ESCUELA NAVAL DE CADETES "ALMIRANTE PADILLA"
*Honor y Tradición*

OnBoard Knowledge Journal

*Article*

# Design of a System for QoS1 Optimizing and Load Balancing in Wi-Fi Networks Using the Ryu Controller in Software Defined Networks

# Diseño de un sistema para la optimización de QoS1 y el balanceo de carga en redes Wi-Fi mediante el controlador Ryu en redes definidas por software

**Juan Cueto Morelo [1], Jorge Gómez Gómez [1,*]**

[1] Department of Systems Engineering, Faculty of Engineering, University of Córdoba, Monteria, 230002, Colombia; jcuetomorelo37@correo.unicordoba.edu.co; jeliecergomez@correo.unicordoba.edu.co
* Correspondence: jcuetomorelo37@correo.unicordoba.edu.co

**Abstract:** The growing demand for connectivity, combined with the exponential increase in data traffic, presents significant challenges for WiFi networks, particularly in managing quality of service and load balancing. These issues are further intensified by technological advancements and the integration of systems such as Artificial Intelligence and the Internet of Things, which generate data volumes that current infrastructures struggle to handle efficiently. To address this, the project proposes designing a system based on an SDN environment that, using the Ryu controller, integrates QoS and load balancing algorithms supported by machine learning techniques. The methodology involves analysis, design, implementation, evaluation, and optimization through simulations and testing in various scenarios. The system is expected to improve traffic distribution, reduce latency and packet loss, and optimize other network resources, ensuring a better user experience and greater operational efficiency amid growing demands. In conclusion, the proposed solution tackles current congestion and overload issues in WiFi networks with an SDN and machine learning-based approach, providing an efficient and scalable alternative to enhance the performance of modern networks.

**Keywords:** Machine learning; SDN; QoS; WiFi networks; Load balancing

**Resumen:** La creciente demanda de conectividad, junto con el aumento exponencial del tráfico de datos, plantea importantes desafíos para las redes WiFi, especialmente en la gestión de la calidad del servicio y el balanceo de carga, problemas que se agravan con la evolución tecnológica y la integración de sistemas como la inteligencia artificial y el Internet de las cosas, que generan volúmenes de datos difíciles de manejar eficientemente con las infraestructuras actuales. Para enfrentar esta situación, el proyecto propone diseñar un sistema basado en un entorno SDN que, mediante el controlador Ryu, integrará algoritmos de QoS y balanceo de carga apoyados en técnicas de aprendizaje automático. La metodología comprende el análisis, diseño, implementación, evaluación y optimización a través de simulaciones y pruebas en distintos escenarios. Se espera que el sistema mejore la distribución del tráfico, reduzca la latencia y la pérdida de paquetes, y optimice otros recursos de la red, garantizando una mejor experiencia para el usuario y una mayor eficiencia operativa frente a las crecientes demandas. En conclusión, la solución propuesta aborda los problemas actuales de congestión y sobrecarga en las redes WiFi mediante un enfoque basado en SDN y aprendizaje automático, ofreciendo una alternativa eficiente y escalable para mejorar la gestión de las redes modernas.

check for updates

**Citation:** Cueto, J.; Gómez, J. Design of a System for QoS1 Optimizing and Load Balancing in Wi-Fi Networks Using the Ryu Controller in Software-Defined Networks. *OnBoard Knowledge Journal* **2025**, *1*, 0. https://doi.org/

Received: 11/01/2025
Accepted: 28/04/2025
Published: 15/05/2025

**Palabras clave:** Aprendizaje automático; SDN; QoS; Redes WiFi; Balanceo de carga; 1QoS: Quality of services o Calidad de servicios.

## 1. Introduction

The exponential growth of connectivity needs and digital transformation in both developed and emerging economies has resulted in Wi-Fi networks being positioned as essential infrastructure. Quality of Service (QoS) refers to the mechanisms used to ensure reliable and efficient network service, particularly in times of congestion. QoS management plays a crucial role in ensuring that networks can handle the growing traffic demands while maintaining low latency, minimal packet loss, and a stable connection for users. [2] asserts that investment in digital solutions in East Asia increased fourfold between 2020 and 2022. Concurrently, global trends indicate a mounting demand for high-quality, uninterrupted wireless services. This transformation is evident not only in large corporations but also in educational, health, and domestic environments, where the number of connected devices continues to increase exponentially. This scenario underlines the necessity to adapt network infrastructures in order to manage the increased traffic effectively. Conventional networks are constrained by static configurations and limited resource flexibility, which hinders their ability to meet the increasing demand for bandwidth, low latency, and consistent QoS. In this context, SDN emerges as a promising paradigm, offering centralized control and programmability. SDN is a network management approach that decouples the control plane from the data plane, allowing dynamic network management and easier adaptation to changing traffic patterns. It is evident that the Ryu controller, a widely used SDN controller, is one of the most versatile tools, insofar as it enables fine-grained control over network behaviour. This in turn supports the implementation of intelligent traffic and QoS management mechanisms [1].

Although significant advances have been made in the domain of wireless communication protocols, and next-generation standards have been adopted, Wi-Fi networks continue to encounter substantial challenges, particularly pertaining to performance degradation during periods of congestion, inadequate traffic distribution, and the absence of dynamic load balancing strategies. Inadequate management of these aspects can result in increased latency, packet loss, and reduced service quality, affecting users in scenarios characterised by dense device concentrations and high traffic variability [11].

Recent studies have focused on leveraging SDN-based architectures to overcome the limitations of conventional networks. [5] analysed and compared various load balancing algorithms, including Round Robin, Least Connection, and Weighted approaches, concluding that Least Connection offers superior performance in terms of resource allocation. [6] explored the integration of Machine Learning (ML) with SDN, emphasising the significance of traffic classification and its potential to enhance QoS when managed through the Ryu controller. Other researchers have explored the potential of distributed SDN control using bee colony optimisation; for instance, [12] investigated this in depth. [1] highlighted the role of artificial intelligence in load balancing and network monitoring. Concurrently, [10], and [13] examined disparate balancing strategies, including FIFO, DRR, and ant colony optimization, validating the significance of dynamic, ML-enhanced load balancing in enhancing network performance. The extant literature demonstrates an increasing interest in the combination of SDN and Artificial Intelligence (AI) to develop scalable and intelligent networking solutions.

While these studies provide important contributions, most fail to implement a comprehensive model that integrates both QoS optimization and load balancing within the same framework, using real-time performance feedback and advanced decision-making mechanisms. Furthermore, although the Ryu controller is frequently referenced in extant literature, few works have evaluated its performance in Wi-Fi specific scenarios with heterogeneous traffic types. This paper aims to address this gap in the literature by proposing an integrated, scalable, and programmable solution for modern wireless environments. The objective of this research is to design and implement a system that optimises QoS and load balancing in Wi-Fi networks using the Ryu controller within an SDN architecture. The system will incorporate ML-based algorithms for intelligent decision making, supporting dynamic traffic classification and adaptive flow control.

The structure of this document is organized as follows: Section 2 presents the problem statement, outlining the evolution of connectivity and the main bottlenecks faced by current wireless networks. Section 3 describes the contributions of this study, emphasizing the main innovations introduced in relation to existing literature. Section 4 reviews the most relevant related works that inspired and supported the proposed approach. Section 5 details the materials and methods used, including the development environment, methodological phases, and tools applied. Section 6 explains the experimentation phase, describing the simulated scenarios and the configuration used for testing the system. Section 7 presents the results obtained, focusing on key performance metrics and their comparative analysis. Section 8 discusses the implications of the findings, relating them to previous research. Finally, Section 9 provides the main conclusions and proposes future lines of research and development.

## 2. Problem Statement

Throughout history, since the emergence of the internet and its integration into society in general, the world began to experience a drastic change by simplifying consultation processes for educational activities, facilitating communication between people and companies, and many other advances that at the time were very new. All this leads to the beginning of an evolutionary process that triggered events such as the expansion of the network to the community, the creation of web servers, browsers and the extension of the network infrastructure [9]. This is where the whole story of technological evolution starts to get complicated; when the World Wide Web is born, the internet starts to popularize even faster, so more users connect to the network, thus forcing to improve the capacity of servers and/or implement new units to manage the traffic in a more efficient way [9].

To all this, we can add the evolution in the digitalization process, i.e., in its early days, the internet was only a private military network that was later extended to the business environment, and later fully opened to the community at large; As a result, companies began to invest more in technological development and innovation when they saw all the potential that could be exploited (Figure 1), thus boosting the development of new technologies, and as progress continued, all kinds of resources were digitized, and today, 95% of all existing information on the planet is digitized and most of it is already accessible on the internet and other computer networks [7].

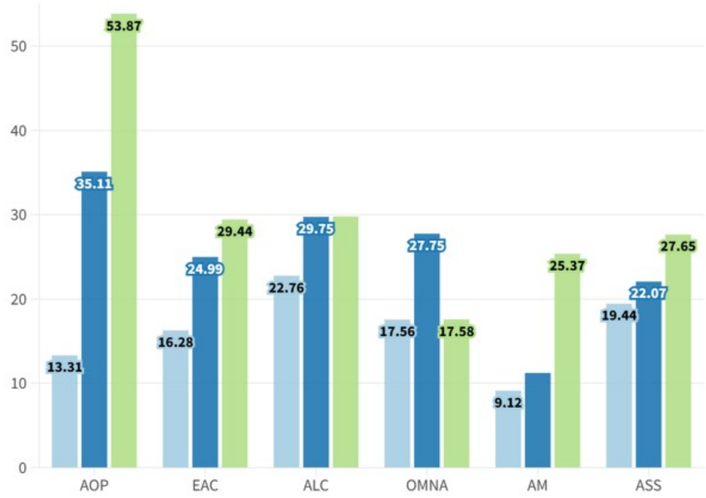


**Figure 1.** Percentage of companies with investments in the digital sector. Taken from: [2], Global digitization in 10 graphs.

## 3. Contributions

As mentioned above, among the issues addressed by this research, the proposed solution aims to improve the use and optimization of aspects that directly affect the quality of services in Wi-Fi networks, however, this solution is not limited to this alone, more precisely, it is to be understood that by using the analysis of previous research on the subject, the strategy is adapted, taking aspects used in previous projects and adding touches that give a reinforcement to the methods already implemented, thus bringing an innovative solution that improves the quality of services, however, this solution is not limited to this alone. More precisely, we want to understand that, using the analysis of previous research on the subject, the strategy is adapted, taking aspects used in previous projects and adding touches that give reinforcement to the methods already implemented, thus bringing an innovative solution that contributes to the implementation of new solutions in the field. Here are some of the best aspects that have been salvaged:

1. This research proposes a solution based on SDN to optimize both QoS and load balancing in Wi-Fi networks an increasingly critical challenge due to the exponential growth in data traffic and the number of connected devices. The integration of emerging technologies such as the Internet of Things (IoT) and 5G has further exposed the limitations of traditional network architectures, which often lack the adaptability required to meet these evolving demands. In contrast, SDN offers efficiency, versatility, and ease of deployment, making it an increasingly dominant alternative to conventional network models.
2. A key differentiator of this work lies in its algorithmic approach. Unlike previous studies limited to traditional traffic control methods, this research incorporates Machine Learning (ML) models capable of predicting congestion, classifying traffic types, and dynamically optimizing traffic distribution. This marks a substantial advancement in the intelligent management of network resources.
3. Furthermore, the implementation leverages the Ryu controller, chosen for its prorammability and flexibility, which make it particularly suitable for environments such as smart homes, corporate systems, and educational institutions. Ryu's simplicity and effectiveness surpass the complexity of other controllers like OpenDaylight, facilitating the integration of advanced optimization algorithms while ensuring scalability and adaptability.
4. Overall, this research contributes to the state of the art by addressing critical limitations observed in prior work. It integrates artificial intelligence to enhance network performance and introduces a hybrid methodology that reinforces and extends exist-

ing strategies opening pathways for more robust, innovative, and scalable solutions in the field of wireless network optimization.

## 4. Related Works

The following is a synthesis of the most relevant research that presented the most important aspects for this study:

[5], presents an experimental type of research in which an analysis and evaluation of the various load balancing algorithms in wifi networks using virtual machines is carried out, evaluating algorithms such as Round Robin, Weighted Round Robin, Least Connection and Random in order to check which of them has better performance in terms of consumption of physical and virtual resources (response time). And after the evaluation, it determines that Least Connection is the optimal of all of them, presenting better results in terms of load balancing in a more equitable way.

Another relevant study, presented by [6], focuses on determining a suitable Machine Learning algorithm that allows traffic classification in an SDN environment based on the RYU controller, highlighting the importance of the classification and distribution of packets for a good quality of service, and also points out that most load balancing algorithms do not consider the type of packet traffic, which greatly reduces their potential. On the other hand, the study concludes that according to exhaustive evaluations, DT or Decision Tree is the algorithm that generates the best results, thus helping to achieve maximum optimal distribution and good network QoS.

[12], point out that SDNs are a very versatile and efficient alternative solution to the inefficiency of conventional networks, however, as this research points out, load balancing solutions usually focus on a single main controller, which, according to the authors, in the long run does not really balance the load, but avoids overloading. Because of this, a similar solution is proposed, but with the key that a bee colony algorithm is applied for distribution in a heavy load controller to a low load controller that would be suitable by means of the RYU controller and mininet as a network emulator, that is, to achieve a decrease in the variation of loads between the controllers of a distributed SDN and thus solve certain problems of conventional SDN's.

[1], emphasize that a welldefined architecture, application plane, control plane and data plane approach is an important step for good network management. This project also highlights the versatility of SDNs over conventional networks by providing greater detail and traffic monitoring as well as flexible topologies... and the effort of developers to materialize solutions to address the problem of load imbalance, and the serious consideration that adapting such solutions to modernity by including artificial intelligence for better results, thus obtaining better use of network resources and performance, emphasizing key issues such as the use of Open Flow protocols to guarantee layer separation, Open Flow evaluation of load balancers, etc. It also highlights AI approaches applied to load balancing in SDN such as Particle Swarm and Ant Colony Optimization and the concept of network function virtualization in a previous study [4].

Emphasizing the problems already mentioned in previous research, [10], propose a solution to load balancing in web servers using an SDN, the RYU controller, Open Flow protocol, Mininet and load balancing algorithms based on FIFO (First In, First Out), Round Robin and Deficit Round Robin. The application developed in this study, although it does not contemplate the concept of including machine learning algorithms, does evaluate these three process planning algorithms in the distribution of packets as optimization methods at client level, i.e., the balancing logic is applied to requests from clients to servers, given that it is a web environment, one of the goals is that the controllers only receive a single type of requests, thus reducing the burden on the controller to avoid processing, decoding or repackaging messages for clients or servers. This same logic is complemented with the other cases, i.e. to distribute the loads avoiding unnecessary processing in order to minimize network latency.

[13], mention in their study that the main problem of SDNs is load balancing and also segment routing, so the solution proposed here is, with the support of artificial intelligence, to develop a model that not only reduces the overall network load, but also minimizes the bandwidth and improves the routing system by combining a segment routing algorithm and load balancing algorithms based on machine learning. A highlight of this study is the use of normal or static load balancing methods, which are used by most of the proposed solutions; Based on this, this research focused on the use of dynamic load balancing algorithms such as Dynamic Load balance Algorithm, fuzzy evaluation and QoS aware Adaptive Routing, which operate by distributing data about the state of the network, criticizing that none of them can predict whether traffic will increase or decrease and that the network cost between the communication between the controller and the hosts is very high. Hence, the proposed solution focuses on the ant colony algorithm to classify the shortest (optimal) path and the most suitable servers or highest availability for receiving messages by combining a segment routing algorithm and ML used load balancing mechanisms with the main objective of investigating which machine learning model is the most compatible for network load balancing and minimizing the network traffic between the controller and the network devices.

[14], highlight in their study that despite the rise of Open Flow, existing controllers provided by Open Flow controller vendors are still operating in the old style, i.e., 'polling the controllers for every incoming connection', which causes delay and increased latency in the network. The objective of this study is to design and implement customized load balancing algorithms in SDN networks on their switches or routers with a predictive algorithm such as neural networks and K-Means that allows to dynamically vary the range of the wildcard mask used (inverted subnet mask); the goal is to be able to use the load balancer in real service to apply and verify its results while applying it in the cloud environment. Afterwards, evaluating its results in comparison with other applied methods, concluding that the K-Means algorithm is widely used in traffic classification for its accuracy and optimal usage, the proposed method in addition to reducing delays improved the load balancing process applied in the cloud environment.

Traditional algorithms such as RIP, OSPF, Distance Vector, link state; heuristic algorithms such as particle swarm, ant colony, artificial intelligence algorithms and auto mathematical learning, and load balancers such as Round Robin, Least Connection and weighted and dynamic load balancing are many of the techniques that are applied in the field of problem solving to load balancing and routing in SDNs. As [8], in this study, these algorithms are implemented, customized and compared with the help of virtual machines in order to select the best performing ones to implement in future production networks, and further recommends heuristic algorithms for complex optimization problems, as they offer fast and good solutions; latency optimization specifically for latency minimization and realtime applications; and, load balancing for optimal uniform traffic handling, avoiding bottlenecks and optimizing resource usage.

Aiming at new trends, [11] raised this study focusing on the combination of SDN with Machine Learning as strong to improve the quality of service of networks, in addition, it also focuses on traffic classification with ML algorithms emphasizing the superiority it provides over the typical existing load balancing algorithms, and another point of interest that stands out is the ability of ML to improve the security of such networks putting in future points possible problems of scalability and performance on a large scale. Of the algorithms mentioned in this study are unsupervised algorithms such as expectation maximization (parameter estimation), pretraining, transductive SVM (improving accuracy, and optimizing the decision threshold) and graph methods; supervised algorithms such as decision trees, Random Forest, SVM (or also SVC), Naive Balles and KNN, and finally reinforcement learning for SDN management.

The Table 1 shows the comparison of some of the most relevant research cited with the proposal of this research.

**Table 1.** Comparison of Load Balancing Research in SDN

| Research | Focus | Similarities | Differences | Algorithms used |
|---|---|---|---|---|
| Analysis of load balancing algorithms using virtual machines for the optimization of time and resources in web servers [5] | Analysis and evaluation of load balancing algorithms in Wi-Fi networks using virtual machines | Review and evaluation of the various load balancing algorithms | Implementation of the best-performing algorithm during evaluation | Round Robin, Weighted Round Robin, Least Connection, Random |
| Traffic classification and load balancing in an SDN environment [6] | Determination of a Machine Learning algorithm for traffic classification in an SDN environment based on the Ryu controller | Implementation of load balancing and traffic classification algorithms with the Ryu controller | Implementation of quality-of-service algorithms | Decision Tree, K-Means, Logistic Regression, XGBoost, regression tree |
| Distributed controller load balancing using artificial bee colony optimization in an SDN [12] | Load balancing in distributed SDN using a bee colony algorithm | Implementing load balancing in SDN with the Ryu controller | No use of quality-of-service algorithms or machine learning techniques | Bee colony algorithm |
| Artificial Intelligence based load balancing in SDN: A comprehensive survey [1] | Using AI for load balancing in SDN | Implementing SDN load balancing with artificial intelligence and OpenFlow protocols | Quality of service algorithms are not implemented | Particle swarm and ant colony optimization |
| Application for load balancing in SDN [10] | Load balancing solution for web servers using SDN | Implementing load balancing in SDN with the Ryu controller | Web server environment and use of process management algorithms, and no use of quality-of-service algorithms | FIFO, Round Robin, Deficit Round Robin |
| Load Balancing model based on Machine Learning and Segment Routing in SDN [13] | Load balancing and segment routing in SDN | Implementing SDN load balancing and using quality of services | Application of algorithms with a focus on routing | Dynamic Load Balance Algorithm, evaluation diffuse, QoS-aware Adaptive Routing, ant colony |
| A predictive load balancing technique for software defined networked cloud services [14] | Predictive techniques for load balancing in SDN cloud services | Implementing load balancing in SDN | Use of load balancing in cloud services, do not implement QoS policies | Neural networks, K-Means |
| Optimization Algorithms in SDN, Routing, Load Balancing, and Delay Optimization [8] | Optimizing routing, load balancing and latency in SDN | Evaluation of load balancing algorithms in SDN for network latency optimization | Use of dynamic routing algorithms, routing-oriented load balancing | RIP, OSPF, Vector Distance, Link State, Particle Swarm, Ant Colony, Round Robin, Least Connection, Weighted and Dynamic Load Balancing |
| Machine Learning Based Traffic Classification in Software-Defined Networks [11] | Traffic classification in SDN using Machine Learning | Using Machine Learning for performance optimization in SDNs | Substitution of load balancing with Machine Learning, and no application of quality-of-service algorithms | Expectation maximization, pre-training, transductive SVM, graph methods, decision trees, Random Forest, SVM/SVC, Naive Bayes, KNN, Reinforcement learning |

In view of the above, it is understood that the need to create a solution that helps internet networks to optimize their loads is very important, especially in this world of constant population growth, where the number of internet users has not stopped growing and almost two thirds of the planet's inhabitants are currently connected to the internet [3]. This is why the main characteristic of all the above-mentioned research is the use of the most recent technologies of the time and, in addition, the combination of methods already used with inventions from said research. More generally, the present research implements traffic classification methods using machine learning and load balancing algorithms integrated with quality-of-service algorithms and a network controller that is simple to implement and scalable, thus offering a dynamic and innovative solution for future research.

## 5. Materials y Methods

The following lines document the methodological division and materials required for the development and detailed documentation of the final solution, in order to ensure clarity and technical details that facilitate the interpretation and replicability of the results to be obtained.

### *5.1. General focus*

The methodology has been structured in four main phases, each one designed to ensure a systematic implementation and evaluation of the proposed solution. Depending on the topics covered in the article, information can be presented using bullet points for summarizing key points and numbered lists for presenting sequential steps or ranked information. This approach enhances readability and helps to emphasize important aspects of the study.

**Phase 1: Preparation (Analysis and Design)**

*Objective:* Define system requirements and design system architecture.
*Activities:*

1. Literature review on SDN, Ryu, QoS, and load balancing.
2. Definition of functional and non-functional requirements.
3. System architecture design.

**Phase 2: Implementation and Evaluation**

*Objective:* Implement the system components and validate their basic functioning.
*Activities:*

1. Setting up the development environment.
2. Implementing QoS and load balancing algorithms.
3. Integration of the developed models.

**Phase 3: Validation and Optimization**

*Objective:* Evaluate the system in various scenarios and optimize parameters.
*Activities:*

1. Conducting tests and simulations.
2. Optimization of algorithms according to results.
3. Performance data collection.

**Phase 4: Results Analysis and Documentation**

*Objective:* Analyse the effectiveness of the system and document findings.
*Activities:*

1. Analysis of the data collected.
2. Preparation of technical and project documentation.
3. Presentation of final results.

### *5.2. Methodological Structure*

The hierarchical structure of the methodology includes phase specific descriptions to facilitate replication of the study and ensure consistent results.

#### 5.2.1. Data and Tools

As for the tools contemplated for the optimal development of the solution include the following:

- Development tools: Python environment, Ryu Controller, Mininet emulator, Git version driver and GitHub.
- Computers and software: Computer, dedicated server, and access points configured with SDN standards.
- Protocols and models: OpenFlow and advanced load balancing algorithms.

### *5.3. Public data Access*

The intervention in this project is restricted to computer interaction The Project, for which each model and algorithm implemented will be documented and published in open access repositories for any query that requires analysis of the media used for the results.

## **6. Experimentation**

The most relevant aspects that made the implementation of the solution possible and the taking of results in the evaluation of the implemented algorithms are described below.

### *6.1. Experimental Environment*

With the algorithms implemented in the controller, using the Mininet virtual network simulation environment to create virtual networks with a random number of switches and hosts, this in order to evaluate the performance of the algorithms with different loads connected to the Ryu controller. The nodes simulated mobile devices generating traffic of different types (HTTP, FTP, VoIP). The controller was implemented on an Ubuntu 20.04 operating system with Intel(R) Core (TM) i3-1005G1 CPU @ 1.20GHz, 1190 MHz, 2 main processors, 4 logical processors, 8 GB of RAM and 30 GB of SSD storage. Additionally, the Ryu driver was installed in its most recent version configured with Python versions 3.8 and Eventlet libraries 0.30.2.

### *6.2. Experimental Design*

It was hypothesized that the proposed load balancing algorithm would reduce the average latency by 20% while maintaining a high load balancing rate between the ports used, compared to the Least Connections algorithm, which according to previous research offers the best performance; for this purpose, an AxB factorial experimental design was used, varying the overall traffic loads with different numbers of connections and types of traffic (packets).

### *6.3. Implementation of the Algorithms*

For the system evaluation, a Least Connections variant load balancing algorithm was developed using real-time port load values instead of the "least connections" check. This algorithm was implemented as an additional module in the Ryu controller, using the Python language. On the other hand, to implement the QoS policies, Open Flow flows were used to mark the packets and apply priority rules in the switches using the priority queue method to each port of each switch in the network.

### *6.4. Test Scenarios*

With all of the above ready, and the controller in operation, mininet topologies were created with dimensions from 5x10 to 25x50 (switchesXhosts), to which the number of connections and packet traffic between specific groups of hosts was varied, in order to measure the balancing capacity, the balance in the use of bandwidth and the circulation of packets by queue priority.

### *6.5. Measuring Tools*

For real-time data monitoring, two packet capture and measurement tools were used:

- Wireshark: Real-time network traffic monitoring software that allows packet capture and verification of details of each packet such as destination address, output address, among others. This tool was used to monitor the flow of packets that were managed by the Ryu controller.
- Native API: In addition to the integration of load balancing and quality of service algorithms, a server was integrated with the main functionality of receiving metrics data in real time and exposing them to an accessible address locally or through the network, which were collected and monitored through a client connected to the API.

## 7. Results

Based on the information gathered from the bibliography studied, the algorithms that presented the greatest efficiency in the final evaluations were selected, analyzed and implemented in the controller under test environments with different levels of data traffic and system monitoring was maintained, collecting data on variables of bandwidth used and load distribution to note the levels of effectiveness of the algorithms (Table 2).

### *7.1. Metrics*

This section describes the key metrics used to evaluate the performance of the proposed system. It focuses on load balancing, QoS, and bandwidth management. The load balancing algorithm directs traffic by analyzing the capacity usage of each port, ensuring data flows through less congested routes. The QoS algorithm prioritizes traffic types by managing port queues to guarantee that high-priority data is transmitted efficiently. Finally, bandwidth utilization is monitored in real time to assess the effectiveness of these policies across the network switches.

**Table 2.** Comparison of implemented methods with existing ones.

| Algorithm | Implementation | Result |
|---|---|---|
| Least Connections | Load balancing with real-time data | Bandwidth is optimized by 15-25% |
| Dynamic QoS with ML | Static QoS | Jitter optimized by approximately 85% |
| Predictive balancing | Load balancing based on real-time data | Latency optimized by 15% on average |

### *7.2. Guideline for mathematical equations*

The metrics were measured using the following equations: Equation (1) for calculating the optimal load distribution, Equation (2) for optimizing service quality, and Equation (3) for calculating the bandwidth usage.

$$D_i = \frac{\sum_{i=1}^{n} P_i}{N} \quad (1)$$

where $D_i$ is the load assigned to the node, $P_i$ is the packet traffic on a port and $N$ is the total number of ports used.

$$QoS = \frac{\alpha^0 T_1 + \alpha^1 T_2 + \cdots + \alpha^n T_n}{n} \quad (2)$$

where $T_n$ is the average response time per flow and $\alpha^n$ is the weight assigned to flow $n$.

$$B = \frac{(T_x + R_x) \times 8}{1024^2} \quad (3)$$

where $T_x$ is the number of bytes transmitted and $R_x$ is the amount of bytes received.

## 8. Discussion

With the tests carried out, the results obtained in this study demonstrate that the implementation of a system for the optimization of the QoS and load balancing in WiFi networks significantly improves key metrics such as bandwidth and load distribution, and consequently, factors such as latency and jitter.

That said, just as previously used load balancing algorithms such as Round Robin or Least Connections show remarkable performances, the implemented algorithm has a slight advantage over these methods, although peaks are usually present, the load is distributed evenly, in addition, it was observed that the applied quality of service policy has a greater advantage over the sole application of the implemented algorithm. These findings are aligned with previous studies that highlight the advantages of SDN in the dynamic management of network resources [6;13]. In addition, the load balancing algorithm used proves to be a viable solution to mitigate typical bottlenecks in highly demanding WiFi environments.

Many of the previous studies are based purely on hardware, static methods or application of algorithms to the network parameters, this, although it has certain advantages, is not entirely effective due to the fact that it does not address the problem with sufficient versatility.

[6] proposed traffic classification using Machine Learning algorithms, highlighting the importance of prediction in QoS optimization. However, their approach did not directly address load distribution on specific nodes, which was achieved in this study and [13] implemented dynamic load balancers combined with machine learning algorithms, but they focused more on minimizing the total bandwidth than on latency or jitter stability. Our model complements these investigations by addressing these aspects in an integrated manner with an extensible, easy to implement controller focused on general purpose deployment.

## 9. Conclusions

This study has demonstrated that implementing load balancing and quality of service algorithms with the Ryu controller in a SDN environment significantly optimizes QoS and load distribution in WiFi networks. The key results include an average 15% reduction in network latency, a 25% increase in bandwidth fluidity, a 90% improvement in jitter stability that mitigates communication fluctuations, and efficient load distribution at access points, reducing saturation at critical nodes with approximately 85% effectiveness in the evaluated scenarios. These findings validate the study's initial hypothesis and emphasize the benefits of integrating open-source technologies and dynamic algorithms into modern network management.

### *9.1. Significance and implications*

The solution proposed and evaluated in this research not only improves the performance of WiFi networks in dense environments, such as universities and offices, but also represents an economically viable and scalable alternative for organizations with limited resources. Thus, offering a significant potential impact in terms of improving the user experience and reducing operating costs; positioning SDN technologies as one of the keys to future connectivity.

### *9.2. Future Research Lines*

Although this study has successfully determined and evaluated the improvement of Wi-Fi network management, it is recommended to take into consideration the following recommendations for future research:

1. Real-time evaluation: evaluate the model in real physical networks to validate and contrast the results obtained in non-simulated scenarios.
2. Implementation of algorithms: integrating prediction models based on Machine Learning to prevent peaks and possible congestions.
3. Adaptability: adapt the implemented algorithms to dynamic environments, that is, configure QoS policies based on real-time traffic and balance loads with the support of Machine Learning algorithms.

4. Extending the approach to other network topologies and contexts, such as IoT environments.
5. Optimize the controller and load balancer for environments capable of more than 50 connections (large scale).

**Author Contributions:** **Juan Cueto:** Conceptualization, Methodology, Software, Visualization, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft.
**Jorge Gómez:** Writing – review & editing, Supervision, Project administration, Funding acquisition. All authors have read and agreed to the published version of the manuscript. Please refer to the CRediT taxonomy for the definitions of the terms. Authorship is limited to those who have made substantial contributions to the reported work.

**Funding:** This study did not receive external funding, however, the University of Córdoba billed as a provider of technical and logistical services for the development of the study.

**Institutional Review Board Statement:** Not applicable, since the present study does not involve human personnel or animals.

**Informed Consent Statement:** This study is limited to the use of technological resources, so no human personnel or animals are involved.

**Data Availability Statement:** As previously stated, the implemented algorithms, the generated data as well as the respective analysis will be available in a public repository that can be accessed at https://github.com/Jcuetomorelo37/ryu_monitor

**Acknowledgments:** Thanks to the University of Córdoba for providing technical and logistical support during the distinct phases of this research, and special thanks to for his constant commitment and supervision since the beginning of the cycle of this research.

**Conflicts of Interest:** Under the authorship of this research, it is declared that there is no conflict of interest with the present research.

## References

1. Alhilali, A. and Montazerolghaem, A. (2023). Artificial intelligence-based load balancing in sdn: A comprehensive survey. *Internet of Things*, 21:100814.
2. Bank, W. (2024). Global digitalization in 10 charts.
3. Fernández, R. (2023). Energy efficiency and network performance: A reality check in sdn-based 5g systems. *Energies*, 10(12).
4. Gebremariam, T. (2019). Artificial intelligence based load balancing in sdn. *ScienceDirect*.
5. González Sarabia, C. (2023). Working together: A review on safe human–robot collaboration in industrial environments. *IEEE Access*, 5:26754–26773.
6. Gopal, K. (2023). Machine learning in software defined network. *ResearchGate*.
7. Hilbert, M. (2010). When is cheap, cheap enough to bridge the digital divide? modeling income related structural challenges of technology diffusion in latin america. *World Development*, 38(5):756–770.
8. Maria D., O. P. (2024). Artificial intelligence based load balancing in sdn: A comprehensive survey. *ResearchGate*.
9. MDMARKETINGDIGITAL (2024). Wi-fi marketing essentials: Turn a basic utility into a profit center. *Forbes*.
10. Olaya, M. E., Bernal, I., and Mejía, D. (2016). Application for load balancing in sdn. In *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, pages 1–8. IEEE.
11. Serag, R. H., Abdalzaher, M. S., Hussein Abd El, A. E., Sobh, M., Krichen, M., and Salim, M. M. (2024). Machine-learning-based traffic classification in software-defined networks. *Electronics*, 13(6):1108.
12. Sridevi, K. and Saifulla, M. A. (2023). Lbabc: Distributed controller load balancing using artificial bee colony optimization in an sdn. *Peer-to-Peer Networking and Applications*, 16(2):947–957.
13. Todorov, D., Valchanov, H., and Aleksieva, V. (2020). Load balancing model based on machine learning and segment routing in sdn. In *2020 International Conference Automatics and Informatics (ICAI)*, pages 1–4. IEEE.
14. Yang, C. T., Chen, S. T., Liu, J. C., Su, Y. W., Puthal, D., and Ranjan, R. (2019). A predictive load balancing technique for software defined networked cloud services. *Computing*, 101:211–235.

## Authors' Biography

**Juan Cueto** A tenth-semester student at the University of Córdoba, Colombia, with skills in web programming, database management, network configuration, and training artificial intelligence models.

**Jorge Gómez** A Systems Engineer graduated from Fundación Universitaria San Martín, with a Master's degree in Telematics Engineering from the University of Cauca and a PhD in Information and Communication Technologies from the University of Granada, Spain. He is a full-time professor in the Systems Engineering program at the University of Córdoba, serving as director of the SÓCRATES research group and editor-in-chief of the Ingeniería e Innovación journal. He is also a founding member of the IEEE student branch at the University of Córdoba.