps, a packet size of 1500 bytes, and a transmission frequency of 30 packets per second, corresponding to 30 frames per second for smooth playback. This type of traffic demands medium QoS priority to ensure continuous data flow while tolerating minor delays. In contrast, emergency alert traffic operates with significantly lower bandwidth requirements of 128 kbps, smaller packet sizes of 256 bytes, and an event-driven transmission frequency of 1 packet per second. Emergency alerts are assigned high QoS priority to guarantee immediate delivery, as they are highly latency-sensitive and must remain below 50 ms for timely response.

Table 6. Comparison of network traffic parameters for real-time video and emergency alerts

Traffic type	Bandwidth	Packet size	Frequency	Priority	Latency requirement
Real-time video	5 Mbps	1500 bytes	30 packets/s	Medium	<30 ms
Emergency alerts	128 kbps	256 bytes	1 packet/s	High	<50 ms

The total latency L calculated by Equation (9) can be expressed as sum of Queueing delay (L_{queue}), Processing delay ($L_{process}$), and Transmission delay ($L_{transmission}$):

$$L = L_{queue} + L_{process} + L_{transmission} \tag{9}$$

The system exhibited low-latency performance, making it suitable for real-time applications like video surveillance. The average latency ranged between 20 and 30 ms, meeting the needs of most real-time monitoring scenarios [26, 27]. Figure 7 shows how latency varies across different test scenarios, emphasizing the low-latency performance of Wi-Fi for real-time applications. The average latency $\overline{L}$ a system can be defined as the mean value of latencies observed across different application scenarios. Mathematically, it is calculated by Equation (10):

$$\overline{L} = \frac{1}{n}\sum_{i=1}^{n} L_i \tag{10}$$

where, L_i : Latency observed in the i^{th} scenario. n: Total number of scenarios (e.g., different application types like video streaming and emergency alerts)

To evaluate the difference in latency $\triangle L$ between these two types of applications, we can use Equation (11):

$$\triangle L = L_2 - L_1 \tag{11}$$

Real-time video applications demonstrate a lower average latency of 25 ms (L_1), whereas emergency alerts have a higher average latency of 30 ms (L_2). the average latency across these two scenarios is calculated as $\overline{L} = \left(\frac{L_1}{L_2}\right)/2 \ = 27.5\,ms$. The calculated difference in latency ($\triangle L = L_2 - L_1 = 5\ ms$) emphasizes the variation in performance requirements between these scenarios.

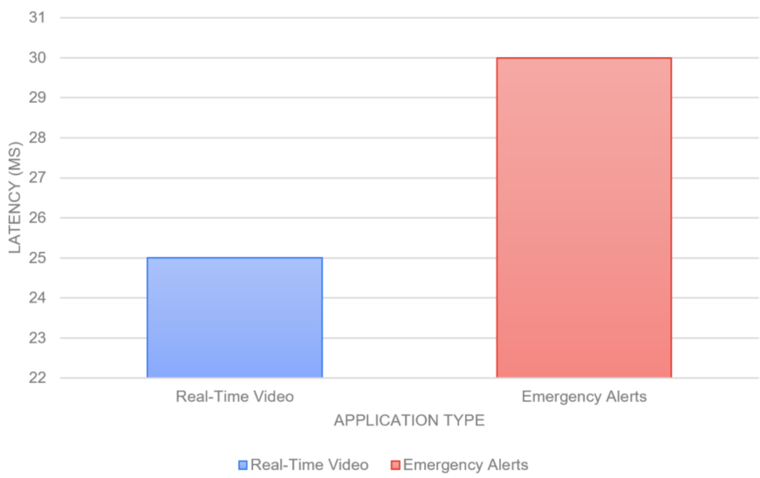

Figure 7. Average latency for real-time applications

This aligns with the visual representation in Figure 7, showing that while Wi-Fi 6 offers low-latency capabilities, the latency for emergency alerts is slightly higher, likely due to the additional priority or handling needed in such critical situations. The analysis confirms that Wi-Fi 6 meets the latency requirements for both application types, making it suitable for real-time monitoring and response systems

5. Conclusions and future work

This paper explored the applications of Wi-Fi technology in Wireless Sensor Networks, focusing on its potential and challenges for IoT implementations. The project highlighted Wi-Fi's advantages, such as high data rates and widespread infrastructure, making it suitable for real-time applications like video surveillance and industrial automation. The performance evaluation showed that modern enhancements like Wi-Fi 6's Target Wake Time can significantly reduce power consumption, achieving up to 40% savings, and improve data reliability through techniques like Forward Error Correction. However, the analysis also revealed limitations, particularly in high-power consumption and susceptibility to interference in dense environments. To address these issues, the study suggested adopting hybrid solutions, such as combining Wi-Fi with ZigBee or LoRa for better energy efficiency and coverage. Wi-Fi faces additional challenges in densely populated urban areas where signal congestion and high levels of interference significantly impact performance. These environments often suffer from overlapping channels and limited spectrum availability, leading to reduced throughput and increased latency. Furthermore, urban deployments must contend with physical obstacles, such as buildings and walls, which weaken signals and cause packet loss. Addressing these limitations requires exploring adaptive mechanisms, such as dynamic channel allocation, advanced interference mitigation techniques, and complementary technologies like mesh networks to maintain reliability and efficiency under such challenging conditions.

In the future work, the authors continue with advanced wireless communication technologies to support WSNs for further developments in specific applications. As the demand for data transmitting in the networks increases including high quality data, high transmission rate, high capacity channels, etc., modulation and multiplexing techniques should be considered in Wi-Fi to be processed at each sensor node to adapt the increasing requirements. Furthermore, future research could focus on combining Wi-Fi with new wireless technologies like 5G to take advantage of their strengths. For example, 5G's ability to provide very fast and reliable connections could improve Wi-Fi's performance in areas like smart cities, self-driving cars, and devices connected to the Internet of Things. A mix of 5G and Wi-Fi could also help manage different types of data, such as video streams and emergency messages, more efficiently. Studying how these systems can work together effectively, handle larger networks, and remain affordable would provide useful insights for building stronger and more flexible communication systems in the future.

Acknowledgments

The authors would like to thank the Thai Nguyen University (project ĐH2024-NV-06) for the support.

Conflict of interest

There is no conflict of interest in this work.

References

[1] A. Coboi, M. T. Nguyen, T. C. Vu, M. D. Nguyen, and D. T. Nguyen, "Zigbee based mobile sensing for wireless sensor networks," *Comput. Netw. Commun.*, vol. 1, no. 2, pp. 325–342, 2023, https://doi.org/10.37256/cnc.1220233923.

[2] M. T. Nguyen, "An energy-efficient framework for multimedia data routing in internet of things (IoTs)," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 6, no. 19, p. e1, 2019, https://doi.org/10.4108/eai.13-6-2019.159120.

[3] B. Rashid and M. H. Rehmani, "Applications of wireless sensor networks for urban areas: A survey," *J. Netw. Comput. Appl.*, vol. 60, pp. 192–219, 2016, https://doi.org/10.1016/j.jnca.2015.09.008.

[4] M. T. Nguyen, "Distributed compressive and collaborative sensing data collection in mobile sensor networks," *Internet Things*, vol. 9, p. 100156, 2020, https://doi.org/10.1016/j.iot.2019.100156.

[5] Z. Al Aghbari, A. M. Khedr, W. Osamy, I. Arif, and D. P. Agrawal, "Routing in wireless sensor networks using optimization techniques: A survey," *Wirel. Pers. Commun.*, vol. 111, pp. 2407–2434, 2020, https://doi.org/10.1007/s11277-019-06993-9.

[6] M. Nguyen, C. Nguyen, and H. T. Tran, "A framework of deploying blockchain in wireless sensor networks," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 9, no. 32, p. 1125, 2022, https://doi.org/10.4108/eetinis.v9i32.1125.

[7] M. T. Nguyen, H. M. Nguyen, A. Masaracchia, and C. V. Nguyen, "Stochastic-based power consumption analysis for data transmission in wireless sensor networks," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 6, no. 19, p. e5, 2019, https://doi.org/10.4108/eai.13-6-2019.159123.

[8] O. N. Samijayani, R. Darwis, S. Rahmatia, A. Mujadin, and D. Astharini, "Hybrid ZigBee and Wi-Fi wireless sensor networks for hydroponic monitoring," in *Proc. Int. Conf. Electr., Commun., Comput. Eng. (ICECCE)*, Istanbul, Turkey, Jun. 12–13, 2020, pp. 1–4, https://doi.org/10.1109/ICECCE49384.2020.9179342.

[9] M. T. Nguyen and K. A. Teague, "Distributed DCT based data compression in clustered wireless sensor networks," in *Proc. 11th Int. Conf. Design Rel. Commun. Netw. (DRCN)*, Kansas City, MO, USA, Mar. 24–27, 2015, pp. 255–258, https://doi.org/10.1109/DRCN.2015.7149022.

[10] R. Musaloiu-E and A. Terzis, "Minimising the effect of Wi-Fi interference in 802.15.4 wireless sensor networks," *Int. J. Sens. Netw.*, vol. 3, no. 1, pp. 43–54, 2008, https://doi.org/10.1504/IJSNet.2008.016461.

[11] C. V. Nguyen, A. E. Coboi, N. V. Bach, A. T. Dang, T. T. Le, H. P. Nguyen, et al., "ZigBee based data collection in wireless sensor networks," *Int. J. Inf. Commun. Technol.*, vol. 10, no. 3, pp. 212–224, 2021, https://doi.org/10.11591/ijict.v10i3.pp212-224.

[12] L. Li, H. Xiaoguang, C. Ke, and H. Ketai, "The applications of Wi-Fi-based wireless sensor network in internet of things and smart grid," in *Proc. 6th IEEE Conf. Ind. Electron. Appl.*, Beijing, China, Jun. 21–23, 2011, pp. 789–793, https://doi.org/10.1109/ICIEA.2011.5975693.

[13] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, et al., "PLoRa: A passive long-range data network from ambient LoRa transmissions," in *Proc. ACM SIGCOMM Conf.*, Budapest, Hungary, Aug. 20–25, 2018, pp. 147–160, https://doi.org/10.1145/3230543.3230567.

[14] A. Moawiah and L. B. Miguel, "Dynamic contention window methods for improved coexistence between LTE and Wi-Fi in unlicensed bands," in *IEEE Wireless Commun. Netw. Conf. Works. (WCNCW)*, Marrakech, Morocco, Apr. 15–18, 2019, https://doi.org/10.1109/WCNCW.2019.8902488.

[15] M. Alhulayil and M. López-Benítez, "Static contention window method for improved LTE-LAA/Wi-Fi coexistence in unlicensed bands," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Fez, Morocco, Oct. 29–Nov. 1, 2019, https://doi.org/10.1109/WINCOM47513.2019.8942468.

[16] S. J. Danbatta and A. Varol, "Comparison of Zigbee, Z-Wave, Wi-Fi, and bluetooth wireless technologies used in home automation," in *Proc. 7th Int. Symp. Digit. Forensics Secur. (ISDFS)*, Barcelos, Portugal, Jun. 10–12, 2019, https://doi.org/10.1109/ISDFS.2019.8757472.

[17] S. S. Murad, R. Badeel, B. B. Abdal, T. Rahman, and T. Al-Quraishi, "Introduction to Wi-Fi 7: A review of history, applications, challenges, economical impact and research development," *Mesopotamian J. Comput. Sci.*, vol. 2024, pp. 110–121, 2024, https://doi.org/10.58496/MJCSC/2024/009.

[18] C. Zeyu, "6G, LIFI and WI-FI wireless systems: Challenges, development and prospects," in *Proc. 18th Int. Comput. Conf. Wavelet Active Media Technol. Inf. Process. (ICCWAMTIP)*, Chengdu, China, Dec. 17–19, 2021, pp. 322–325, https://doi.org/10.1109/ICCWAMTIP53232.2021.9674090.

[19] M. T. Nguyen and K. A. Teague, "Neighborhood based data collection in wireless sensor networks employing compressive sensing," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Hanoi, Vietnam, Oct. 15–17, 2014, pp. 198–203, https://doi.org/10.1109/ATC.2014.7043383.

[20] E. J. Oughton, W. Lehr, K. Katsaros, I. Selinis, D. Bubley, J. Kusuma, "Revisiting wireless internet connectivity: 5G vs Wi-Fi 6," *Telecommun. Policy*, vol. 45, no. 5, p. 102127, 2021, https://doi.org/10.1016/j.telpol.2021.102127.

[21] L. Samal and P. Bute, "Wireless network for industrial application using ESP32 as gateway," in *Proc. 14th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Delhi, India, Jul. 6–8, 2023, pp. 1–5, https://doi.org/10.1109/ICCCNT56998.2023.10306864.

[22] P. Shirisha, G. S. Kumarr, K. Shivanjan, K. S. R. Krishna, K. Vineela, "IoT based Wi-Fi fingerprint door lock system with raspberry pi & webcam," *MATEC Web Conf.*, vol. 392, 2024, https://doi.org/10.1051/matecconf/202439201066.

[23] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002, https://doi.org/10.1016/S1389-1286(01)00302-4.

[24] H. T. Tran, C. V. Nguyen, N. T. Phung, and M. T. Nguyen, “Mobile agents assisted data collection in wireless sensor networks utilizing ZigBee technology,” *Bull. Electr. Eng. Inform.*, vol. 12, no. 2, pp. 1127–1136, 2023, https://doi.org/10.11591/eei.v12i2.4541.

[25] M. T. Nguyen and K. A. Teague, “Compressive sensing based energy-efficient random routing in wireless sensor networks,” in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Hanoi, Vietnam, Oct. 15–17, 2014, pp. 187–192, https://doi.org/10.1109/ATC.2014.7043381.

[26] E. C. Chan, G. Baciu, and S. C. Mak, “Using Wi-Fi signal strength to localize in wireless sensor networks,” in *Proc. WRI Int. Conf. Commun. Mobile Comput. (CMC)*, Kunming, China, Jan. 6–8, 2009, pp. 538–542, https://doi.org/10.1109/CMC.2009.233.

[27] Y. Yu, F. Han, Y. Bao, and J. Ou, “A study on data loss compensation of Wi-Fi-based wireless sensor networks for structural health monitoring,” *IEEE Sens. J.*, vol. 16, no. 10, pp. 3811–3818, 2016, https://doi.org/10.1109/JSEN.2015.2512846.