Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 6, December 2025, pp. 4913~4924
ISSN: 2302-9285, DOI: 10.11591/eei.v14i6.9895

Design and emulation of an SDN network with opendaylight to improve QoS in a peruvian financial institution

Juan David Indigoyen Roncal, Christian Ovalle Paulino
Department de Ingeniería, Facultad de Ingeniería de Telecomunicaciones, Universidad Tecnológica del Perú, Lima, Perú

Article Info

Article history:

Received Jan 15, 2025
Revised Sep 29, 2025
Accepted Oct 14, 2025

Keywords:

Financial networks
OpenDaylight
Quality of service
Software-defined networking
Software-defined-wide area network

ABSTRACT

This study presents the design and emulation of a software-defined networking (SDN) architecture using the OpenDaylight controller to enhance the quality of service (QoS) in a Peruvian financial institution. The main objective is to overcome limitations of traditional networks, including high latency, limited bandwidth, and packet loss, which hinder the efficiency of financial services. The proposed SDN architecture was implemented and tested through simulations in the Eve-NG platform, where key performance parameters—latency, throughput, and packet loss—were measured. Results demonstrated significant improvements, with latency reduced by up to 40%, stable throughput maintained at 10 Mbps across all branches, and a noticeable reduction in packet loss. These outcomes validate the feasibility of adopting SDN in financial environments to support critical services and ensure operational continuity. Furthermore, the findings emphasize SDN's role in modernizing network infrastructures, improving user experience, and aligning local financial institutions with international technological trends. Future research may explore alternative SDN controllers, scalability in larger topologies, and integration with emerging technologies such as network function virtualization (NFV).

This is an open access article under the CC BY-SA license.

Corresponding Author:

Christian Ovalle Paulino
Department de Ingeniería, Facultad de Ingeniería de Telecomunicaciones
Universidad Tecnológica del Perú
Lima, Perú
Email: dovalle@utp.edu.pe

1. INTRODUCTION

The exponential growth in the number of users has recently driven companies to explore innovative network designs and advanced technologies [1]. Although computational bandwidth continues to evolve with most data centers currently operating above 20 Gbps and next-generation technologies aiming at 800 Gbps these advances can also create bottlenecks in memory and routing systems [2]. In financial institutions, where service continuity, low latency, and efficient data management are essential, traditional network architectures often face challenges such as congestion, limited scalability, and inefficient resource allocation.

Software-defined networking (SDN) has emerged as a disruptive paradigm in telecommunications and network management, offering flexibility, scalability, and programmability by decoupling the control plane from the data plane [3]. OpenFlow, as the main SDN protocol, enables dynamic flow management between controllers and switches [4]. Several open-source SDN controllers have been developed, including NOX [5], POX [6], Beacon [7], Floodlight [8], RYU [9], ONOS [10], and OpenDaylight [11]. Among these, OpenDaylight has established itself as a modular and extensible platform capable of addressing the challenges associated with managing complex networks in real time.

Previous research has shown that the implementation of SDN in wide area networks (SD-WAN) enables the optimization of routing protocols, including border gateway protocol (BGP), and even suggests the possibility of using external entities to manage inter-internet service provider (ISP) routing [12]. These advances have contributed to more efficient load distribution, particularly through algorithms such as K-Path [13], resulting in a significant reduction in congestion and latency [14]. Beyond these mechanisms, recent technological developments such as virtualization, heuristic optimization, and deep learning have opened new opportunities to further enhance SD-WAN performance. For instance, the application of dynamic quality of service (QoS) policies and local search algorithms has proven effective for bandwidth allocation and route selection, thereby improving user experience and ensuring greater compliance with service level agreements (SLAs) [15].

Despite these advances, existing routing strategies still face important challenges. Centralized algorithms benefit from a global network view, allowing the calculation of optimal routes, but they lack scalability in large environments. On the other hand, distributed algorithms offer better scalability but frequently consume more resources and do not always guarantee optimal results. To overcome these drawbacks, hybrid approaches integrating heuristics and machine learning have been proposed, enabling real-time route calculation and adaptive traffic management in unpredictable scenarios [16]. These perspectives highlight the importance of continuing to improve load-balancing mechanisms and validate their applicability across diverse platforms and topologies, in order to achieve more efficient resource discovery and transmission [17].

In parallel, applied studies have demonstrated the benefits of SDN-based load balancing for enhancing QoS. One notable example is the plug-n-serve system, which reduces client–server response times by allowing the controller to dynamically install routing rules [18]. Although this system improves flexibility and latency, its limited scalability restricts its deployment in large-scale environments. To address this limitation, other proposals suggest the use of algorithms capable of calculating wildcard rules, reducing the processing load on the SDN controller. This method combines partitioning algorithms to define rules and transition algorithms to adjust them according to traffic policies [19]. While effective in relatively simple networks, these solutions must evolve to handle the complexity of large-scale, high-traffic environments.

More recent approaches, such as Koerner's multi-OpenFlow model, have explored distributing traffic across multiple services, thereby improving the efficiency of handling high traffic volumes [20]. However, despite these contributions, the practical application of such mechanisms in production-oriented environments—particularly in financial institutions—remains limited. Implementing SDN in this context could enable centralized and flexible network management, providing improvements in performance, resource optimization, and responsiveness compared to traditional architectures [21]. This paper addresses that gap by designing and emulating an SDN-based architecture, aligned with Pereira's guidelines, to strengthen infrastructure agility and ensure higher service quality satisfaction in a Peruvian financial institution [22]–[25].

2. METHOD

This study was conducted to design and emulate a SDN architecture using OpenDaylight, with the aim of improving QoS parameters in a Peruvian financial company. The methodology combined industry best practices with SDN-based strategies to build a network capable of reducing latency, ensuring consistent throughput, and minimizing packet loss. All simulations were performed using the Eve-NG platform, which provides realistic multi-vendor environments and efficient packet transfer between devices [20].

2.1. Research stages

The method was structured in three main stages:

a. Initial data collection: the data from the last month of the company's network will be analyzed, including information about the current topology, the equipment used, average traffic, and existing QoS parameters.
b. Network emulation: a similar network will be designed and emulated using Mininet and Eve-NG, integrating the OpenDaylight controller, which will be programmed to manage QoS policies.
c. Traffic simulation and measurement: traffic will be generated using Iperf, measuring key parameters such as:

- Throughput: the data transfer rate is successful in Mbps.
- Flow completion time (FCT): total time from the moment the first packet of a flow leaves the source until the last packet arrives at the destination, measured in seconds [22], [26].

QoS parameters analyzed:

- Latency: is the time it takes for a packet to travel from the source to the destination. Reducing latency is crucial to improving the speed of critical transactions, such as real-time financial data, and optimizing the user experience in services like online banking.
- Packet loss: reducing packet loss increases network reliability, decreases errors in critical transactions, and improves operational efficiency.

2.2. Current network analysis

The financial company currently employs a hub-and-spoke topology, routing all traffic through the central office via the provider’s multiprotocol label switching (MPLS) network. While common, this design increases latency due to centralized routing. Figure 1 illustrates the existing topology, and Table 1 summarizes the measured QoS parameters across offices.

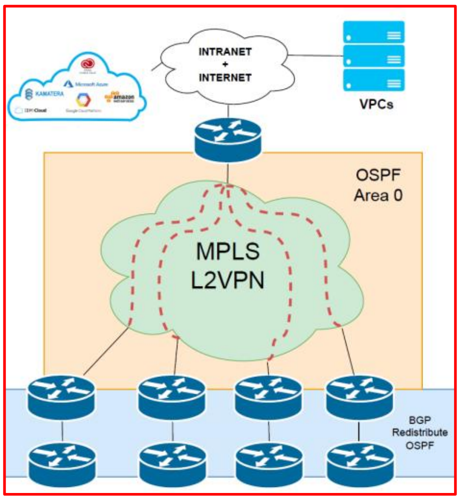

Figure 1. Current network topology

Table 1. QoS parameters in the current network

Offices	Latency (ms)	Average throughtput (Mbps)	Packet loss (%)
Header	35	15	0
Office A	60	2	3
Office B	54	1	0.15
Office C	57	2	1
Office D	75	2.5	0

The current network design of the financial company routes traffic to the Internet or cloud services through the provider's MPLS network, also passing through the main office's internal network. This topology, known as “Hub and Spoke,” is widely used by various organizations. However, it has a significant drawback: requiring all branches to route traffic through the central office increases the network latency. Table 1 shows the QoS parameters recorded at each office.

2.3. Equipment selection

The current Cisco ISR 4221 devices are to be replaced with Cisco C1111 routers, which provide higher throughput, SD-WAN support, and OpenFlow compatibility. Table 2 compares both devices, and Figure 2 depicts the proposed model.

- Throughput: the Cisco C1111 offers up to 1 Gbps, significantly higher than the 35 Mbps of the ISR 4221. This improvement allows for handling larger data volumes, which is crucial for an SDN network that processes and redirects large amounts of traffic efficiently. For the financial company, higher throughput means better management of traffic demands during periods of high activity without compromising service quality.
- SD-WAN and OpenFlow compatibility: the Cisco C1111's compatibility with SD-WAN and OpenFlow allows optimal integration with the OpenDaylight controller, enabling centralized and automated traffic management. OpenFlow enables the SDN controller to send specific flow rules to the router, optimizing

routing and dynamically applying QoS policies. This is particularly beneficial for a financial company that requires flexibility and efficiency in managing network resources.

– Processing capacity and memory: the Cisco C1111 has 8 GB of RAM, compared to the 4 GB in the ISR 4221. This increased memory allows for handling complex SDN tasks and packet processing without overloading the system, improving the network's real-time response. This is crucial for avoiding bottlenecks and ensuring the network can quickly adjust to the company's changing needs.

Table 2. Comparison between C1111 and ISR4221

Equipment	C1111	ISR 4221
RAM	8 GB	4 GB
OS	IOS XE	IOS
SDWAN	Yes	No
Througput	1 Gbps	35 Mbps

Figure 2. Cisco C1111

2.4. Software-defined networking architecture overview

The SDN architecture separates the control plane (centralized in the controller) from the data plane (executed by routers). After evaluating multiple open-source controllers, OpenDaylight was selected due to its support for recent OpenFlow versions, graphical interface, and multi-OS compatibility. Table 3 summarizes the comparison and Figure 3 presents the proposed architecture.

Table 3. Comparison of controller types

Controllers	NOX	POX	Beacon	Floodlight	OpenDaylight
Development Language	C++	Python	Java	Java	Java
Openflow	1.0	1.0	1.0	1.0-1.5	1.0–1.5
GUI	No	Python+y web	Web	Web	Web
SO	Linux	Linux, Mac OS y Windows	Linux, Mac OS y Windows	Linux, Mac OS y Windows	Linux, Mac OS y Windows

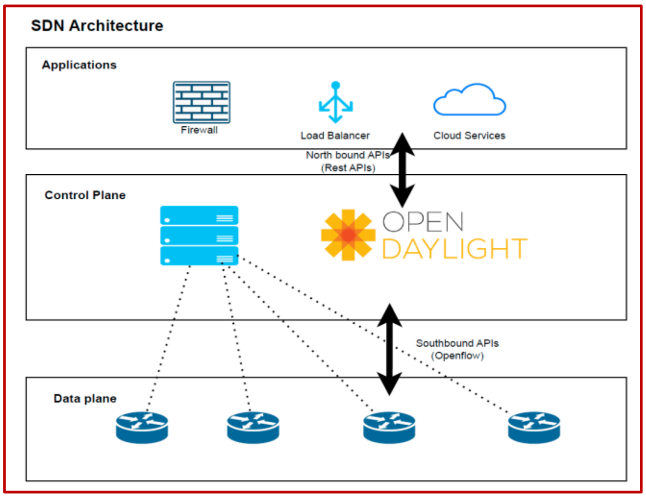

Figure 3. Proposed SDN architecture

The new architecture proposes running security applications (firewalls and access control policies), cloud services, load balancing, and routing via REST APIs. As part of the research, a REST API has been developed that optimizes routing by implementing dynamic traffic policing and shaping policies to improve throughput, reduce latency, and minimize packet loss. Figure 4 shows the proposed SD-WAN topology.

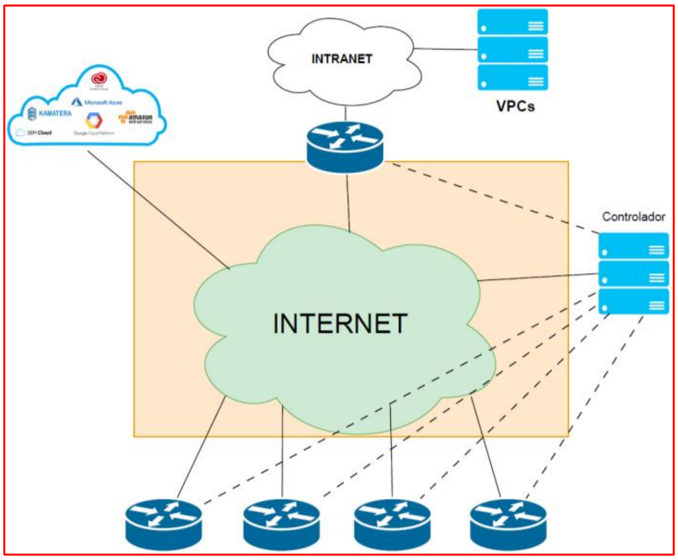

Figure 4. Proposed topology

2.5. Simulation tools

In this new topology, remote offices connect directly to the internet, facilitating interaction with the clients' virtual clouds. The OpenDaylight controller will communicate with the developed application and send flow entries to the routers, allowing them to configure their flow tables and optimize QoS. Table 4 summarizes this comparison.

- Simulation and emulation: once the solution and elements are defined, the emulation will begin. It will utilize the Eve-NG network emulator, which allows interaction with various vendors and provides a more realistic simulation.
- Packet tracer: a network simulator developed by Cisco that is ideal for creating and testing basic and advanced network configurations, learning network concepts, and preparing certifications like CCNA. limited to cisco devices.
- GNS3: a network emulator that works with real operating system images, offering a closer-to-real network experience. It supports multiple vendors and is highly customizable.
- Eve-NG: a multi-vendor network emulation platform that allows the creation of complex environments with devices from different manufacturers. It is ideal for advanced testing and more realistic lab scenarios.

Table 4. Comparison between

Description	Packet tracer	GNS3	Eve-NG
Software	Simulator	Emulator	Emulator
Devices	Cisco only	Cisco (Dynamips+QEMU)	Multi-vendor
Resource usage	Low	Medium	High

Figure 5 shows how all the routers will be connected to the controller, represented in Eve-NG as the Internet cloud. For the controller, a virtual machine based on Linux was created, and OpenDaylight was downloaded from the Linux Foundation's website. OpenDaylight will be responsible for routing and making any decisions in the control plane, as demonstrated in Figure 6.

For traffic simulation, Jperf was used, which is a batch file that simplifies the use of Iperf and sent traffic at the maximum bandwidth for each branch. Figure 7 shows the use of this batch file for traffic saturation testing.

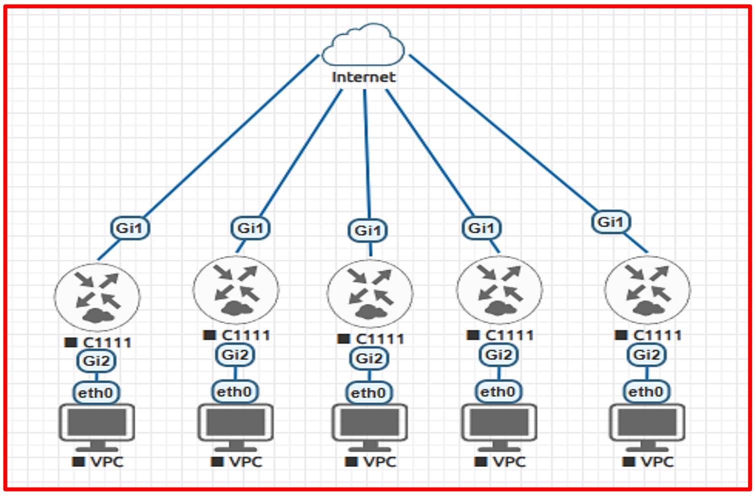

Figure 5. Topology in Eve-NG

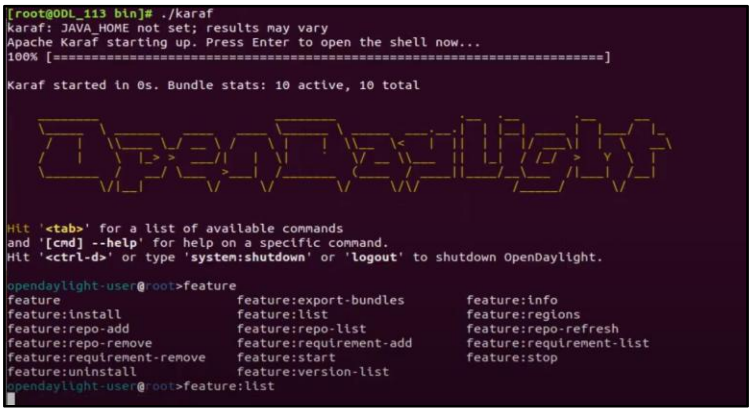

Figure 6. OpenDaylight controller

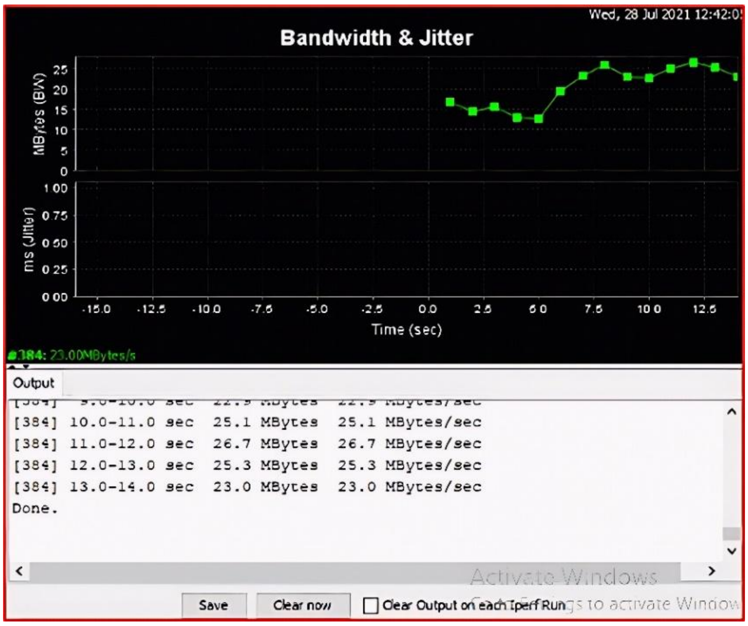

Figure 7. Traffic saturation test in Jperf

3. RESULTS

Table 5 summarizes the QoS parameters obtained after implementing the SDN-based architecture in the financial institution's network. After the implementation of the SDN architecture in the financial company's network, significant improvements were observed in the QoS indicators, especially in latency, throughput, and packet loss. These improvements are described.

Table 5. Comparison of QoS parameters in the SD-WAN network

Offices	Latency (ms)	Average throughput (Mbps)	Packet loss (%)
Header	7	50	0
Office A	17	10	1.25
Office B	10	10	0
Office C	13	10	0
Office D	32	10	0

3.1. Latency

Latency showed the most substantial improvement. Compared to the traditional network, where the headend averaged 35 ms and branch offices ranged from 54–75 ms, the SDN-based network reduced these values to 7–32 ms, representing an average 40% reduction. This improvement is attributed to centralized traffic management by the OpenDaylight controller, which optimizes routing and avoids router overload. During traffic spikes of 10,000 packets/s, the traditional network suffered congestion and packet drops, while the SDN controller dynamically redistributed traffic to mitigate bottlenecks. The transmission latency can be estimated as:

$$Latency\ Time\ =\ RTT\ x\ Number\ of\ Hops$$

when a client performs an online banking transaction that involves multiple steps, such as authentication, data transfer to the server, and bank confirmation, each step requires a client-server interaction, generating a round-trip time (RTT).

3.1.1. Latency in the traditional network

Each router has 10,000 routes in its routing table. The processor works at 80% of its capacity due to constant route calculations and packet processing. Router load (traditional network): average time per packet: 2 ms (intensive processing due to searching in a large table). CPU load: 80%, handling 1,000 packets per second.

3.1.2. Latency in the SD-WAN network

The centralized controller sends specific flow rules to each router, reducing the size of the routing tables to only those necessary for current traffic (~2,000 routes). The routers' processing is significantly reduced, lowering the CPU load.

Router load (SD-WAN network): average time per packet: 0.5 ms (simplified rules). CPU load: 20%, handling 2,000 packets per second more efficiently. In the SDN network, the router's CPU load decreases by 60%, allowing it to handle twice the traffic without service degradation.

3.1.3. Impact of latency

When a client performs an online banking transaction involving multiple steps: authentication, data transfer to the server, and bank confirmation, each step requires a client-server interaction, generating an RTT.

- RTT in the traditional network (before SDN): 60 ms.
- RTT in the optimized network with SDN: 10 ms.

The transaction has 6 steps (login, authentication, balance check, account selection, confirmation, and notification). Total latency time (for transmission only) is calculated as:

$$Total\ Latency\ Time\ =\ RTT\ x\ Number\ of\ Steps$$

- Traditional network: 50 ms×6=300 ms
- SDN network: 10 ms×6=60 ms

The user experience can be slower in the traditional network, especially during traffic peaks. With SDN, total latency is reduced by 80%, resulting in a smoother experience. Higher latency in a bank transfer could delay confirmation, leading to uncertainty for the user or even errors in automated operations.

Reducing latency minimizes these risks and allows for a quicker system response, essential for customer trust and satisfaction in financial services. If this example is extended to a network with thousands of simultaneous users, optimizing the time can significantly improve the banking platform's responsiveness see Figure 8. Lower latency improves the user experience for critical company applications, such as online banking and other real-time financial services.

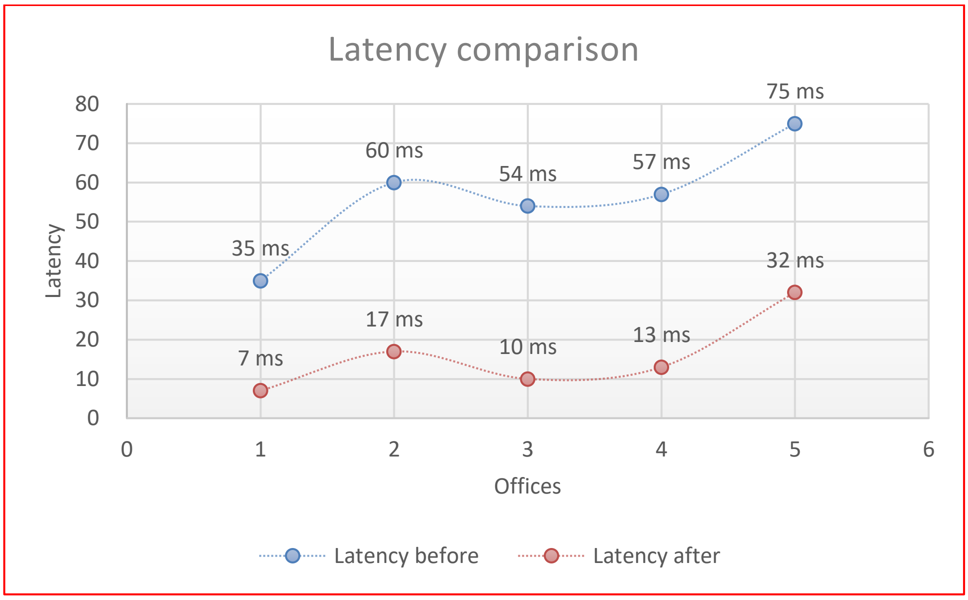

Figure 8. Latency comparison

3.2. Throughput

Throughput, or the network's transmission capacity, improved significantly after the implementation of SDN. While contracted speeds were rarely reached in the traditional network due to congestion and router limitations, the SDN network achieves consistent performance even during peak demand periods.

3.2.1. Throughput in the software-defined networking network

With the SDN network, throughput increased significantly across all locations:
- The offices now consistently reach the contracted 10 Mbps.
- The headend reaches up to 50 Mbps, especially for cloud applications. The SDN controller adjusts routing policies in real-time, redistributing traffic and avoiding bottlenecks. Critical flows, such as banking transactions, are prioritized while less important traffic is redirected to alternative routes.

Impact calculation during a traffic peak:
- Packet volume: 10,000 per second.
- Processing time per packet: 0.5 ms.
- Accumulated delay: 10,000×0.5=5,000 ms (5 seconds).

This represents a 75% reduction in response times compared to the traditional network.

3.2.2. Throughput in the traditional network

In the traditional network, reaching the contracted speed of 10 Mbps was a constant challenge due to congestion and the limited processing capacity of the routers. The routers faced high CPU loads, causing significant delays. During traffic peaks, response times increased, affecting critical applications such as real-time transaction approvals. Extreme congestion filled the router buffers, causing packet loss and decreasing system reliability, see Figure 9.

Impact calculation during a traffic peak:
- Packet volume: 10,000 per second.
- Processing time per packet: 2 ms.
- Accumulated delMay: total: 10,000×2=20,000 ms (20 seconds of accumulated delay).

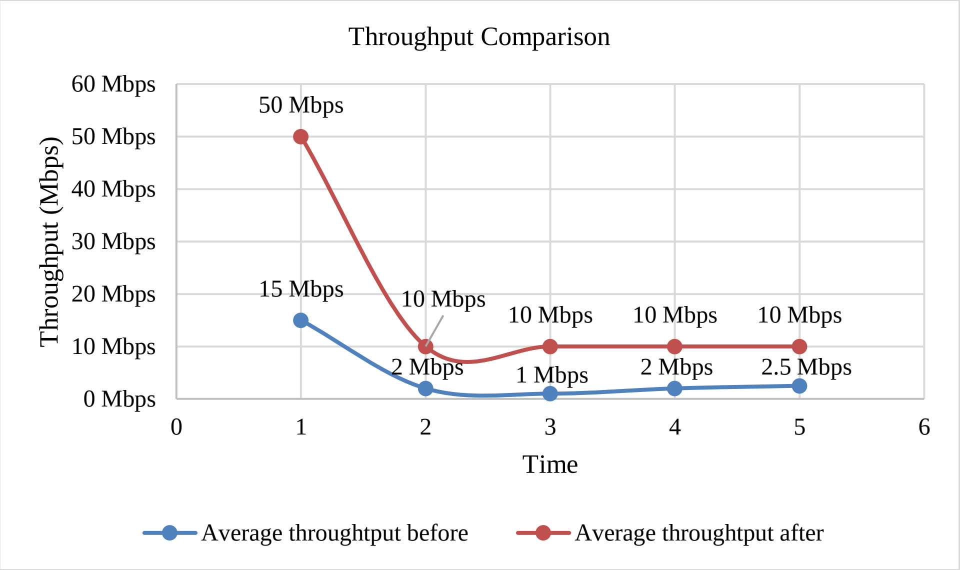

Figure 9. Throughput comparison

3.3. Packet loss

As for packet loss, a significant reduction was observed. In the traditional network, some offices experienced loss rates of up to 3%, which affected connection reliability. With the new SDN architecture, packet loss is virtually nonexistent across all locations, except for office A, where a slight percentage persists due to possible physical issues, such as fiber attenuation or the area being difficult to access. This improvement strengthens network stability and ensures safer and more reliable data transmission for the company's operations, see Figure 10.

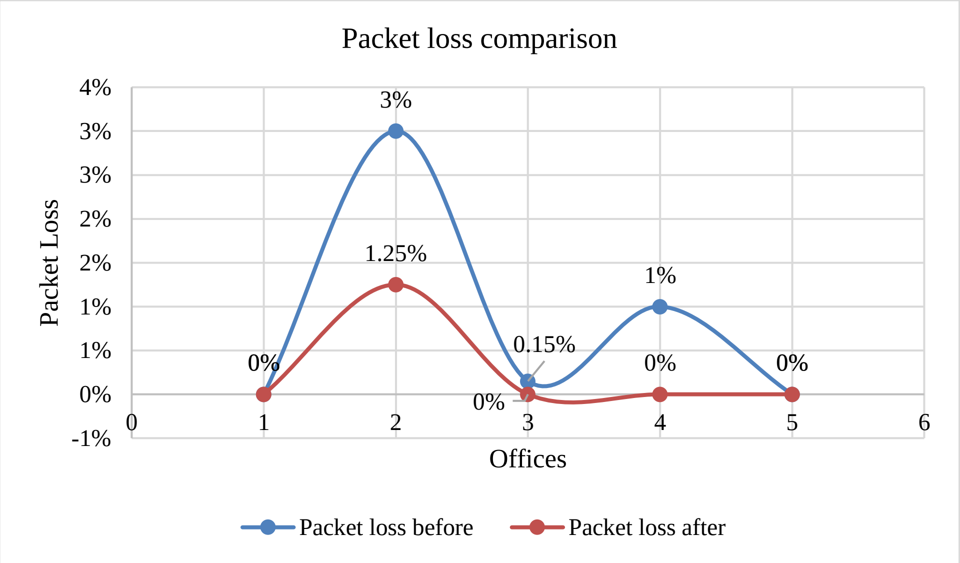

Figure 10. Packet loss comparison

4. DISCUSSION

The findings of this study confirm that implementing an SDN architecture can significantly enhance QoS in financial networks, particularly with respect to latency, throughput, and packet loss. These results are consistent with previous studies, such as [27], which demonstrated how SDN enables dynamic load balancing and traffic optimization in critical networks through centralized control and advanced QoS policies. Similar work in Latin America has emphasized that SDN can reduce the operational complexity of traditional architectures while improving flexibility during traffic spikes [28].

In the Peruvian financial sector, most institutions continue to rely on conventional MPLS-based infrastructures, which are associated with high operational costs, scalability limitations, and increased latency due to hub-and-spoke topologies. Furthermore, the dependency on managed services from external providers often limits customization and responsiveness. The results of this study suggest that an SDN-based approach, using controllers such as OpenDaylight, could effectively address these challenges by reducing average latency from 54–75 ms to 7–32 ms and sustaining a stable throughput of 10 Mbps under heavy loads.

From a regional perspective, countries such as Brazil and Chile are already advancing in the adoption of SDN in banking institutions, demonstrating improvements in operational efficiency and customer experience on digital platforms [29], [30]. These cases, combined with the present findings, reinforce the feasibility and necessity of adopting SDN in Peru to remain competitive and meet the increasing demands for network performance, scalability, and security in the financial sector.

5. CONCLUSION

This study demonstrated that the implementation of an SDN architecture in a financial institution network led to substantial improvements in QoS. Specifically, latency was reduced by 40%, throughput was stabilized at the contracted 10 Mbps across all locations, and packet loss was virtually eliminated. These enhancements ensure that the network can manage intensive traffic loads without service degradation, while also supporting critical financial applications such as secure online transactions and real-time processing.

The results validate the viability of SDN technologies in the Peruvian financial context, where traditional MPLS-based architectures still predominate. By adopting SDN, financial institutions can reduce operational costs, optimize service delivery, and align their infrastructures with global trends in digital banking. Future work should focus on evaluating the performance of alternative SDN controllers (e.g., ONOS and RYU), extending the architecture to larger-scale deployments, and testing interoperability with hybrid cloud environments. Such efforts would provide further evidence of the scalability and adaptability of SDN in diverse financial and enterprise contexts.

ACKNOWLEDGEMENTS

We especially thank the Technological University of Peru for its continued support throughout the development of this project, whose academic guidance and institutional collaboration were essential to the completion of this work.

FUNDING INFORMATION

The authors state that no external funding was received to support the research presented in this article.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Juan David Indigoyen Roncal	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	
Christian Ovalle Paulino	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest regarding the publication of this work.

DATA AVAILABILITY

Data availability is not applicable to this study, as no new datasets were generated or analyzed. All simulation results produced during the research are included within the article.

REFERENCES

[1] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 87–98, 2014, doi: 10.1145/2602204.2602219.

[2] I. Nenov, "Software-Defined Wide-Area Networks in the Coevolution of Network and Computational Bandwidth," *Comptes Rendus de L'Academie Bulgare des Sciences*, vol. 75, no. 12, pp. 1805–1812, Dec. 2022, doi: 10.7546/CRABS.2022.12.12.

[3] N. McKeown *et al.*, "OpenFlow: Enabling Innovation in Campus Networks," *Computer Communication Review*, vol. 38, no. 2, pp. 69–74, Mar. 2008, doi: 10.1145/1355734.1355746.

[4] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Computer Networks*, vol. 72, pp. 74–98, 2014, doi: 10.1016/j.comnet.2014.07.004.

[5] N. Gude *et al.*, "NOX: Towards an Operating System for Networks," *Computer Communication Review*, vol. 38, no. 3, pp. 105–110, Jul. 2008, doi: 10.1145/1384609.1384625.

[6] R. Patel, P. Patel, P. Shah, B. Patel, and D. Garg, "Software Defined Network (SDN) Implementation with POX Controller," in *3rd International Conference on Smart Electronics and Communication, ICOSEC 2022 - Proceedings*, pp. 65–70, Oct. 2022, doi: 10.1109/ICOSEC54921.2022.9952123.

[7] D. Erickson, "The Beacon OpenFlow controller," in *HotSDN 2013 - Proceedings of the 2013 ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, Aug. 2013, pp. 13–18, doi: 10.1145/2491185.2491189.

[8] T. D. Nadeau and K. Gray, "SDN: software defined networks; an authoritative review of network programmability technologies," Sebastopol, CA, USA: O'Reilly Media, 2013.

[9] T. V. K. Buyakar, H. Agarwal, B. R. Tamma, and A. A. Franklin, "Prototyping and Load Balancing the Service Based Architecture of 5G Core Using NFV," in *2019 IEEE Conference on Network Softwarization (NetSoft)*, Paris, France, 2019, pp. 228-232, doi: 10.1109/NETSOFT.2019.8806648.

[10] ON.LAB, "Introducing ONOS—a SDN network operating system for Service Providers," White Paper, vol. 1, p. 14, 2014. [Online]. Available: http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf. (Accessed: Jul. 7, 2025).

[11] Linux Foundation, "OpenDaylight." [Online]. Available: https://www.opendaylight.org/. (Accessed: Jul. 7, 2025).

[12] V. Kotronis, A. Gämperli, and X. Dimitropoulos, "Routing centralization across domains via SDN: A model and emulation framework for BGP evolution," *Computer Networks*, vol. 92, pp. 227–239, Dec. 2015, doi: 10.1016/j.comnet.2015.07.015.

[13] N. Demaku and A. Dermaku, "Improving Load Balancing and Scalability by Implementing Path Selection on BGP Using Multi SD-WAN," *Journal of Communications*, vol. 17, no. 4, pp. 250–259, 2022, doi: 10.12720/jcm.17.4.250-259.

[14] L. M. Elguea and F. M-Rios, "A new method to optimize BGP routes using SDN and reducing latency," *Procedia Computer Science*, vol. 135, pp. 163–169, 2018, doi: 10.1016/j.procs.2018.08.162.

[15] P. T. A. Quang, J. Leguay, X. Gong, and X. Huiying, "Global QoS Policy Optimization in SD-WAN," in *2023 IEEE 9th International Conference on Network Softwarization: Boosting Future Networks through Advanced Softwarization, (NetSoft)*, Madrid, Spain, Jun. 2023, pp. 202–206, doi: 10.1109/NetSoft57336.2023.10175407.

[16] M. C. Saxena, M. Sabharwal, and P. Bajaj, "Exploring path computation techniques in software-defined networking: a review and performance evaluation of centralized, distributed, and hybrid approaches," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 9s, pp. 553–567, Aug. 2023, doi: 10.17762/ijritcc.v11i9s.7468.

[17] P. T. A. Quang *et al.*, "Routing and QoS Policy Optimization in SD-WAN," *arXiv preprint*, 2022, doi: 10.48550/arXiv.2209.12515.

[18] M. Koerner and O. Kao, "Multiple service load-balancing with OpenFlow," in *2012 IEEE 13th International Conference on High Performance Switching and Routing (HPSR)*, Belgrade, Serbia, Jun. 2012, pp. 210–214, doi: 10.1109/HPSR.2012.6260852.

[19] R. Wang, D. Butnariu, and J. Rexford, "OpenFlow-based server load balancing gone wild," *USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services, Hot-ICE*, 2011.

[20] H. Riggs, A. Khalid, and A. I. Sarwat, "An Overview of SDN Issues—A Case Study and Performance Evaluation of a Secure OpenFlow Protocol Implementation," *Electronics*, vol. 14, no. 16, pp. 1-35, 2025, doi: 10.3390/electronics14163244.

[21] G. Pereira and E. Gamess, "Lineamientos para el Despliegue de Redes SDN/OpenFlow," *Revista Venezolana de Computación*, vol. 4, no. 2, pp. 21–33, 2017.

[22] E. Amiri, E. Alizadeh, and M. H. Rezvani, "Controller selection in software defined networks using best-worst multi-criteria decision-making," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1506–1517, Aug. 2020, doi: 10.11591/eei.v9i4.2393.

[23] J. C. Chico, D. Mejia and I. Bernal, "Implementación de un Prototipo de una Red Definida por Software (SDN) Empleando una Solución Basada en Hardware Digital TV View project Basic protypes for HPC View project," *XXV Jornadas en Ingeniería Eléctrica y Electrónica*, vol. 25, pp. 356–365, 2014.

[24] E. E. Haleplidis, E. K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," *Internet Research Task Force (IRTF)*, pp. 1–35, 2015.

[25] H. T. Zaw and A. H. Maw, "Traffic management with elephant flow detection in software defined networks (SDN)," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 3203–3211, Aug. 2019, doi: 10.11591/ijece.v9i4.pp3203-3211.

[26] M. Alizadeh *et al.*, "CONGA: Distributed congestion-aware load balancing for datacenters," in *Proceedings of the ACM SIGCOMM Conference*, pp. 503–514, 2014, doi: 10.1145/2619239.2626316.

[27] M. Karakus and A. Durresi, "Quality of Service (QoS) in Software Defined Networking (SDN): A survey," *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017, doi: 10.1016/j.jnca.2016.12.019.

[28] H. S. G. de Freitas, M. Schwarz, J. Bezerra, and J. Ibarra, "Deploying SDN experiments in Latin America: the ONOS and SDN-IP application use case at AmLight," *Sexta Conferencia de Directores de Tecnología de Información*, 2016.

[29] D. F. T. Pontes, M. F. Caetano, G. P. R. Filho, L. Z. Granville, and M. A. Marotta, "On the Transition of Legacy Networks to SDN - An Analysis on the Impact of Deployment Time, Number, and Location of Controllers," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM),* Bordeaux, France, 2021, pp. 367-375.
[30] H. Kim and N. Feamster, "Improving network management with software defined networking," in *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114-119, Feb. 2013, doi: 10.1109/MCOM.2013.6461195.

BIOGRAPHIES OF AUTHORS

Juan David Indigoyen Roncal is a 9th-cycle Telecommunications Engineering student. His analytical aptitude for information management allows him to perform effectively in various fields. With strong logical and mathematical reasoning skills, along with the ability to synthesize and deduce information, he aims to demonstrate his full potential and is open to teamwork. He specializes in communication networks for businesses and service providers and is CCNA certified. He can be contacted at email: u20200788@utp.edu.pe.

Dr. Christian Ovalle Paulino is an associate professor at the Faculty of Engineering of the Technological University of Peru, Lima-Peru. He has a Ph.D. in Systems Engineering with a specialization in artificial intelligence. His research areas are process mining, business data analysis and pattern recognition. He is CEO of the 7D consultancy dedicated to the Investigation of intelligent solutions. He has participated in different research projects, receiving awards from the Peruvian Ministry of Defense and the Armed Forces Army for the best general researcher, which is a technology-based company and his innovative products received national and international recognition. He has filed a number of patents and industrial designs on his innovative ideas. His research interests include data mining, artificial intelligence, image/signal processing, bibliometrics, and pattern recognition. He can be contacted at email: dovalle@utp.edu.pe.