A REVIEW OF COMPARISON OF SDN (SOFTWARE-DEFINED NETWORKING) Vs. TRADITIONAL NETWORKING FOR TRAFFIC MANAGEMENT

Shashank Yadav[1], Mrs. Sahreen Hizab[2]

[1]Master of Technology, Computer Science and Engineering, *Sagar Institute of Technology and Management, Barabanki, India*
[2]*Assistant Professor, Department of Computer Science and Engineering, Sagar Institute of Technology and Management, Barabanki, India*

***

Abstract - *The rapid growth of cloud computing, Internet of Things (IoT), multimedia services, and data-intensive applications has significantly increased the complexity of network traffic management. Traditional networking architectures rely on tightly coupled control and data planes, distributed routing protocols, and static configuration mechanisms, which often limit flexibility, scalability, and real-time adaptability. In contrast, Software-Defined Networking (SDN) introduces architectural decoupling, centralized control logic, and network programmability, enabling dynamic and fine-grained traffic engineering. This review paper provides a systematic comparison between SDN and traditional networking paradigms with specific emphasis on traffic management mechanisms, including congestion control, load balancing, Quality of Service (QoS) enforcement, scalability, and security considerations. Existing literature is critically analyzed to evaluate performance improvements, operational complexity, and deployment challenges associated with both approaches. The study synthesizes findings from experimental, simulation-based, and hybrid implementations to highlight architectural trade-offs and practical limitations. Furthermore, emerging trends such as AI-driven traffic optimization and hybrid SDN deployments are discussed. The review aims to offer a structured foundation for researchers and network designers in selecting appropriate traffic management strategies for modern and future network infrastructures.*

Key Words: Software-Defined Networking (SDN), Traditional Networking, Traffic Management, Traffic Engineering, Quality of Service (QoS), Network Scalability.

1. INTRODUCTION

1.1 Background: Evolution of Networking Paradigms

The evolution of computer networking has been driven by the growing demand for scalability, reliability, and efficient data delivery. Early network architectures were primarily hardware-centric, where routing decisions and packet forwarding were tightly integrated within proprietary devices. Traditional networking relies on distributed control mechanisms, in which each router independently executes routing protocols such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP) to determine optimal paths (Medhi and Ramasamy, 2017). While this distributed paradigm ensures robustness, it introduces operational complexity and limited flexibility in dynamic traffic environments.

1.1.1 Emergence of Software-Defined Networking

Software-Defined Networking (SDN) emerged as a paradigm shift to address the limitations of conventional architectures. By decoupling the control plane from the data plane, SDN centralizes network intelligence within a programmable controller, enabling dynamic traffic engineering and simplified network management (Kreutz et al., 2015). The introduction of Open Flow as a southbound interface allowed standardized communication between controllers and forwarding devices (McKeown et al., 2008). This architectural abstraction facilitates network programmability, rapid policy deployment, and global traffic visibility, marking a significant departure from static, hardware-driven designs.

Figure-1: Software-Defined Networking

1.2 Importance of Traffic Management in Modern Networks

Traffic management plays a pivotal role in ensuring optimal network performance, particularly in environments characterized by heterogeneous applications such as cloud services, real-time streaming, and Internet of Things (IoT) systems. Effective traffic management mechanisms regulate bandwidth allocation, congestion avoidance, and routing efficiency to maintain Quality of Service (QoS) guarantees (Akyildiz et al., 2014). In traditional networks, QoS enforcement is typically achieved through static configuration of queuing disciplines, traffic shaping, and protocol-based path selection.

1.2.1 Quality of Service, Performance, and Scalability Considerations

Modern networks demand low latency, minimal packet loss, and high throughput to support mission-critical applications. However, distributed routing decisions may lead to suboptimal path utilization and slower adaptation to congestion (Feamster, Rexford and Zegura, 2014). SDN addresses these challenges by enabling centralized traffic optimization and fine-grained flow control. The controller's global network view allows proactive congestion mitigation and dynamic reconfiguration, thereby improving scalability and resource utilization (Kreutz et al., 2015). Consequently, traffic management has become a strategic function rather than a reactive mechanism.

1.3 Motivation for Comparing SDN with Traditional Networking

The coexistence of legacy infrastructure and emerging SDN deployments has created a need for systematic comparison. Traditional networking remains widely adopted due to its maturity, protocol stability, and proven reliability. Conversely, SDN promises enhanced programmability, automation, and improved traffic engineering capabilities (McKeown et al., 2008). Despite these advantages, SDN introduces new concerns related to controller scalability, security vulnerabilities, and deployment complexity (Kreutz et al., 2015).

A comparative analysis is therefore necessary to evaluate architectural trade-offs, operational efficiency, and real-world applicability. Understanding performance differences under various traffic scenarios—such as congestion-heavy environments or latency-sensitive applications—can guide network designers in selecting appropriate paradigms.

2. METHODOLOGY

2.1 Research Design and Review Framework

This review adopts a systematic literature review (SLR) methodology to ensure transparency, reproducibility, and comprehensive coverage of existing research comparing Software-Defined Networking (SDN) and traditional networking for traffic management. A systematic approach is widely recommended for synthesizing evidence in technology-oriented domains because it minimizes selection bias and enhances analytical rigor (Kitchenham and Charters, 2007). The review process was structured into identification, screening, eligibility assessment, and synthesis phases, consistent with established review protocols (Snyder, 2019).

2.1.1 Criteria for Selecting Literature Sources

Peer-reviewed journal articles, conference proceedings, and high-impact survey papers were collected from reputable digital libraries, including IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and Scopus-indexed journals. These databases were selected due to their extensive coverage of networking, communication systems, and software-defined architectures (Kreutz et al., 2015).

The literature search focused primarily on publications from 2008 to 2024. The year 2008 was selected as the starting point because it marks the introduction of OpenFlow, which significantly influenced SDN research (McKeown et al., 2008). Search queries included combinations of keywords such as: "Software-Defined Networking", "Traditional Networking", "Traffic Management", "Traffic Engineering", "QoS in SDN", "Congestion Control", and "Network Scalability". Boolean operators (AND/OR) were used to refine search outcomes and ensure relevance.

2.2 Inclusion and Exclusion Criteria

To maintain focus and analytical depth, explicit inclusion and exclusion filters were applied during the screening stage.

2.2.1 Inclusion Criteria

Studies were included if they:

- Directly addressed traffic management, traffic engineering, congestion control, or QoS mechanisms.
- Presented comparative analysis between SDN and traditional networking, or provided measurable performance evaluation of either paradigm.
- Included experimental, simulation-based, analytical, or real-world deployment results.

2.2.2 Exclusion Criteria

Studies were excluded if they:

- Focused solely on unrelated SDN aspects such as virtualization without traffic management relevance.
- Addressed purely hardware-level optimizations without architectural comparison.
- Lacked empirical or analytical validation.

Applying these filters aligns with best practices in systematic reviews, ensuring that only methodologically sound and contextually relevant studies contribute to the synthesis (Kitchenham and Charters, 2007).

2.3 Summary of Review Process

The initial database search yielded approximately 420 publications. After removing duplicates and conducting title and abstract screening, 210 papers remained. Full-text eligibility assessment further refined the selection to 95 high-relevance studies. Finally, 72 papers were included in the qualitative and comparative synthesis based on strict relevance to traffic management mechanisms.

2.3.1 Classification Strategy

Selected papers were classified into four thematic categories:

- Comparative Architectural Studies – Direct comparisons of SDN and traditional networking performance.
- SDN-Based Traffic Optimization Approaches – Controller-based congestion control, load balancing, and QoS enforcement mechanisms.
- Traditional Traffic Engineering Enhancements – MPLS-based optimization, protocol refinements, and distributed routing improvements.
- Hybrid and Transitional Models – Partial SDN deployment or hybrid architectures integrating legacy systems.

This thematic classification facilitated structured comparison and critical synthesis. Quantitative metrics such as latency, throughput, packet loss, scalability, and control overhead were extracted where available to support analytical evaluation. The classification approach is consistent with systematic mapping studies commonly adopted in networking research (Snyder, 2019).

3. FUNDAMENTALS AND ARCHITECTURAL OVERVIEW

3.1 Traditional Networking

Traditional networking architectures are built on vertically integrated devices in which the control plane and data plane coexist within the same physical hardware. In this model, routers and switches independently make forwarding decisions based on locally computed routing tables and distributed protocol exchanges. This tightly coupled design has been the foundation of Internet architecture for decades due to its robustness and decentralized fault tolerance (Medhi and Ramasamy, 2017). However, the distributed nature of decision-making often limits global network visibility and centralized optimization.

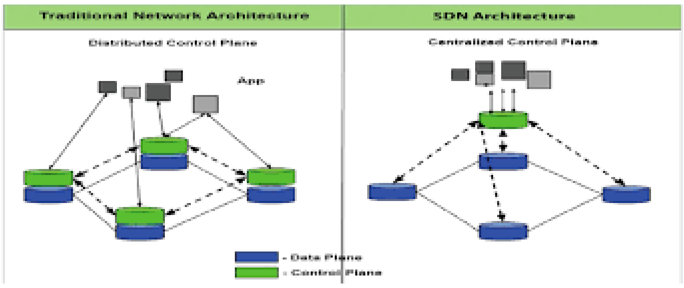

Figure-2: Traditional Networking

3.1.1 Control and Data Plane Coupling

In conventional networks, the control plane—responsible for routing decisions—and the data plane—responsible for packet forwarding—are embedded within the same networking device. Each router executes routing algorithms to compute shortest paths and updates its forwarding table accordingly. While this approach ensures autonomy and resilience, it complicates network-wide traffic optimization because no single entity possesses a complete global topology view (Feamster, Rexford and Zegura, 2014). Consequently, traffic management relies on protocol convergence and distributed coordination, which may introduce latency in dynamic conditions.

3.1.2 Typical Protocols: OSPF, BGP and MPLS

Traditional networking employs standardized routing and traffic engineering protocols. OSPF is a link-state interior gateway protocol used for intra-domain routing, enabling routers to compute shortest paths using Dijkstra's algorithm. BGP functions as an inter-domain routing protocol that manages path selection between autonomous systems. Multiprotocol Label Switching (MPLS) enhances traffic engineering by introducing label-switched paths, allowing more deterministic forwarding and resource reservation (Medhi and Ramasamy, 2017). Although MPLS improves traffic control granularity, it still operates within a distributed control framework.

3.1.3 Traffic Management Mechanisms

Traffic management in traditional networks relies on static and dynamic routing adjustments, congestion control algorithms, and Quality of Service (QoS) configurations. Mechanisms such as traffic shaping, packet scheduling (e.g., weighted fair queuing), and access control lists are configured device-by-device. Congestion control is primarily managed at transport layer protocols such as TCP, while network-layer routing adapts based on link-state updates (Akyildiz et al., 2014). Despite their maturity, these mechanisms may lack rapid adaptability to sudden traffic fluctuations due to protocol convergence delays and limited centralized oversight.

3.2 Software-Defined Networking (SDN)

Software-Defined Networking represents a paradigm shift by separating network intelligence from forwarding hardware. Unlike traditional networking, SDN decouples the control plane from the data plane, centralizing decision-making within a software-based controller. This architectural abstraction enables programmable, flexible, and application-aware traffic management (Kreutz et al., 2015). SDN introduces logically centralized control while maintaining distributed data forwarding devices.

3.2.1 Decoupled Control and Data Planes

The defining feature of SDN is the separation of the control logic from forwarding elements. The control plane resides in a controller that maintains a global view of network topology and traffic conditions. Data plane devices act as simple forwarding elements that execute flow rules installed by the controller. This decoupling facilitates rapid policy deployment, centralized optimization, and dynamic traffic engineering (McKeown et al., 2008). By abstracting hardware functions, SDN enables network programmability through software applications.

3.2.2 Centralized vs Distributed Controllers

Although SDN is often described as centralized, practical deployments may adopt logically centralized but physically distributed controller architectures to enhance scalability and reliability. A single centralized controller simplifies management but introduces potential bottlenecks and single points of failure. Distributed controller frameworks, on the other hand, partition network domains while maintaining synchronization among controllers (Kreutz et al., 2015). The choice between centralized and distributed control impacts latency, fault tolerance, and traffic optimization efficiency.

3.2.3 Key Protocols: OpenFlow

OpenFlow is one of the earliest and most widely adopted southbound interfaces in SDN. It enables communication between the controller and forwarding devices by defining standardized flow table entries and matching rules (McKeown et al., 2008). Through OpenFlow, controllers can install, modify, or remove forwarding rules dynamically based on traffic conditions. Beyond OpenFlow, other programmable interfaces and intent-based frameworks have emerged to enhance flexibility and interoperability.

3.2.4 SDN Traffic Management Primitives

SDN introduces programmable traffic management primitives such as flow-based routing, dynamic path computation, centralized congestion monitoring, and application-aware QoS enforcement. Unlike traditional per-device configuration, policies can be applied network-wide via controller applications. The global visibility provided by the controller enables proactive congestion avoidance, load balancing, and real-time traffic reconfiguration (Akyildiz et al., 2014). These capabilities significantly improve adaptability and resource utilization, particularly in data center and cloud environments.

4. TRAFFIC MANAGEMENT IN TRADITIONAL VS SDN NETWORKS

4.1 Traffic Engineering Mechanisms

Traffic engineering (TE) aims to optimize network resource utilization while satisfying performance constraints such as latency, throughput, and reliability. The implementation of TE differs fundamentally between traditional distributed architectures and SDN-based programmable networks.

4.1.1 Traditional: Static and Dynamic Routing Strategies

In traditional networking, traffic engineering is primarily achieved through distributed routing protocols and manual configuration. Static routing provides predictable path selection but lacks adaptability. Dynamic routing protocols such as OSPF and IS-IS compute shortest paths based on link-state information, while BGP governs inter-domain routing decisions (Medhi and Ramasamy, 2017). MPLS enhances TE by enabling label-switched paths that allow explicit routing and bandwidth reservation. However, these mechanisms rely on local decision-making and protocol convergence, which may limit responsiveness to rapid traffic changes (Feamster, Rexford and Zegura, 2014).

4.1.2 SDN: Controller-Based Traffic Optimization

SDN introduces centralized traffic engineering through a logically centralized controller with global topology awareness. The controller dynamically computes optimal paths and installs flow rules in forwarding devices using protocols such as OpenFlow (McKeown et al., 2008). This centralized model enables fine-grained flow-level management, proactive congestion avoidance, and real-time path reconfiguration. Research demonstrates that SDN-based TE improves bandwidth utilization and reduces latency by leveraging global network state information (Akyildiz et al., 2014).

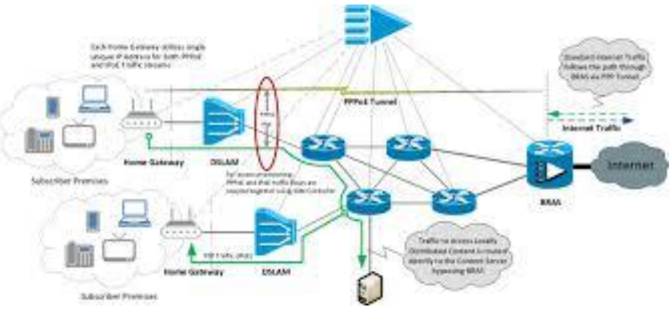

Figure-3: SDN: Controller-Based Traffic Optimization

4.2 Congestion Control and Load Balancing

Efficient congestion control and load balancing are critical for maintaining network stability and preventing packet loss under high traffic demand.

4.2.1 Algorithms in Traditional Networking

In conventional networks, congestion control is largely handled at the transport layer, particularly through TCP congestion control algorithms such as Reno and Cubic. Network-layer load balancing is achieved through mechanisms like Equal-Cost Multi-Path (ECMP) routing, which distributes traffic across multiple shortest paths (Medhi and Ramasamy, 2017). While these approaches provide distributed resilience, they lack coordinated global optimization and may result in uneven load distribution under dynamic traffic conditions.

4.2.2 SDN Approaches: Global Network View and Programmability

SDN enhances congestion control by utilizing the controller's global visibility to monitor link utilization and adjust flow rules dynamically. Load balancing can be implemented through centralized algorithms that redistribute traffic based on real-time metrics. Programmability allows integration of machine learning models for traffic prediction and adaptive routing (Kreutz et al., 2015). Consequently, SDN-based congestion management is more proactive compared to the reactive mechanisms of traditional TCP-based control.

4.3 Quality of Service (QoS) Enforcement

QoS mechanisms ensure that critical applications receive priority handling and guaranteed performance levels.

4.3.1 Queuing and Prioritization Techniques in Traditional Networks

Traditional QoS enforcement relies on device-level configuration, including techniques such as priority queuing, weighted fair queuing (WFQ), traffic shaping, and policing. These mechanisms are configured individually on routers and switches, often requiring significant administrative effort (Akyildiz et al., 2014). While effective, such configurations may lack network-wide coordination and rapid adaptability.

4.3.2 SDN Policy Frameworks and Dynamic QoS

In SDN environments, QoS policies are centrally defined and enforced across the network through programmable interfaces. The controller can dynamically modify flow entries to prioritize traffic based on application requirements. Policy abstraction and intent-based networking further enhance automated QoS provisioning (Kreutz et al., 2015). This centralized enforcement enables consistent QoS across heterogeneous network segments and reduces manual configuration overhead.

4.4 Scalability and Performance

Scalability remains a fundamental concern for both traditional and SDN architectures, particularly in large-scale enterprise and cloud networks.

4.4.1 Limits in Traditional Distributed Control

Distributed control mechanisms inherently scale with network size, but they introduce complexity in routing convergence and configuration management. As network topologies grow, maintaining consistent policy enforcement and optimal routing becomes increasingly challenging (Feamster, Rexford and Zegura, 2014). Protocol convergence delays may also affect performance during topology changes.

4.4.2 SDN Approaches: Hierarchical Controllers and Flow Management

SDN addresses scalability challenges through hierarchical or distributed controller architectures that divide network control into logical domains. Flow aggregation techniques and proactive rule installation reduce control-plane overhead (Kreutz et al., 2015). Although centralized control improves optimization efficiency, controller placement and synchronization remain active research areas to ensure low latency and fault tolerance.

4.5 Security Considerations for Traffic Management

Security is a critical aspect of traffic management, as malicious traffic patterns can disrupt network performance and compromise reliability.

4.5.1 Attack Surfaces and Mitigation in Traditional Networks

Traditional networks face threats such as Distributed Denial of Service (DDoS) attacks, route hijacking, and misconfiguration vulnerabilities. Security mechanisms typically include access control lists (ACLs), intrusion detection systems, and protocol-level authentication (Medhi and Ramasamy, 2017). However, decentralized management can complicate coordinated defense strategies.

4.5.2 SDN Security Risks and Defense Approaches

While SDN enhances visibility and centralized monitoring, it introduces new attack vectors, particularly targeting the controller. Controller compromise, southbound interface exploitation, and flow rule manipulation represent significant risks (Kreutz et al., 2015). To mitigate these threats, distributed controller replication, secure communication channels (e.g., TLS), and anomaly detection frameworks are implemented. Despite these safeguards, ensuring controller resilience remains essential to maintain reliable traffic management.

5. LITERATURE REVIEW

5.1 Survey and Comparative Studies

The comparative analysis of Software-Defined Networking (SDN) and traditional networking has attracted significant scholarly attention, particularly in the context of traffic management and traffic engineering. Foundational survey studies provide a comprehensive overview of architectural differences, programmability advantages, and implementation challenges associated with SDN (Kreutz et al., 2015). These surveys often position SDN as a paradigm shift from distributed control toward centralized intelligence, emphasizing improved global visibility and flexible policy enforcement.

5.1.1 Key Studies Comparing SDN and Traditional Architectures

Feamster, Rexford and Zegura (2014) critically examined the intellectual evolution of programmable networks, highlighting how SDN addresses limitations in traditional distributed routing. Akyildiz et al. (2014) specifically analyzed traffic engineering mechanisms, demonstrating that SDN-based centralized optimization can outperform traditional OSPF/MPLS-based routing under dynamic traffic conditions. Empirical comparisons frequently rely on simulation platforms such as Mininet or real-world testbeds to measure throughput, latency, and packet loss across architectures. Results generally indicate that SDN achieves better adaptability and load balancing efficiency, though concerns regarding controller scalability and single points of failure persist (Kreutz et al., 2015).

5.1.2 Summary of Methodologies and Findings

Methodologically, existing studies employ a combination of analytical modeling, experimental testbeds, and simulation-based performance evaluation. Metrics such as link utilization, flow completion time, jitter, and control overhead are commonly assessed. Comparative findings suggest that while traditional networks demonstrate stability and maturity, SDN provides superior traffic reconfiguration speed and centralized optimization capabilities (Akyildiz et al., 2014). However, performance gains are often context-dependent, particularly in large-scale or latency-sensitive environments.

5.2 SDN-Focused Traffic Management Solutions

Research dedicated to SDN-based traffic management emphasizes programmability and centralized intelligence as primary enablers of optimization.

5.2.1 Controller-Based Strategies and AI/ML Integration

Controller-based traffic engineering strategies leverage the global network view to compute optimal routing paths dynamically. Centralized optimization algorithms allocate bandwidth and mitigate congestion proactively. Recent studies incorporate Artificial Intelligence (AI) and Machine Learning (ML) techniques within SDN controllers to predict traffic patterns and automate decision-making processes (Kreutz et al., 2015). Such approaches improve adaptability in cloud and data center networks by enabling intelligent flow scheduling and anomaly detection.

5.2.2 SDN Traffic Prediction and Adaptive Path Reconfiguration

Adaptive path reconfiguration is a defining feature of SDN-enabled traffic management. Using real-time monitoring data, controllers dynamically modify forwarding rules to avoid congestion hotspots. Research demonstrates that predictive traffic modeling enhances routing efficiency and reduces packet loss compared to static or reactive mechanisms (Akyildiz et al., 2014). Nevertheless, controller processing overhead and synchronization delays remain key performance considerations in large-scale deployments.

5.3 Traditional Networking Traffic Engineering Enhancements

Although SDN has gained prominence, substantial advancements have been made within traditional networking to improve traffic management efficiency.

5.3.1 Advanced MPLS-TE Solutions

Multiprotocol Label Switching with Traffic Engineering (MPLS-TE) represents a significant enhancement in traditional architectures. MPLS-TE enables explicit path selection and bandwidth reservation, allowing operators to optimize resource utilization across backbone networks (Medhi and Ramasamy, 2017). Constraint-based routing and fast reroute mechanisms further enhance reliability and performance. Despite these improvements, configuration complexity and limited global optimization capabilities remain challenges.

5.3.2 QoS Enhancements with Classic Protocols

Quality of Service in traditional networks has evolved through advanced queuing disciplines, traffic shaping mechanisms, and differentiated services (DiffServ). These mechanisms provide prioritization and bandwidth guarantees for latency-sensitive applications (Akyildiz et al., 2014). However, QoS enforcement often requires manual per-device configuration, which may reduce operational agility compared to SDN-based centralized policy management.

5.4 Hybrid and Transitional Approaches

Recognizing the practical challenges of full SDN adoption, researchers have explored hybrid networking models that integrate SDN with legacy infrastructure.

5.4.1 Partial SDN Deployments

Hybrid approaches typically deploy SDN controllers to manage specific network segments—such as data center cores—while retaining traditional routing protocols at the edge. This incremental strategy reduces deployment risk and leverages existing infrastructure investments (Feamster, Rexford and Zegura, 2014). Such architectures allow gradual transition without disrupting established services.

5.4.2 Performance Trade-Off Analysis

Comparative evaluations of hybrid models reveal trade-offs between flexibility and complexity. While hybrid architectures improve traffic optimization within SDN-managed segments, interoperability challenges and control-plane coordination overhead may limit overall efficiency (Kreutz et al., 2015). Empirical findings suggest that hybrid models offer a pragmatic balance between innovation and stability, particularly in enterprise and service-provider environments.

6. DISCUSSION

6.1 Synthesis of Findings: Strengths and Weaknesses

The comparative analysis of traditional networking and Software-Defined Networking (SDN) reveals fundamental architectural trade-offs that directly influence traffic management performance. Traditional networking demonstrates robustness, protocol maturity, and decentralized fault tolerance due to its distributed control architecture (Medhi and Ramasamy, 2017). Its long-standing deployment history ensures operational stability and interoperability across heterogeneous infrastructures. However, limited global visibility and dependence on protocol convergence restrict rapid adaptability to dynamic traffic conditions (Feamster, Rexford and Zegura, 2014).

In contrast, SDN offers centralized intelligence, fine-grained flow control, and programmability, enabling dynamic traffic engineering and rapid policy enforcement (Kreutz et al., 2015). The controller's global network view allows optimized routing decisions and proactive congestion mitigation. Nevertheless, SDN introduces challenges such as controller scalability, potential single points of failure, and increased control-plane communication overhead (McKeown et al., 2008). Thus, while SDN excels in flexibility and optimization, traditional networking remains advantageous in resilience and operational familiarity.

6.2 Practical Implications for Network Operators

From an operational perspective, the choice between SDN and traditional networking depends on deployment context, performance requirements, and administrative capabilities. Network operators managing large-scale data centers or cloud infrastructures benefit from SDN's centralized orchestration and automation features, which reduce manual configuration complexity (Kreutz et al., 2015). Policy-driven traffic management and rapid reconfiguration enhance service agility and reduce downtime.

Conversely, service providers operating legacy backbone networks may prefer traditional architectures due to established protocol reliability and incremental upgrade feasibility (Medhi and Ramasamy, 2017). Transitioning to SDN requires investment in controller infrastructure, staff training, and security reinforcement. Therefore, hybrid deployment models often represent a pragmatic compromise, enabling gradual migration while preserving operational continuity (Feamster, Rexford and Zegura, 2014).

6.3 Traffic Adaptation, Flexibility and Management Overhead

6.3.1 Differences in Traffic Adaptation and Flexibility

Traditional networks adapt to traffic changes through distributed routing updates and transport-layer congestion control. While effective, these mechanisms are reactive and dependent on protocol convergence intervals. As network size increases, adaptation speed may degrade (Medhi and Ramasamy, 2017). SDN, by contrast, facilitates near real-time traffic adaptation via centralized monitoring and programmable rule updates. The controller can reconfigure forwarding paths proactively based on global utilization metrics (Akyildiz et al., 2014). This programmability significantly enhances flexibility in handling bursty or unpredictable traffic patterns.

6.3.2 Management Overhead Considerations

Management overhead differs substantially between the two paradigms. Traditional networking requires device-level configuration, resulting in higher administrative effort and potential configuration inconsistencies. However, control-plane operations are distributed, reducing the risk of centralized bottlenecks. In SDN environments, centralized policy enforcement reduces manual configuration but increases reliance on controller performance and secure communication channels (Kreutz et al., 2015). Excessive flow rule updates or large-scale networks may impose processing burdens on controllers, affecting latency and scalability.

6.4 Challenges in Fair Benchmarking

Accurately comparing SDN and traditional networking performance presents methodological challenges. Many empirical studies rely on simulation tools such as Mininet or controlled laboratory testbeds, which may not fully capture real-world traffic heterogeneity and hardware constraints (Akyildiz et al., 2014). Simulation-based evaluations often idealize network conditions, potentially overstating SDN performance gains. Conversely, real-world deployments introduce unpredictable traffic patterns, legacy integration issues, and hardware limitations that influence outcomes.

Additionally, benchmarking metrics vary across studies, including throughput, latency, jitter, control overhead, and scalability indicators. The absence of standardized evaluation frameworks complicates cross-study comparison (Kreutz et al., 2015). Therefore, future research should emphasize reproducible experimentation, standardized datasets, and hybrid evaluation environments that combine simulation accuracy with realistic deployment conditions.

7. CONCLUSION

This review critically compared Software-Defined Networking (SDN) and traditional networking architectures with specific emphasis on traffic management mechanisms, including traffic engineering, congestion control, Quality of Service (QoS), scalability, and security. Traditional networking, built on distributed control and protocol-driven routing, offers robustness, maturity, and decentralized fault tolerance. However, its limited global visibility and slower adaptation to dynamic traffic patterns restrict optimization efficiency. In contrast, SDN introduces architectural decoupling and centralized programmability, enabling fine-grained flow management, proactive congestion mitigation, and rapid policy enforcement. Empirical findings across the literature indicate that SDN generally achieves superior flexibility and resource utilization, particularly in data center and cloud environments. Nevertheless, controller scalability, security vulnerabilities, and deployment complexity remain significant concerns. Hybrid architectures have emerged as a practical transition strategy, balancing innovation with operational stability. Overall, the choice between SDN and traditional networking depends on performance requirements, network scale, and administrative capabilities, with SDN demonstrating clear advantages in environments demanding agility and dynamic traffic optimization.

8. LIMITATIONS

This review is limited by its reliance on previously published studies, many of which employ simulation-based or small-scale experimental testbeds that may not fully represent real-world deployment conditions. Variations in benchmarking methodologies, performance metrics, and evaluation environments complicate direct comparison across studies. Additionally, rapid advancements in SDN technologies and controller platforms may render some findings time-sensitive. The review primarily focuses on traffic management aspects and does not deeply explore economic cost analysis, vendor-specific implementations, or emerging paradigms such as programmable data planes (e.g., P4). These factors may influence practical adoption decisions beyond architectural performance considerations.

REFERENCES

1. Akyildiz, I.F., Lee, A., Wang, P., Luo, M. and Chou, W. (2014) 'A roadmap for traffic engineering in software-defined networks', Computer Networks, 71, pp. 1–30.
2. Feamster, N., Rexford, J. and Zegura, E. (2014) 'The road to SDN: An intellectual history of programmable networks', ACM SIGCOMM Computer Communication Review, 44(2), pp. 87–98.
3. Kitchenham, B. and Charters, S. (2007) Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report. Keele University and Durham University.
4. Kreutz, D., Ramos, F.M.V., Verissimo, P., Rothenberg, C.E., Azodolmolky, S. and Uhlig, S. (2015) 'Software-defined networking: A comprehensive survey', Proceedings of the IEEE, 103(1), pp. 14–76.
5. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J. (2008) 'OpenFlow: Enabling innovation in campus networks', ACM SIGCOMM Computer Communication Review, 38(2), pp. 69–74.
6. Medhi, D. and Ramasamy, K. (2017) Network Routing: Algorithms, Protocols, and Architectures. 2nd edn. Burlington: Morgan Kaufmann.
7. Snyder, H. (2019) 'Literature review as a research methodology: An overview and guidelines', Journal of Business Research, 104, pp. 333–339.
8. Al-Fares, M., Loukissas, A. and Vahdat, A. (2008) 'A scalable, commodity data center network architecture', ACM SIGCOMM Computer Communication Review, 38(4), pp. 63–74.
9. Benson, T., Akella, A. and Maltz, D.A. (2010) 'Network traffic characteristics of data centers in the wild', Proceedings of the ACM SIGCOMM Conference, pp. 267–280.
10. Casado, M., Freedman, M.J., Pettit, J., Luo, J., McKeown, N. and Shenker, S. (2007) 'Ethane: Taking control of the enterprise', ACM SIGCOMM Computer Communication Review, 37(4), pp. 1–12.
11. Curtis, A.R., Mogul, J.C., Tourrilhes, J., Yalagandula, P., Sharma, P. and Banerjee, S. (2011) 'DevoFlow: Scaling flow management for high-performance networks', ACM SIGCOMM Computer Communication Review, 41(4), pp. 254–265.

12. Dixit, A., Hao, F., Mukherjee, S., Lakshman, T.V. and Kompella, R. (2013) 'Towards an elastic distributed SDN controller', ACM SIGCOMM Computer Communication Review, 43(4), pp. 7–12.

13. Fang, W., Peterson, L., Rexford, J., Freedman, M.J. and Walker, D. (2013) 'Overcoming the limitations of SDN with hybrid networking', Proceedings of HotNets, pp. 1–7.

14. Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N. and Shenker, S. (2008) 'NOX: Towards an operating system for networks', ACM SIGCOMM Computer Communication Review, 38(3), pp. 105–110.

15. Heller, B., Sherwood, R. and McKeown, N. (2012) 'The controller placement problem', Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Software Defined Networks, pp. 7–12.

16. Jain, R., Mendiratta, V., Ong, J., Poretsky, S., Tofighbakhsh, R., Zhuang, L. and Hadi Salim, J. (2013) 'B4: Experience with a globally deployed software defined WAN', ACM SIGCOMM Computer Communication Review, 43(4), pp. 3–14.

17. Jarschel, M., Oechsner, S., Schlosser, D., Pries, R., Goll, S. and Tran-Gia, P. (2011) 'Modeling and performance evaluation of an OpenFlow architecture', Proceedings of ITC, pp. 1–7.

18. Kandula, S., Sengupta, S., Greenberg, A., Patel, P. and Chaiken, R. (2009) 'The nature of data center traffic', Proceedings of the ACM SIGCOMM Conference, pp. 202–208.

19. Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T. and Shenker, S. (2010) 'Onix: A distributed control platform for large-scale production networks', Proceedings of OSDI, pp. 1–14.

20. Lantz, B., Heller, B. and McKeown, N. (2010) 'A network in a laptop: Rapid prototyping for software-defined networks', Proceedings of HotNets, pp. 1–6.

21. Liu, H., Xu, Y., Wang, G., Liu, D., Jin, H., Liu, C. and Liao, X. (2015) 'Flow scheduling for cloud data centers: A survey', IEEE Communications Surveys & Tutorials, 18(1), pp. 1–22.

22. Mao, M. and Humphrey, M. (2012) 'A performance study on the VM startup time in the cloud', Proceedings of IEEE CLOUD, pp. 423–430.

23. Nadeau, T.D. and Gray, K. (2013) SDN: Software Defined Networks. Sebastopol: O'Reilly Media.

24. Phemius, K., Bouet, M. and Leguay, J. (2014) 'DISCO: Distributed multi-domain SDN controllers', Proceedings of IEEE NOMS, pp. 1–4.

25. Qazi, Z.A., Tu, C.-C., Chiang, L., Miao, R., Sekar, V. and Yu, M. (2013) 'SIMPLE-fying middlebox policy enforcement using SDN', ACM SIGCOMM Computer Communication Review, 43(4), pp. 27–38.

26. Rexford, J. and Dovrolis, C. (2010) 'Future Internet architecture: Clean-slate versus evolutionary research', Communications of the ACM, 53(9), pp. 36–40.

27. Sezer, S., Scott-Hayward, S., Chouhan, P.K., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M. and Rao, N. (2013) 'Are we ready for SDN? Implementation challenges for software-defined networks', IEEE Communications Magazine, 51(7), pp. 36–43.

28. Shahzad, M., Liu, A.X., Sun, Q. and Liu, T. (2016) 'A survey on network traffic monitoring and analysis', IEEE Communications Surveys & Tutorials, 18(4), pp. 1–26.

29. Shin, S. and Gu, G. (2012) 'Attacking software-defined networks: A first feasibility study', Proceedings of HotSDN, pp. 165–166.

30. Tootoonchian, A. and Ganjali, Y. (2010) 'HyperFlow: A distributed control plane for OpenFlow', Proceedings of INM/WREN, pp. 1–6.

31. Van Adrichem, N.L.M., Doerr, C. and Kuipers, F.A. (2014) 'OpenNetMon: Network monitoring in OpenFlow software-defined networks', Proceedings of IEEE NOMS, pp. 1–8.

32. Wang, R., Butnariu, D. and Rexford, J. (2011) 'OpenFlow-based server load balancing gone wild', Proceedings of Hot-ICE, pp. 1–6.

33. Yu, M., Rexford, J., Freedman, M.J. and Wang, J. (2010) 'Scalable flow-based networking with DIFANE', ACM SIGCOMM Computer Communication Review, 40(4), pp. 351–362.

34. Zhang, Y., Beheshti, N., Beliveau, L., Lefebvre, G., Manghirmalani, R., Mishra, R., Patney, R., Subrahmaniam, A. and Truchan, P. (2013) 'Steering: A software-defined networking for inline service chaining', Proceedings of ICNP, pp. 1–10.