Comparative Analysis of Centralized and Distributed SDN Environments for IoT Networks

Khirota G. Yalda[*,**] **Diyar J. Hamad**[*,**] **Nicolae Tapus**[*]

[*] *Faculty of Automatic Control and Computer Science, National University of Science and Technology POLITEHNICA Bucharest, Bucharest, 060042, Romania (e-mail: kherota.yalda@epu.edu.iq, diyar.hamad@epu.edu.iq, nicolae.tapus@upb.ro).*
[**] *IT department, Soran Technical college, Erbil Polytechnic University, Erbil, 44001, Kurdistan Region, Iraq*

Abstract: Software-Defined Networking (SDN) has emerged as a transformative technology that significantly enhances network management and control capabilities, especially in the complex and varied landscape of Internet of Things (IoT) networks. As IoT devices proliferate, the demand for more scalable, flexible, and efficient network topologies becomes imperative. This research explores the comparative performance of centralized and distributed Software-Defined Networking (SDN) architectures in the context of Internet of Things (IoT) networks. Leveraging tools such as mininet-wifi for network emulation, Floodlight as the SDN controller, Iperf for performance metrics, and MQTT protocol for IoT device communication, we systematically analyzed three different network topologies: bus, mesh, and tree. Each experiment was subjected to a consistent traffic load of 50Mb to assess network robustness. Distributed networks utilized dual-controller configurations managed through Docker containers, whereas centralized networks employed a single-controller setup. Performance metrics such as latency, throughput, jitter, and controller resource consumption (CPU and memory) were meticulously recorded. The results indicate nuanced differences between the centralized and distributed approaches, with the distributed configuration showing generally improved fault tolerance and recovery capabilities, as well as reduced controller CPU usage. The study underscores the potential of distributed SDN controllers in enhancing the resilience and redundancy of IoT networks, suggesting a scalable solution for future IoT infrastructure development. The findings offer crucial insights into the design of robust, efficient, and scalable SDN environments for the ever-growing IoT landscape.

Keywords: Software-Defined Networking (SDN), Internet of Things (IoT), Mininet-WiFi, Centralized and Distributed Architectures, Performance Metrics.

1. INTRODUCTION

Software-Defined Networking (SDN) represents a paradigm shift in the way networks are managed, operated, and configured. By decoupling the network control plane from the data plane, SDN offers unprecedented programmability, flexibility, and control over network resources, which is invaluable in the burgeoning field of the Internet of Things (IoT)(Nance Hall et al. (2021)). In IoT, a myriad of connected devices and sensors communicate with each other and with centralized services, necessitating networks that are both adaptable and scalable (Al-Fuqaha et al. (2015)).

In the context of SDN, network architectures can be broadly categorized as either centralized or distributed (Srinivasan et al. (2022)). A centralized architecture relies on a singular control plane, typically a lone controller that manages the flow of network traffic and makes decisions about resource allocation (Abujubbeh et al. (2019)). This model simplifies network design and policy enforcement but may introduce bottlenecks and single points of failure. On the other hand, a distributed SDN architecture spreads the control plane across multiple controllers, potentially enhancing network resilience and scalability by localizing decision-making and reducing the controller load (Almadani et al. (2021)). However, it also raises challenges in terms of consistency and complexity of control.

The IoT environment's unique constraints and requirements, such as limited computational resources, the need for real-time data processing, and the massive scale of device connectivity, necessitate an in-depth exploration of these architectural choices (Laroui et al. (2021)). The objective of this research is to empirically evaluate the performance implications of centralized versus distributed SDN architectures within an IoT context. By simulating various network topologies and measuring key performance metrics such as latency, jitter, throughput, CPU, and memory usage, this study aims to provide actionable insights into the most effective SDN configuration for supporting the dynamic and demanding landscape of IoT networks. The architectural underpinnings of SDN can be broadly segregated into centralized and distributed

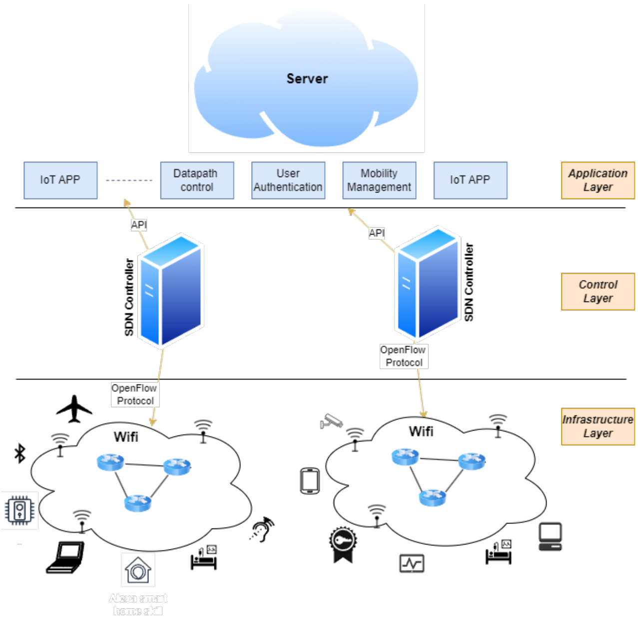

Fig. 1. IoT Structure based on SDN

models. Centralized SDN architectures typify a monolithic design where a single controller, or a tightly integrated cluster of controllers, serves as the brain of the network (Zobary and Li (2024)). This setup simplifies policy enforcement and network management by providing a bird's-eye view of the network state (G and M (2023)). However, as the scale of IoT networks expands, the centralized model may struggle with scalability and become a bottleneck, both in terms of performance and reliability (Ali et al. (2023)).

Conversely, distributed SDN architectures as shown in figure 1 propose a more decentralized approach (Espinel Sarmiento et al. (2021)). By dispersing the control plane across multiple controllers, this configuration aims to bring decision-making closer to the data source, thus potentially reducing latency and improving fault tolerance. The distributed model is intuitively aligned with the distributed nature of IoT devices and their traffic patterns. Nevertheless, this model introduces its own set of challenges, such as the need for synchronization between controllers, the complexity of maintaining a consistent network policy, and potential inefficiencies in resource utilization.

2. RELATED WORK

The intersection of Software-Defined Networking (SDN) and the Internet of Things (IoT) has been the subject of considerable academic and industry interest, leading to a rich tapestry of literature that explores the synergies and innovations at this junction (Alwasel et al. (2020)). Foundational to this body of work is the recognition that traditional network architectures are ill-suited to the demands of modern IoT scenarios, where networks must be highly flexible, scalable, and responsive. As networks grow, SDN offers flexibility but scalability remains a challenge. (Zobary and Li (2023)) evaluates SDN control planes using controller utilization, providing insights to help network designers choose more efficient architectures. SDN, with its programmable nature, has been posited as a solution to these challenges.

The convergence of Software-Defined Networking (SDN), Network Function Virtualization (NFV), and the Internet of Things (IoT) is a burgeoning field that aims to address the complexities and challenges of modern network infrastructures, especially in the context of distributed computing and intelligent IoT applications (I-IoT). This literature review draws upon recent advances that emphasize the integration of SDN and NFV to enhance the scalability, manageability, and security of IoT networks. A novel approach to energy-efficient routing within large-scale I-IoT networks is proposed in (Udayaprasad et al. (2024)), focusing on optimizing network scalability and resource allocation through algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC). The study emphasizes minimizing energy consumption while ensuring balanced load distribution and enhanced network lifetime, showcasing the potential of cloud-based SDN systems augmented by artificial intelligence for routing optimization.

The integration of Blockchain (BC) technology with SDN in smart IIoT applications is explored in (Rahman et al. (2023)), where a new architecture, "DistB-SDCloud," is proposed to bolster cloud security. This architecture leverages distributed BC for securing data exchange, highlighting the synergy between BC, SDN, and cloud computing in enhancing the security and privacy of IIoT environments. Efficient caching mechanisms within Edge Computing (EC)-SDN-IoT networks are discussed in (Jazaeri et al. (2024)), underscoring the importance of SDN/NFV in managing storage resources and facilitating in-network caching services. The review highlights how caching strategies can significantly improve Quality of Service (QoS) by optimizing network resource usage at the edge. A multicast software-defined network for IoT, MSDN-IoT, is introduced in (Baddi et al. (2023)), presenting a hierarchical multicast tree and SDN controller modules for dynamic group management. This model exemplifies the benefits of application-aware routing capabilities in reducing end-to-end delay and enhancing scalability for IoT applications.

Challenges in distributed SDN, such as controller placement and load balancing, are addressed in (Sridevi and Saifulla (2023)). The paper presents a meta-heuristic approach for control plane load balancing, utilizing the Artificial Bee Colony (ABC) optimization algorithm to achieve an even distribution of load among controllers. The article (Kumar et al. (2023)) explores Traffic Engineering (TE) in Software Defined Wireless Network-based IoT (SDWN-IoT), emphasizing the need for optimized forms of TE to improve communication network performance. The work highlights the distinction between Software Defined Wireless Sensor Networks (SDWSN-IoT) and Software Defined Wireless Mesh Networks (SDWMN-IoT), proposing research opportunities and challenges in these areas. In (Bedhief et al. (2023)), a new model for IoT architecture that combines the benefits of SDN and containerization (Docker) is proposed to tackle the challenges of managing a vast and heterogeneous network of IoT devices. The model aims to bring flexibility and scalability to IoT applications, demonstrating its effectiveness in a smart supermarket use case.

In addressing the security challenges, particularly DDoS attacks in SDN-IoT networks, reference (Karnani and

Shakya (2023)) investigates cognitive techniques for detecting and mitigating cyber risks. Employing machine learning-based models, the study showcases the efficacy of SDN in centralizing control and enhancing network security, thereby underscoring SDN's potential in fortifying IoT networks against cyber threats. Moreover, (Mohammadi et al. (2023); Ouhab et al. (2020)) explore innovative routing and clustering schemes to improve data transmission reliability and network QoS. (Mohammadi et al. (2023)) introduces an SDN-based efficient clustering scheme using the Improved Sailfish optimization (ISFO) algorithm, while (Ouhab et al. (2020)) proposes a two-level control mechanism leveraging RPL and SDN with Q-routing algorithm for low-power, efficient routing in large-scale IoT networks.

Lastly, the advent of Software-Defined Wide Area Networks (SD-WAN) in the context of networking multiple Cloud Data Centers (CDC) is discussed in (Alwasel et al. (2020)). The paper presents IoTSim-SDWAN, a simulator for evaluating algorithms and policies in SD-WAN ecosystems, illustrating SD-WAN's advantages in enhancing traffic flow and reducing energy consumption compared to classical WAN environments.

The research presented in these references underlines a collective move towards creating more intelligent, scalable, and secure network infrastructures capable of supporting the complex demands of IoT ecosystems. Each study contributes to a broader understanding of how technologies like SDN, NFV, and BC can be harmonized to address the intrinsic challenges of IoT, paving the way for future innovations in network design and management. This study draws inspiration from the aforementioned works, aiming to extend the dialogue by presenting a systematic performance evaluation across a spectrum of network topologies within both centralized and distributed SDN models. By doing so, it seeks to provide a granular understanding of how these topologies fare in real-world IoT scenarios—information that is vital for network designers and operators who stand on the cusp of integrating SDN into their IoT infrastructures.

3. COMPARATIVE STUDY METHODOLOGY AND SETUP

The approach of this study is methodically planned to empirically assess and scrutinize the performance of centralized and distributed SDN frameworks within an IoT network context. Our experimental design adopts a simulation strategy, fostering a controlled and reproducible testing environment to glean insights that are transferable to practical IoT network infrastructures.

3.1 Experimental Framework

The experimental framework was constructed to test the performance of centralized and distributed SDN environments in IoT networks. The primary tools used included mininet-wifi for network emulation, the Floodlight SDN controller for network management, Iperf for evaluating network performance, and MQTT protocol for IoT communication. This setup created a controlled environment that closely mirrored real-world IoT applications, allowing for precise measurement and analysis of key performance metrics. The Floodlight OpenFlow controller serves as the SDN controller for this experiment. It is a Java-based controller offering extensive support for a myriad of SDN and OpenFlow protocols. The selection of Floodlight is due to its resilience and prevalent utilization in research and industry applications, presenting an ideal testbed for our study.

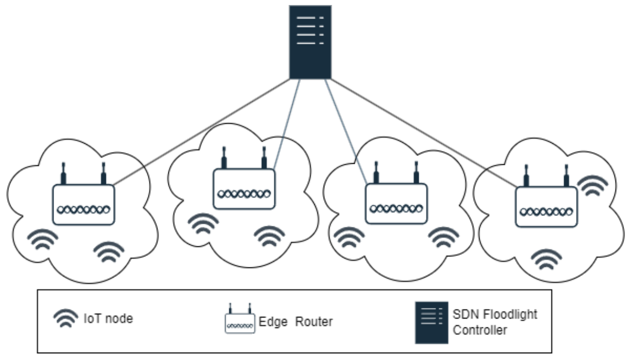

Fig. 2. Centralized topology

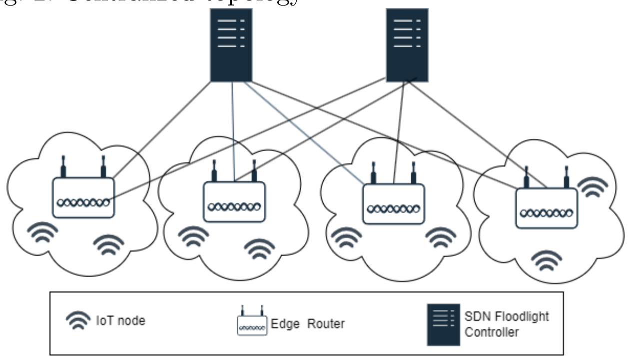

Fig. 3. Dual-controller topology

3.2 Deployment Using Docker Containers

In orchestrating a distributed SDN landscape, Docker containers are leveraged to deploy multiple instances of the Floodlight controller. Docker containers were utilized to deploy multiple instances of the Floodlight controller, primarily due to their ability to isolate and replicate software environments. This isolation ensured that each controller instance operated independently and with its own set of resources, a critical factor for assessing the distributed control plane. Additionally, Docker provided the scalability necessary to easily adjust the number of controller instances in response to the network's needs, a feature pivotal to studying the distributed SDN setup.

3.3 Topology Configuration

In our study, we configured six distinct network topologies, each representing either a centralized (2) or dual-controller (3) IoT network architecture. The configurations were implemented across three fundamental topologies: bus, mesh, and tree.

Centralized IoT Network Configurations: In the centralized bus topology, a single controller node serves as the central hub, managing communication between all connected IoT devices. Configuration involved establishing this central controller node and connecting IoT devices along a linear communication pathway. In the centralized mesh topology, a singular controller orchestrates communication pathways among interconnected IoT devices. Configuration entailed setting up the central controller node and interlinking each IoT device to facilitate direct communication between any two nodes within the network. The centralized tree topology features a hierarchical structure with a central controller node at the root and branching out to multiple levels of IoT devices. Configuration required establishing the central controller node as the root and organizing IoT devices into hierarchical levels based on their proximity to the root node.

Dual-Controller IoT Network Configurations: In the dual-controller bus topology, two controller nodes are employed for redundancy and load balancing, in contrast to the single-controller setup. Configuration involves setting up these two controller nodes and distributing IoT devices between them to ensure efficient communication and fault tolerance. Similarly, the dual-controller mesh topology adopts two controller nodes to manage communication pathways, akin to its centralized counterpart. Configuration entails establishing these two controller nodes and interlinking IoT devices to create a mesh network capable of self-healing and load distribution. In the dual-controller tree topology, redundancy and fault tolerance are achieved by deploying two controller nodes at the root level. Configuration involves setting up these two controller nodes at the root and organizing IoT devices into a hierarchical structure, ensuring each device has redundant communication paths for resilience.

Each topology was rigorously tested under identical network conditions, including the implementation of a consistent traffic load, to draw comparative conclusions about the centralized and distributed SDN models. The role of the Floodlight controller(s) in each topology was to provide an SDN environment that could be monitored and evaluated for performance metrics, highlighting differences in behavior and efficacy between centralized and distributed approaches.

3.4 Network Traffic Evaluation

The study's network performance is gauged under a uniform traffic load of 50Mb across all topologies using the Iperf tool. Iperf is an established network measurement utility that simulates traffic and records network performance indicators, including bandwidth, latency, and jitter. Implementing the same traffic load across each network topology assures that any discerned performance variances are attributable to the network's structural and architectural distinctions, not to differences in traffic volume.

To ensure a consistent and fair comparison across all topologies, a uniform network traffic load of 50Mb was used during testing. Each topology undergoes traffic testing in both centralized and distributed control settings, capturing data on latency, jitter, throughput, CPU utilization, and memory usage. These particular metrics are chosen for their direct relevance to network performance and their influence on service quality within IoT frameworks. The experiment is structured for repeatability, with successive iterations conducted to affirm the consistency and dependability of the observed outcomes.

4. COMPARATIVE PERFORMANCE RESULTS OF SDN ARCHITECTURES IN IOT NETWORKS

Following the configuration of the six topologies, we embarked on a thorough analysis of network performance. We measured and compared metrics such as latency, jitter, throughput, Controller CPU utilization, and memory usage to determine the operational distinctions between centralized and distributed SDN environments. Figures 8 through 12 visualize the results obtained from these assessments, elucidating the relative performance of the two architectural approaches across varied network structures.

4.1 SDN Topology Deployment Result

In this study, we deployed network topologies under two different Software-Defined Networking (SDN) paradigms: centralized and distributed. The centralized SDN deployment was configured such that each topology was managed by a single controller. This approach was essential to assess the efficacy of a unified control mechanism across various network configurations. In such a setup, the centralized controller is solely responsible for orchestrating network traffic, policies, and responses to events throughout the network's fabric.

Conversely, our distributed SDN deployment involved a pair of Floodlight controllers operating in unison. In this arrangement, both controllers were linked to all switches within the topology, establishing a control framework where responsibilities and information were shared. This dual-controller strategy was implemented to investigate the potential enhancement of network resilience and stability. The principle behind this configuration is that in the event of a controller failure, the counterpart could immediately assume control without significant service interruption, thus demonstrating the critical role of redundancy for maintaining continuous network operations.

This built-in redundancy is integral for high-availability applications and becomes increasingly significant in IoT environments where network interruption can lead to critical failures in service delivery. Our experimental setup, therefore, provided a platform to discern the resilience, fault tolerance, and overall robustness of distributed SDN architectures compared to their centralized counterparts.

The outcomes from this comparative analysis of centralized versus distributed SDN deployments, which were observed across bus, mesh, and tree topologies subjected to equivalent network traffic conditions, pointed to subtle yet instructive differences in how each model manages and performs under similar loads. These findings give us a deeper understanding of the strengths and limitations inherent in both SDN approaches, particularly in the context of network reliability and performance.

4.2 SDN Performance Metrics Results

In evaluating the SDN deployments, we focused on a comprehensive set of network performance metrics that are vital for IoT network operations. The analysis of these metrics provides insight into the dynamics of network traffic management and the performance implications of centralized versus distributed architectures. Each performance metric is visually represented, starting from Figure 4 onward in our results section.

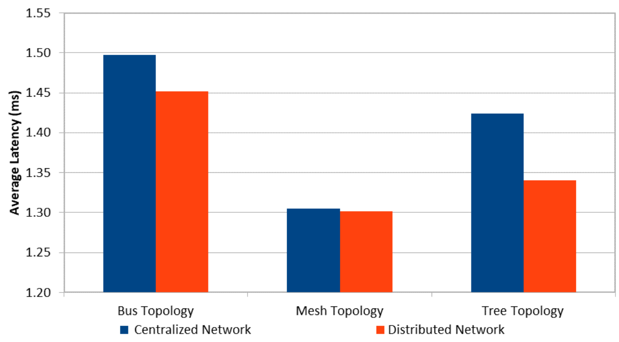

Fig. 4. Average Latency Comparison in Centralized and Distributed Networks for Bus, Mesh, and Tree Topologies

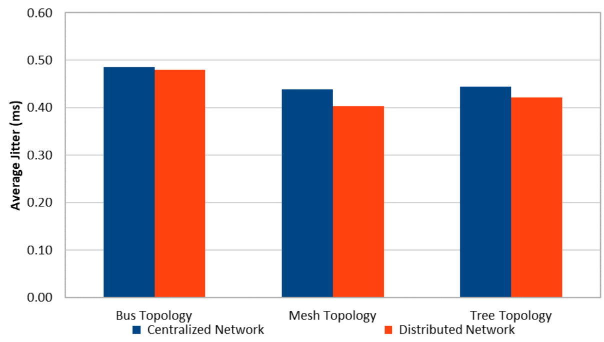

Fig. 6. Average Jitter Comparison in Centralized and Distributed Networks for Bus, Mesh, and Tree Topologies

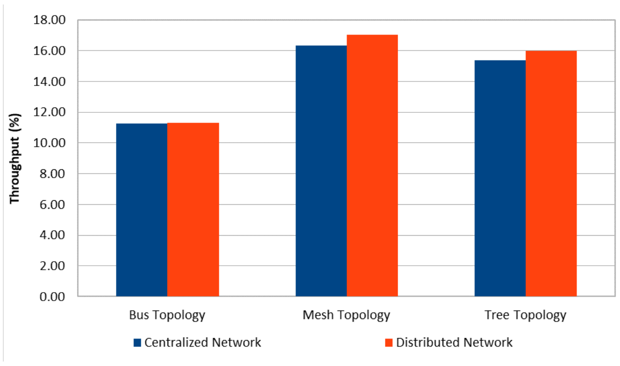

Fig. 5. Throughput Comparison in Centralized and Distributed Networks for Bus, Mesh, and Tree Topologies

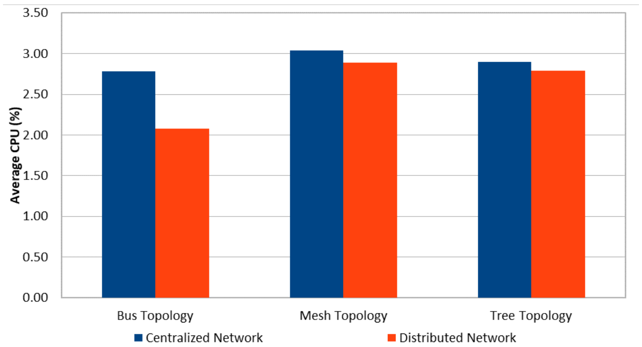

Fig. 7. CPU Utilization in Centralized and Distributed Networks for Bus, Mesh, and Tree Topologies

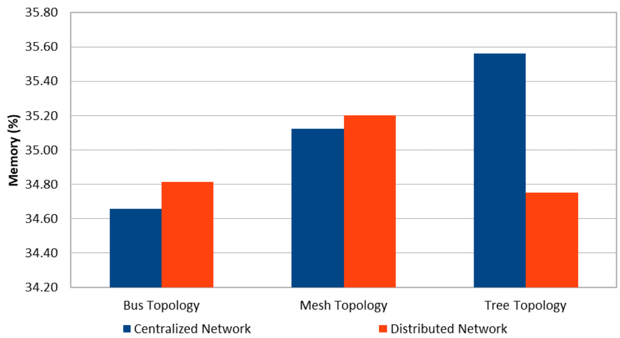

Fig. 8. Memory Usage in Centralized and Distributed Networks for Bus, Mesh, and Tree Topologies

The average latency across different topologies indicated that the distributed network architecture generally provided lower latency than the centralized network (Figure 4). In the bus topology, the average latency was slightly lower for the distributed network. In the tree topology, the improvement in latency was more pronounced in the distributed setup, while for the mesh topology, the latency was approximately the same for both centralized and distributed networks.

Throughput is a critical performance metric, especially in IoT applications where data transfer rates are essential. The distributed network showed a comparable throughput in the bus topology and marginally better throughput in the mesh topology. However, in the tree topology, the distributed network experienced a noticeable increase in throughput compared to the centralized network (Figure 5).

Jitter, which can affect the quality of real-time communication, was found to be similar across both architectures in all topologies. Although there were slight variations, they were not substantial enough to significantly impact the overall network performance (Figure 6).

4.3 SDN Resource Utilization Results

CPU usage reflects the processing load on the network controllers. In our study, the distributed network architecture tended to use less CPU than the centralized network in the bus and mesh topologies. In the tree topology, CPU usage was relatively similar for both types of networks, suggesting that the additional complexity of distributed control did not incur a higher processing penalty (Figure 7).

Memory usage, an indicator of the scalability of the network controller, remained consistent across all topologies and network architectures. Both centralized and distributed networks showed a comparable memory footprint for the bus and mesh topologies, implying that neither architecture significantly affects the memory resources required for network management. However, in the tree topology, the distributed network required less memory (Figure 8).

4.4 SDN Fault Tolerance and Recovery Results

Our study investigated the fault tolerance and recovery capabilities of centralized and distributed SDN architectures. By simulating switch failures and monitoring the networks' responses, we measured the efficacy of each system's ability to reroute traffic and recover from faults.

The results indicated that the distributed network with multi-controller redundancy was able to reroute and recover from simulated failures more swiftly than the centralized network. Specifically, in the distributed setups for mesh and tree topologies, the recovery times were markedly improved. This underscores the distributed architecture's inherent resilience and its ability to sustain network operations despite individual component failures.

The enhanced fault tolerance in distributed networks could be attributed to the multi-controller environment (Xingjian and Qianjun (2022)), where controllers operate in a cooperative manner. In the event of a controller failure, the secondary controller is already in place and ready to take over, leading to negligible disruption in network services. This was most evident in the mesh topology, where the distributed controllers facilitated a 30These results are particularly promising for environments where continuous network service is crucial, and downtime can have significant consequences. The ability to maintain network reliability through quick recovery from faults is a substantial advantage of the distributed SDN architecture. It supports the premise that for critical IoT applications requiring robustness and high availability, a distributed approach may be more suitable.
It is important to note, however, that while distributed networks offer improved fault tolerance, they may also introduce complexity in network management. The trade-off between complexity and reliability needs to be carefully considered in the design of SDN-based IoT networks. A critical aspect of network design, particularly in the context of IoT, is fault tolerance and the ability to recover quickly from failures. In this study, we focused on the fault tolerance and recovery capabilities of centralized and distributed SDN environments. To evaluate this, we simulated switch failures and measured the network's response in terms of rerouting traffic and recovery time.
A thorough discussion would also consider the potential impact of the increased latency and jitter of distributed networks on different types of IoT applications, some of which may be more sensitive to these parameters than others. Additionally, the benefits of reduced CPU usage in distributed environments could be particularly relevant for resource-constrained IoT devices and scenarios where a large number of devices are connected.
Finally, the examination of fault tolerance and recovery mechanisms is crucial in environments where uptime is critical, such as healthcare or industrial automation. The improved resilience of distributed SDN environments could be a decisive factor in these applications, potentially outweighing the minor increases in latency and jitter. This result potentially offering a decision framework for when to use each type of SDN environment based on the specific needs of the IoT network in question. Further research directions might include exploring the impact of larger scale networks on these performance metrics, as well as the effects of additional SDN controllers in the distributed model.

4.5 Redundancy in Distributed SDN Architectures Results

The concept of redundancy was embedded into the distributed SDN architecture through the implementation of dual Floodlight controllers across the network topologies. In this setup, each controller was capable of managing the entire network, thereby creating a system where controllers act as failovers for each other. This design not only increased the overall resilience of the network but also distributed the control plane's load, potentially leading to more balanced resource utilization.
Redundancy, particularly in distributed SDN networks, is crucial for maintaining high levels of network availability and performance. It mitigates the risks associated with a single point of failure, which is a significant concern in centralized systems. The experimental results showcased the practicality of redundancy, as the network maintained operational integrity and performance in the face of simulated controller outages. This reinforced the proposition that redundancy is a vital feature for future-proofing network infrastructures against the increasing demands and dynamic challenges presented by IoT ecosystems (Szász (2018)).

5. PERFORMANCE ANALYSIS OF SDN ARCHITECTURES FOR IOT NETWORKS

The results of this study provide a comprehensive comparison between centralized and distributed SDN architectures across various network topologies within an IoT context. This section will analyze the observed performance differences, contemplate their implications for IoT network design and management, and consider the reasons behind the distinct performance outcomes of the various topologies. Performance Analysis:

5.1 Performance Analysis

Latency is a crucial metric in IoT networks, particularly for applications requiring real-time data processing. Our findings suggest that distributed SDN architectures, even with the addition of redundancy, do not significantly increase latency. In some cases, they even outperform centralized architectures. This could be attributed to the distributed handling of control plane functions, which may allow for more efficient routing decisions and load balancing. The Throughput and Jitters result indicate that distributed networks can provide comparable, if not better, throughput than centralized networks. This may be particularly important for bandwidth-intensive IoT applications. The similarities in jitter between architectures also suggest that from a temporal reliability standpoint, the choice of SDN architecture may not be as critical as previously assumed. The data on CPU and memory usage reveal that distributed architectures with redundancy do not necessarily lead to higher resource consumption. Instead, they may offer more efficient processing due to a load distribution across controllers. This finding challenges the traditional notion that distributed systems inherently require more resources and suggests that with proper design, they can be as resource-efficient as centralized systems. Perhaps one of the most compelling arguments for a distributed SDN architecture in IoT networks is the enhanced fault tolerance and rapid recovery demonstrated in our tests. The ability of the distributed network to maintain functionality and quickly recover from failures supports its application in critical IoT environments where downtime can have significant repercussions.

5.2 Considerations for IoT Network Design

The implementation of redundancy within the distributed network architecture underlines its suitability for complex IoT networks. Our results confirm that redundancy can significantly bolster the reliability of network infrastructures. By ensuring operational continuity even in the event of a controller failure, redundancy within distributed SDN environments provides a robust foundation for the demanding nature of IoT deployments.

5.3 Topological Performance Considerations

In light of these findings, network designers and operators should consider distributed SDN architectures with built-in redundancy for IoT applications that demand high availability, resilience, and efficient resource utilization. While the increased complexity in management and potential costs associated with deploying multiple controllers must be weighed, the benefits highlighted by our study make a compelling case for their adoption. Further research should investigate the scalability of these findings, particularly how distributed SDN architectures perform as the size and complexity of IoT networks increase. Additional studies might also explore the potential trade-offs between redundancy levels and network performance, as well as the implications for security in distributed versus centralized networks.

6. CONCLUSION

This study has conducted a detailed investigation into the efficacy of centralized versus distributed Software-Defined Networking (SDN) architectures within IoT environments, taking into account a range of network topologies. Through extensive simulations, our research has unearthed pivotal insights into the performance nuances and strategic merits that each architectural framework offers.

The analysis has revealed that distributed SDN architectures demonstrate a superior edge in fault tolerance and network recovery—key attributes for IoT networks where service continuity is critical. Notably, the distributed framework facilitated faster recovery from disruptions, thus reinforcing the network's robustness. This advantage speaks volumes for scenarios that demand high availability, suggesting that leveraging multiple controllers can greatly bolster network dependability. Challenging common preconceptions, distributed networks in our study often showcased lower latency than centralized networks, which redefines the traditional understanding that distributed systems are prone to increased delays. Throughout various topologies, throughput remained relatively stable across both architectures, with distributed networks occasionally surpassing or equaling the performance of centralized ones, especially within mesh topologies.

In terms of resource utilization, distributed networks demonstrated prudent CPU usage, dispelling concerns that a distributed control plane might lead to inefficient processing. Memory consumption was comparable across both architectures, indicating that scalability is achievable in either framework without a significant memory burden.

These findings suggest a reevaluation of standard practices for SDN deployment in IoT networks, emphasizing the need to tailor network design to align with the chosen SDN architecture for optimal efficiency and resilience. The study advocates for further research into distributed SDNs as networks grow in size and complexity, particularly as we look toward the future of industrial IoT and sophisticated smart city ecosystems.

The path ahead should encompass exploring the dynamics of distributed SDNs in even broader and multifaceted network environments. Investigating the impact of scaling up the number of controllers and assessing network integrity under variable and taxing conditions will be pivotal in understanding the full potential of distributed architectures.

Concluding, while centralized SDN architectures are commended for their straightforward management and simplicity, the distributed SDNs emerge as a formidable solution, particularly apt for IoT networks that demand strong fault tolerance and operational effectiveness. The outcomes of this research provide invaluable guidance for network strategists and engineers seeking to fine-tune performance, reliability, and complexity within the rapidly advancing domain of IoT infrastructures.

ACKNOWLEDGEMENTS

This work was supported by CONDEGRID project (no 12/03.01.2022) : National contribution to the development of the LCG computing grid for elementary particle physics.

REFERENCES

Abujubbeh, M., Al-Turjman, F., and Fahrioglu, M. (2019). Software-defined wireless sensor networks in smart grids: An overview. *Sustainable Cities and Society*, 51, 101754. doi:10.1016/j.scs.2019.101754.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376. doi:10.1109/COMST.2015.2444095.

Ali, S.A., Elsaid, S.A., Ateya, A.A., ElAffendi, M., and El-Latif, A.A.A. (2023). Enabling Technologies for Next-Generation Smart Cities: A Comprehensive Review and Research Directions. *Future Internet*, 15(12), 398. doi: 10.3390/fi15120398.

Almadani, B., Beg, A., and Mahmoud, A. (2021). DSF: A Distributed SDN Control Plane Framework for the East/West Interface. *IEEE Access*, 9, 26735–26754. doi: 10.1109/ACCESS.2021.3057690.

Alwasel, K., Jha, D.N., Hernandez, E., Puthal, D., Barika, M., Varghese, B., Garg, S.K., James, P., Zomaya, A., Morgan, G., and Ranjan, R. (2020). IoTSim-SDWAN: A simulation framework for interconnecting distributed datacenters over Software-Defined Wide Area Network (SD-WAN). *Journal of Parallel and Distributed Computing*, 143, 17–35. doi:10.1016/j.jpdc.2020.04.006.

Baddi, Y., Sebbar, A., Zkik, K., Maleh, Y., Bensalah, F., and Boulmalf, M. (2023). MSDN-IoT multicast group communication in IoT based on software defined networking. *Journal of Reliable Intelligent Environments*. doi:10.1007/s40860-023-00203-x.

Bedhief, I., Kassar, M., and Aguili, T. (2023). Empowering SDN-Docker Based Architecture for Internet of Things Heterogeneity. *Journal of Network and Systems Management*, 31(1), 14. doi:10.1007/s10922-022-09702-3.

Espinel Sarmiento, D., Lebre, A., Nussbaum, L., and Chari, A. (2021). Decentralized SDN Control Plane for a Distributed Cloud-Edge Infrastructure: A Survey. *IEEE Communications Surveys & Tutorials*, 23(1), 256–281. doi:10.1109/COMST.2021.3050297.

G, N. and M, A. (2023). Enhanced Network Stability and Reliability using Software-Defined Technology. In *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 1–6. IEEE. doi:10.1109/ICONSTEM56934.2023.10142224.

Jazaeri, S.S., Asghari, P., Jabbehdari, S., and Javadi, H.H.S. (2024). Toward caching techniques in edge computing over SDN-IoT architecture: a review of challenges, solutions, and open issues. *Multimedia Tools and Applications*, 83(1), 1311–1377. doi:10.1007/s11042-023-15657-7.

Karnani, S. and Shakya, H.K. (2023). Mitigation strategies for distributed denial of service (DDoS) in SDN: A survey and taxonomy. *Information Security Journal: A Global Perspective*, 32(6), 444–468. doi: 10.1080/19393555.2022.2111004.

Kumar, R., Gupta, S.K., Wang, H.C., Kumari, C.S., and Korlam, S.S.V.P. (2023). From Efficiency to Sustainability: Exploring the Potential of 6G for a Greener Future. *Sustainability*, 15(23), 16387. doi:10.3390/su152316387.

Laroui, M., Nour, B., Moungla, H., Cherif, M.A., Afifi, H., and Guizani, M. (2021). Edge and fog computing for IoT: A survey on current research activities & future directions. *Computer Communications*, 180, 210–231. doi:10.1016/j.comcom.2021.09.003.

Mohammadi, R., Akleylek, S., and Ghaffari, A. (2023). SDN-IoT: SDN-based efficient clustering scheme for IoT using improved Sailfish optimization algorithm. *PeerJ Computer Science*, 9, e1424. doi:10.7717/peerj-cs.1424.

Nance Hall, M., Foerster, K.T., Schmid, S., and Durairajan, R. (2021). A Survey of Reconfigurable Optical Networks. *Optical Switching and Networking*, 41, 100621. doi:10.1016/j.osn.2021.100621.

Ouhab, A., Abreu, T., Slimani, H., and Mellouk, A. (2020). Energy-efficient clustering and routing algorithm for large-scale SDN-based IoT monitoring. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 1–6. IEEE. doi: 10.1109/ICC40277.2020.9148659.

Rahman, A., Islam, M.J., Band, S.S., Muhammad, G., Hasan, K., and Tiwari, P. (2023). Towards a blockchain-SDN-based secure architecture for cloud computing in smart industrial IoT. *Digital Communications and Networks*, 9(2), 411–421. doi:10.1016/j.dcan.2022.11.003.

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. doi:10.1007/s12083-023-01448-2.

Srinivasan, G., Kumar, R.S., and Muthukaruppasamy, S. (2022). Evaluation of additional power loss reduction in dg integrated optimal distribution network. *Journal of Control Engineering and Applied Informatics*, 24(1), 68–76.

Szász, C. (2018). Reconfigurable hardware technology: an emerging paradigm for combined software-hardware fault-tolerance implementation. *Journal of Control Engineering and Applied Informatics*, 20(3), 99–108.

Udayaprasad, P.K., Shreyas, J., Srinidhi, N.N., Kumar, S.M.D., Dayananda, P., Askar, S.S., and Abouhawwash, M. (2024). Energy Efficient Optimized Routing Technique With Distributed SDN-AI to Large Scale I-IoT Networks. *IEEE Access*, 12, 2742–2759. doi: 10.1109/ACCESS.2023.3346679.

Xingjian, F. and Qianjun, Z. (2022). Fault-tolerant tracking control for nonlinear ncss with time-delay and data loss failure in dual-channel based on iterative learning. *Journal of Control Engineering and Applied Informatics*, 24(1), 26–36.

Zobary, F. and Li, C. (2023). A mathematical model for sdn control plane scalability evaluation based on controller utilization. *Journal of Control Engineering and Applied Informatics*, 25(4), 14–21. doi: 10.61416/ceai.v25i4.8783.

Zobary, F. and Li, C. (2024). Two weighting methods for software-defined network controllers ranking as a multicriteria decision-making problem. *Journal of Control Engineering and Applied Informatics*, 26. doi: 10.61416/ceai.v26i2.8924.