Comparative Study of Wireless Communication Protocols for Lower-Limb Exoskeleton Systems

Sebastian Garcia
Department of Electrical Engineering, Electronics, and Telecommunications
Universidad de Cuenca
Cuenca, Ecuador
sebastiangarciar@ucuenca.edu.ec

Juan Diego Belesaca
Department of Electrical Engineering, Electronics, and Telecommunications
Universidad de Cuenca
Cuenca, Ecuador
juan.belesaca@ucuenca.edu.ec

Fabian Astudillo-Salinas
Department of Electrical Engineering, Electronics, and Telecommunications
Universidad de Cuenca
Cuenca, Ecuador
fabian.astudillos@ucuenca.edu.ec

Luis I. Minchala-Avila
Department of Electrical Engineering, Electronics, and Telecommunications
Universidad de Cuenca
Cuenca, Ecuador
ismael.minchala@ucuenca.edu.ec

Jhon Calle
Biomedic Engineering Research Group
Univerdad Politécnica Salesiana
Cuenca, Ecuador
jcalle@ups.edu.ec

***Abstract*—The purpose of this study is to evaluate the effectiveness of three wireless communication methods in the context of a lower-limb exoskeleton system, comparing ESP-NOW in broadcast mode, traditional UDP/IP, and communication using Imocap-GIS devices. For this purpose, four different experimental architectures were designed, each comprising a controller (Jetson Nano) and a variable number of slave devices (nodes). The results indicate that ESP-NOW exhibited significantly lower latency times than UDP, positioning it as potentially more suitable for applications in wireless personal area networks that demand fast response times, such as real-time control systems. The experimental findings conclude that the ESP-NOW transmission protocol is prone to increased packet loss as the network expands, especially when the number of nodes increases from one to four, with a maximum loss of 8.6%. These observations emphasize the critical role of protocol selection, tailored to the application's specific requirements and the network environment's intricacies, in ensuring the reliability and efficiency of wireless communication systems.**

***Keywords*—ESP-NOW, Linux, real-time, UDP/IP, Inertial sensors**

I. Introduction

The development of exoskeletons has increasingly been recognized as a critical area of research within the scientific community, driven primarily by the versatility of these systems in performing various functions such as movement assistance and continuous rehabilitation [1]. These devices are often developed in response to the challenges posed by occupational tasks that involve repetitive movements or require individuals to maintain fixed positions for extended periods. Exoskeletons intend to extend the duration individuals can sustain such positions without health repercussions while stimulating innovative solutions to address physical and occupational challenges across different work settings.

Developments in wearable robotics often aim to enhance physical capabilities by designing systems that support or augment natural strength or speed [2]. This integration of exoskeleton technology with rehabilitation needs, prompted by the demands of specific occupations, underlines the profound motivation for ongoing research in this field. This endeavor not only aims to elevate the quality of life and occupational health but also focuses on enhancing exoskeletons at various levels, including mechanics, control systems, and communications.

A critical component related to the performance of exoskeletons is the control unit, which interfaces sensors' signals with actuators' commands, typically through communication systems. For instance, reference [3] describes the deployment of a CAN communication network for communicating a centralized controller unit with distributed controllers installed within the joints of a lower limb exoskeleton.

The authors of [4] designed an exoskeleton aimed at assisting the movement cycle of the lower extremities, utilizing kinematic analysis to ensure precise trajectory replication. The system incorporates Maxon motors and harmonic drive gearboxes, powered by Imocap-GIS inertial sensors. These sensors transmit positional data to a centralized controller via WiFi using the UDP protocol [5].

Further developments in this field include designing controllers with mobile interfaces to aid gait rehabilitation for individuals with lower limb disabilities, utilizing technologies like electromyography (EMG) and artificial neural networks (ANNs) to detect and interpret muscle activity [6], [7]. These approaches have led to significant advancements in control systems, incorporating novel sensors for better capturing motion from the limbs and improving classifiers of movement intention detection.

Inertial sensors, such as accelerometers and gyroscopes, provide data on orientation, acceleration, and movement intentions, essential for the dynamic adjustment of exoskeletons to specific user needs. This data must be accurately and quickly communicated to the exoskeleton controller. Therefore, the communication protocol remains crucial in the development of these systems [8], [9]. For example, UDP offers simplicity and quick responses but lacks error control and MQTT ensures reliability and control at the cost of higher latency and bandwidth needs.

The ESP-NOW protocol has emerged as a particularly effective solution for wireless environments, optimizing performance in systems with stringent energy, rate, and latency requirements. However,its implementation is restricted to compatible devices [10]. The integration of wireless inertial sensors using the ESP-NOW protocol underscores the potential of these technologies beyond exoskeletons into broader biomedical applications [11], [12].

Based on the considerations above, this work proposes using wireless inertial sensors based on the ESP32 platform [13] to communicate with a Linux-based exoskeleton controller (Jetson Nano) using the ESP-NOW protocol. The objective is to reduce communication latency, maintain data integrity, and facilitate sensor deployment through wireless personal area communication.

This paper is structured to provide a comprehensive overview of the field, beginning with a background in Section II, followed by detailed methodologies in Section III, and culminating with the presentation of results in Section IV. The final section, Section V, summarizes the findings and discusses their broader implications for the field.

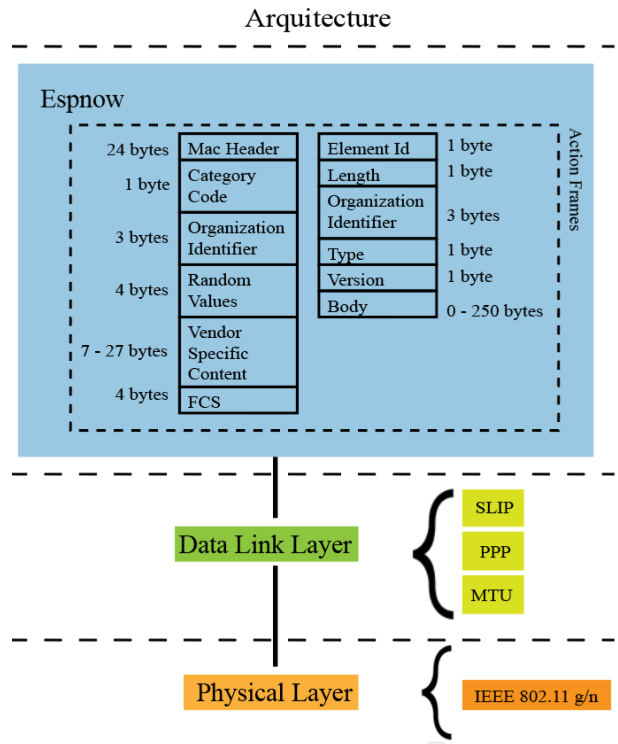

Fig. 1. Arquitecture ESP-NOW.

II. Background

This section presents brief details about inertial measurement units and the ESP-NOW communication protocol, which are key elements of this research. The fundamental theoretical and conceptual principles that support the application and relevance of these technologies in the study context are addressed.

A. Inertial sensors

Inertial sensors or Inertial Measurement Units (IMUs) have become increasingly popular and are now widely integrated in a variety of devices, including mobile phones, tablets, and smartwatches [14]. IMUs have been fundamental in the development of innovative and versatile solutions for various technological and assistance needs. In research, the application of IMUs ranges from devices designed to assist people with disabilities to teleoperation with a wide range of robots [15].

Within the IMU sensors, the gyroscope and the accelerometer are key components. The gyroscope is responsible for measuring the angular velocity or the rate of angular change on the three spatial axes: X, Y, and Z [15], and the accelerometer captures the linear acceleration experienced by an object in the three-dimensional space, using gravitational acceleration as its primary reference [16]. In addition, it allows the identification of abrupt movements, speed variations, or displacements at different angles [17].

The MPU-6050 IMU has been widely used in exoskeleton systems due to its robust performance and versatility [5], [18], [19]. One of its main features is its high-speed data transmission capability, supporting up to 200 samples per second. In addition, it includes a digital filter to reduce noise, enhancing the accuracy and reliability of data acquisition. For these reasons, the MPU-6050 is utilized in this work.

B. ESP-NOW protocol

ESP-NOW is a wireless communication protocol developed by Espressif that facilitates data transmission between ESP8266 or ESP32 devices [20]. This protocol is set up to offer a direct connection between devices without needing a traditional Wi-Fi router, making it ideal for point-to-point communications in Internet of Things (IoT) applications [21]. It is also notable for its energy efficiency, as it consumes minimal power, which is essential for applications that aim to achieve extended battery life.

The ESP-NOW protocol is structured into three layers: the physical layer, the media access control (MAC) layer, and the application layer [22] (see Fig. 1). The physical layer manages wireless data transmission, including signal modulation, transmission power, operating frequency, and modulation schemes for data exchange between ESP-NOW devices. The MAC layer coordinates data transmission, detects collisions, and efficiently delivers data packets among devices within the ESP-NOW network. Finally, the application layer handles user-specific interactions and provides interfaces and services for implementing specific applications on ESP-NOW devices.

Communication through the ESP-NOW protocol can be performed in unicast, multicast, and broadcast modes [23]. This work focuses on the application of ESP-NOW to real-time systems, such as exoskeleton systems, utilizing the broadcast mode due to its wide and rapid dissemination capabilities. To establish broadcast communication, the ESP-NOW protocol is first initialized on the data-emitting devices. Then, transmission is configured using the MAC broadcast address, which allows data to be sent to all devices within range without the need to specify individual recipients. In this way, it ensures that all devices with ESP-NOW enabled and on the same channel receive the transmitted information.

III. Materials and Methods

This section presents the experimental design that supports the research, the specific procedures applied, and the resources used. The description of this section is detailed in Fig. 2. It establishes the foundations for the study's replicability by the scientific community interested in developing real-time communication systems, such as exoskeleton systems with wireless personal area network.

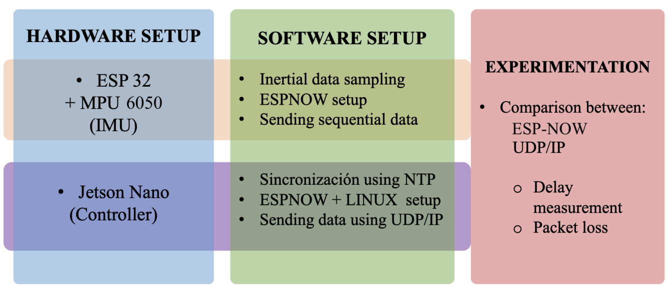

Fig. 2. Methodology used for the development of this work

Fig. 3 illustrates the communication process and development environment used in this work. Four IMUs were configured in the first stage using I2C connections between the ESP32 platform and an MPU-6050 sensor. A Jetson Nano controller based on Linux with a wireless interface was integrated in the second stage.

Regarding software setup, the IMUs were programmed with the ESP-NOW communication protocol, setting up the sampling of inertial data each 5 ms (200 Hz) and the sequential transmission of aggredated data each 25 ms in different time slots defined for each IMU unit; data is aggregated to reduce the overhead and energy consumption [24]. On the other hand, the Jetson Nano control unit runs over the Ubuntu 23.04 operating system, including a library of the ESP-NOW communication protocol applied to Linux [**?**]. This library allows only ESP-NOW in broadcast mode, significantly reducing communication latency and complexity. This improvement is achieved because broadcast mode does not use ACK (Acknowledgment) and backoff procedure in the data link layer, which typically introduces latency.

The experiments in this work were meticulously designed, focusing on two key metrics. The first metric is the delay between capturing inertial data and its arrival in the controller. The second metric is the amount of lost packets in the communication. To measure delay, we implemented an Network Time Protocol (NTP) server into the Jetson Nano controller to synchronize time slots for each IMU and measure delay end-

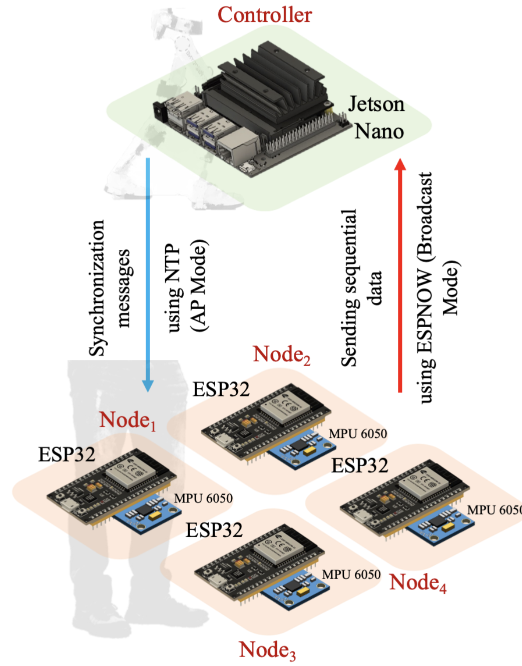

Fig. 3. Communication processes between the controller and the IMUs.

to-end. For lost packets, we used sequential identifiers for each transmitted data, ensuring a reliable data collection process.

For a comprehensive comparison, we evaluated three protocols: ESP-NOW with Linux, UDP/IP, and the factory communication of Imocap-GIS devices (UDP). These experiments were conducted with a Jetson Nano controller acting as master and one to four IMU nodes as slaves. Below, we detail each configured device.

A. Node configuration

In this work, the IMU used is the MPU-6050 (Section II-A). The configuration of this unit was carried out carefully, setting specific parameters for the inertial data sampling frequency at 200 Hz (5 ms per sample) and limiting the accuracy of the gyroscope to one decimal place. This limitation helps reduce the visual complexity in interpreting the information. It optimizes data processing efficiency while effectively adapting to the scale and magnitude of typical movements associated with assistance in the lower limbs [25].

The ESP-NOW protocol is configured after the synchronization process. This protocol is configured based on the 802.11g standard, set to the maximum allowed data rate of 54 Mbps. This technical choice is geared towards optimizing wireless transmission performance, meaning reduced airtime for packets and, consequently, lower latency.

Regarding the configuration of the ESP32 linked to the MPU-6050 sensor, we work with two primary processes: synchronization and data transfer, as presented in Fig. 3 and detailed in the Algorithm 1. The process begins with I2C communication with the MPU-6050 sensor, followed by the synchronization setup and process 802.11g standard setup, and finally, the recursive process of capturing and sending data.

Algorithm 1 Configuration and transmission of inertial data from nodes.

MPU5060_setup() ▷ Configuration of I2C connection with MPU 5060
Synchronization_Processes() ▷ Connection to the NTP server for synchronization
ESP-NOW_80211gSetup() ▷ Configuration of ESP-NOW under 802.11g on ESP32
while true do
    $inetial\_data \leftarrow$ **from_MPU5060**()
    $timestamp \leftarrow$ **from_NTP**()
    Send_data($inertial\_data$, $timestamp$, $identifier$)
end while

B. Controller configuration

In the context of this work, the controller used is the Jetson Nano based on Linux OS. This device, known for its processing power and energy efficiency, serves as the cornerstone of the system, managing the reception and processing of data from nodes.

We used an ESP-NOW library specifically implemented for Linux [2]. The monitor mode configuration was established on channel 1 in the 2.4 GHz band through the wireless interface. The ESP-NOW protocol ensures low-latency wireless communication between the controller and nodes, thus establishing a solid foundation for real-time data transmission.

Additionally, an NTP server has been configured in the Jetson Nano. This server is essential for synchronizing the nodes and maintaining an accurate time reference among the devices, which is critical for temporal coherence and sequence in the capture of inertial data.

C. Design and evaluation of experiments

In this section, we detail the design and execution of comparative experiments to evaluate three wireless communication protocols for a personal area network in the context of a lower-limb exoskeleton system. The assessed protocols include UDP communication from Imocap-GIS devices, UDP/IP, and ESP-NOW with Linux.

In this context, experiments were designed to measure delay and Packet Loss Ratio (PLR). PLR represents the ratio between the number of lost packets and the total number of sent packets, expressed as a percentage. Four independent experiments were designed for the end-to-end delay measurement experiments. Each consists of a controller (Jetson Nano) and a variable number of slave devices (nodes). The specific configurations included one controller and one slave, one controller and two slaves, one controller and three slaves, and finally, one controller and four slaves. The choice of four slaves as the maximum number was based on representing a typical scenario of the lower limb exoskeleton, where capturing data in key joints, such as the knee and hip, requires high precision.

In each experiment, data transmission was carried out sequentially between the slave devices. Each IMU captured data at 5 ms intervals and transmitted it in time windows spaced 25 ms apart. The reason for using a 25 ms interval per sensor is that, by transmitting the data from the four sensors in a staggered manner, it fits within the 100 ms allowed for the complete transmission [26]. This strategy was implemented to avoid interference when increasing the number of IMUs, thus ensuring the integrity of the captured data. The use of temporal windows allowed for proper synchronization and avoided conflicts in data transmission, which is fundamental for accurately assessing the delay.

On the other hand, the experiment was designed for packet loss, with each IMU sending packets with sequential identifiers to the controller. These experiments were performed using the same four architectures for delay measurement.

Each experiment was carried out 20 times, and for each one, 30 000 packets were sent. The extensive number of repetitions and packets sent ensured a robust and reliable dataset for the analysis, allowing for meaningful conclusions to be drawn from the experiment.

IV. Results and discusion

In this section, graphs and comparative tables are presented to illustrate the delay time and packet loss percentage for the proposed communication system.. The delay time was evaluated using a system consisting of an IMU node and a Jetson controller, employing the ESP-NOW and UDP communication protocols. Regarding the packet loss percentage, a variable architecture was implemented, including a number of IMU nodes ranging from one to four.

A. Delay results and analysis

In Fig. 4, the box plots represent the delay for the ESP-NOW and UDP protocols implemented in this experimental proposal. The red box plot depicts the latency time using the ESP-NOW transmission protocol with Linux, and the blue box plot represents the UDP protocol. The experiment was conducted as follows: first, the communication test was configured using either the UDP or ESP-NOW protocol. Then, data transmission from the slave node to the master node (controller) was initiated, and the duration of this transmission was monitored. This process was carried out over 12.5 minutes (30,000 packed each 25 ms) was repeated in 20 trials.

Table I summarizes the delay results. ESP-NOW protocol presents an average latency between transmission and reception of 13.36 ms, with a maximum latency value of 20.9 ms and a minimum of 8.76 ms. Notably, the concentration of latency times in the experiments falls within an interquartile range from 19.01 ms to 11.84 ms. On the other hand, the UDP communication protocol presents an average latency time of 42.89 ms, with a maximum of 98.78 ms and a minimum of 9.37 ms. Unlike the experiments with ESP-NOW, in the case of UDP, the interquartile range spans from 22.68 ms to 78.56 ms.

The results show more significant variability in latency times with the UDP protocol. This is expected since UDP operates at the transport layer and requires more processing time. Therefore, in systems that require low latency times on the order of 20 ms, it is more appropriate to use ESP-NOW

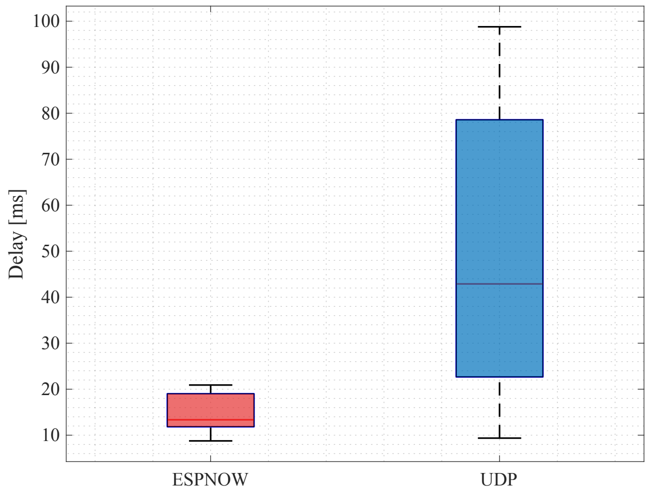

Fig. 4. Delay of ESP-NOW vs UDP.

TABLE I
DELAY OF ESP-NOW VS UDP.

Protocol	Average Time (ms)	Maximun Delay (ms)	Minimum Delay (ms)	Interquartile Range (ms)
ESP-NOW	**13.36**	**20.9**	**8.76**	**19.01 - 11.84**
UDP	42.89	98.78	9.37	78.56 - 22.68

as the wireless communication mechanism between the IMUs and the Linux controller.

B. Packet loss analysis

Fig. 5 presents box plots of PLR with the ESP-NOW and UDP protocols. The experiment's objective was to quantify packet loss with an increasing number of nodes. The experiment was conducted as follows: first, the communication protocol to be used, UDP or ESP-NOW, was selected. Next, data transmission was initiated with a single node in master-slave mode, and over 20 trials were conducted, similar to the previous section. Subsequently, additional nodes were sequentially added, reaching four nodes, to generate the results in Fig. 5.

Table I summarizes the percent lost packets results. Regarding the packet loss results using the ESP-NOW protocol, it is noted that with a single node, the loss ranges from a minimum of 0.5% to a maximum of 2%, with an average loss of 1% and an interquartile range from 0.8% to 1.4%. On the contrary, packet loss using UDP is negligible, registering at only 0. 1%. Upon analyzing the results with two nodes, a significant increase in packet loss is observed when using the ESP-NOW protocol. The loss range is broad, with a minimum value of 1.4% and a maximum of 3.4%, and an average loss of 2.7%, characterized by an interquartile range fluctuating between 1.7% and 3.6%. Conversely, the UDP protocol exhibits a minimal loss of 0.2%. Continuing this line of analysis, examining the results with three nodes reveals a notable increase in packet loss. This loss varies from a minimum of 1.8% to a maximum

TABLE II
PLR OF ESP-NOW VS UDP

Protocol	Node	Avg. PLR (%)	Max. PLR (%)	Min. PLR (%)	Range (%)
ESP-NOW	1	1	2	0.5	0.8 - 1.4
	2	2.7	3.4	1.4	1.7 - 3.6
	3	3.7	5.2	1.8	2.8 - 4.4
	4	7.7	8.6	5.8	6.8 - 8.8
UDP	1	**0.1**	**0.1**	**0**	**0 - 0.1**
	2	**0.1**	**0.2**	**0**	**0 - 0.2**
	3	**0.2**	**0.3**	**0**	**0 - 0.3**
	4	**0.3**	**0.4**	**0**	**0 - 0.4**

of 5.2%, with an average loss of 3.7%, characterized by an interquartile range between 2.8% and 4.4%. On the other hand, the UDP protocol shows a minimal loss of 0.3%. Finally, considering packet loss with four nodes, the following results are observed: the minimum loss is 5.8% and the maximum is 8.6%, with an average of 7.7%. This pattern is characterized by a range that varies between 6.8% and 8.8%. In contrast, the UDP protocol presents a minimal loss of 0.4%

The findings of this study provide a detailed insight into packet loss for the ESP-NOW and UDP transmission protocols in a wireless personal area communication context between nodes and a hub (Jetson Nano). A clear pattern of increased PLR was observed for both protocols as the number of nodes in the network progressively increased. Specifically, with ESP-NOW, PLR ranged from a minimum of 0.5% to a maximum of 8.6% with four nodes, averaging 7.7%. On the other hand, UDP exhibited a minimal and consistent loss compared to a minimum percentage of 0.1% and a maximum of 0.4%. Evidently, packet loss significantly affects the performance of the ESP-NOW communication protocol, particularly as network complexity increases. This is mainly due to the broadcast mode used, which does not consider packet retransmission or

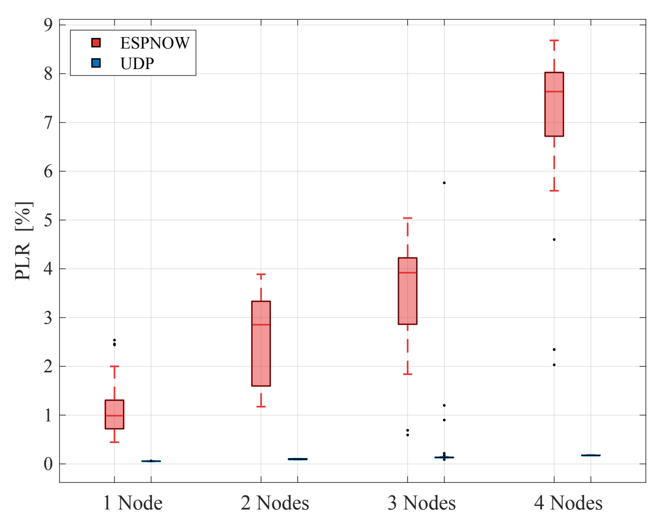

Fig. 5. PLR of ESP-NOW vs UDP.

contention process.

Based on the results obtained, the ESP-NOW protocol shows less jitter (delay fluctuation) compared to UDP. Furthermore, ESP-NOW presents a notable improvement in response times. This reduced jitter and enhanced performance in response times position ESP-NOW as a superior option for wireless personal area networks. These characteristics are especially critical in applications that require fast response times and minimal latency, such as real-time control systems, including exoskeleton systems, where critical latency is a determining factor for effective and safe operation.

V. Conclusion

Choosing a transmission protocol in wireless communication systems is crucial for achieving optimal performance, particularly concerning latency and data consistency. This study demonstrates that ESP-NOW, when used in broadcast mode, significantly outperforms UDP in terms of latency, achieving an average of 13.36 ms compared to UDP's 42.89 ms, and exhibiting less jitter. These characteristics make ESP-NOW particularly well-suited for real-time control systems, such as those used in exoskeletons, where rapid response times are critical.

However, the study also reveals that ESP-NOW's performance degrades with increased network size, as evidenced by a rising packet loss rate, which reaches a maximum of 8.6% when the number of nodes increases from one to four. In contrast, UDP maintains minimal and stable packet loss, demonstrating greater robustness and efficiency under higher network loads.

These findings underscore the importance of carefully selecting a protocol that aligns with the specific requirements of the application and the network environment. While ESP-NOW may be ideal for small-scale applications with stringent latency requirements, UDP is more reliable for larger, more complex networks.

References

[1] M. Callejas-Cuervo, M. A. Vélez-Guerrero, and W. J. P. Holguín, "Arquitectura de un sistema de medición de bioparámetros integrando señales inerciales-magnéticas y electromiográficas," *Revista Politécnica*, vol. 14, no. 27, pp. 93–102, 2018.

[2] P. Chinmilli, S. Redkar, W. Zhang, and T. Sugar, "A review on wearable inertial tracking based human gait analysis and control strategies of lower-limb exoskeletons," *Int. Robot. Autom. J*, vol. 3, no. 7, p. 00080, 2017.

[3] H. Morales, A. Cordova, I. Minchala, and F. Astudillo-Salinas, "Implementation and performance evaluation of a library to improve the communication with maxon motor's," *Revista Digital Novasinergia*, vol. 6, no. 1, pp. 95–104, 2023.

[4] C. Feicán, C. Saquicela, and J. Calle-Siguencia, "Kinematic validation of a lower limb exoeskeleton using computer simulation," in *2020 IEEE ANDESCON*, pp. 1–5, IEEE, 2020.

[5] J. Calle-Siguencia, M. Callejas-Cuervo, and S. García-Reino, "Integration of inertial sensors in a lower limb robotic exoskeleton," *Sensors*, vol. 22, no. 12, p. 4559, 2022.

[6] F. Astudillo, J. Charry, I. Minchala, and S. Wong, "Lower limbs motion intention detection by using pattern recognition," in *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, pp. 1–6, IEEE, 2018.

[7] E. Mora-Tola, J. Loja-Duchi, A. Ordonez-Torres, A. Vazquez-Rodas, F. Astudillo-Salinas, and L. I. Minchala, "Robotic knee exoskeleton prototype to assist patients in gait rehabilitation," *IEEE Latin America Transactions*, vol. 18, no. 09, pp. 1503–1510, 2020.

[8] M. Ghislieri, L. Gastaldi, S. Pastorelli, S. Tadano, and V. Agostini, "Wearable inertial sensors to assess standing balance: A systematic review," *Sensors*, vol. 19, no. 19, p. 4075, 2019.

[9] X. Zhang, Y. Shi, and Y. Zhang, "A can-based inertial sensor network for lower limb exoskeleton," in *2014 IEEE international conference on communiction problem-solving*, pp. 473–476, IEEE, 2014.

[10] P. Garofalo, A. G. Cutti, M. V. Filippi, S. Cavazza, A. Ferrari, A. Cappello, and A. Davalli, "Inter-operator reliability and prediction bands of a novel protocol to measure the coordinated movements of shoulder-girdle and humerus in clinical settings," *Medical & biological engineering & computing*, vol. 47, no. 5, pp. 475–486, 2009.

[11] M. A. Beltrán Saavedra and F. M. Tacuri Merchán, "Desarrollo de un sistema iot para la adquisición de datos biomecánicos de un patinador de velocidad, utilizando dispositivos esp32," B.S. thesis, Universidad de Cuenca.

[12] A. A. Süzen, B. Duman, and B. Şen, "Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–5, IEEE, 2020.

[13] M. Olinski, A. Gronowicz, M. Ceccarelli, and D. Cafolla, "Human motion characterization using wireless inertial sensors," in *New Advances in Mechanisms, Mechanical Transmissions and Robotics: Proceedings of The Joint International Conference of the XII International Conference on Mechanisms and Mechanical Transmissions (MTM) and the XXIII International Conference on Robotics (Robotics' 16)*, pp. 401–408, Springer, 2017.

[14] L. Vera and G. O. Lombardero, "Caracterización de sensores para aplicaciones en biomecánica," 2018.

[15] M. Acevedo and J. Stevens, "Implementación de un sistema basado en sensores inerciales para la caracterización biomecánica de la marcha normal y de pacientes con amputación del miembro inferior," 2021.

[16] M. Callejas-Cuervo, A. Ruiz-Olaya, M. Vélez-Guerrero, and A. Alarcón-Aldana, "Platform for biomechanical data capture and analysis using inertial sensors: Calibration and validation plataforma de captura y análisis de información biomecánica usando sensores inerciales: Calibración y validación,"

[17] J. Aventin Roig, "Red inalámbrica de sensores inerciales para stream de registros goniométricos en biomecánica," 2015.

[18] N. Lacuata, B. O'Dell, A. John, C. Pelletier, D. Jefferson, R. Lineberger, and M. Sharifi, "Design and fabrication of a modular, lightweight, and portable upper limb exoskeleton for shoulder and elbow," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 87622, p. V005T06A063, American Society of Mechanical Engineers, 2023.

[19] G. A. Wicaksono, M. R. Rosa, and M. H. Barri, "Implementation of exoskeleton robots for upper limb rehabilitation based on indonesian anthropometry," in *2022 2nd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, pp. 139–143, IEEE, 2022.

[20] R. Pasic, I. Kuzmanov, and K. Atanasovski, "Esp-now communication protocol with esp32," *Izzivi prihodnost*, vol. 6, 3 2021.

[21] J. A. Ruiz Sandoval, "Diseño e implementación de un sistema de alarma de intrusión basado en el protocolo esp-now de internet de las cosas," 2021.

[22] M. G. Gómez, "Sistema de control de casa inteligente utilizando el protocolo esp-now: Smart-home control system using the esp-now protocol," *Investigación y Ciencia Aplicada a la Ingeniería*, vol. 4, no. 24, pp. 1–6, 2021.

[23] T. N. Hoang, S.-T. Van, and B. Nguyen, "Esp-now based decentralized low cost voice communication systems for buildings," in *2019 international symposium on electrical and electronics engineering (ISEE)*, pp. 108–112, IEEE, 2019.

[24] S. Randhawa and S. Jain, "Data aggregation in wireless sensor networks: Previous research, current status and future directions," *Wireless Personal Communications*, vol. 97, pp. 3355–3425, 2017.

[25] J. D. Paredes Altamirano, "Prototipo de un exoesqueleto para rehabilitación de miembros inferiores," B.S. thesis, 2021.

[26] J. Beil, I. Ehrenberger, C. Scherer, C. Mandery, and T. Asfour, "Human motion classification based on multi-modal sensor data for lower limb exoskeletons," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5431–5436, IEEE, 2018.