A Survey of Advanced Concepts and Technologies in Software-Defined Networking and Network Function Virtualization

www.surveyx.cn

Abstract

This survey paper delves into the advanced concepts and technologies within Software-Defined Networking (SDN) and Network Function Virtualization (NFV), focusing on their integration into cloud-native environments. Key areas of exploration include the separation of control and data planes, exemplified by OpenFlow, and the flexibility offered by the P4 programming language, which enhance network programmability and scalability. The survey highlights the transformative impact of these technologies on modern networking paradigms, particularly in optimizing resource management and addressing the demands of next-generation networks. Despite these advancements, challenges persist in security, scalability, and interoperability, with issues such as topology poisoning and centralized vulnerabilities requiring further research. The integration of machine learning and sustainability into network slicing, as well as AI-driven solutions for optimization, are identified as promising avenues for future exploration. The paper underscores the significance of frameworks like ContentFlow and Fast Failover in improving content delivery and recovery times, respectively. By examining these technologies and their implications, the survey provides insights into the current landscape and future directions of SDN and NFV, emphasizing their role in shaping flexible, efficient, and secure network infrastructures capable of meeting contemporary and future application demands.

1 Introduction

1.1 Purpose and Scope of the Survey

This survey examines advanced concepts and technologies in Software-Defined Networking (SDN) and Network Function Virtualization (NFV), particularly their integration into cloud-native environments. The primary aim is to analyze how SDN and NFV are transforming network architectures within cloud computing, intentionally excluding non-cloud-based networking technologies and legacy systems to maintain relevance to contemporary trends [1]. The survey also explores the adaptation of SDN, NFV, and network slicing in integrated terrestrial and non-terrestrial networks, highlighting their critical roles and challenges in emerging 6G networks [2].

The scope includes a historical overview of SDN's evolution, its functional architecture, and the significance of OpenFlow standards, thereby underscoring SDN's impact on modern networking paradigms [3]. Additionally, it addresses the integration of cloud virtualization with communication networks, illustrating the interplay between advanced SDN and NFV concepts and cloud environments [4].

The survey identifies limitations in existing SDN architectures concerning resilience, scalability, and extensibility, particularly in multi-domain networks [5]. It also investigates emerging challenges in SDN control planes, focusing on scalability, security vulnerabilities, and the necessity for multitenancy

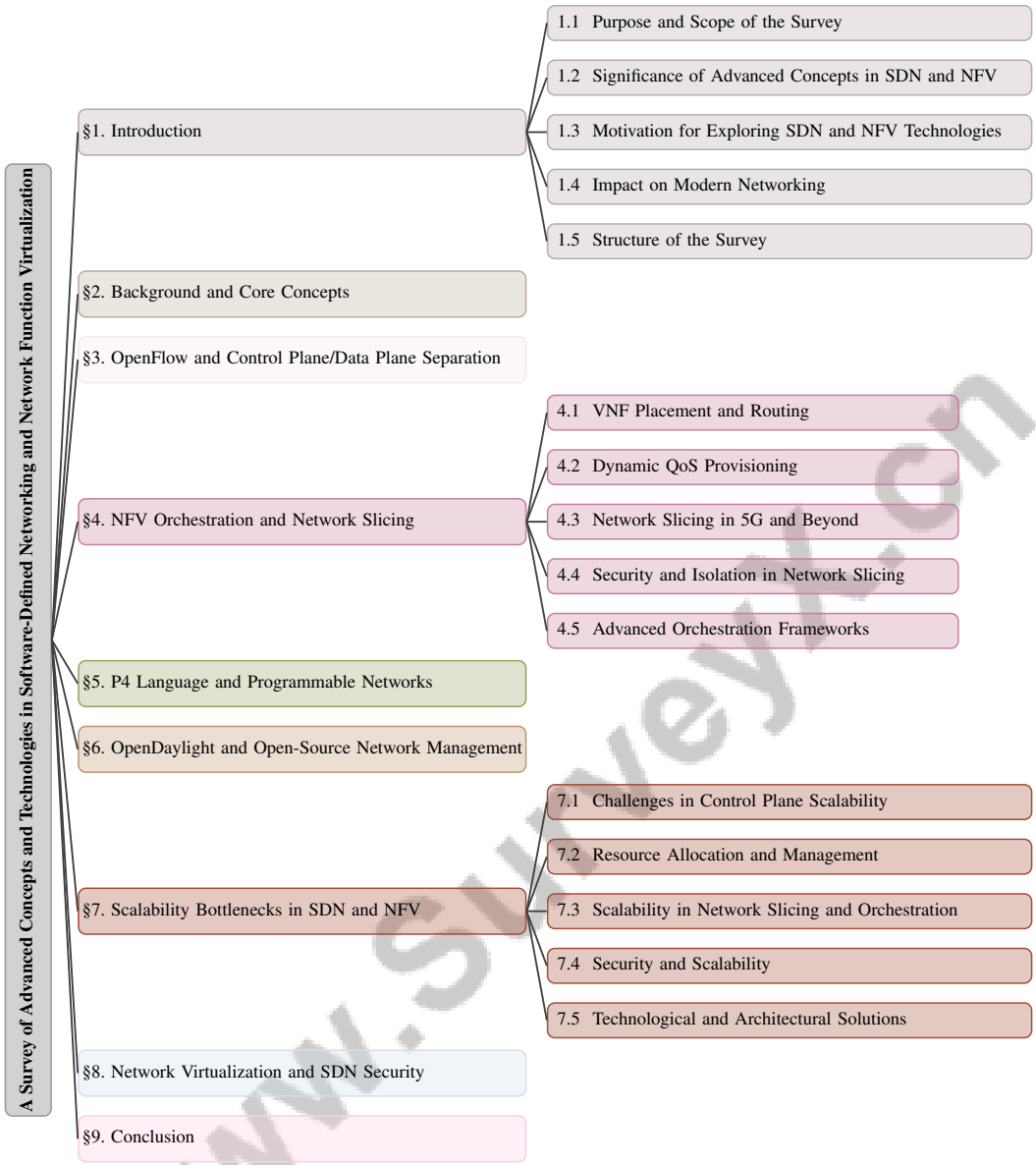


Figure 1: chapter structure

in cloud contexts [6]. Moreover, it addresses challenges faced by Internet Service Providers (ISPs) in managing Quality of Experience (QoE) for Over-The-Top (OTT) multimedia applications [7] and the complexities of end-to-end orchestration and management of network slices in 5G, Fog, Edge, and cloud computing environments [8].

1.2 Significance of Advanced Concepts in SDN and NFV

The integration of SDN and NFV is crucial in transforming the management and efficiency of modern networking infrastructures. These technologies are essential for addressing the complexities associated with growing demands for network resources and diverse application requirements. SDN enhances network configurability and responsiveness by decoupling the control plane from the data plane, providing a dynamic framework that surpasses traditional networking paradigms [9].

In multimedia services, the adoption of SDN and NFV significantly improves Quality of Experience (QoE) management, especially with the rise of Over-The-Top (OTT) content [7]. The ability to create isolated end-to-end networking abstractions through techniques like Cloud Network Slicing (CNS)

further illustrates the transformative potential of these technologies [4]. In 5G networks, SDN and NFV are vital for secure and efficient operations, providing frameworks for managing complex and heterogeneous environments [10].

The shift from host-centric to content-centric approaches through the integration of Information-Centric Networking (ICN) with SDN addresses the limitations of legacy IP networks. Secure routing within SDN frameworks is essential for ensuring the integrity and reliability of network services [11]. Furthermore, addressing vulnerabilities in SDN architectures, such as topology poisoning attacks, is critical for maintaining network security [12]. The increasing demand for reliable multicast solutions for streaming services to multiple users simultaneously underscores the importance of robust multicast capabilities [13]. Additionally, cooperative security approaches for IoT devices are pivotal in mitigating vulnerabilities arising from collusive cyberattacks, shaping the future of secure networking [14].

Advancements in network slicing, particularly regarding automation and cost-efficiency, signify substantial progress in orchestrating virtual network functions [8]. Technologies like ContentFlow are crucial for enabling content-based routing, further demonstrating their importance in future networking landscapes [15]. Collectively, these advanced concepts in SDN and NFV are instrumental in developing flexible, secure, and efficient network infrastructures that meet the demands of contemporary applications.

1.3 Motivation for Exploring SDN and NFV Technologies

The exploration of SDN and NFV is motivated by the urgent need for networks that are highly adaptable and programmable, capable of responding dynamically to the evolving demands of modern services. This is particularly critical in the context of 5G and anticipated 6G applications, where existing mobile architectures fall short in meeting diverse requirements such as enhanced mobility, robust security, and efficient resource utilization [16]. The integration of 5G communication, Artificial Intelligence (AI), and Internet of Things (IoT) technologies poses significant challenges in constructing scalable infrastructures for heterogeneous IoT applications across multiple domains [16].

Key motivations for advancing SDN and NFV technologies include the necessity for automation and dynamic policy management to optimize network performance [17]. Existing Multi-Protocol Label Switching (MPLS)-based methods lack flexibility, necessitating the development of dynamic solutions that leverage SDN capabilities [18]. Additionally, the insufficient integration of wireless networking support within SDN constrains the programmability and management of wireless networks compared to wired counterparts, highlighting the need for enhanced wireless support [19].

The vulnerability of IoT systems to collusive cyberattacks further drives research into SDN and NFV, emphasizing the need for cooperative security approaches [14]. The demand for practical solutions to evaluate network slicing functionalities in 5G mobile networks is also a significant motivator in this research area [20]. Furthermore, exploring SDN traffic measurement solutions aims to address security issues and enhance the efficiency and robustness of Software Defined Networks (SDN) and Software Defined Cellular Networks (SDCN) [21].

Additionally, the need for scalable and automated methods to manage network topologies that incorporate both Information-Centric Networking (ICN) and SDN is a critical driver of research efforts [22]. The integration challenges of ICN into existing networking architectures necessitate scalable and efficient paradigms [23]. The optimization of traffic workload and network slicing in 5G networks, focusing on managing increasing traffic demands and ensuring quality of service (QoS), further underscores the motivation for exploring these technologies [24]. Collectively, these factors highlight the pressing need for continued innovation in SDN and NFV to address evolving challenges and opportunities in modern networking environments.

1.4 Impact on Modern Networking

SDN and NFV have fundamentally transformed modern networking paradigms by enhancing flexibility, efficiency, and adaptability. These technologies facilitate dynamic resource allocation within heterogeneous network environments, effectively addressing the diverse demands of contemporary IT infrastructures [7]. The programmability inherent in SDN, particularly within the data plane,

enables significant advancements in network customization and optimization, overcoming historical challenges in network development [3]. By separating control from data forwarding, SDN redefines traditional network architectures, improving routing efficiency and protocol adaptation in heterogeneous environments [2].

In 5G networks, SDN and NFV play vital roles in implementing key components such as Next Generation Radio Access Network (NG-RAN), Multi-access Edge Computing (MEC), and virtualized Evolved Packet Core (vEPC), all essential for secure and efficient network operations [10]. The integration of these technologies with emerging paradigms like Information-Centric Networking (ICN) addresses challenges in delivering services over new architectures and meeting 5G requirements [25].

Moreover, SDN and NFV facilitate the orchestration and management of network slices, addressing complexities in resource provisioning, mobility management, and wireless resource virtualization in multi-domain IoT environments. The integration of machine learning into SDN traffic measurement solutions enhances security and efficiency, representing a promising research avenue [21].

Despite these advancements, challenges remain, particularly in single-controller SDN scenarios, where controller unavailability can lead to system instability [26]. Additionally, the need for improved accuracy in predictions and the ability to handle diverse data types underscores the versatility of SDN and NFV for various applications [1]. The ongoing evolution of these technologies emphasizes their critical role in shaping the future of networking, providing essential tools for constructing flexible, secure, and efficient infrastructures capable of thriving in an increasingly complex digital landscape.

1.5 Structure of the Survey

This survey is meticulously organized to provide a comprehensive exploration of advanced concepts and technologies within SDN and NFV. It is divided into distinct sections, each focusing on key themes and technological advancements. Initially, the survey introduces the purpose, scope, and significance of SDN and NFV, establishing the motivation for research and their impact on modern networking paradigms. Following this, background and core concepts are discussed, including the separation of control and data planes, the role of OpenFlow, and the significance of NFV orchestration and network virtualization.

Subsequent sections delve into specific technologies and challenges, such as OpenFlow's role in control plane/data plane separation, orchestration of virtual network functions, and network slicing. The survey further explores the P4 programming language and its applications in programmable networks, emphasizing its role in network customization and integration with NFV. The role of OpenDaylight as an open-source platform for SDN management is analyzed, highlighting its architecture, features, and contributions to traffic management and security enhancements.

The survey also addresses scalability bottlenecks in SDN and NFV, examining challenges in control plane scalability, resource allocation, and security. This is followed by a discussion on network virtualization and SDN security, focusing on integration strategies, security challenges, and future research directions. The survey concludes with a summary of key findings, emphasizing advancements and challenges in SDN and NFV technologies, along with potential future research directions and their impact on the evolution of networking. This structured approach aligns with the methodology outlined in the research, which organizes current studies into descriptive, correlational, and interventional stages [27].The following sections are organized as shown in Figure 1.

2 Background and Core Concepts

2.1 Separation of Control and Data Planes

The decoupling of control and data planes is central to Software-Defined Networking (SDN), enhancing network programmability and flexibility. This separation allows the control plane, responsible for decision-making and policy enforcement, to function independently from the data plane, which handles packet forwarding. This architectural shift facilitates simplified IP network configurations, reducing operational costs and enabling dynamic policy adjustments without hardware alterations [28, 23]. SDN's centralized control enhances scalability and adaptability, as seen in applications like

ContentFlow, which manages content routing by identity [15], and fosters IoT device cooperation for improved security [14].

However, the centralized nature of SDN controllers presents security challenges, such as vulnerability to network fingerprinting by remote attackers [29] and link discovery manipulation [12]. The lack of standardized northbound interfaces (NBIs) can lead to vendor lock-in, complicating SDN application development and deployment. Ensuring consistent network states across distributed controllers is crucial, particularly in scenarios involving network partitioning [23]. Addressing these issues requires innovative approaches to enhance decision-making and improve security and efficiency in SDN environments.

2.2 OpenFlow and Network Programmability

OpenFlow has been instrumental in advancing SDN by providing a standardized interface that enables the separation of control and data planes, thereby enhancing network programmability. This separation allows centralized management, where control plane policies dynamically configure network behavior, fostering flexibility and adaptability [30, 31]. OpenFlow's architecture supports network virtualization and scalability, further augmented by programming languages like SNAP for stateful packet processing [32]. Despite its benefits, OpenFlow faces challenges such as packet processing delays in Open vSwitch [33] and control path bottlenecks in traffic management [34].

Machine learning frameworks like NeuRoute enhance dynamic routing capabilities, addressing the need for predictive solutions [35]. OpenFlow remains pivotal in SDN, driving innovations in management and security. FPGA-based switches compliant with OpenFlow demonstrate high-speed packet processing with minimal resource use, enhancing scalability [6]. Integrating programmable data planes with OpenFlow, as seen in frameworks like ORACLE, improves DDoS detection and mitigation [36]. Automating the translation of high-level security policies into flow entries exemplifies OpenFlow's capability to ensure runtime security policy enforcement [37]. However, maintaining flow entries for numerous concurrent flows and identifying OpenFlow controllers remain critical challenges [38, 39].

2.3 Network Function Virtualization (NFV) and Orchestration

Network Function Virtualization (NFV) transforms network management by deploying functions as software services on generic hardware, enhancing agility, scalability, and cost-effectiveness, especially in 5G and future 6G networks [40]. NFV orchestration, facilitated by frameworks like NFV MANO, manages Virtual Network Functions (VNFs), ensuring optimal resource allocation and service delivery [40]. Integrating NFV with SDN enhances programmability and resource management, enabling dynamic bandwidth allocation and efficient traffic management [15, 24].

Challenges in NFV include data plane acceleration, crucial for high-throughput environments. Hybrid modular switches address these limitations by balancing programmability and throughput [40]. NFV orchestration frameworks also enhance QoE management for multimedia streaming, leveraging adaptive streaming approaches and SDN/NFV capabilities [24]. Decentralized access control mechanisms strengthen security by providing secure authentication and monitoring of OpenFlow applications [13].

2.4 Network Virtualization and Its Impact

Network virtualization is pivotal in modern network architecture evolution, providing flexibility and efficiency for managing complex environments. By abstracting physical infrastructure, virtualization enables the creation of multiple virtual networks on a single physical network, each with distinct resources and policies, facilitating efficient resource management and predictable performance [41]. OpenFlow integration within virtualization frameworks is significant for IoT applications, enabling dynamic configurations and context-aware services [42]. High-level rule-based languages simplify network control rule definitions, enhancing accessibility across OpenFlow controllers [43].

In multi-tenant environments, hypervisors ensure logical isolation and resource allocation, though their performance impact requires further understanding [44]. Advanced frameworks like HyMoS achieve high throughput with advanced programmability, minimizing performance penalties [45]. Network virtualization benefits from application-level insights in SDN traffic management, en-

hancing bandwidth allocation and QoS enforcement [46]. Machine learning optimizes data center architectures, improving overall performance [17].

In 5G networks, network virtualization is integral to managing and orchestrating network slices, supporting diverse service requirements and enabling efficient resource utilization [4]. Categorizing studies based on management, orchestration, pricing models, and 5G integration underscores virtualization's comprehensive impact on network architecture evolution. Network virtualization facilitates the creation of adaptable, scalable, and high-performance networks, essential for meeting QoS requirements in next-generation networks, including 6G [47, 48, 41, 49, 50]. Implementing these technologies introduces challenges, such as maintaining low control plane latencies and addressing resource interference, which must be managed to optimize performance and reliability.

In recent years, the emergence of OpenFlow has revolutionized network management and architecture. To better understand its implications, Figure 2 illustrates the hierarchical structure of key concepts related to OpenFlow, effectively categorizing the benefits, challenges, innovations, and case studies associated with this technology. This figure not only highlights enhancements in network architecture, fault tolerance, scalability, and security but also emphasizes advanced network solutions that OpenFlow facilitates. Furthermore, it underscores the transformative impact of OpenFlow on network performance, customization, and security through its diverse applications and innovations. Such a comprehensive overview allows for a deeper appreciation of how OpenFlow reshapes the landscape of modern networking.

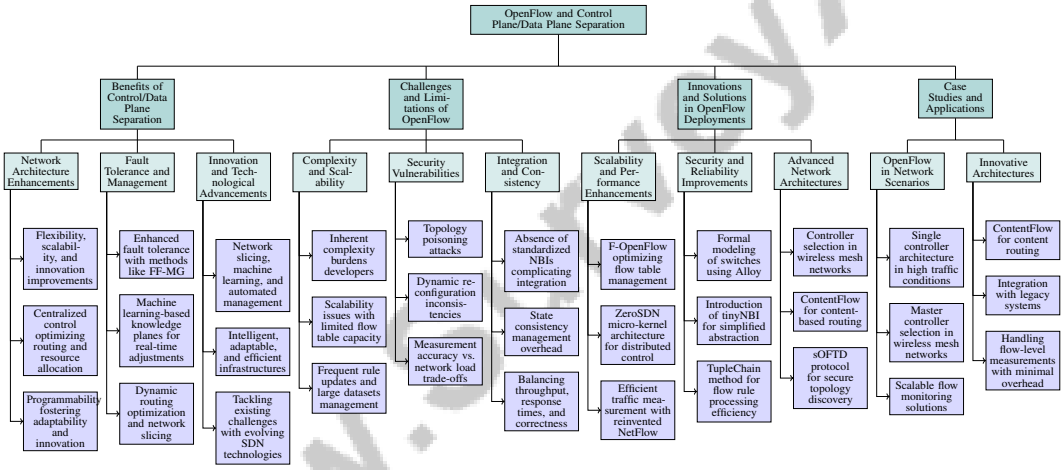


Figure 2: This figure illustrates the hierarchical structure of key concepts related to OpenFlow and the separation of control and data planes. It categorizes the benefits, challenges, innovations, and case studies associated with OpenFlow, highlighting enhancements in network architecture, fault tolerance, scalability, security, and advanced network solutions. The figure underscores the transformative impact of OpenFlow in network performance, customization, and security through its applications and innovations.

3 OpenFlow and Control Plane/Data Plane Separation

3.1 Benefits of Control/Data Plane Separation

The separation of control and data planes in Software-Defined Networking (SDN) significantly enhances network architectures by improving flexibility, scalability, and innovation. This decoupling allows for centralized control, which optimizes routing and resource allocation, as demonstrated by the ContentFlow framework [15]. The programmability of data plane devices, supported by standardized approaches, fosters adaptability and innovation in network solutions [51].

Enhanced fault tolerance is exemplified by the Fast Failover for Multicast Groups (FF-MG) method, which ensures service continuity by redirecting traffic to backup paths during link failures [13]. Machine learning-based knowledge planes enable real-time adjustments, increasing network management flexibility and agility [17].

The separation also facilitates dynamic routing optimization and network slicing, as seen in methods like Deep Q-Network based Dynamic Clustering and Placement (DDCP) [24]. Performance evaluations of SDN controllers focusing on metrics such as TCP and UDP throughput and average RTT underscore the importance of efficient control plane operations for improved network responsiveness [28].

This architectural shift not only enhances orchestration and flexibility in network resource management but also drives innovation through advanced technologies such as network slicing, machine learning, and automated management systems. It enables intelligent, adaptable, and efficient network infrastructures capable of meeting dynamic demands across industries, ensuring optimized performance and enhanced security while scaling resources in real-time [52, 17, 3, 53, 54]. As SDN technologies evolve, they tackle existing challenges, fostering robust environments for contemporary and future applications.

3.2 Challenges and Limitations of OpenFlow

OpenFlow, a foundational protocol in Software-Defined Networking (SDN), faces several challenges that impact its practical application. A major issue is the inherent complexity of the protocol, which burdens developers with managing variability and complicates network application development [55]. This complexity is exacerbated by the physical separation of data and control planes, complicating attackers' efforts to identify the controller type without direct access, thus posing security risks [39].

As illustrated in Figure 3, the primary challenges and limitations in implementing OpenFlow can be categorized into complexity issues, scalability constraints, and security vulnerabilities, highlighting associated research insights. Scalability is constrained by the limited capacity of flow tables in OpenFlow switches, which hampers the maintenance of extensive flow statistics without overloading resources [56]. The challenge of managing frequent rule updates and large datasets while maintaining high lookup speeds further complicates scalability [57]. Additionally, benchmarks often overlook critical aspects such as first packet processing time, essential for evaluating SDN controller performance [28].

Security vulnerabilities, especially topology poisoning attacks that exploit the control plane for link discovery manipulation, are a significant concern [12]. Although OpenFlow's dynamic reconfiguration capabilities enhance flexibility, they may introduce inconsistencies and potential security breaches [40]. The trade-off between measurement accuracy and network load presents challenges, as existing methods either overload the network or fail to provide accurate, timely measurements [56].

The absence of standardized northbound interfaces (NBIs) complicates application integration, resulting in dependencies on specific SDN platforms and limiting interoperability [55]. Managing state consistency across replicas introduces overhead, impacting synchronization traffic and network performance [58]. Current consistency models struggle to balance high throughput and low response times with decision-making correctness, presenting significant challenges for SDN applications [59].

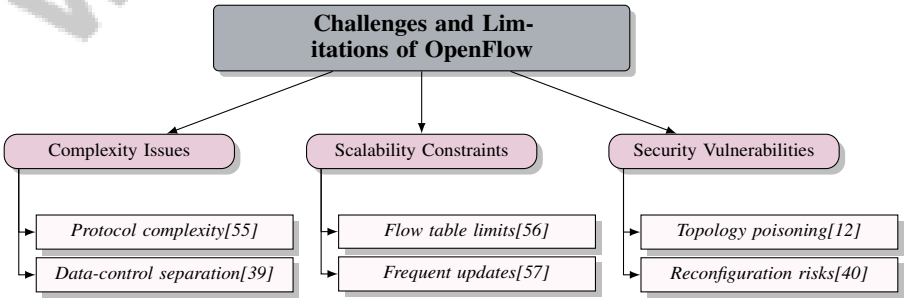


Figure 3: This figure illustrates the primary challenges and limitations in implementing OpenFlow, categorized into complexity issues, scalability constraints, and security vulnerabilities, highlighting associated research insights.

3.3 Innovations and Solutions in OpenFlow Deployments

Recent innovations in OpenFlow deployments focus on enhancing scalability, security, and performance to address existing challenges. F-OpenFlow, which integrates tuple-space search with a classification model, optimizes flow table management by reducing overhead associated with flow rule matching [60]. ZeroSDN employs a micro-kernel architecture to distribute control logic across network elements, addressing centralization challenges and enhancing scalability and resilience [9].

In traffic measurement, the reinvention of NetFlow for OpenFlow environments introduces efficient traffic sampling methods that reduce overhead and resource requirements, enabling effective flow measurements without modifying the OpenFlow specification [38]. The formal modeling of OpenFlow switches using Alloy provides a robust framework for verifying network properties, enhancing reliability and security [30].

The introduction of tinyNBI, a minimal Northbound Interface, simplifies OpenFlow abstraction configuration by offering a low-level API, reducing complexity in application development [55]. The TupleChain method enhances flow rule processing efficiency by using a tuple graph to guide lookups, maintaining competitive performance in high-performance networks [57].

A novel approach to controller selection in wireless mesh networks reduces the risk of inconsistencies during master selection by eliminating the need for controller communication, enhancing network stability [61]. These advancements significantly improve scalability, security, and efficiency in SDN environments. Innovations such as ContentFlow enable content-based routing over legacy IP architectures, while Network Function Parallelism (NFP) optimizes service function chaining in cloud networks, achieving notable latency reductions. Exploring hypervisor performance in multi-tenant SDNs reveals complexities in resource allocation and performance isolation, and the introduction of a more secure topology discovery protocol, sOFTD, addresses critical limitations of the existing OpenFlow Discovery Protocol. Collectively, these developments pave the way for more robust and adaptable SDN infrastructures [62, 15, 63, 44, 64].

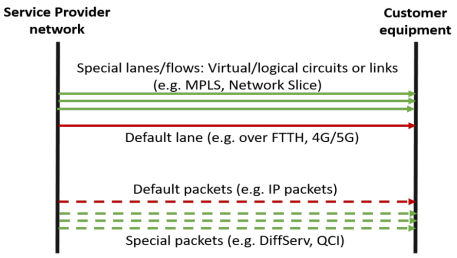
3.4 Case Studies and Applications

OpenFlow's widespread adoption across various network scenarios highlights its flexibility and effectiveness. One case study models OpenFlow-based Software-Defined Networks (SDNs) using a single controller architecture, effectively capturing feedback interactions within OpenFlow networks under high traffic conditions, crucial for maintaining stability and performance in fluctuating environments [31].

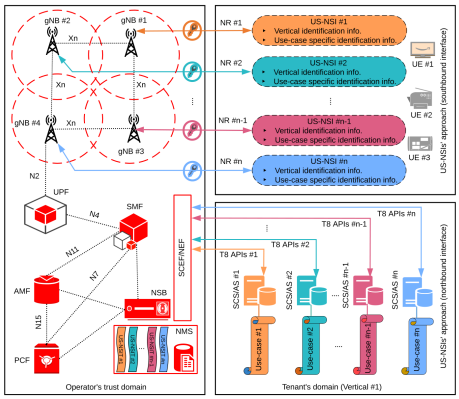
In wireless mesh networks, master controller selection is critical for stability and efficiency. Experiments using a network emulator (CORE) and physical testbeds (NITOS and w-iLab.t) have evaluated master selection mechanisms under varying conditions, demonstrating OpenFlow's adaptability to changing topologies and optimal control plane operations [61].

These case studies underscore OpenFlow's adaptability in managing complex network environments. Innovative architectures like ContentFlow enable content routing based on content names rather than traditional methods, enhancing network programmability and integration with legacy systems. The development of scalable flow monitoring solutions compatible with OpenFlow switches illustrates OpenFlow's capability to handle flow-level measurements while minimizing resource overhead, emphasizing its versatility across diverse scenarios [38, 15].

As depicted in Figure 4, the examination of "OpenFlow and Control Plane/Data Plane Separation; Case Studies and Applications" presents two illustrative case studies. The first, "Service Provider Network and Customer Equipment with Differentiated Services and Logical Circuits," visually represents a service provider network interfacing with customer equipment, emphasizing service differentiation through distinct lanes or flows, separate from the default lane used in FTTH or 4G/5G networks. This setup highlights flexibility and efficiency in managing network traffic and service delivery. The second, "US-NSIs Approach in 5G Network," details User-Specific Network Services (US-NSIs) within a 5G network, showcasing the division between the operator's trust domain and the tenant's domain, with critical network functions like the Access and Mobility Management Function (AMF) and Session Management Function (SMF) playing pivotal roles. Together, these examples underscore the transformative potential of OpenFlow and control/data plane separation in enhancing network performance, customization, and security [65, 66].



(a) Service Provider Network and Customer Equipment with Differentiated Services and Logical Circuits[65]



(b) US-NSIs Approach in 5G Network[66]

Figure 4: Examples of Case Studies and Applications

4 NFV Orchestration and Network Slicing

Category	Feature	Method
VNF Placement and Routing	Resource Management Collaborative Security	DDCP[24] CIS-SDN[14]
Dynamic QoS Provisioning	Resource Allocation Strategies	MNSAC[67], ATMF[46], ENSPLA[68], JRA[69], ML-KP[17], CALB-VNF[70]
Network Slicing in 5G and Beyond	Continuous Performance Optimization Adaptive Resource Management	PCLANSA[71] NS[72], SFI2[73], OVNFD[74], MBLP[75]
Security and Isolation in Network Slicing	Security Enhancement Resource Management	SIDM[76], OSM-WG[77], LASENS[78], TS-QKD-NFV[79], MNSO[52], N/A[12] IMCA[80], REShare[81]
Advanced Orchestration Frameworks	Resource Optimization Network Integration	VNF-PR[47], 5GT-SO[82] S3[83]

Table 1: This table presents a comprehensive overview of various methods and strategies employed in Network Function Virtualization (NFV) and network slicing, categorized into key areas such as VNF Placement and Routing, Dynamic QoS Provisioning, Network Slicing in 5G and Beyond, Security and Isolation in Network Slicing, and Advanced Orchestration Frameworks. Each category highlights specific features and methods, showcasing the diversity of approaches in optimizing network performance, resource management, and security in contemporary networking environments.

The orchestration of Network Function Virtualization (NFV) is pivotal for effective resource management, ensuring optimal performance and service delivery. Table 1 provides a detailed summary of the methods and strategies utilized in NFV orchestration and network slicing, illustrating their application across different categories such as VNF placement, dynamic QoS provisioning, and security enhancement. Additionally, Table 6 offers a comprehensive comparison of various methods employed in VNF placement and routing, dynamic QoS provisioning, and network slicing, underscoring their roles in enhancing resource management, security, and scalability within 5G and beyond. This section delves into the intricacies of Virtual Network Function (VNF) placement and routing, crucial for navigating the complexities of dynamic network environments. By exploring various strategies for VNF placement, we gain valuable insights into their impact on NFV system efficiency. The subsequent subsections detail innovative approaches that enhance VNF routing and placement within contemporary networking frameworks.

4.1 VNF Placement and Routing

Optimizing the placement and routing of VNFs is essential in NFV environments, particularly within 5G networks where low latency, high throughput, and Quality of Service (QoS) are critical. The orchestration of network slices across federated domains necessitates seamless integration of networking, computing, and storage resources to maintain performance and service quality [68]. Innovative methods like the DDCP dynamically calculate optimal SDN controller locations,

Method Name	Optimization Strategies	Resource Management	Network Efficiency
ENSPLA[68]	Heuristic Algorithm	Resource Usage	Low Execution Times
DDCP[24]	Dynamic Clustering Placement	Efficient Resource Utilization	Minimizing Control Latency
CALB-VNF[70]	Hashing Functions	Session Affinity	High Throughput
JRA[69]	Minimize Network Cost	Resource Allocation	Optimize Allocation
CIS-SDN[14]	-	-	Low Latency, High Throughput

Table 2: Comparison of various VNF placement and routing methods in NFV environments, focusing on optimization strategies, resource management, and network efficiency. The table highlights key methods such as ENSPLA, DDCP, CALB-VNF, JRA, and CIS-SDN, emphasizing their contributions to enhancing network performance and resource utilization.

improving VNF placement and routing efficiency [24]. Connection-aware Load Balancing for VNF Chains (CALB-VNF) employs hashing functions in load balancers to efficiently route traffic while maintaining session affinity, vital for NFV traffic management [70].

Frameworks optimizing VNF placement within physical substrate networks (PSNs) aim to minimize operational costs while achieving resource consumption and QoS objectives [68]. Communication Service Providers (CSPs) must efficiently allocate resources to satisfy slice requests from multiple tenants, maximizing resource utilization and minimizing costs [69]. Security is also a key consideration, as cooperative approaches enable IoT devices to share attack information with SDN controllers, enhancing NFV security [14]. The virtualization of Radio Access Network (RAN) resources into slices, complicated by limited spectrum availability and the need for tenant security and isolation, highlights the complexities of managing VNFs in multi-tenant environments [84].

As illustrated in Figure 5, which depicts the hierarchical structure of VNF placement and routing strategies, these strategies can be categorized into optimization strategies, resource management, and network efficiency. The figure highlights key methods and concepts such as DDCP, CALB-VNF, ENSPLA, joint resource allocation, RAN resource slicing, and cooperative IoT security, emphasizing their roles in enhancing network performance and resource utilization in NFV environments. Table 2 presents a comparative analysis of different VNF placement and routing methods, showcasing their optimization strategies, resource management techniques, and network efficiency characteristics, which are critical for effective NFV management in 5G networks. Advancing VNF placement and routing strategies is crucial for effective NFV management. These strategies must address the growing complexity of modern networking applications, which demand dynamic resource allocation and rapid provisioning. By optimizing VNF orchestration, including their placement within network slices and traffic routing, operators can enhance network efficiency, ensure performance isolation, and meet diverse service requirements in real-time. This evolution transforms traditional network architectures into programmable platforms supporting multi-tenancy and customized service delivery [47, 68, 41].

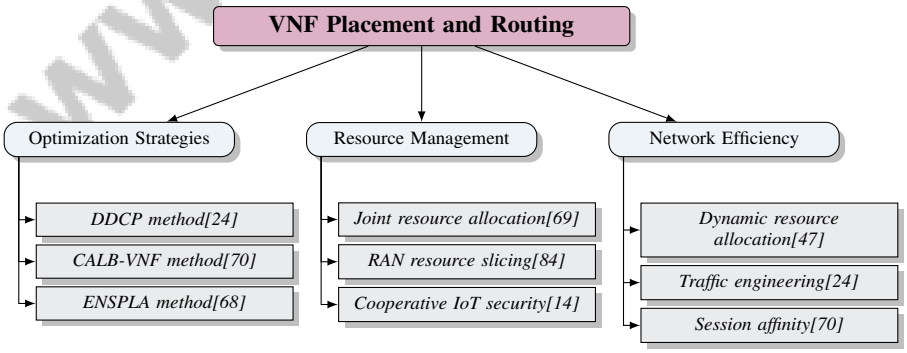


Figure 5: This figure illustrates the hierarchical structure of VNF placement and routing strategies, categorizing them into optimization strategies, resource management, and network efficiency. It highlights key methods and concepts such as DDCP, CALB-VNF, ENSPLA, joint resource allocation, RAN resource slicing, and cooperative IoT security, emphasizing their roles in enhancing network performance and resource utilization in NFV environments.

Method Name	Resource Management	Dynamic Adaptation	Service Isolation
ENSPLA[68]	Optimize Resource Usage	Quickly Adapt Changing	Network Slice Placement
T-S3RA[85]	Deep Learning Models	Dynamic Flow Offloading	Secure Slicing
ATMF[46]	Network Resource Allocation	Real-time Monitoring	-
ML-KP[17]	Optimize Resource Allocation	Dynamic Policy Adjustments	-
JRA[69]	Resource Allocation Optimization	-	Slice Requests Management
CALB-VNF[70]	Efficient Load Distribution	Dynamically Adjust Changes	-
MNSAC[67]	Resource Utilization	Machine Learning	Network Slicing Controller

Table 3: Comparison of various methods for dynamic QoS provisioning in NFV environments, focusing on resource management, dynamic adaptation, and service isolation capabilities. The table highlights the unique approaches and contributions of each method to optimize resource allocation and enhance service delivery in network slicing scenarios.

4.2 Dynamic QoS Provisioning

Dynamic Quality of Service (QoS) provisioning in NFV environments is crucial for optimizing resource allocation and ensuring efficient service delivery. A key challenge is managing the arrival of multiple, volatile network slice requests, especially in edge computing scenarios, while adhering to strict end-to-end latency requirements [68]. Effective slice isolation and management are necessary for integrating multiple network operators on shared infrastructure to meet diverse service demands [85]. Table 3 provides a comprehensive comparison of different methods employed for dynamic QoS provisioning in NFV environments, illustrating their effectiveness in resource management, dynamic adaptation, and service isolation.

Innovative strategies like the Application Traffic Management Framework (ATMF) allow applications to request specific QoS treatments from the SDN controller, enabling dynamic network resource management to meet application-specific QoS requirements [46]. Machine learning-based Knowledge Planes facilitate dynamic policy adjustments, enhancing QoS provisioning through real-time network adaptations based on predictive analytics [17].

Optimizing resource allocation for network slicing in NFV-enabled networks is furthered by a joint resource and admission management algorithm, minimizing network costs for CSPs while managing slice requests from tenants [69]. This approach ensures efficient resource utilization and SLA compliance by dynamically adapting to varying slice demands. Methods for scaling VNFs without code modifications demonstrate efficient traffic distribution management using load balancers, preserving session state crucial for service continuity [70]. Multi-queuing systems enhance resource allocation by enabling differentiated handling of slice requests based on tenant characteristics and slice types, ensuring precise fulfillment of diverse service demands [67].

4.3 Network Slicing in 5G and Beyond

Method Name	Resource Management	Service Customization	Network Scalability
NS[72]	Resource Allocation Optimization	Tailored Network Slices	Evolve And Adapt
PCLANSA[71]	Dynamic Resource Scaling	Tailor Network Slices	Enhancing Scalability Efficiency
REShare[81]	Dynamic Embedding Algorithm	Tailor Network Slices	Evolving Network Demands
S3[83]	Dynamic Resource Allocation	Customizable Apis	Flexible Management System
MBLP[75]	Resource Allocation Optimization	Flexible Routing Guarantees	5G And Beyond
OVNFD[74]	Resource Allocation Optimization	Varying Service Requirements	Dynamic Resource Allocation
SFI2[73]	Machine Learning Optimizations	Slice-as-a-service	End-to-end Orchestration

Table 4: Comparison of Various Network Slicing Methods in 5G and Beyond, Highlighting Resource Management, Service Customization, and Network Scalability. The table provides an overview of different methodologies and their approaches to optimizing resource allocation, customizing services, and enhancing network scalability to meet the evolving demands of modern applications.

Network slicing is a transformative advancement in 5G mobile networks, allowing the creation of multiple virtual networks on shared infrastructure. This capability enables operators to customize network slices to meet specific service demands, essential for managing the diverse connectivity requirements of modern applications, including industrial settings [72]. The flexibility of network slicing accommodates varied applications, from ultra-reliable low-latency communications to enhanced mobile broadband services, optimizing resource utilization and service delivery [71]. Table 4 presents a comparative analysis of network slicing methods, showcasing their strategies for resource

management, service customization, and network scalability in the context of 5G and future network architectures.

Dynamic service embedding within network slices is critical for efficient resource management. Algorithms like REShare optimize resource allocation by considering time-varying service loads, enhancing adaptability and efficiency [81]. The integration of satellite networks into 5G through network slicing frameworks extends terrestrial networks' reach and capabilities, reflecting the evolution of networks to meet diverse service requirements [83].

Optimal formulations addressing resource allocation and latency constraints significantly enhance network slicing performance in 5G and beyond. These formulations balance resource allocation, ensuring latency requirements are met while optimizing VNF deployment [75]. Joint optimization of VNF deployment and resource allocation supports efficient VNF chain management, improving resource utilization across network slices [74]. As network slicing architectures evolve to meet the demands of 5G and future 6G applications, enhanced scalability and flexibility in network management become increasingly important [73]. Small-scale 5G testbeds implementing network slicing functionalities significantly advance network management practices, providing valuable insights into the practical application of network slicing strategies [20].

4.4 Security and Isolation in Network Slicing

Security and isolation are fundamental in deploying network slicing within 5G and future communication networks, ensuring multiple slices coexist on shared infrastructure without interference or security breaches. Effective isolation is critical for maintaining the integrity and performance of each slice, given the diverse requirements and SLAs associated with different applications [80]. Enhancing security involves using slice isolation to mitigate DDoS attacks, ensuring resource allocation guarantees both security and performance [76].

The integration of SDN, NFV, and cloud technologies in platforms like 5GIIK provides comprehensive support for multi-tenancy and dynamic slice provisioning, essential for maintaining security and isolation in network slicing environments [86]. However, challenges remain in utilizing physical network and computing infrastructure efficiently to provide reliable and secure connections to cyber-physical systems [78]. The OSM-WireGuard framework enhances security through effective traffic isolation, ensuring confidentiality in virtualized network functions [77].

Security risks associated with NFV deployment necessitate secure methods to protect the transfer of network functions stored as software images in remote data centers [79]. The adaptability of solutions like REShare to changing network conditions allows for efficient resource usage and lower operational costs, crucial for maintaining isolation and security in dynamic environments [81]. Despite advancements, limitations persist, including complexities in managing inter-domain connectivity and the need for standardized interfaces and protocols for seamless integration [52]. Challenges posed by topology poisoning in SDN environments, particularly regarding link discovery protocol manipulation, highlight the need for robust security measures to safeguard network integrity [12].

4.5 Advanced Orchestration Frameworks

Method Name	Resource Allocation	Network Integration	Scalability and Adaptability
S3[83]	Dynamic Resource Allocation	Efficient Integration 5G	Flexible Management System
VNF-PR[47]	Resource Utilization Inefficiencies	-	Scalability Challenges
5GT-SO[82]	Resource Orchestration Capabilities	Not Explicitly Mentioned	Flexible Orchestration Mechanism

Table 5: Comparison of Advanced Orchestration Frameworks in Terms of Resource Allocation, Network Integration, and Scalability. This table presents an analysis of various frameworks, including S3, VNF-PR, and 5GT-SO, highlighting their approaches to resource management, integration with 5G networks, and adaptability to changing network demands.

Advanced orchestration frameworks are vital for optimizing NFV and network slicing management, addressing the intricate challenges of resource allocation and service delivery in complex network environments. These frameworks employ sophisticated algorithms to enhance VNF placement and chaining, ensuring robust and scalable network operations. A notable innovation is the Satellite Slice as a Service (S3) framework, which mutualizes satellite infrastructure to enable efficient

integration with 5G networks, showcasing potential for extending network capabilities beyond terrestrial boundaries [83].

Developing advanced algorithms for resource allocation is essential for enhancing scalability and security in network slicing. Future work should refine these algorithms and architecture to better integrate network slices, improving overall NFV environment efficiency [84]. Addressing limitations in current studies, particularly in management and orchestration across multiple network domains, is vital for advancing network slicing practices [20].

The proactive closed-loop algorithm PCLANSA optimizes resource allocation across diverse network slices in 5G and Beyond 5G (B5G) environments, ensuring adherence to QoS requirements and enhancing service assurance. By dynamically scaling VNFs and efficiently managing resources in real-time, PCLANSA addresses the critical need for performance and reliability in network slicing. This approach minimizes resource utilization and reduces over-provisioning, underscoring the significance of advanced orchestration strategies within the SDN and NFV framework [87, 80, 88, 71]. Future research should enhance PCLANSA's adaptability to rapidly changing network conditions and explore integrating advanced machine learning techniques for more accurate resource prediction.

Predictive modeling in resource allocation, such as the temporal characterization of VR traffic, reduces overprovisioning and improves QoS, illustrating the benefits of adaptive resource management in advanced orchestration frameworks. The scalability and efficiency of frameworks designed for managing ICN topologies within SDN networks enhance the deployment of Information-Centric Networking (ICN) by allowing integration with existing protocols without requiring modifications. This underscores the critical role of seamless orchestration in bridging traditional network architectures with innovative ICN solutions, as demonstrated by studies proposing dynamic topology management strategies and experimental implementations within SDN environments. These advancements facilitate intelligent management of ICN-enabled networks, ensuring operators can leverage their legacy infrastructure while evolving towards more content-centric networking paradigms [46, 23, 22, 15].

As shown in **??**, the integration of advanced orchestration frameworks plays a pivotal role in optimizing network functionalities and service delivery. The first example, "Internet Services Network Architecture," illustrates a traditional network setup where a provider orchestrates internet services through a cloud-based management system, featuring components like firewalls, routers, and switches interconnected via secure tunneling mechanisms for efficient data transmission. In contrast, the "5G Network Architecture with Multiple Vertical Services and Multiple Administrative Domains" represents a dynamic structure reflecting modern 5G technology demands, supporting multiple vertical services across various administrative domains linked to service operations and management platforms. Together, these examples underscore the evolution of network architectures towards more sophisticated, flexible, and secure frameworks capable of meeting the diverse needs of contemporary digital ecosystems [47, 82]. Additionally, Table 5 provides a comparative analysis of advanced orchestration frameworks, focusing on their resource allocation strategies, network integration capabilities, and scalability features.

Feature	VNF Placement and Routing	Dynamic QoS Provisioning	Network Slicing in 5G and Beyond
Resource Management	Dynamic Allocation	Efficient Allocation	Optimal Allocation
Security Features	Cooperative Iot Security	Slice Isolation	Confidentiality Ensured
Scalability	Multi-tenant Environments	Edge Computing Scenarios	5G And Future

Table 6: Comparison of VNF Placement, Dynamic QoS Provisioning, and Network Slicing Methods in 5G Environments. This table highlights the distinct features across three critical domains: resource management, security features, and scalability, providing insights into their applications and benefits in next-generation network architectures.

5 P4 Language and Programmable Networks

5.1 Introduction to P4 and Programmable Networks

The P4 programming language revolutionizes programmable networks by offering a protocol-independent approach to packet processing, crucial for network operators seeking customization beyond hardware and protocol constraints [89]. By abstracting network protocol complexities, P4 enables dynamic device reconfiguration, supporting sophisticated functionalities such as middleboxes

[90]. P4's robust type system and whole program analysis enhance security and facilitate reliable network configurations [91]. It supports functions from basic forwarding to advanced traffic management, optimizing performance when integrated with hardware accelerators like P4-compatible NICs and PCI-e [45].

As illustrated in Figure 6, the key features, applications, and challenges associated with P4 and programmable networks are highlighted, showcasing the protocol independence, dynamic reconfiguration, and security enhancements offered by P4. The applications include middleboxes, traffic management, and memory management, while the challenges involve security risks, API verification, and efficient lookup solutions. However, the security landscape of programmable dataplanes, including P4, introduces new attack surfaces necessitating careful management [92]. Verified APIs, such as the P4R-Type API, perform static checks to ensure conformity between P4 tables and actions, maintaining the security and integrity of P4-enabled networks [93]. Moreover, P4's integration with advanced memory management techniques, such as FastReact, showcases its versatility in supporting diverse networking applications [94]. The TupleChain flow lookup scheme exemplifies P4's efficiency in handling complex operations through optimized lookups and updates [57].

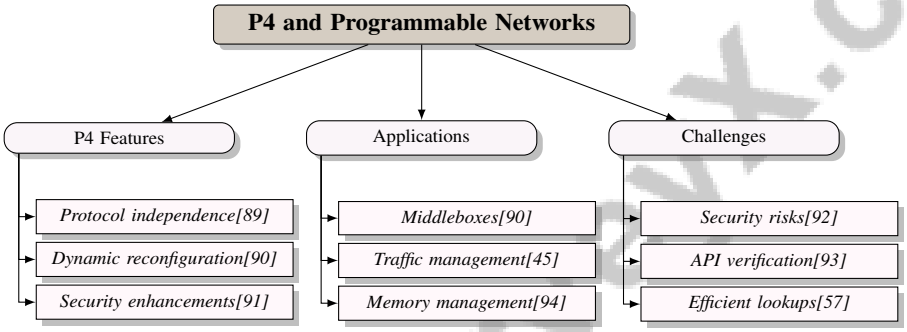


Figure 6: This figure illustrates the key features, applications, and challenges associated with P4 and programmable networks, highlighting the protocol independence, dynamic reconfiguration, and security enhancements offered by P4. The applications include middleboxes, traffic management, and memory management, while the challenges involve security risks, API verification, and efficient lookup solutions.

5.2 P4's Role in Network Customization

P4 facilitates network customization by providing a framework for precise packet processing task management, adaptable to new protocols and enabling field reconfiguration without hardware constraints [89]. This adaptability is crucial in dynamic environments where rapid protocol changes enhance performance and responsiveness. P4's architecture supports dynamic reconfiguration, essential for advanced traffic management and security operations, allowing real-time adaptation independent of protocols or hardware. This enhances efficiency in network application management, supporting a wide array of custom packet processing functions and facilitating rapid innovation [45, 95, 89]. Integration with tools like P4R-Type leverages Scala's type system to enforce correctness in P4Runtime operations, ensuring adherence to predefined specifications and enhancing operational reliability [93]. Frameworks such as tinyNBI simplify development by providing a consistent interface that abstracts OpenFlow's complexities, facilitating seamless P4 integration with existing infrastructures [55].

5.3 Integration of P4 with Network Function Virtualization

Integrating P4 with Network Function Virtualization (NFV) significantly advances network programmability and flexibility. P4 abstracts hardware complexities, providing a unified approach to packet processing, essential for addressing modern networks' dynamic needs [89]. This abstraction allows seamless NFV integration, enabling customizable and efficient virtualized network functions deployment. P4 programmable switches offer greater flexibility and customization than traditional and SDN devices, optimizing NFV deployments [95]. The dynamic reconfiguration of network devices facilitates implementing complex functions and policies without hardware modifications, crucial for adapting to changing conditions. The P4R-Type system enhances integration by providing

a formal typing system ensuring well-typed programs do not perform invalid P4Runtime operations, improving programmable networks' reliability and security [93]. Furthermore, P4's integration with platforms like NetIDE enhances programmability and interoperability in NFV deployments, enabling cohesive SDN application functionality [96].

5.4 Security and Reliability in Programmable Networks

Programmable networks, especially those utilizing P4, offer flexibility in network management but also pose significant security and reliability challenges. P4's protocol-independent packet processing enables dynamic device reconfiguration, aligning with modern environments' evolving needs [89]. However, this flexibility can introduce vulnerabilities, necessitating robust security measures to mitigate threats [92]. Managing security in dynamic environments where novel attack vectors exploit device programmability is a primary challenge. Innovations like LineSwitch, an OpenFlow extension, enhance security by efficiently managing TCP connections and reducing memory overhead, protecting against saturation attacks [97]. Ensuring reliable operations involves effective countermeasures against emerging threats. The P4R-Type API offers significant improvements by providing static verification of P4Runtime operations to prevent common errors associated with weakly-typed APIs [93]. By ensuring compliance with predefined specifications, P4R-Type enhances network functions' reliability, minimizing operational disruptions.

6 OpenDaylight and Open-Source Network Management

6.1 OpenDaylight Architecture and Features

OpenDaylight (ODL) stands out as a premier open-source platform for managing Software-Defined Networking (SDN) environments, offering a modular architecture that enhances flexibility and extensibility through diverse plugins and modules [28]. Its compatibility with multiple southbound protocols, notably OpenFlow, abstracts hardware complexities, facilitating seamless control of network devices. Extensions such as TIME4 further enhance OpenFlow's capabilities to support time-sensitive networking, which is crucial for synchronizing network elements [98]. This adaptability ensures straightforward deployment across varied network environments with minimal modifications [99].

OpenDaylight's robustness and scalability are demonstrated in experimental setups measuring bootstrapping times across different topologies, underscoring its efficiency in managing complex configurations [22]. The platform also enhances network resilience through algorithms for backup forwarding rules, crucial for disaster recovery scenarios [100]. Moreover, its compliance with OpenFlow standards is validated by scalable switch architectures that ensure high performance [6]. Implementations using OpenFlow and the NOX controller on Mininet testbeds further bolster network reliability under adverse conditions [101].

6.2 Traffic Management and Automation

OpenDaylight plays a pivotal role in optimizing traffic management and automating network operations within SDN environments. It facilitates dynamic Quality of Service (QoS) provisioning and centralized traffic flow control, enhancing resource utilization in multi-tenant networks. Its programmability, combined with protocols like OpenFlow, enables efficient separation of control and data planes, thus prioritizing network traffic and minimizing disruptions during frequent path reconfigurations [98, 46, 28, 102, 38]. The platform's modular architecture supports seamless integration of various network functions, enabling effective traffic control and automation.

OpenDaylight's support for multiple southbound protocols, including OpenFlow, offers a standardized interface for managing traffic flows, ensuring efficient data packet routing. The integration of algorithms for backup forwarding rules further enhances network resilience, particularly during link failures [100]. Its automation capabilities are exemplified through support for time-sensitive networking (TSN) extensions like TIME4, ensuring precise synchronization of network elements for low-latency and high-reliability applications [98]. The platform's capability to automate network operations is further illustrated by experimental setups that measure bootstrapping time across varying topology sizes, highlighting its efficiency in managing and automating network configurations [22].

6.3 Security Enhancements through OpenDaylight

OpenDaylight significantly boosts security in SDN environments by offering a robust framework for advanced security measures. Its modular architecture integrates adaptive security features that respond in real-time to network fluctuations, protecting against threats like DDoS and intrusion attacks. This approach enhances the overall security posture, leveraging intelligent microservices and machine learning techniques, especially in next-generation networks such as 5G and 6G, where dynamic network slicing requires flexible security solutions [103, 104]. The platform's management of multiple southbound protocols, including OpenFlow, centralizes the enforcement of security policies across the network.

A key security enhancement is the implementation of a failure management framework that reduces reliance on the central controller, enabling instantaneous recovery and zero packet loss, thereby enhancing resilience against security breaches [105]. By decentralizing certain control functions, OpenDaylight mitigates risks associated with single points of failure, often targeted in attacks. Support for TSN extensions, such as TIME4, further enhances security for time-critical applications by ensuring precise synchronization of network elements, crucial for maintaining communication integrity in high-security environments [98]. Compatibility with existing OpenFlow controllers allows tailored deployment of security policies across diverse network environments, effectively addressing specific threats and vulnerabilities [99].

6.4 OpenDaylight in Virtualized and Integrated Networks

OpenDaylight plays a crucial role in managing virtualized and integrated network environments, providing a flexible platform that supports dynamic orchestration of network resources. Its modular architecture enables seamless integration with various network functions, such as firewalls and load balancers, enhancing the management of virtualized infrastructures. This design supports high-throughput capabilities through advanced hardware and P4-compatible Network Interface Cards while maintaining programmability for optimal placement of Virtual Network Functions (VNFs) across network slices. Such flexibility is essential for meeting the demands of next-generation networks and managing resources across multiple administrative domains [47, 45, 104, 106, 52].

The platform's ability to interface with various southbound protocols, including OpenFlow, enhances its management of diverse network environments. This integration allows for incorporating virtualized network functions alongside existing physical infrastructures, optimizing resource utilization and performance isolation in multi-tenant scenarios. Leveraging OpenFlow, the platform enables centralized API-driven network management while supporting advanced features like content routing and performance monitoring [15, 44, 64, 107, 108]. This capability is vital for deploying advanced networking solutions like network slicing and multi-access edge computing, which require seamless coordination of virtual and physical components.

As illustrated in Figure 7, OpenDaylight's hierarchical structure emphasizes its role in managing virtualized and integrated networks, highlighting key aspects such as platform integration, resource management, and network performance. Moreover, OpenDaylight's support for advanced orchestration frameworks facilitates dynamic resource allocation across multiple network domains, empowering operators to optimize the placement and chaining of VNFs. These frameworks address complexities introduced by Network Functions Virtualization (NFV), ensuring efficient resource utilization and enabling dynamic scaling and load balancing of VNFs to meet varying demands. Compliance with service-level agreements (SLAs) enhances overall network performance through improved resource management and reduced latency, particularly in large-scale, cloud-native, and edge-enabled networks [47, 109, 68, 70, 1].

The platform's robust adaptability to diverse network topologies and configurations positions it as an optimal solution for managing integrated network environments, where seamless coordination of multiple functions and services is crucial. This adaptability supports advanced architectures like network slicing, allowing tailored provisioning of resources across various industries with distinct service requirements. By leveraging SDN and NFV components, OpenDaylight enhances orchestration and management capabilities, ensuring efficient resource allocation and improved performance in complex, multi-domain scenarios [52, 66, 47, 104]. Its ability to automate network tasks and manage complex configurations ensures that virtualized and integrated networks operate efficiently, meeting the demands of modern and emerging applications.

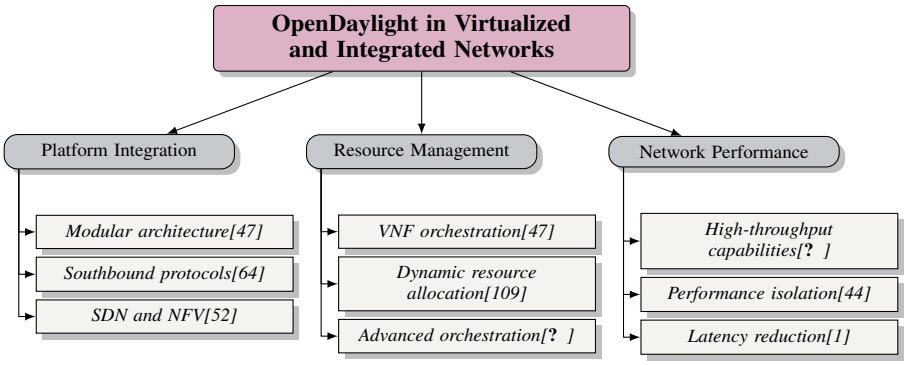


Figure 7: This figure illustrates the hierarchical structure of OpenDaylight's role in managing virtualized and integrated networks, highlighting key aspects such as platform integration, resource management, and network performance.

7 Scalability Bottlenecks in SDN and NFV

Understanding scalability bottlenecks in Software-Defined Networking (SDN) and Network Function Virtualization (NFV) is crucial as network size and complexity increase. This section delves into the critical issues affecting control plane scalability in modern network architectures.

7.1 Challenges in Control Plane Scalability

The control plane in SDN faces significant scalability challenges as networks grow. A primary concern is the limited capacity of Ternary Content Addressable Memory (TCAM) in switches, leading to excessive flow entries and lookup delays, which degrade performance [57]. Efficient resource solutions are needed to support SDN functionalities while meeting throughput and latency demands [9]. Variability across OpenFlow versions complicates application development and increases error potential, compounded by data processing latencies [55, 110]. The lack of a common abstraction layer further complicates programmability across different data plane devices, hindering scalability [51].

Dynamic traffic and real-time routing adjustments pose additional scalability challenges [24]. The interdependence of virtual node and link embedding complicates resource allocation, increasing fragmentation risk [111]. The inability to optimize all slices dynamically limits decision-making, posing management challenges [112]. In variable environments with fluctuating link capacities, maintaining performance consistency is challenging, exacerbating scalability issues [107]. Secure path reliance, as in FARSec, highlights difficulties in maintaining scalability amid dynamic conditions [11]. Hypervisor overheads in multi-tenant environments impact scalability, particularly in OpenFlow-based setups [44].

Innovative approaches are necessary to enhance control plane scalability. Optimizing controller placements in large networks provides a robust framework for improving scalability and demonstrates practical applicability [21]. The exponential growth of content items in Information-Centric Networking (ICN) also presents significant scalability challenges, complicating name-based routing [23]. The framework for network slicing in 5G emphasizes architectural principles that enable dedicated and shared slice coexistence, crucial for scalability [84]. In IoT contexts, enabling cooperative security by sharing attack information across networks presents specific scalability challenges [14].

7.2 Resource Allocation and Management

Efficient resource allocation and management are critical for optimizing SDN and NFV performance in complex, dynamic networks. Traditional strategies struggle to meet evolving requirements, necessitating adaptive, intelligent solutions. Integrating intelligent agents into network slicing enhances real-time decision-making and resource allocation, improving service agility and efficiency across domains like 5G/6G, IoV, and IIoT [103, 52, 113, 87, 54].

Dynamic bandwidth allocation in MPLS networks through a centralized SDN controller illustrates SDN's capability to enhance resource utilization by reallocating resources based on demand [15, 41]. However, sharing resources across service providers complicates VNF and SFC orchestration and scheduling. Effective resource management in multi-domain environments relies on collaboration among slice owners, network controllers, and cloud providers to orchestrate network slices across various domains [52, 8, 114, 4]. This coordination balances traffic-fairness with computing-fairness while ensuring efficient resource allocation.

Managing stateful operations remains challenging due to frequent central controller communication, which can introduce delays. Strategies minimizing communication overhead while ensuring optimal resource allocation are essential. Proposed methods for optimizing resource allocation in VNF environments address VNF interdependencies, improving embedding performance and resource utilization [109, 47, 115, 1].

In large-scale networks, managing tasks like content routing and network slicing poses computational challenges impacting runtime performance, especially in architectures supporting advanced functionalities like content management and caching [15, 4]. Existing methods often suffer from inefficiency due to large problem sizes or inadequate problem structure consideration. Managing multiple NSIs and potential orchestration overheads are significant challenges in resource allocation and management.

Orchestrating virtual networks across providers introduces complexities, particularly in negotiation and provisioning. The federated cloud network architecture underscores the need for effective resource management across domains, facilitating network slice orchestration and ensuring seamless service delivery through a multi-layered framework integrating SDN and NFV technologies [52, 4]. Challenges in network slicing include technical implementation difficulties, efficient resource allocation management, and business issues like cost optimization and revenue generation.

Efficient resource allocation and management strategies are vital for SDN and NFV operation. Addressing dynamic provisioning challenges and optimizing resource utilization significantly enhances network performance and ensures high-quality service delivery across applications. The main challenge lies in satellite network management complexity and rigidity, hindering effective network slicing application [83].

7.3 Scalability in Network Slicing and Orchestration

Scalability in network slicing and NFV orchestration is critical as networks evolve to accommodate diverse service requirements. Orchestrating network slices involves complex resource allocation and management across domains and technologies, presenting significant scalability challenges. This process requires a sophisticated architecture integrating SDN and NFV to manage network slice lifecycles effectively. Current approaches struggle to address service demand intricacies in multi-domain environments, emphasizing the need for innovative solutions leveraging LLMs and machine learning for optimized resource orchestration and enhanced collaboration [52, 8, 109, 113, 54]. Efficient resource allocation strategies adapting to fluctuating demands while maintaining service quality are essential.

The Optimal Cross-Slice Orchestration (OCSO) method optimizes resource allocation across slices but relies on accurate request modeling, which may not reflect real-world conditions, leading to suboptimal allocation if conditions deviate [116]. This limitation highlights the need for adaptive, resilient orchestration frameworks.

Scalability concerns are exacerbated by managing multiple NSIs across federated environments. Orchestrating NSIs requires seamless coordination among domains, each with its own resources and policies. The complexity of orchestrating resources and functionalities within network slicing and VNFs increases orchestration overhead, hindering scalability and efficiency [54, 47, 113].

Integrating advanced technologies like machine learning and AI into orchestration frameworks enhances scalability by enabling predictive resource allocation and dynamic adaptation to network conditions. Technologies like Digital Twin (DT) frameworks and network slicing mechanisms facilitate real-time monitoring and analysis of performance and demand patterns, improving slice management and enabling tailored virtual networks for applications. This capability enhances scalability by optimizing resource allocation and adapting to dynamic service requirements across

environments like 5G, Fog, and Cloud computing. By integrating SDN and NFV, these technologies support a sophisticated orchestration architecture managing multiple slices across domains, improving network efficiency and responsiveness [117, 8, 52, 67].

7.4 Security and Scalability

The interplay between security and scalability in SDN and NFV environments is crucial in influencing network infrastructure design and deployment. The centralized SDN control plane facilitates efficient management but presents scalability challenges and security vulnerabilities. As the central management point, the controller is susceptible to threats, including DDoS attacks, which can saturate buffers and degrade performance [97]. Solutions like LineSwitch address these issues by reducing memory usage for connection management, enhancing resilience.

Innovative approaches like FloRa offer high detection accuracy and low resource overhead, suitable for real-time SDN deployment [118]. Its minimal resource consumption underscores the importance of resource-efficient solutions in maintaining security and scalability. Similarly, AMS-HD provides low memory and computational overhead, enabling effective DDoS detection without performance compromise [119].

Managing security across network slices complicates scalability efforts. Strong resource isolation is vital to prevent unauthorized access and minimize DDoS impacts on unaffected slices [76]. Existing methods often struggle to achieve adequate isolation, leading to vulnerabilities and performance degradation [77].

Integrating independent security layers, as seen in approaches outside traditional TLS implementations, offers additional safeguards against TLS risks, enhancing the overall security framework against evolving threats [120]. Solutions like Gwardar exemplify innovative approaches for securing SDN and NFV infrastructures, detecting and responding to threats at data and control plane levels without relying on a trusted NOS [121].

Despite advancements, limitations remain, such as reliance on accurate simulations for predicting network behavior. Discrepancies in simulator accuracy can lead to unexpected conditions, necessitating robust verification mechanisms [122]. While methods like Routing Verification as a Service (RVaaS) detect misbehavior, they do not prevent it, highlighting the need for proactive security measures [123].

7.5 Technological and Architectural Solutions

Addressing scalability bottlenecks in SDN and NFV requires integrating technological innovations and architectural advancements. A critical focus is developing programmable solutions that scale in large networks. These solutions must accommodate various protocols and offer user-friendly programming interfaces to manage modern networks' complexity. Integrating adaptive algorithms for timeout selection and packet sampling enhances flow monitoring scalability and efficiency, significantly reducing flow entries and OpenFlow messages [38].

Optimizing flow management through MPLS-based flow aggregation significantly reduces flow entries and OpenFlow messages, demonstrating a 96

Developing hybrid architectures combining hardware and software solutions is promising, as these architectures enhance data plane flexibility and efficiency, addressing scalability and performance limitations [3]. Introducing adaptive consistency approaches, autonomously adjusting consistency levels, offers a novel way to manage distributed systems efficiently, contrasting with static models that may not scale well in dynamic environments [59].

Regarding resource management, the BiVNE method, a nested bilevel optimization approach, simultaneously considers VNoE and VLiE, aiming to reduce fragmentation and improve profit for providers. This approach highlights optimizing resource allocation strategies to enhance scalability and efficiency [111]. Its effectiveness lies in leveraging hashing to ensure consistent traffic flow through the same function instances, enabling efficient load distribution [70].

Implementing network slicing architectures like the Network Virtualization Substrate (NVS) and Software Defined Mobile Network Control (SDMC) supports scalability by enabling dynamic

resource allocation across slices [84]. These architectures facilitate efficient resource management, ensuring service-level agreements are met while optimizing performance.

8 Network Virtualization and SDN Security

The integration of network virtualization with Software-Defined Networking (SDN) marks a pivotal advancement in network architecture, significantly enhancing operational efficiency and resource management. This synergy enables multiple virtual networks to coexist on a single physical infrastructure, allowing dynamic configurations tailored to specific service requirements. Understanding this integration's implications is crucial, particularly concerning improvements in flexibility, scalability, and security.

8.1 Integration of Network Virtualization with SDN

The integration of network virtualization with SDN substantially enhances network capabilities by enabling flexible, efficient, and scalable resource management, crucial for service-oriented architectures in modern networks like 5G. Network virtualization abstracts the physical infrastructure, allowing independent operation of multiple virtual networks, each with distinct resources and policies, thereby improving resource management and performance predictability [44]. Real-time monitoring and dynamic network adjustments are integral, as demonstrated by multi-technology monitoring schemes that create continuous feedback loops to enhance performance [124]. The AetherFlow framework expands beyond traditional wired-centric models by incorporating wireless protocols, enhancing network capabilities and application support [19].

Leveraging SDN's programmability enhances security and resilience, as seen in the Flexible Intrusion Detection and Treatment System (IDTS), which uses SDN controllers for dynamic threat response, essential for secure operations in multi-tenant environments [44]. Moreover, integrating ContentFlow with SDN exemplifies the advantages of content-based routing for efficient data flow [15]. In 5G networks, this integration supports network slicing, enabling operators to create tailored virtual networks for diverse services.

The S3 framework highlights the potential of this integration by supporting both integrated and standalone operational modes, thereby enhancing network capabilities and management flexibility [83]. This synergy represents a significant advancement in network management, equipping infrastructures to meet contemporary and future application demands.

8.2 Security Challenges in SDN Environments

Security challenges in SDN environments stem from architectural innovations like the decoupling of control and data planes and the programmability of network functions. A significant vulnerability arises from the exposure of the control plane to attacks, particularly flooding attacks that can disrupt operations [29]. The centralized nature of SDN controllers makes them prime targets; compromising a controller can lead to widespread network disruption [12].

While programmability offers flexibility, it introduces new security challenges. Programmable data planes, such as those enabled by P4, alter the attack surface, necessitating robust security measures [45]. Accurate flow classification and real-time data reliance pose challenges in variable environments [24]. In multi-tenant scenarios, ensuring the integrity of network slices and deployed Virtual Network Functions (VNFs) is critical, as any compromise can lead to privilege escalation and unauthorized access [40]. Current access control systems often depend on specific SDN controllers, leading to inflexibility and vulnerability to denial of service attacks, underscoring the need for decentralized, robust access control mechanisms [75].

The dynamic nature of SDN complicates resource management and security in multi-tenant situations. Many studies overlook the complexities of managing dynamic resources and their security implications, potentially leading to vulnerabilities in slice isolation and resource allocation [73]. Furthermore, the assumption of reliability in RAN components may not hold in real-world scenarios, risking security breaches and network instability [125].

8.3 Strategies for Secure SDN Deployments

Securing SDN deployments necessitates a comprehensive approach incorporating advanced monitoring, robust authentication protocols, and strategic isolation techniques. Leveraging SDN controller programmability for flexible monitoring and dynamic threat response is pivotal, enabling real-time configuration adjustments to enhance resilience against various threats [126]. Optimizing network slice isolation is essential, particularly in multi-tenant environments, to balance security, cost, and performance, ensuring slices remain isolated and minimizing security breach impacts [53].

Integrating advanced security protocols, such as protocol dialecting, enhances OpenFlow protocol security through per-message authentication without modifying existing protocols [120]. Combining WireGuard with Open Source MANO (OSM) provides a secure communication framework addressing specific SDN security challenges [77]. The implementation of Quantum Key Distribution (QKD) with NFV orchestration over SDN-controlled optical networks secures network function image transmission, leveraging quantum cryptography to protect sensitive data [79]. Addressing vulnerabilities in programmable dataplanes requires systematic security analyses to identify threats and propose countermeasures, enhancing the overall security framework of SDN networks [92]. Continuous monitoring and proactive defense strategies are essential to counteract passive inference attacks [127].

Machine learning enhances network security through real-time data processing and predictive analytics, enabling threat anticipation and mitigation before they materialize [17]. Future research could focus on optimizing existing algorithms for efficiency and exploring their applications across diverse network scenarios. By integrating these strategies, network operators can enhance the security and robustness of SDN deployments, ensuring reliable operations in the face of evolving threats.

8.4 Threat Detection and Mitigation Techniques

In SDN environments, the dynamic architecture and centralized control present unique challenges for threat detection and mitigation. Effective strategies are crucial for safeguarding network operations against evolving threats. Integrating machine learning techniques enhances detection capabilities by analyzing traffic patterns and identifying anomalies indicative of security breaches. These techniques utilize real-time data processing for predictive analytics, enabling proactive threat mitigation [17].

Advanced monitoring frameworks, such as the SDN-based flexible on-the-fly monitoring system, facilitate real-time detection of network anomalies, allowing operators to adjust monitoring parameters dynamically [126]. Additionally, protocol dialecting in the OpenFlow protocol enhances security through per-message authentication, mitigating risks associated with unauthorized access and data manipulation [120].

Implementing Quantum Key Distribution (QKD) alongside NFV orchestration over SDN-controlled optical networks provides robust protection for sensitive data transmission, employing quantum cryptography to secure network function images and mitigate data interception threats [79]. Addressing vulnerabilities in programmable dataplanes, particularly those using P4, necessitates systematic security analyses to identify potential attack vectors and develop countermeasures. This proactive approach ensures secure network configurations against emerging threats, maintaining the integrity of SDN environments [92]. The potential for passive inference attacks underscores the need for continuous monitoring and robust defense mechanisms, enabling operators to counteract such threats effectively [127].

8.5 Future Directions in SDN Security

Future research in SDN security is poised to enhance the robustness and adaptability of network infrastructures across several key areas. Developing online adaptations of the Service Overlay Forest Dynamic Adaptation (SOFDA) framework to manage dynamic scenarios in SDN is promising, effectively addressing changing user demands and network conditions [128]. Additionally, focusing on dynamic resource allocation strategies will be crucial, particularly in integrating resource types beyond traditional bandwidth and computational power, thereby enhancing SDN deployment efficiency and scalability [129].

Developing detection mechanisms for inference attacks is another critical area, as these pose significant threats to SDN environments. Improved flow table management designs can mitigate vulnera-

bilities associated with such attacks, enhancing SDN security frameworks [127]. The integration of advanced cryptographic techniques, including multi-hop Quantum Key Distribution (QKD), offers avenues for secure communications over longer distances, reinforcing NFV deployment security [79].

Collectively, these research directions underscore the necessity for ongoing innovation in SDN security, emphasizing the importance of adaptability to meet diverse application demands, optimizing resource allocation for efficient performance, and developing advanced threat detection mechanisms leveraging machine learning and intelligent microservices. By addressing these challenges, future developments can ensure that SDN environments remain secure, resilient, and capable of meeting the demands of evolving network landscapes [103, 15, 41, 126].

9 Conclusion

The exploration of advanced concepts within Software-Defined Networking (SDN) and Network Function Virtualization (NFV) has underscored their pivotal role in redefining modern networking paradigms. By decoupling control and data planes, exemplified by OpenFlow, these technologies have significantly enhanced network programmability and scalability. This transformation is further facilitated by programming languages like P4, which enable extensive customization and efficient resource management, crucial for the demands of next-generation networks. However, the centralized nature of SDN poses security challenges, such as topology poisoning, necessitating robust detection mechanisms to safeguard network integrity. The integration of machine learning into network slicing, exemplified by architectures like SFI2, offers promising avenues for improving efficiency and security. Content-based routing frameworks like ContentFlow demonstrate potential in optimizing content delivery, while methods such as Fast Failover enhance multicast session reliability, vital for streaming services. Incorporating Information-Centric Networking within SDN, as seen in the CONET framework, further advances content delivery efficiency. Future research should focus on refining predictive analytics through machine learning, optimizing network slicing, and enhancing resource utilization in hybrid cloud environments, thereby driving the evolution of SDN and NFV technologies.

References

[1] Swarna Bindu Chetty, Avishek Nag, Ahmed Al-Tahmeesschi, Qiao Wang, Berk Canberk, Johann Marquez-Barja, and Hamed Ahmadi. Optimized resource allocation for cloud-native 6g networks: Zero-touch ml models in microservices-based vnf deployments, 2024.

[2] Mohamed Lamine Lamali, Nasreddine Fergani, Johanne Cohen, and Hélia Pouyllau. Path computation in multi-layer networks: Complexity and algorithms, 2016.

[3] Enio Kaljic, Almir Maric, Pamela Njemcevic, and Mesud Hadzialic. A survey on data plane flexibility and programmability in software-defined networking, 2019.

[4] Leandro C. de Almeida, Paulo Ditarso Maciel Jr au2, and Fábio L. Verdi. Cloud network slicing: A systematic mapping study from scientific publications, 2020.

[5] Kévin Phemius, Mathieu Bouet, and Jérémie Leguay. Disco: Distributed multi-domain sdn controllers, 2013.

[6] Sasindu Wijeratne, Ashen Ekanayake, Sandaruwan Jayaweera, Danuka Ravishan, and Ajith Pasqual. Scalable high performance sdn switch architecture on fpga for core networks, 2019.

[7] Alcardo Alex Barakabitze, Nabajeet Barman, Arslan Ahmad, Saman Zadtootaghaj, Lingfen Sun, Maria G. Martini, and Luigi Atzori. Qoe management of multimedia streaming services in future networks: A tutorial and survey, 2019.

[8] Adel Nadjaran Toosi, Redowan Mahmud, Qinghua Chi, and Rajkumar Buyya. Management and orchestration of network slices in 5g, fog, edge and clouds, 2018.

[9] Frank Dürr, Thomas Kohler, Jonas Grunert, and Andre Kutzleb. Zerosdn: A message bus for flexible and light-weight network control distribution in sdn, 2016.

[10] Ta-Hao Ting, Tsung-Nan Lin, Shan-Hsiang Shen, and Yu-Wei Chang. Guidelines for 5g end to end architecture and security issues, 2019.

[11] Jorge López, Charalampos Chatzinakis, Marc Cartigny, and Claude Poletti. Software defined networking flow admission and routing under minimal security constraints, 2023.

[12] Mingming Chen, Thomas La Porta, Teryl Taylor, Frederico Araujo, and Trent Jaeger. Manipulating openflow link discovery packet forwarding for topology poisoning, 2024.

[13] Jorik Oostenbrink, Niels L. M. van Adrichem, and Fernando A. Kuipers. Fast failover of multicast sessions in software-defined networks, 2017.

[14] Garegin Grigoryan, Yaoqing Liu, Laurent Njilla, Charles Kamhoua, and Kevin Kwiat. Enabling cooperative iot security via software defined networks (sdn), 2018.

[15] Abhishek Chanda and Cedric Westphal. Contentflow: Mapping content to flows in software defined networks, 2013.

[16] Ronghua Xu, Yu Chen, Xiaohua Li, and Erik Blasch. A secure dynamic edge resource federation architecture for cross-domain iot systems, 2022.

[17] Mujahid Sultan, Dodi Imbuido, Kam Patel, James MacDonald, and Kumar Ratnam. Designing knowledge plane to optimize leaf and spine data center, 2020.

[18] Mohammad Mahdi Tajiki, Behzad Akbari, Nader Mokari, and Luca Chiaraviglio. Sdn-based resource allocation in mpls networks: A hybrid approach, 2018.

[19] Muxi Yan, Jasson Casey, Prithviraj Shome, Alex Sprintson, and Andrew Sutton. Ætherflow: Principled wireless support in sdn, 2015.

[20] Ali Esmaeily and Katina Kralevska. Small-scale 5g testbeds for network slicing deployment: A systematic review, 2021.

[21] MD Samiul Islam, Mojammel Hossain, and Mohammed AlMukhtar. A survey on sdn & sdcn traffic measurement: Existing approaches and research challenge, 2022.

[22] George Petropoulos, Konstantinos V. Katsaros, and Maria-Evgenia Xezonaki. Openflow-compliant topology management for sdn-enabled information centric networks, 2017.

[23] Stefano Salsano, Nicola Blefari-Melazzi, Andrea Detti, Giacomo Morabito, and Luca Veltri. Information centric networking over sdn and openflow: Architectural aspects and experiments on the ofelia testbed, 2013.

[24] El Hocine Bouzidi. *Data-driven Dynamic Optimization of Traffic Workload and Network Slicing for 5G Networks and beyond*. PhD thesis, Université Gustave Eiffel, 2021.

[25] Ravishankar Ravindran, Asit Chakraborti, Syed Obaid Amin, Aytac Azgin, and Guoqiang Wang. 5g-icn : Delivering icn services over 5g using network slicing, 2016.

[26] Ermin Sakic and Wolfgang Kellerer. Response time and availability study of raft consensus in distributed sdn control plane, 2019.

[27] Sandra Scott-Hayward. Trailing the snail: Sdn controller security evolution, 2017.

[28] Gereltsetseg Altangerel, Tugsjargal Chuluuntsetseg, and Dashdorj Yamkhin. Performance analysis of sdn controllers: Pox, floodlight and opendaylight, 2021.

[29] Heng Cui, Ghassan O. Karame, Felix Klaedtke, and Roberto Bifulco. Fingerprinting software-defined networks, 2015.

[30] Saber Mirzaei, Sanaz Bahargam, Richard Skowyra, Assaf Kfoury, and Azer Bestavros. Using alloy to formally model and reason about an openflow network switch, 2016.

[31] Kashif Mahmood, Ameen Chilwan, Olav N. Østerbø, and Michael Jarschel. On the modeling of openflow-based sdns: The single node case, 2014.

[32] Mina Tahmasbi Arashloo, Yaron Koral, Michael Greenberg, Jennifer Rexford, and David Walker. Snap: Stateful network-wide abstractions for packet processing, 2016.

[33] Danish Sattar and Ashraf Matrawy. An empirical model of packet processing delay of the open vswitch, 2017.

[34] Carmelo Cascone, Luca Pollini, Davide Sanvito, and Antonio Capone. Traffic management applications for stateful sdn data plane, 2015.

[35] Abdelhadi Azzouni, Raouf Boutaba, and Guy Pujolle. Neuroute: Predictive dynamic routing for software-defined networks, 2017.

[36] Sebastián Gómez Macías, Luciano Paschoal Gaspary, and Juan Felipe Botero. Oracle: Collaboration of data and control planes to detect ddos attacks, 2020.

[37] Yunfei Meng, Changbo Ke, Zhiqiu Huang, Guohua Shen, Chunming Liu, and Xiaojie Feng. A practical runtime security policy transformation framework for software defined networks, 2023.

[38] José Suárez-Varela and Pere Barlet-Ros. Reinventing netflow for openflow software-defined networks, 2017.

[39] Abdelhadi Azzouni, Othmen Braham, Nguyen Thi Mai Trang, Guy Pujolle, and Raouf Boutaba. Fingerprinting openflow controllers: The first step to attack an sdn control plane, 2017.

[40] Igor Buzhin, Veronica Antonova, Yury Mironov, Vladislav Gnezdilov, Eldar Gaifutdinov, and Mikhail Gorodnichev. An information security monitoring and management system for 5g and 6g networks based on sdn/nfv, 2022.

[41] Andreas Alfred Blenk. *Towards virtualization of software-defined networks: Analysis, modeling, and optimization*. PhD thesis, Technische Universität München, 2018.

[42] Theo Kanter, Rahim Rahmani, and Arif Mahmud. Conceptual framework for internet of things' virtualization via openflow in context-aware networks, 2014.

[43] Mehdi Mohammadi, Ala Al-Fuqaha, and Zijiang James Yang. A high-level rule-based language for software defined network programming based on openflow, 2017.

[44] Arsany Basta, Andreas Blenk, Wolfgang Kellerer, and Stefan Schmid. Logically isolated, actually unpredictable? measuring hypervisor performance in multi-tenant sdns, 2017.

[45] Ashkan Aghdai, Yang Xu, and H. Jonathan Chao. Design of a hybrid modular switch, 2017.

[46] Hamidreza Almasi and Hossein Ajorloo. A framework for application-aware networking by delegating traffic management of sdns, 2017.

[47] Meihui Gao, Bernardetta Addis, Mathieu Bouet, and Stefano Secci. Optimal orchestration of virtual network functions, 2017.

[48] Sahar Ammar, Chun Pong Lau, and Basem Shihada. An in-depth survey on virtualization technologies in 6g integrated terrestrial and non-terrestrial networks, 2023.

[49] András Faragó. Optimization of virtual networks, 2020.

[50] Ning Zhang, Peng Yang, Shan Zhang, Dajiang Chen, Weihua Zhuang, Ben Liang, Xuemin, and Shen. Software defined networking enabled wireless network virtualization: Challenges and solutions, 2017.

[51] Debobroto Das Robin and Javed I. Khan. Toward an abstract model of programmable data plane devices, 2020.

[52] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf. On multi-domain network slicing orchestration architecture federated resource control, 2022.

[53] Stan Wong, Bin Han, and Hans D. Schotten. 5g network slice isolation, 2022.

[54] Rodrigo Moreira, Flavio de Oliveira Silva, Tereza Cristina Melo de Brito Carvalho, and Joberto S. B. Martins. Intelligent data-driven architectural features orchestration for network slicing, 2024.

[55] C. Jasson Casey, Andrew Sutton, and Alex Sprintson. tinynbi: Distilling an api from essential openflow abstractions, 2014.

[56] Nguyen Minh Tri, Masahiro Shibata, and Masato Tsuru. Effective route scheme of multicast probing to locate high-loss links in openflow networks, 2020.

[57] Yanbiao Li, Neng Ren, Xin Wang, Yuxuan Chen, Xinyi Zhang, Lingbo Guo, and Gaogang Xie. Tuplechain: Fast lookup of openflow table with multifaceted scalability, 2024.

[58] German Sviridov, Marco Bonola, Angelo Tulumello, Paolo Giaccone, Andrea Bianco, and Giuseppe Bianchi. Local decisions on replicated states (loader) in programmable data planes: programming abstraction and experimental evaluation, 2020.

[59] Ermin Sakic and Wolfgang Kellerer. Impact of adaptive consistency on distributed sdn applications: An empirical study, 2019.

[60] Yiheng Su, Ting Peng, Xiaoxun Zhong, and Lianming Zhang. Matching model of flow table for networked big data, 2020.

[61] Stefano Salsano, Giuseppe Siracusano, Andrea Detti, Claudio Pisa, Pier Luigi Ventre, and Nicola Blefari-Melazzi. Controller selection in a wireless mesh sdn under network partitioning and merging scenarios, 2014.

[62] Sanjeev Singh and Rakesh Kumar Jha. A survey on software defined networking: Architecture for next generation network, 2020.

[63] Ankur Chowdhary and Dijiang Huang. Sdn based network function parallelism in cloud, 2018.

[64] Abdelhadi Azzouni, Nguyen Thi Mai Trang, Raouf Boutaba, and Guy Pujolle. Limitations of openflow topology discovery protocol, 2017.

[65] Emeka Obiodu and Nishanth Sastry. From atm to mpls and qci: The evolution of differentiated qos standards and implications for 5g network slicing, 2020.

[66] Mohammad Asif Habibi, Bin Han, Faqir Zarrar Yousaf, and Hans D. Schotten. How should network slice instances be provided to multiple use cases of a single vertical industry?, 2021.

[67] Bin Han, Vincenzo Sciancalepore, Xavier Costa-Perez, Di Feng, and Hans D. Schotten. Multiservice-based network slicing orchestration with impatient tenants, 2020.

[68] Jose Jurandir Alves Esteves, Amina Boubendir, Fabice Guillemin, and Pierre Sens. Edge-enabled optimized network slicing in large scale networks, 2020.

[69] Sina Ebrahimi, Abulfazl Zakeri, Behzad Akbari, and Nader Mokari. Joint resource and admission management for slice-enabled networks, 2019.

[70] Jiefei Ma, Windhya Rankothge, Christian Makaya, Mariceli Morales, Frank Le, and Jorge Lobo. A comprehensive study on load balancers for vnf chains horizontal scaling, 2018.

[71] Nguyen Phuc Tran, Oscar Delgado, and Brigitte Jaumard. Proactive service assurance in 5g and b5g networks: A closed-loop algorithm for end-to-end network slicing, 2024.

[72] Anders Ellersgaard Kalør, René Guillaume, Jimmy Jessen Nielsen, Andreas Mueller, and Petar Popovski. Network slicing for ultra-reliable low latency communication in industry 4.0 scenarios, 2017.

[73] Joberto S. B. Martins, Tereza C. Carvalho, Rodrigo Moreira, Cristiano Both, Adnei Donatti, João H. Corrêa, José A. Suruagy, Sand L. Corrêa, Antonio J. G. Abelem, Moisés R. N. Ribeiro, Jose-Marcos Nogueira, Luiz C. S. Magalhães, Juliano Wickboldt, Tiago Ferreto, Ricardo Mello, Rafael Pasquini, Marcos Schwarz, Leobino N. Sampaio, Daniel F. Macedo, José F. de Rezende, Kleber V. Cardoso, and Flávio O. Silva. Enhancing network slicing architectures with machine learning, security, sustainability and experimental networks integration, 2023.

[74] Antonio De Domenico, Ya-Feng Liu, and Wei Yu. Optimal virtual network function deployment for 5g network slicing in a hybrid cloud infrastructure, 2020.

[75] Wei-Kun Chen, Ya-Feng Liu, Antonio De Domenico, Zhi-Quan Luo, and Yu-Hong Dai. Optimal network slicing for service-oriented networks with flexible routing and guaranteed e2e latency, 2021.

[76] Danish Sattar and Ashraf Matrawy. Towards secure slicing: Using slice isolation to mitigate ddos attacks on 5g core network slices, 2019.

[77] Simen Haga, Ali Esmaeily, Katina Kralevska, and Danilo Gligoroski. 5g network slice isolation with wireguard and open source mano: A vpnaas proof-of-concept, 2020.

[78] Qiang Liu, Tao Han, and Nirwan Ansari. Learning-assisted secure end-to-end network slicing for cyber-physical systems, 2019.

[79] Alejandro Aguado, Emilio Hugues-Salas, Paul Anthony Haigh, Jaume Marhuenda, Alasdair B. Price, Philip Sibson, Jake E. Kennard, Christopher Erven, John G. Rarity, Mark G. Thompson, Andrew Lord, Reza Nejabati, and Dimitra Simeonidou. First experimental demonstration of secure nfv orchestration over an sdn-controlled optical network with time-shared quantum key distribution resources, 2016.

[80] Panagiotis Promponas and Leandros Tassiulas. Network slicing: Market mechanism and competitive equilibria, 2023.

[81] Gil Einziger, Gabriel Scalosub, Carla Fabiana Chiasserini, and Francesco Malandrino. Virtual service embedding with time-varying load and provable guarantees, 2022.

[82] K. Antevski, J. Martín-Pérez, Nuria Molner, C. F. Chiasserini, F. Malandrino, P. Frangoudis, A. Ksentini, X. Li, J. SalvatLozano, R. Martínez, I. Pascual, J. Mangues-Bafalluy, J. Baranda, B. Martini, and M. Gharbaoui. Resource orchestration of 5g transport networks for vertical industries, 2018.

[83] Youssouf Drif, Emmanuel Chaput, Emmanuel Lavinal, Pascal Berthou, Boris Tiomela Jou, Olivier Gremillet, and Fabrice Arnal. An extensible network slicing framework for satellite integration into 5g, 2020.

[84] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker. Network slicing to enable scalability and flexibility in 5g mobile networks, 2017.

[85] Ali J. Ramadhan. T-s3ra: traffic-aware scheduling for secure slicing and resource allocation in sdn/nfv enabled 5g networks, 2021.

[86] Ali Esmaeily, Katina Kralevska, and Danilo Gligoroski. A cloud-based sdn/nfv testbed for end-to-end network slicing in 4g/5g, 2020.

[87] Eduardo S. Xavier, Nazim Agoulmine, and Joberto S. B. Martins. On modeling network slicing communication resources with sarsa optimization, 2023.

[88] Eliseu Silva Torres, Rafael F. Reale, Leobino N. Sampaio, and Joberto S. B. Martins. A sdn/openflow framework for dynamic resource allocation based on bandwidth allocation model, 2021.

[89] Pat Bosshart, Dan Daly, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. Programming protocol-independent packet processors, 2014.

[90] Chen Tian, Alex X. Liu, Ali Munir, Jie Yang, and Yangming Zhao. Openfunction: Data plane abstraction for software-defined middleboxes, 2016.

[91] C. Jasson Casey, Andrew Sutton, Gabriel Dos Reis, and Alex Sprintson. Eliminating network protocol vulnerabilities through abstraction and systems language design, 2013.

[92] Andrei-Alexandru Agape, Madalin Claudiu Danceanu, Rene Rydhof Hansen, and Stefan Schmid. Charting the security landscape of programmable dataplanes, 2018.

[93] Jens Kanstrup Larsen, Roberto Guanciale, Philipp Haller, and Alceste Scalas. P4r-type: a verified api for p4 control plane programs (technical report), 2023.

[94] Jonathan Vestin, Andreas Kassler, and Johan Åkerberg. Fastreact: In-network control and caching for industrial control networks using programmable data planes, 2018.

[95] Elie F. Kfoury, Jorge Crichigno, and Elias Bou-Harb. An exhaustive survey on p4 programmable data plane switches: Taxonomy, applications, challenges, and future trends, 2021.

[96] Pedro A. Aranda Gutierrez, Roberto Doriguzzi-Corin, and Elisa Rojas. Lessons learnt from the netide project: Taking sdn programming to the next level, 2017.

[97] Moreno Ambrosin, Mauro Conti, Fabio De Gaspari, and Radha Poovendran. Lineswitch: Efficiently managing switch flow in software-defined networking while effectively tackling dos attacks, 2015.

[98] Tal Mizrahi and Yoram Moses. Time4: Time for sdn, 2016.

[99] Volkan Yazici, M. Oguz Sunay, and Ali O. Ercan. Controlling a software-defined network via distributed controllers, 2014.

[100] Niels L. M. van Adrichem, Farabi Iqbal, and Fernando A. Kuipers. Computing backup forwarding rules in software-defined networks, 2016.

[101] An Xie, Xiaoliang Wang, Guido Maier, and Sanglu Lu. Designing a disaster-resilient network with software defined networking, 2016.

[102] Mohammad Sajid Shahriar, Faisal Ahmed, Genshe Chen, Khanh D. Pham, Suresh Subramaniam, Motoharu Matsuura, Hiroshi Hasegawa, and Shih-Chun Lin. Prioritized multi-tenant traffic engineering for dynamic qos provisioning in autonomous sdn-openflow edge networks, 2024.

[103] Rodrigo Moreira, Rodolfo S. Villaca, Moises R. N. Ribeiro, Joberto S. B. Martins, Joao Henrique Correa, Tereza C. Carvalho, and Flavio de Oliveira Silva. An intelligent native network slicing security architecture empowered by federated learning, 2024.

[104] Xueli An, Riccardo Trivisonno, Hans Einsiedler, Dirk von Hugo, Kay Haensge, Xiaofeng Huang, Qing Shen, Daniel Corujo, Kashif Mahmood, Dirk Trossen, Marco Liebsch, Filipe Leitao, Cao-Thanh Phan, and Frederic Klamm. End-to-end architecture modularisation and slicing for next generation networks, 2016.

[105] Antonio Capone, Carmelo Cascone, Alessandro Q. T. Nguyen, and Brunilde Sansò. Detour planning for fast and reliable failure recovery in sdn with openstate, 2015.

[106] Ernesto Abarca, Johannes Grassler, Gregor Schaffrath, and Stefan Schmid. A federated cloudnet architecture: The pip and the vnp role, 2013.

[107] Anees Al-Najjar, Furqan Hameed Khan, and Marius Portmann. Network traffic control for multi-homed end-hosts via sdn, 2020.

[108] S. H. Warraich, Z. Aziz, H. Khurshid, R. Hameed, A. Saboor, and M. Awais. Sdn enabled and openflow compatible network performance monitoring system, 2020.

[109] Mathieu Leconte, Georgios S Paschos, Panayotis Mertikopoulos, and Ulaş C Kozat. A resource allocation framework for network slicing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 2177–2185. IEEE, 2018.

[110] Giuseppe Bianchi, Marco Bonola, Antonio Capone, Carmelo Cascone, and Salvatore Pontarelli. Towards wire-speed platform-agnostic control of openflow switches, 2014.

[111] Yingying Guan, Qingyang Song, Weijing Qi, Ke Li, Lei Guo, and Abbas Jamalipour. Multidimensional resource fragmentation-aware virtual network embedding in mec systems interconnected by metro optical networks, 2023.

[112] Bin Han, Di Feng, Lianghai Ji, and Hans D. Schotten. A profit-maximizing strategy of network resource management for 5g tenant slices, 2017.

[113] Abdulhalim Dandoush, Viswanath Kumarskandpriya, Mueen Uddin, and Usman Khalil. Large language models meet network slicing management and orchestration, 2024.

[114] Jose Ordonez-Lucena, Oscar Adamuz-Hinojosa, Pablo Ameigeiras, Pablo Muñoz, Juan J. Ramos-Muñoz, Jesús Folgueira Chavarria, and Diego Lopez. The creation phase in network slicing: From a service order to an operative network slice, 2018.

[115] Tu N. Nguyen, Kashyab J. Ambarani, and My T. Thai. Optimizing resource allocation and vnf embedding in ran slicing, 2022.

[116] Dinh Thai Hoang, Dusit Niyato, Ping Wang, Antonio De Domenico, and Emilio Calvanese Strinati. Optimal cross slice orchestration for 5g mobile services, 2017.

[117] Xinyu Huang, Haojun Yang, Shisheng Hu, and Xuemin Shen. Digital twin-driven network architecture for video streaming, 2024.

[118] Ankur Mudgal, Abhishek Verma, Munesh Singh, Kshira Sagar Sahoo, Erik Elmroth, and Monowar Bhuyan. Flora: Flow table low-rate overflow reconnaissance and detection in sdn, 2024.

[119] Sarwan Ali, Maria Khalid Alvi, Safi Faizullah, Muhammad Asad Khan, Abdullah Alshanqiti, and Imdadullah Khan. Detecting ddos attack on sdn due to vulnerabilities in openflow, 2020.

[120] Michael Sjoholmsierchio, Britta Hale, Daniel Lukaszewski, and Geoffrey G. Xie. Strengthening sdn security: Protocol dialecting and downgrade attacks, 2020.

[121] Arash Shaghaghi, Salil S. Kanhere, Mohamed Ali Kaafar, and Sanjay Jha. Gwardar: Towards protecting a software-defined network from malicious network operating systems, 2018.

[122] Qiang Liu, Nakjung Choi, and Tao Han. Atlas: Automate online service configuration in network slicing, 2022.

[123] Liron Schiff, Kashyap Thimmaraju, and Stefan Schmid. Routing-verification-as-a-service (rvaas): Trustworthy routing despite insecure providers, 2016.

[124] Yanni Ou, Matthew Davis, Alejandro Aguado, Fanchao Meng, Reza Nejabati, and Dimitra Simeonidou. Optical network virtualisation using multi-technology monitoring and sdn-enabled optical transceiver, 2018.

[125] Cyril Shih-Huan Hsu, Danny De Vleeschauwer, and Chrysa Papagianni. Sla decomposition for network slicing: A deep neural network approach, 2024.

[126] Maxli Campos and Joberto Martins. A sdn-based flexible system for on-the-fly monitoring and treatment of security events, 2018.

[127] Junyuan Leng, Yadong Zhou, Junjie Zhang, and Chengchen Hu. An inference attack model for flow table capacity and usage: Exploiting the vulnerability of flow table overflow in software-defined network, 2015.

[128] Jian-Jhih Kuo, Shan-Hsiang Shen, Ming-Hong Yang, De-Nian Yang, Ming-Jer Tsai, and Wen-Tsuen Chen. Service overlay forest embedding for software-defined cloud networks, 2017.

[129] Yingyu Li, Anqi Huang, Yong Xiao, Xiaohu Ge, Sumei Sun, and Han-Chieh Chao. Federated orchestration for network slicing of bandwidth and computational resource, 2020.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.