Architectural Strategies for Implementing and Automating Service Function Chaining (SFC) in Multi-Cloud Environments

ARUNKUMAR VELAYUTHAM [1]

[1] Cloud Software Development Engineer and Technical Lead at Intel, Arizona, USA

Published: 2020

Abstract

Service Function Chaining (SFC) represents a paradigm shift in the deployment, management, and automation of network services, enabling a dynamic approach to connecting virtual network functions (VNFs) in a prescribed sequence. Businesses adopting multi-cloud environments to leverage different cloud providers face significant challenges in implementing and automating SFC across these distributed infrastructures. These include managing the complexities of orchestrating SFC across heterogeneous cloud platforms, ensuring consistent performance, optimizing resource allocation, and maintaining security. This paper explores architectural strategies for implementing and automating SFC in multi-cloud environments, focusing on optimizing deployment, orchestration, and scalability. It examines the components and frameworks required to achieve seamless SFC automation, such as Network Function Virtualization (NFV), Software-Defined Networking (SDN), and cloud-native technologies like Kubernetes. The paper also discusses the importance of policy-driven orchestration, dynamic scaling, and integration of AI/ML techniques for performance optimization. This research also proposes the use of cross-layer coordination and programmable data planes to enhance SFC deployment in multi-cloud environments. The goal of the paper is to demonstrate how to create a robust and adaptive SFC architecture that efficiently operates in multi-cloud setups for enhancing service delivery, reducing operational costs, and improving network agility.

1. INTRODUCTION

In recent years, there has been a growing trend to virtualize network, storage, and computational resources. This shift is driven by the need for more flexible, scalable, and cost-effective solutions to manage and operate modern networks. However, despite the virtualization of these resources, the underlying network infrastructure remains largely managed through physical means. Network operators are facing increasing challenges in meeting the growing demands of users and network traffic on traditional network architectures. While end users continuously demand lower costs for data services, Internet Service Providers (ISPs) are encountering rising capital expenditures (CAPEX) and operational expenditures (OPEX) associated with maintaining and expanding increasingly complex network infrastructures. The surge in mobile devices and the emergence of novel networking paradigms, such as the Internet of Things (IoT), have significantly increased user traffic, adding further complexity to these networks. This rapid increase in traffic and service demands is pushing traditional network models to their limits.

Network Function Virtualization (NFV) has emerged as a potential solution to these challenges by allowing network services to be deployed as virtualized services. NFV enables network operators to run these services on virtualized hardware rather than relying on specialized, proprietary equipment. This reduces costs and increases flexibility, allowing services to be dynamically deployed and scaled to meet varying demands. With NFV, operators can reduce the reliance on costly hardware-based solutions and instead implement essential network functions as software-based services. These network functions can include services like firewalls, load balancers, intrusion detection systems, and packet inspection tools, among others, which are typically implemented as virtualized network functions (VNFs) [1, 2] .

The rapid advancements in cloud computing have had a profound impact on the deployment and management of computational and storage resources. Cloud computing allows these resources to be distributed across geographically diverse areas, bringing them closer to end users and improving the performance and reliability of services. Application Service Providers (ASPs) can now deploy their services across multiple datacenters that are geographically dispersed, thus improving the overall user experience by reducing latency and enhancing redundancy. However, despite these advancements in cloud infrastructure, the traditional network model that underpins these services remains static and rigid, lacking the capabilities for dynamic auto-configuration and real-time adaptability. This inflexibility

Table 1. Key Challenges in Traditional Network Models

Challenge	Cause
Dependence on Physical Topology	Networks are tightly coupled with physical layouts, limiting flexibility and scalability.
Manual Service Provisioning	Adding or updating services requires manual intervention, which is time-consuming and error-prone.
Static Path Provisioning	Static routes are pre-defined, making it difficult to adapt to changing traffic patterns.
Complex Load Balancing	Load balancing requires manual configuration, leading to inefficiencies and errors.
High CAPEX and OPEX	Rising costs associated with maintaining and expanding network infrastructures.
Limited Scalability and Flexibility	Traditional networks struggle to meet the dynamic needs of modern traffic demands.

poses several challenges for network operators as they attempt to scale their networks to meet increasing demands [3].

The traditional ISP network model suffers from several limitations, including dependence on physical topology, the need for manual intervention in adding or updating services, and the reliance on static path provisioning. Furthermore, load balancing within traditional networks often requires manual configuration, which can be time-consuming and prone to error. These limitations make it difficult for ISPs to respond quickly to changing traffic patterns and user demands. However, emerging technologies such as Software Defined Networking (SDN) and NFV offer new tools to help address these challenges. SDN, in particular, provides a more flexible and programmable network architecture by decoupling the control plane from the data plane. This allows for more efficient traffic management, dynamic routing, and simplified network configuration, which can improve both performance and scalability.

NFV and SDN are highly complementary technologies, both of which are driving the evolution of modern network management. While NFV focuses on the virtualization of network functions, SDN centralizes the control of network traffic, enabling more precise and efficient forwarding of data. The combination of these two technologies allows ISPs to deploy virtualized network services over flexible, programmable infrastructures that can be dynamically adjusted to meet changing demands. This approach significantly reduces the time and resources required to deploy new services, update existing services, or manage network traffic in real-time.

One key development in this area is the concept of Service Function Chaining (SFC), which defines an ordered sequence of network functions to be applied to specific traffic flows. SFC allows network operators to specify the exact sequence of processing steps that traffic must pass through, such as firewalls, load balancers, and proxies. This level of control enables operators to optimize the delivery of services and ensure that traffic is processed efficiently and securely. The SFC architecture is made up of several key components that work together to manage and orchestrate service chains. These components include service classifiers, which categorize traffic flows; service encapsulation mechanisms, which define how traffic is processed; and service function paths, which specify the sequence of functions to be applied.

The orchestration of service chains is a critical aspect of SFC. Orchestration involves defining the service chains and constructing the necessary service paths to ensure that traffic is routed through the correct sequence of functions. This process requires coordination between multiple components, including control and policy planes, which define the rules and policies governing the behavior of the service chains. These policies dictate which network functions should be applied to specific types of traffic and ensure that the service chains comply with predefined service-level agreements (SLAs) and regulatory requirements.

The data plane in SFC handles the actual forwarding of traffic through the service chains. It ensures that traffic is steered through the correct sequence of service functions in accordance with the rules set by the control plane. This process requires real-time traffic steering and the use of metadata to guide traffic through the appropriate paths. The data plane architecture is often accessible through open APIs, allowing third-party tools and services to integrate with the system and enhance its capabilities. This open architecture approach fosters innovation and allows for greater customization of service chains.

The deployment and management of service chains involve several steps, beginning with the instantiation of the service functions themselves. Service chains are instantiated by defining an ordered set of virtualized service functions that correspond to specific processing requirements. These service functions are then deployed dynamically based on the needs of the network, allowing for flexible and on-demand service provisioning. The service chain orchestration system selects the appropriate service functions, defines the service paths, and configures the necessary control and data plane mechanisms to direct traffic through the chain. This dynamic approach to service deployment reduces the need for manual configuration and allows for faster, more efficient service delivery.

Once the service chains are operational, the system continuously monitors traffic flows and adjusts the service chains as needed to maintain performance and meet SLAs. The control plane ensures that traffic is routed through the correct sequence of functions and that the service chains operate as intended. In addition, the control plane can make real-time adjustments to service paths in response to changing network conditions, ensuring optimal performance and resource utilization. This automated approach to network management allows operators to quickly respond to changes in traffic patterns and service demands, improving the overall efficiency and reliability of the network.

2. SERVICE FUNCTION CHAINING (SFC) ARCHITECTURE

The primary architectural concept of Service Function Chaining (SFC) is the clear separation between the logical SFC overlay and the data plane. As defined in RFC 7665, which outlines the specification of an SFC architecture, this separation ensures

Table 2. Benefits of Network Function Virtualization (NFV)

Benefit	Detail
Reduced Costs	Decreases reliance on specialized hardware, lowering CAPEX and OPEX.
Increased Flexibility	Allows dynamic deployment and scaling of network services.
Improved Service Agility	Enables faster deployment of new services and updates to existing ones.
Hardware Independence	Services can run on generic, virtualized hardware rather than specialized, proprietary equipment.
Enhanced Scalability	VNFs can be scaled up or down based on demand, improving resource utilization.
Simplified Network Management	Centralized management of virtualized services simplifies network operations.

Table 3. Comparison of NFV and SDN Technologies

Aspect	NFV	SDN
Focus	Virtualization of network functions	Centralization of network control
Primary Objective	Replace hardware functions with software	Decouple control plane from data plane
Key Benefit	Cost reduction, increased flexibility	Programmable, dynamic traffic management
Typical Applications	Firewalls, load balancers, intrusion detection	Dynamic routing, efficient traffic management
Complementarity	Runs VNFs on SDN-managed infrastructures	Provides flexible, programmable paths for VNFs [4]

Table 4. Components of Service Function Chaining (SFC)

Component	Detail
Service Classifiers	Categorize traffic flows for appropriate processing.
Service Encapsulation	Defines how traffic is processed through the service chain [5].
Service Function Paths	Specifies the sequence of network functions applied to traffic.
Control Plane	Defines rules and policies for traffic routing through service chains.
Data Plane	Handles the forwarding of traffic according to control plane rules [6].
Orchestration System	Manages the deployment, configuration, and adjustment of service chains to ensure compliance with SLAs and dynamic network requirements [7].

that the operations related to packet handling are distinct from the realization of Service Function Paths (SFPs). In this context, SFPs are constructed as an abstraction of the packet handling operations, such as packet forwarding. One of the key benefits of this architecture is that it remains independent of the underlying network topology, meaning that any topological changes in the network do not affect the functioning of SFC operations. This abstraction allows SFC to maintain consistent performance and operational stability even when the physical network undergoes modifications.

The SFC architecture includes several components responsible for managing traffic along the SFP. These components include classifiers, service function forwarders, service function nodes, and proxies when necessary. Each of these components plays a crucial role in ensuring that the SFC operates smoothly, allowing traffic to be efficiently directed through the defined sequence of service functions [8].

The classifier (CL) is a fundamental component in the SFC architecture, tasked with the initial classification of incoming traffic. This classification process enables the filtering of different traffic types based on predefined policy profiles. Once the traffic is classified at the ingress point, this classification result is used throughout the SFC to avoid reclassification at every subsequent element. Typically, Service Path Identifiers (SPIs) are inserted into packets to facilitate this process, ensuring that the next SFC element can process packets based on the initial classification decision. This approach streamlines the handling of packets through the service chain, reducing processing overhead and enhancing overall efficiency [9].

Another critical component of the SFC architecture is the Service Function Forwarder (SFF). The SFF is responsible for managing the forwarding of traffic between different SFC components over the SFC overlay. Forwarding decisions within the SFC are made based on traffic flow identifiers or by matching

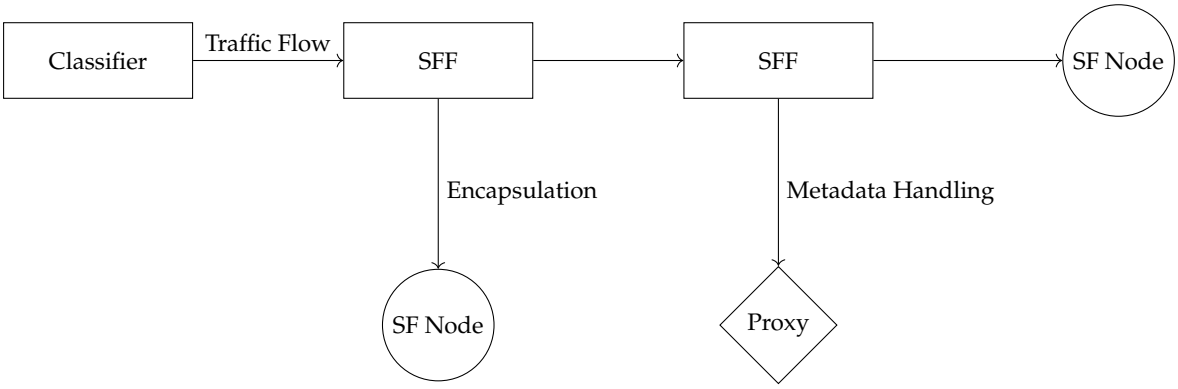

Fig. 1. Service Function Chaining (SFC) Architecture: Components and Data Flow

specific rules configured within the SFFs. These forwarding operations determine the next element in the SFP, guiding traffic through the appropriate sequence of service functions. In cases where additional information such as path identifiers or metadata needs to be associated with packets, an encapsulation process is utilized to maintain the connection and communication between SFC elements.

The SFC proxy serves as an intermediary between the SFF and service functions (SFs) that are not inherently SFC-aware. The role of the proxy is to enable communication within the SFC framework, integrating non-SFC-aware SFs into the service chain. The proxy accomplishes this by implementing necessary SFC functions, such as adding or consuming metadata on behalf of the SFs. This capability allows for the seamless incorporation of diverse service functions into the chain, regardless of their native SFC compatibility.

Service Function (SF) nodes are the locations where the actual service functions are deployed. These nodes can be implemented in various forms, including as physical hardware devices or virtualized nodes within a cloud or data center environment. A single SF node may host one or more service functions, providing the flexibility to scale service capabilities as needed. This versatility ensures that SF nodes can be tailored to meet the specific performance and resource requirements of different service functions within the chain.

The SFC architecture's ability to abstract and separate the logical operations of service chaining from the physical network infrastructure provides significant operational benefits. By decoupling service paths from the underlying network topology, SFC enables dynamic and flexible service deployment, which is resilient to changes in the physical environment. This architectural approach enhances the agility of network operators, allowing them to rapidly adapt to evolving service demands and traffic patterns without the need for extensive reconfiguration of the physical network.

3. ARCHITECTURAL STRATEGIES FOR SFC IN MULTI-CLOUD ENVIRONMENTS

A. NFV and SDN for SFC Automation

Network Function Virtualization (NFV) and Software-Defined Networking (SDN) have become essential technologies for automating Service Function Chaining (SFC) in multi-cloud environments. These technologies enable network operators to decouple network functions from dedicated hardware and manage network traffic in a centralized and dynamic way. As networks evolve, the integration of NFV and SDN has become increasingly important for achieving agility, scalability, and cost-efficiency in service delivery. In a traditional network architecture, deploying and managing network services like firewalls, load balancers, and intrusion detection systems often require proprietary hardware and manual configuration. However, with NFV and SDN, these services can be deployed as software-based Virtual Network Functions (VNFs), and traffic can be routed dynamically through these services using a centralized control mechanism. The integration of these technologies facilitates the automation of SFC, reducing manual intervention and allowing for more responsive and flexible network management.

NFV plays a central role in this process by decoupling network functions from the physical hardware they traditionally run on. Instead of relying on proprietary hardware appliances, NFV allows these network functions to be deployed as software on general-purpose servers. This decoupling is especially beneficial in multi-cloud environments, where VNFs can be deployed across different cloud platforms, allowing for more distributed and flexible service deployment. By abstracting the underlying hardware, NFV enables network operators to provision, scale, and manage network services more easily, without the constraints of physical infrastructure. In a multi-cloud context, this ability to deploy VNFs across different clouds is valuable, as it allows operators to optimize resource usage and take advantage of the different capabilities and geographies of various cloud providers.

SDN complements NFV by providing centralized control over network traffic. In traditional networks, traffic routing is often governed by static rules embedded in individual network devices, making it difficult to respond dynamically to changing network conditions. SDN separates the control plane, which makes decisions about how traffic should flow, from the data plane, which handles the actual forwarding of traffic. This separation allows for centralized management of the network, enabling network administrators to define policies and dynamically steer traffic through the appropriate VNFs based on real-time network conditions and service requirements. SDN enables greater flexibility in how traffic is routed through the network and allows for more responsive and efficient management of data flows, especially in the context of SFC.

The combination of NFV and SDN is especially powerful for automating SFC in multi-cloud environments. SFC refers to the process of chaining together a sequence of network services—such as firewalls, load balancers, and intrusion detection systems—that traffic must pass through in order to meet specific

service requirements. Traditionally, configuring and managing these service chains required significant manual intervention, as operators needed to ensure that traffic was correctly routed through each service in the chain. However, by integrating NFV and SDN, this process can be automated. NFV allows the network functions themselves to be deployed as virtualized services, while SDN provides the mechanism for dynamically steering traffic through these services in the correct order.

In multi-cloud environments, where VNFs may be deployed across different cloud providers, the integration of NFV and SDN is useful. SDN controllers can dynamically adjust routing paths based on the availability and performance of VNFs deployed across different clouds. For example, if one VNF in a particular cloud becomes overloaded or experiences a failure, the SDN controller can automatically reroute traffic to a different VNF in another cloud, ensuring that the service chain continues to operate without interruption. This ability to dynamically adjust traffic flows based on real-time network conditions is a key benefit of integrating NFV and SDN, as it allows for more resilient and adaptive service delivery.

One of the core principles of SFC is the separation of the logical service chaining operations from the physical network infrastructure. This separation is made possible by the integration of NFV and SDN. With NFV, network functions are abstracted from the physical hardware, allowing them to be deployed and managed independently of the underlying infrastructure. SDN, on the other hand, abstracts the control of traffic flows from the individual network devices, allowing traffic to be managed centrally and dynamically. This separation of concerns allows for greater flexibility in how service chains are deployed and managed, as it decouples the logical service functions from the physical network topology.

The SDN controller plays a central role in this architecture by managing the traffic flows between VNFs. It continuously monitors the state of the network and makes real-time decisions about how traffic should be routed through the service chain. Based on predefined policies and real-time network conditions, the SDN controller can dynamically adjust routing paths to ensure that traffic flows through the appropriate VNFs in the correct order. This centralized control is essential for automating SFC, as it allows the network to adapt to changing conditions without requiring manual intervention.

One of the key challenges in automating SFC is ensuring that VNFs deployed across different cloud environments can interoperate seamlessly. This is where the integration of NFV and SDN becomes valuable. By abstracting both the network functions and the control of traffic flows, NFV and SDN enable VNFs deployed in different clouds to be managed as part of a single, cohesive service chain. The SDN controller can dynamically route traffic between VNFs in different clouds, based on factors such as performance, availability, and proximity to the end user. This ability to dynamically adjust service chains across multiple cloud environments allows for more efficient use of resources and better overall service performance.

In addition to enabling greater flexibility and scalability, the integration of NFV and SDN for SFC automation also helps reduce costs. By decoupling network functions from dedicated hardware, NFV allows operators to use general-purpose servers to deploy their network services, which can lead to significant cost savings compared to traditional hardware-based solutions. Additionally, the centralized control provided by SDN reduces the need for manual configuration and management of network devices, further lowering operational expenses. This combination of reduced capital expenditures (CAPEX) and operational expenditures (OPEX) makes NFV and SDN an attractive solution for network operators looking to improve efficiency and reduce costs.

Automation of SFC through NFV and SDN also improves the overall reliability and resilience of the network. Because VNFs can be deployed across multiple cloud environments, the network can automatically reroute traffic to avoid overloaded or failing VNFs, ensuring that service chains continue to operate even in the face of hardware failures or network congestion. This dynamic routing capability allows for more resilient service delivery and helps ensure that service-level agreements (SLAs) are consistently met, even under challenging conditions.

Security is another area where NFV and SDN can enhance the automation of SFC. By centralizing control of the network, SDN provides greater visibility into traffic flows and allows for more granular enforcement of security policies. For example, security functions such as firewalls and intrusion detection systems can be integrated into the service chain, and traffic can be dynamically steered through these functions based on predefined security policies. This centralized control makes it easier to enforce consistent security policies across the network and quickly respond to potential threats. Additionally, because VNFs are deployed as software, they can be updated or replaced more easily than hardware-based solutions, allowing for more timely responses to emerging security vulnerabilities.

The flexibility of NFV and SDN also allows for the deployment of custom service chains tailored to the specific needs of individual applications or users. For example, a service chain for a latency-sensitive application might include VNFs optimized for low-latency traffic processing, while a service chain for a security-sensitive application might include additional security functions such as deep packet inspection or encryption. By leveraging the flexibility of NFV and SDN, network operators can create custom service chains that meet the specific performance, security, and reliability requirements of different applications.

In multi-cloud environments, the ability to dynamically adjust service chains across different clouds is important. Different cloud providers offer different capabilities in terms of performance, cost, and geographic coverage. By integrating NFV and SDN, network operators can optimize their use of these cloud resources by dynamically routing traffic through VNFs deployed in the most suitable cloud environment for a given service. For example, traffic from users in a particular geographic region might be routed through VNFs deployed in a nearby cloud to reduce latency, while traffic for a less latency-sensitive application might be routed through a more cost-effective cloud provider. This ability to dynamically optimize service delivery across multiple clouds allows for more efficient use of resources and better overall service performance.

The integration of NFV and SDN for SFC automation also enables greater agility in deploying new services. In traditional networks, deploying a new service often requires significant time and effort to configure the necessary hardware and routing policies. However, with NFV and SDN, new services can be deployed as VNFs, and traffic can be dynamically routed through these services using the SDN controller. This automation significantly reduces the time and effort required to deploy new services, allowing network operators to respond more quickly to changing business needs or customer demands [10].

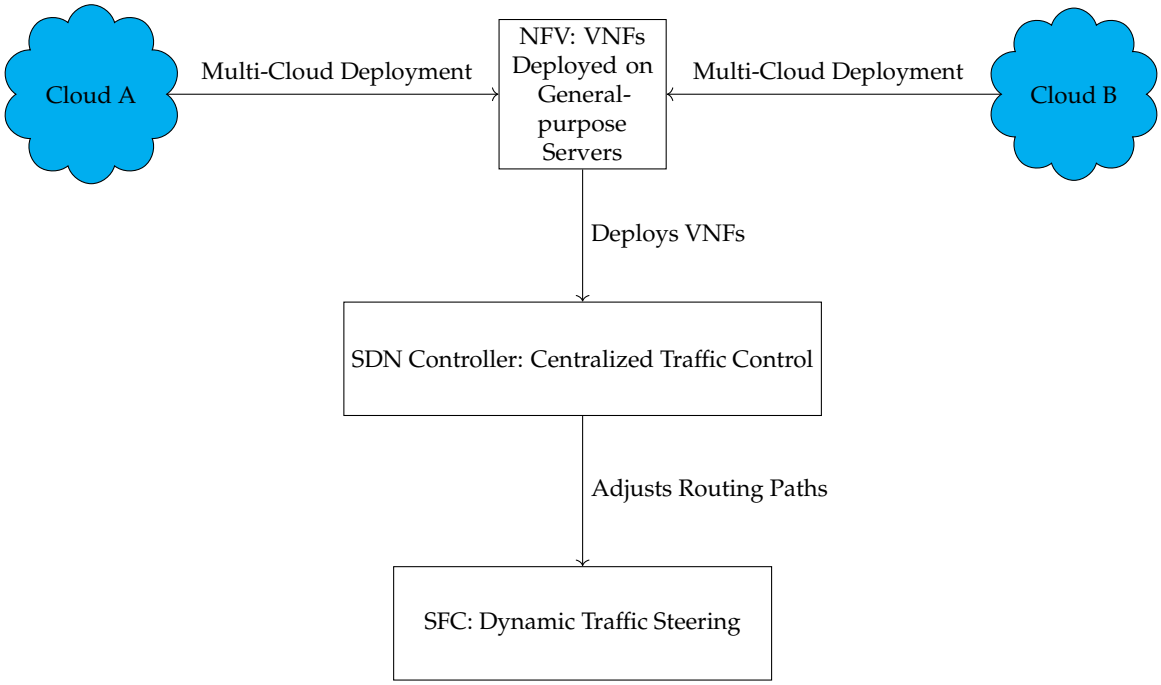

Fig. 2. Integration of NFV and SDN for SFC Automation in Multi-Cloud Environments

B. Cloud-Native Approaches: Kubernetes and Microservices

Cloud-native approaches, including the use of Kubernetes and microservices, are transforming how network services are deployed and managed in the context of Service Function Chaining (SFC). Kubernetes, a widely adopted container orchestration platform, plays a key role in managing cloud-native applications and services. Its extensibility and robust ecosystem make it an effective tool for orchestrating Virtual Network Functions (VNFs) in multi-cloud environments. Kubernetes supports features like service mesh and network policies, which are critical for managing the flow of traffic between different service functions within an SFC. By leveraging these features, Kubernetes can provide fine-grained control over how VNFs are deployed, scaled, and connected, enabling more efficient and dynamic service chaining [11].

Kubernetes allows VNFs to be packaged as containers, making them easier to deploy and manage across diverse cloud environments. This container-based approach abstracts the underlying infrastructure, enabling VNFs to run consistently regardless of the specific cloud provider or environment. Kubernetes handles the scheduling, scaling, and lifecycle management of these containerized VNFs, automating many of the tasks that would traditionally require manual intervention. This automation is valuable in multi-cloud scenarios, where VNFs may need to be deployed across multiple cloud platforms to optimize performance, cost, or compliance with geographic regulations.

Service mesh is a key feature of Kubernetes that enhances the management of microservices and VNFs within an SFC. A service mesh provides a dedicated infrastructure layer for managing service-to-service communication, enabling features like load balancing, traffic routing, and observability. This infrastructure layer operates independently of the application code, allowing for consistent and secure communication between VNFs without requiring changes to the VNFs themselves. Service meshes can also enforce network policies that dictate how traffic should be routed between different VNFs, supporting the requirements of SFCs by ensuring that traffic flows through the correct sequence of services.

The microservices architecture further enhances the flexibility and scalability of VNFs by decomposing them into smaller, independent components. This approach contrasts with traditional monolithic VNFs, which bundle multiple functions into a single, tightly coupled unit. By breaking VNFs into microservices, each function can be independently deployed, scaled, and managed, allowing for more granular control over service delivery. This modularity is beneficial in SFC implementations, where different services may need to be scaled or updated independently to respond to changing traffic conditions or performance requirements [12].

Microservices enable a more flexible chaining of services across clouds, as each microservice can be independently managed and routed. For example, in a multi-cloud environment, a load-balancing microservice could be deployed in one cloud, while a security microservice might be deployed in another, closer to the data source. Kubernetes can manage these distributed microservices, ensuring that they work together seamlessly as part of the overall service chain. This approach allows operators to take advantage of the unique strengths of different cloud platforms, optimizing the deployment of VNFs based on factors like cost, performance, and regulatory requirements [13].

Decomposing VNFs into microservices also simplifies the process of scaling and updating individual service functions. Since each microservice operates independently, scaling one component does not impact the others, reducing the risk of service disruption. This independence also allows for continuous integration and continuous deployment (CI/CD) pipelines, enabling faster updates and improvements to VNFs. In the context of SFC, this means that individual service functions can be updated or replaced without disrupting the overall service chain, enhancing the agility and responsiveness of network operations.

The use of microservices also enhances the resilience of SFC

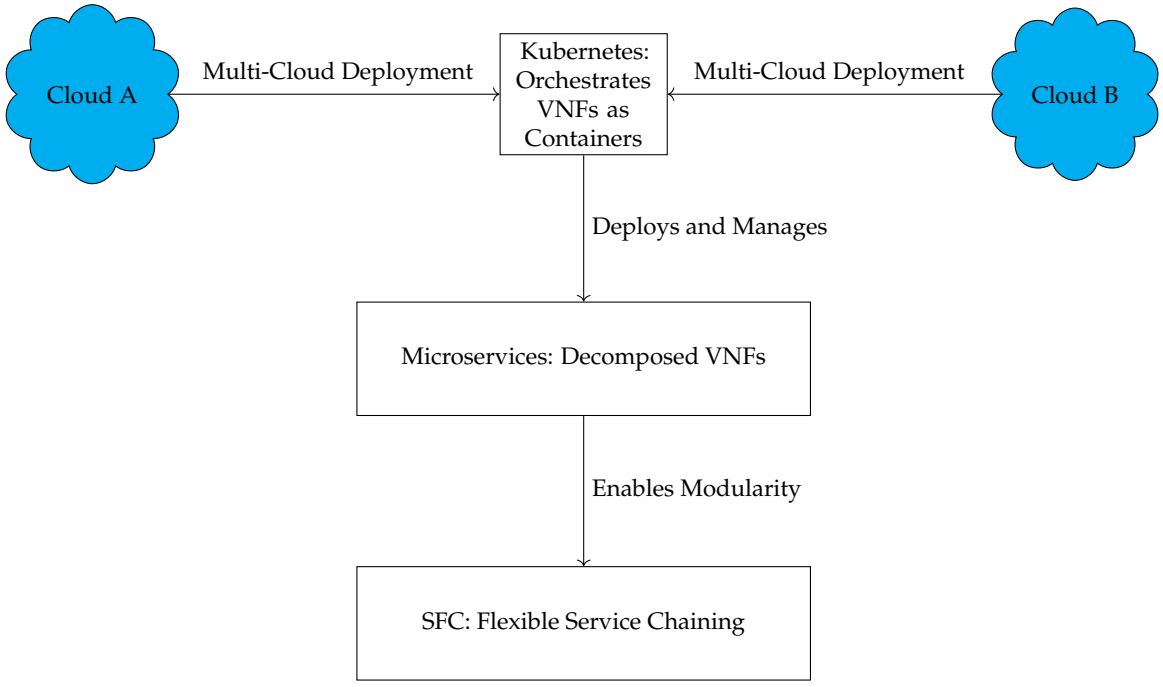

Fig. 3. Kubernetes and Microservices in Cloud-Native SFC: Orchestrating VNFs and Enabling Modularity

implementations. By deploying microservices across multiple clouds, operators can build redundancy into their service chains, reducing the impact of failures or performance issues in any single cloud. If a particular microservice becomes unavailable, Kubernetes can automatically reroute traffic to an alternative instance, ensuring that the service chain continues to function as intended. This built-in redundancy improves the reliability of network services and helps maintain service continuity in the face of infrastructure disruptions.

Kubernetes and microservices also support observability, a key aspect of managing complex SFCs. Observability involves monitoring the performance, health, and traffic flows of VNFs and microservices, providing real-time insights into how the service chain is functioning. Kubernetes integrates with various monitoring and logging tools, allowing operators to track metrics, detect anomalies, and respond proactively to performance issues. This visibility is essential for maintaining the efficiency and reliability of SFCs, as it enables operators to quickly identify and address bottlenecks or failures within the service chain.

C. Policy-Driven Orchestration and Automation

Policy-driven orchestration and automation are key elements in managing complex Service Function Chaining (SFC) environments, especially in multi-cloud settings. Intent-Based Networking (IBN) and automation tools like Ansible, Terraform, and Helm help network operators define and implement high-level policies that guide the orchestration of VNFs and other network components. These technologies enable administrators to simplify network management by focusing on broad operational goals and service requirements rather than going into the specifics of configuring individual network devices and services.

Intent-Based Networking (IBN) is a policy-driven approach that allows administrators to define high-level intents, which describe the desired behavior or state of the network. These intents are then automatically translated into detailed network configurations. For instance, an intent might specify that traffic should pass through a set of security and load-balancing functions, regardless of the physical layout of the network. The orchestration framework takes this high-level intent and implements it by configuring the necessary VNFs, routing paths, and other SFC components.

This approach simplifies the management of SFC by abstracting away the complexity of individual configurations. Instead of manually configuring each network element, administrators can define what they want to achieve, and the orchestration system handles the specifics. This is useful in multi-cloud environments, where VNFs are deployed across different cloud providers with varying infrastructure and configuration requirements. By using IBN, network operators can ensure consistent service delivery across diverse environments without needing to manually adjust configurations for each cloud.

Automation tools such as Ansible, Terraform, and Helm further support policy-driven orchestration by automating the deployment and management of VNFs and SFC components. Ansible is widely used for automating IT tasks, including the configuration of network devices and application deployment. In the context of SFC, Ansible can automate the setup of VNFs, define service chains, and configure the policies that govern traffic flows. Its agentless architecture allows it to work effectively in multi-cloud environments, interacting with cloud APIs without the need for additional software on managed devices.

Terraform automates the provisioning and management of cloud infrastructure using a declarative configuration language. This approach allows administrators to define infrastructure as code, specifying the desired state of their environment in configuration files. In multi-cloud SFC deployments, Terraform can automate the provisioning of VNFs across different cloud providers, ensuring that each service function is deployed correctly and connected according to policy requirements. This reduces the complexity of managing distributed SFCs by providing a unified configuration that spans multiple environments.

Helm, a package manager for Kubernetes, helps manage

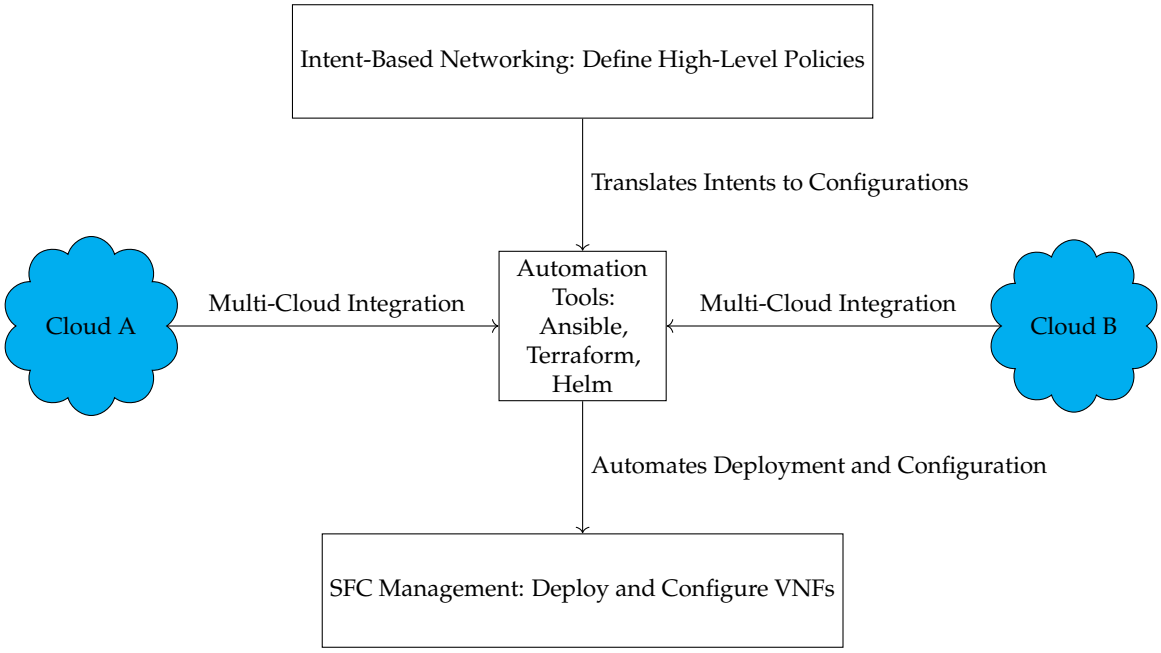

Fig. 4. Policy-Driven Orchestration and Automation in Multi-Cloud SFC: Integration of IBN and Automation Tools

VNFs deployed as containers. Helm charts allow operators to package, configure, and deploy VNFs on Kubernetes clusters, providing a consistent method for managing these components across different deployments. Helm simplifies updates and lifecycle management of VNFs, ensuring that they are consistently deployed and configured, which is important for maintaining the overall integrity of the service chain.

These automation tools work together with intent-based policies to streamline the orchestration of SFCs. For example, a high-level intent might specify that traffic needs to traverse a sequence of VNFs, such as a firewall, load balancer, and security inspection function. Ansible can automate the configuration of each VNF according to this policy, Terraform can handle the deployment across different clouds, and Helm can manage their operation within Kubernetes clusters. This coordinated approach helps align network configurations with high-level policies, simplifying management and reducing the potential for errors.

Policy-driven orchestration and automation also make multi-cloud SFC more adaptable to changes. Updates to network policies can be applied across the entire infrastructure without the need to manually reconfigure individual network elements, allowing operators to respond more quickly to changes in traffic loads, security threats, or business needs. Additionally, automation helps ensure that network services are consistently implemented according to the defined intents, supporting compliance with performance, security, and regulatory requirements.

Automation tools also enhance consistency across deployments, which is essential in multi-cloud environments where configurations can vary significantly between providers. By using scripts that are version-controlled and tested, operators can ensure that deployments are repeatable and predictable, minimizing discrepancies and helping maintain service reliability.

D. Security and Scalability

Security and compliance are important aspects of implementing Service Function Chaining (SFC) across multi-cloud environments. End-to-end security policies are essential to protect data as it moves through the service chain. Encryption ensures that data remains secure during transmission between Virtual Network Functions (VNFs), preventing unauthorized access. Access controls manage which users or services can interact with specific VNFs, reducing the chance of unauthorized changes or data breaches. Security monitoring allows for real-time visibility into traffic, identifying potential threats and helping operators respond swiftly to issues.

Compliance management in multi-cloud environments is complicated by the fact that these environments often span multiple jurisdictions with different regulatory requirements. Automated compliance management tools can assist in ensuring that SFC deployments meet these requirements. These tools can track compliance, automatically adjust configurations to stay within regulatory guidelines, and generate reports for audits, reducing the administrative burden on operators.

Scalability in SFC is supported by cloud-native auto-scaling capabilities. VNFs can scale up or down in response to traffic demand, ensuring efficient use of resources. This elastic scaling helps maintain performance during periods of high traffic and reduces costs during lower demand periods by adjusting resource allocation automatically.

Resilience is achieved through load balancing and failover strategies. Load balancers distribute traffic across multiple instances of a VNF, which helps to prevent overloading a single instance and ensures consistent performance. Failover mechanisms ensure that if one instance fails, traffic is automatically rerouted to another, maintaining service continuity without manual intervention. These strategies enhance the overall reliability of the SFC.

4. OPTIMIZATION OF DEPLOYMENT, ORCHESTRATION, AND SCALABILITY

The optimization of deployment, orchestration, and scalability is crucial for enhancing the performance and efficiency of net-

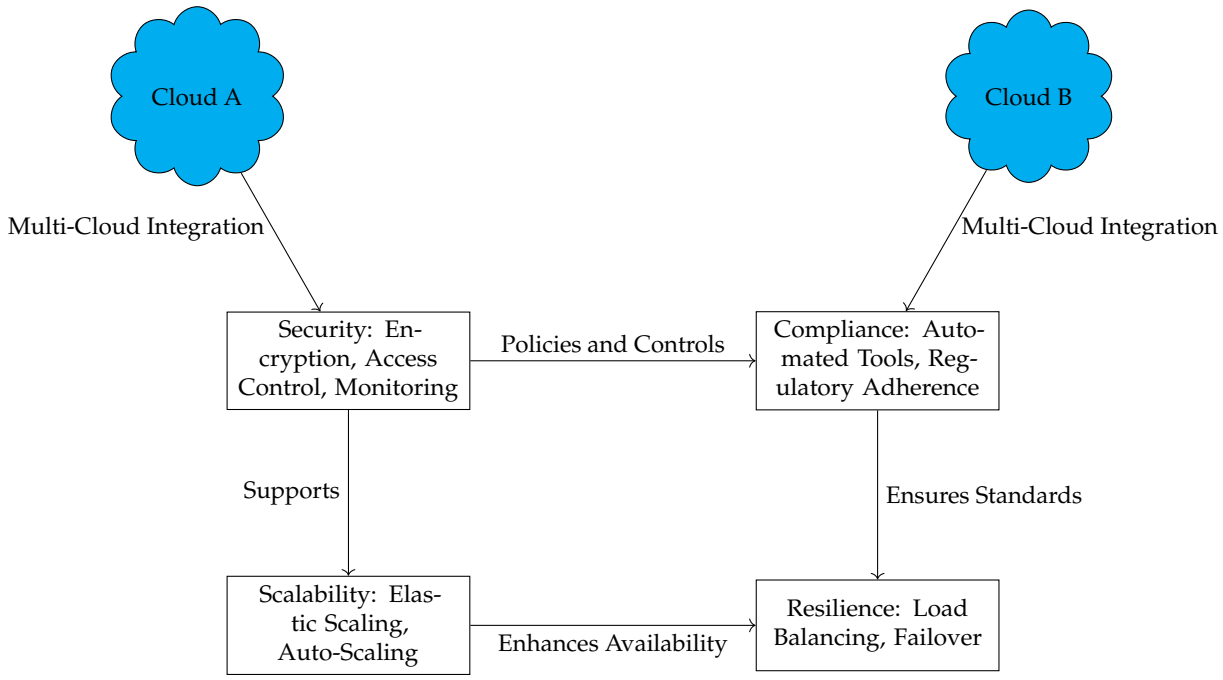

Fig. 5. Security, Compliance, Scalability, and Resilience in Multi-Cloud SFC

work functions and services, especially in dynamic and complex environments like multi-cloud and hybrid cloud setups. Key strategies include deploying a decoupled architecture, enhancing orchestration through advanced tools, and implementing scalability mechanisms that adapt to fluctuating demands. These approaches aim to reduce dependencies, simplify management, and ensure seamless scalability, ultimately enhancing the overall performance of Service Function Chains (SFCs) [14].

Deployment optimization focuses on refining how Virtual Network Functions (VNFs) and other network components are deployed and managed within an SFC. It seeks to streamline processes, minimize dependencies, and leverage architectural designs that foster flexibility and speed. Key components of deployment optimization include the implementation of decoupled architecture and the use of a unified control plane.

Implementing a decoupled architecture is one of the fundamental steps in optimizing deployment. In a decoupled setup, VNFs are designed to be independently deployable and manageable, meaning that they can operate without tight integration with other functions. This independence allows network operators to update or replace specific VNFs without disrupting the entire service chain, significantly reducing downtime and maintenance complexity. Decoupled architectures foster agility, enabling faster deployment of new services or updates, and improving the overall resilience of the network. Moreover, decoupling reduces interdependencies among VNFs, mitigating the risk of cascading failures and simplifying the troubleshooting process. As a result, the decoupled architecture is not just a deployment optimization strategy but also a means to enhance the robustness and adaptability of the network [15].

A unified control plane is another crucial element of deployment optimization in environments where multiple clouds or hybrid clouds are utilized. The unified control plane provides a centralized management interface that spans across different cloud environments, offering a single point of control for all network functions. This centralized approach simplifies the deployment, management, and orchestration of SFCs, as it consolidates the complexities associated with multi-cloud environments into a unified framework. By having a single, cohesive control plane, network operators can more effectively manage resources, enforce policies, and optimize the configuration of VNFs across disparate cloud platforms. This reduces the operational overhead, minimizes the potential for misconfigurations, and ensures consistent performance across all deployments. Additionally, a unified control plane facilitates the seamless integration of new VNFs, making it easier to scale and adapt to evolving network requirements [16].

Orchestration plays a vital role in managing the deployment, scaling, and operational aspects of VNFs within an SFC. Enhancements in orchestration are essential for achieving a more streamlined and efficient network function management process. This includes leveraging advanced orchestration platforms that offer multi-cluster management and integrating service meshes that provide deeper insights and control over the network functions.

Multi-cluster management is an advanced orchestration capability that allows for the coordination and control of VNFs across multiple clusters, often distributed across different cloud environments. Platforms like Kubernetes Federation enable this type of orchestration, allowing network operators to manage resources and configurations consistently across clusters. This approach is beneficial in multi-cloud setups where VNFs are deployed across various geographical regions or cloud providers. Multi-cluster management ensures that policies are uniformly enforced and that VNFs can be deployed, scaled, and updated consistently, regardless of their location. This enhances the reliability and performance of SFCs, as it reduces the complexity of managing disparate clusters and provides a unified view of the entire network function deployment landscape. Moreover, it simplifies compliance and security management, as policies can be centrally defined and automatically propagated to all clusters.

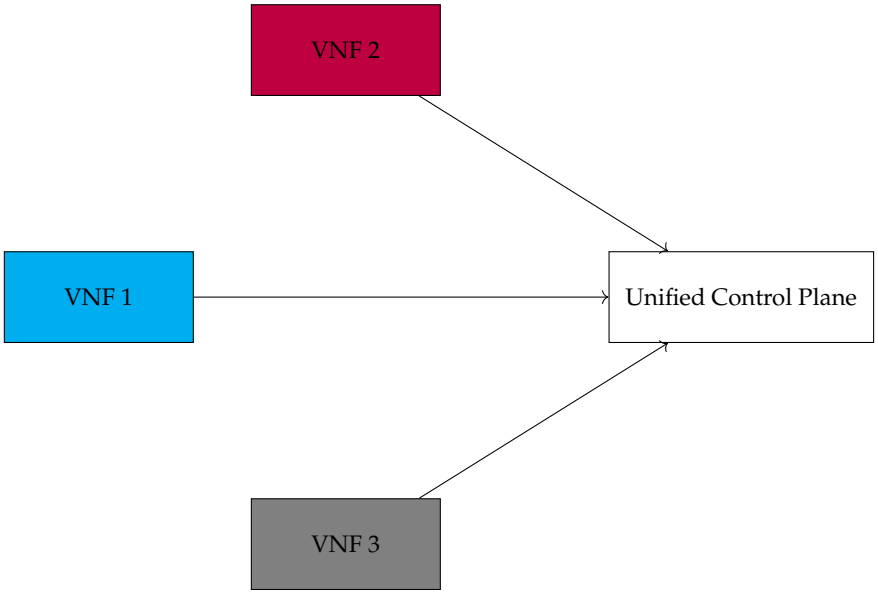

Fig. 6. Decoupled Architecture with Independent VNFs

Service mesh integration is another orchestration enhancement that provides significant benefits for managing VNFs within an SFC. Service meshes, such as Istio, offer advanced features like observability, traffic management, and security that are essential for modern microservices architectures. By integrating service meshes into the orchestration layer, network operators gain enhanced visibility into the behavior and performance of VNFs, enabling more precise monitoring and control. Service meshes also facilitate sophisticated traffic management capabilities, such as load balancing, failover, and circuit breaking, which are critical for maintaining optimal performance and reliability of SFCs. Additionally, service meshes provide a uniform layer of security across all VNFs, including features like mutual TLS, access control, and policy enforcement, thereby enhancing the security posture of the network. Integrating a service mesh into the orchestration layer not only improves the operational efficiency of SFCs but also provides a robust framework for managing complex, distributed network functions.

Scalability is a key consideration in the design and operation of SFCs in environments where demand can fluctuate significantly. Effective scalability mechanisms enable VNFs to handle varying loads without compromising performance or reliability. Key scalability approaches include horizontal and vertical scaling, as well as cross-cloud traffic optimization techniques that enhance the performance of SFCs across multi-cloud environments.

Horizontal and vertical scaling are fundamental mechanisms for adjusting the capacity of VNFs based on performance requirements. Horizontal scaling involves adding more instances of a VNF to distribute the load, thereby enhancing the overall throughput and reliability of the service chain. This type of scaling is effective in cloud environments where resources can be dynamically allocated and deallocated based on demand. Horizontal scaling provides a flexible way to respond to traffic spikes, ensuring that VNFs can maintain performance levels even during periods of high demand. On the other hand, vertical scaling focuses on increasing the resources (such as CPU, memory, or storage) allocated to existing VNF instances. Vertical scaling is advantageous when a VNF requires more computational power or memory to handle increased loads, but where deploying additional instances may not be practical or cost-effective. Both scaling strategies play a critical role in maintaining the performance and availability of SFCs, allowing network operators to tailor resource allocation to meet specific performance objectives [17].

Cross-cloud traffic optimization is a scalability strategy that addresses the challenges of managing SFCs across multiple cloud environments. In multi-cloud setups, VNFs often need to communicate across different cloud regions or providers, which can introduce latency and degrade performance. Techniques such as traffic engineering and path optimization are employed to minimize these issues, ensuring that data flows are efficiently managed across cloud boundaries. Traffic engineering involves the strategic routing of traffic through the most optimal paths, reducing latency and avoiding congested routes. Path optimization further enhances this process by dynamically selecting the best routes based on real-time network conditions, such as bandwidth availability, latency, and packet loss rates. These optimizations are crucial for maintaining the performance of SFCs, as they ensure that VNFs can communicate efficiently across diverse cloud environments. By reducing the latency and improving the quality of cross-cloud communications, traffic optimization techniques enhance the overall scalability and responsiveness of the network.

The optimization of deployment, orchestration, and scalability is essential for the efficient operation of SFCs in modern cloud environments. By implementing a decoupled architecture, leveraging unified control planes, and enhancing orchestration with multi-cluster management and service meshes, network operators can significantly improve the deployment and management of VNFs. Scalability mechanisms, including horizontal and vertical scaling and cross-cloud traffic optimization, further ensure that SFCs can adapt to changing demands without sacrificing performance. These strategies collectively contribute to a more agile, resilient, and high-performing network infrastructure, capable of meeting the evolving needs of modern digital services.

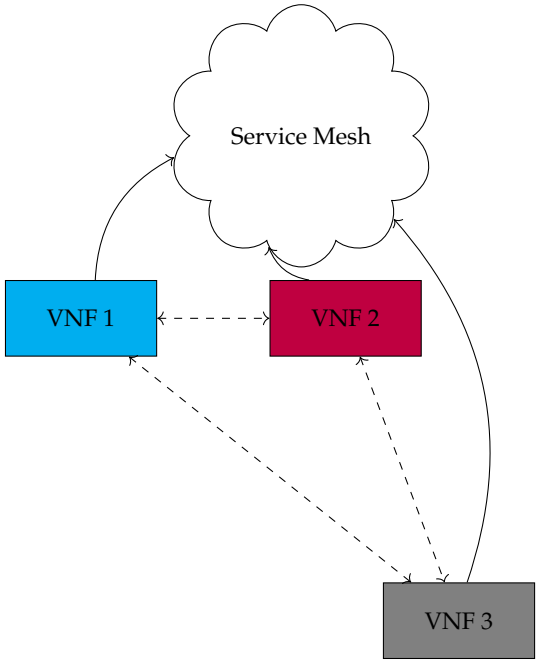

Fig. 7. Service Mesh Integration Enhancing Traffic Management

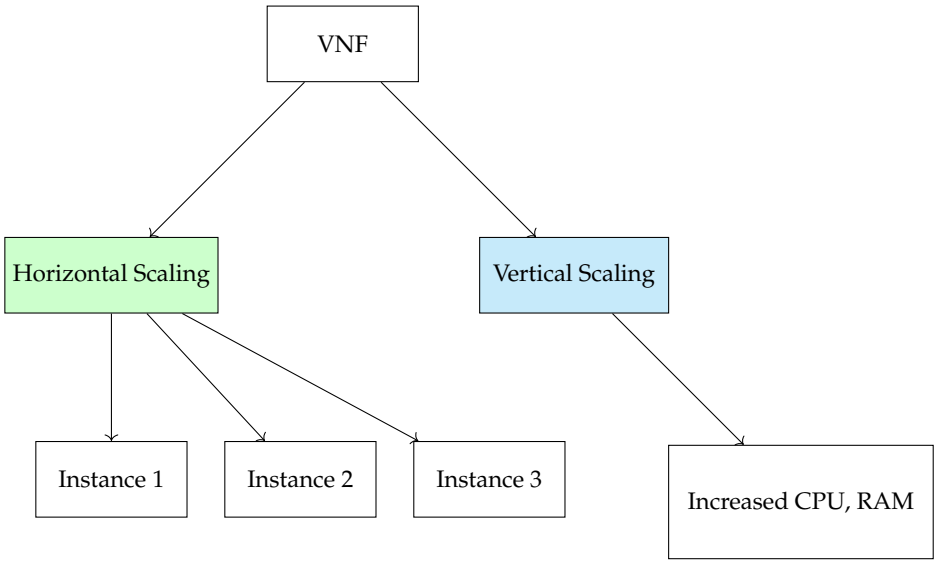

Fig. 8. Horizontal vs. Vertical Scaling of VNFs

5. PROPOSED ADDITION: CROSS-LAYER COORDINATION AND PROGRAMMABLE DATA PLANES FOR ENHANCED SFC IN MULTI-CLOUD ENVIRONMENTS

Traditional approaches often suffer from fragmented control, where the application, network, and infrastructure layers operate in silos, leading to inefficiencies and suboptimal performance. This proposal introduces cross-layer coordination and programmable data planes as critical additions to SFC management, addressing the challenges of disjointed orchestration and static traffic configurations. By integrating these advanced strategies, SFC can achieve improved adaptability, performance, and scalability, paving the way for more robust multi-cloud deployments.

A. Cross-Layer Coordination for Integrated SFC Management

Cross-layer coordination represents a holistic approach to SFC management that integrates the application, network, and infrastructure layers into a unified control framework. This integration addresses the challenges of disjointed management, where each layer typically operates with its own set of tools, policies, and protocols, often resulting in fragmented service chains that are difficult to manage and optimize. By unifying these layers, cross-layer coordination enhances the ability to dynamically manage and adapt SFCs in real-time, ensuring that services are consistently optimized across the entire network stack.

One of the key components of cross-layer coordination is the use of integrated orchestration frameworks that enable seamless

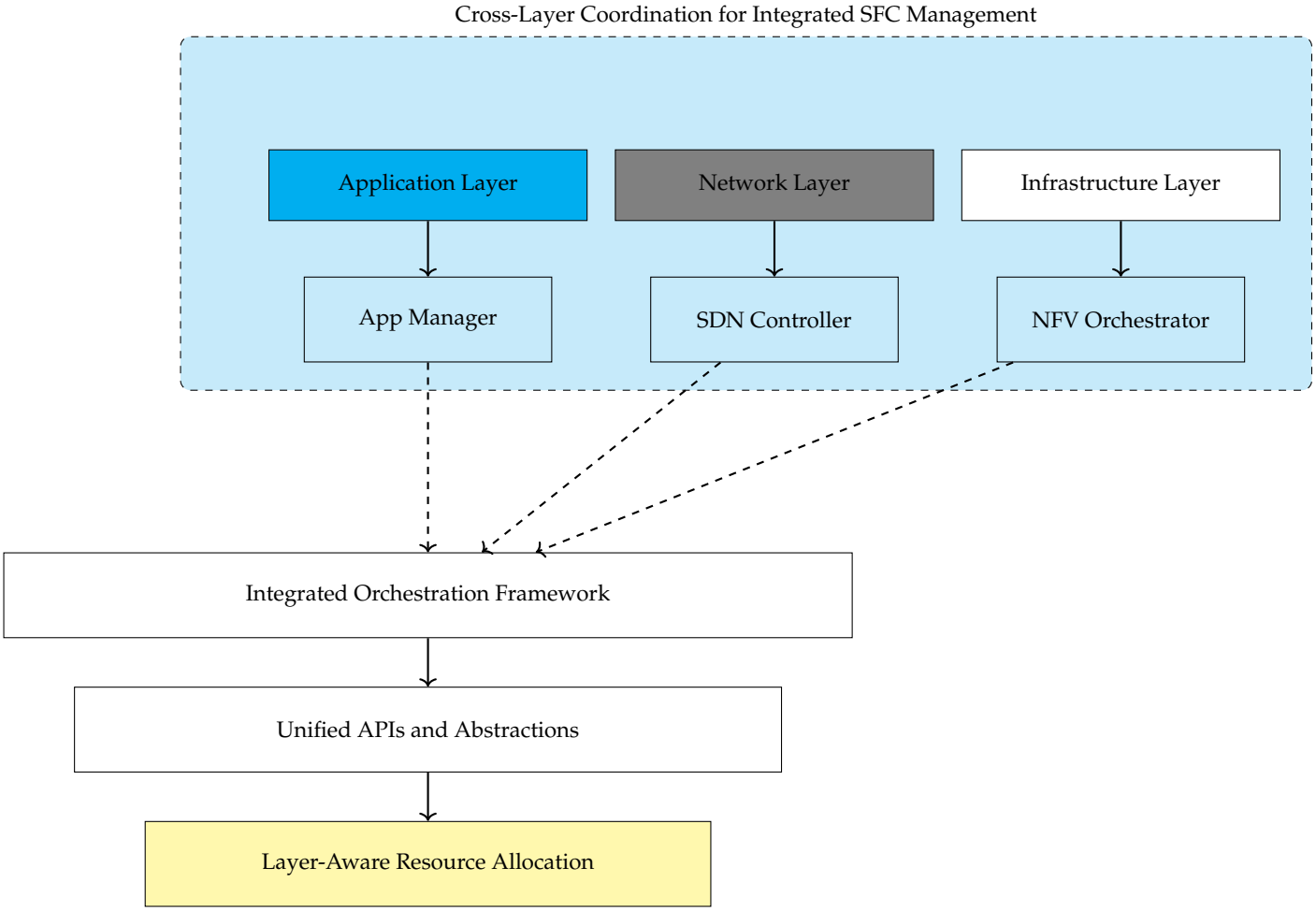

Fig. 9. Cross-Layer Coordination for Integrated SFC Management: The diagram groups Application, Network, and Infrastructure layers within an encompassing node, illustrating their integration and interactions through an Integrated Orchestration Framework, Unified APIs, and Layer-Aware Resource Allocation.

communication between Software-Defined Networking (SDN) controllers, Network Function Virtualization (NFV) orchestrators, and application-level managers. These frameworks create a cohesive environment where information and control signals can flow between layers, allowing for real-time adjustments to SFC paths based on current network conditions, application demands, and infrastructure availability. For example, an SDN controller can dynamically reconfigure network paths in response to congestion detected by the NFV orchestrator, or an application manager can adjust its resource requests based on feedback from lower network layers. This level of integration not only streamlines the management of SFC but also improves the overall responsiveness of the system, as adjustments can be made on-the-fly without the need for manual intervention.

Integrated orchestration frameworks help bridge the gap between different layers, facilitating coordinated decision-making that optimizes SFC performance holistically rather than in isolation. This is important in multi-cloud environments, where SFC components may span multiple providers, each with its own management interfaces and policies. By providing a unified orchestration layer, cross-layer coordination ensures that SFCs are consistently managed and optimized across all clouds, reducing complexity and improving the predictability of service delivery.

Unified APIs and abstractions play a critical role in enabling cross-layer coordination by providing consistent interfaces for managing SFCs across multiple layers and cloud environments. In traditional setups, each layer may expose its own set of APIs, often leading to compatibility issues and increased management overhead. Unified APIs, on the other hand, offer a standardized approach that abstracts the underlying complexity, allowing different components of the SFC to communicate and interact seamlessly. This abstraction simplifies the integration of new services and functions, making it easier to deploy and manage SFCs in multi-cloud settings.

Unified abstractions also enhance the programmability of SFCs by providing higher-level constructs that can be used to define policies, allocate resources, and configure network paths in a way that is agnostic of the underlying infrastructure. This reduces the need for specialized knowledge of each cloud provider's specific tools and APIs, enabling more streamlined management processes. For instance, a single unified API could be used to adjust bandwidth allocations, prioritize traffic, or modify security settings across all layers of the SFC, significantly reducing the time and effort required to implement such changes.

Resource allocation in SFCs traditionally focuses on individual network components, often neglecting the broader impact on the overall service chain. Layer-aware resource allocation

addresses this by considering factors from all network layers, allowing for more coordinated and efficient distribution of resources. By taking into account application requirements, network conditions, and infrastructure capabilities, layer-aware resource allocation ensures that SFCs are optimized not only for performance but also for cost and resource utilization.

This approach enables more intelligent resource management strategies, such as dynamically scaling VNFs based on application-layer insights or rerouting traffic in response to infrastructure-layer changes like server outages or bandwidth constraints. Layer-aware resource allocation can also help balance workloads across multiple clouds, leveraging each provider's strengths to achieve optimal performance. For example, compute-intensive tasks might be routed to a cloud with superior processing capabilities, while latency-sensitive operations are directed to geographically closer data centers. This level of coordination ensures that SFCs are continuously adjusted to align with the overall objectives of performance, efficiency, and cost-effectiveness [18].

B. Programmable Data Planes for Dynamic Traffic Steering

While cross-layer coordination enhances the overall management of SFCs, programmable data planes introduce a level of flexibility and control at the network layer that is crucial for optimizing traffic flow. Traditionally, data planes have been relatively static, with fixed configurations that are slow to adapt to changing network conditions. Programmable data planes, however, offer a dynamic alternative, allowing for real-time adjustments to traffic steering policies based on the current state of the network. This adaptability is valuable in multi-cloud environments, where traffic patterns can vary widely and unexpected changes in load or connectivity can impact service performance [18]. s

Programmable data planes enable real-time traffic adaptation through technologies like P4 (Programming Protocol-independent Packet Processors) and eBPF (Extended Berkeley Packet Filter). These technologies allow for the dynamic reconfiguration of data routing at the packet level, providing unprecedented control over how traffic flows through the network. For instance, P4 allows network operators to define custom packet processing rules that can be modified on-the-fly, adapting to changes in traffic conditions, security requirements, or application demands.

Real-time traffic adaptation enhances the performance of SFCs by ensuring that traffic is always routed through the most optimal paths, minimizing latency, and avoiding congestion. This is important in multi-cloud environments, where the availability and performance of network paths can vary significantly depending on the underlying cloud provider and regional infrastructure. By leveraging programmable data planes, SFCs can dynamically adjust their traffic patterns to optimize performance across all clouds, ensuring that service quality is maintained even in the face of changing network conditions.

Programmable data planes also provide the ability to implement advanced traffic engineering techniques, such as load balancing, traffic prioritization, and fault tolerance, directly within the data plane. This level of granularity allows for more precise control over how traffic is handled, enabling SFCs to meet stringent performance and reliability requirements. For example, traffic can be automatically rerouted in response to link failures or performance degradation, ensuring continuous service availability without the need for manual intervention.

Another significant advantage of programmable data planes is their ability to support customizable SFC policies. Unlike traditional data planes, which rely on predefined, static configurations, programmable data planes allow network operators to define and modify policies that govern how traffic is handled within the SFC. These policies can be tailored to meet specific performance, security, or compliance requirements, providing a level of customization that is difficult to achieve with conventional approaches.

Customizable SFC policies enable fine-tuned traffic steering that can be adjusted in response to real-time analytics and monitoring data. For example, traffic destined for critical applications can be given priority over less important flows, or sensitive data can be routed through paths with enhanced security measures. This flexibility is valuable in multi-cloud environments, where different providers may offer varying levels of performance, security, and compliance capabilities. By customizing SFC policies, network operators can ensure that traffic is always handled in a way that aligns with the specific needs of the service, regardless of the underlying cloud infrastructure.

Additionally, programmable data planes support the implementation of security policies that can be enforced at the data plane level, providing an additional layer of protection for SFCs. For example, network operators can define rules that automatically drop or reroute malicious traffic, mitigating the impact of security threats without affecting legitimate flows. This level of programmability enhances the overall security posture of the SFC, making it more resilient to attacks and other disruptions.

6. CONCLUSION

Service Function Chaining (SFC) has gained significant traction in modern networking, especially as businesses embrace multi-cloud strategies. SFC allows for the ordered and dynamic chaining of various network functions like firewalls, load balancers, and security gateways, which are essential components of today's complex network architectures. The goal of SFC is to enhance service delivery by streamlining how network functions are managed and deployed, reducing latency, and increasing the agility of network services. However, as companies deploy SFC in multi-cloud environments, they encounter unique challenges that necessitate advanced architectural strategies to ensure seamless and efficient operation.

Multi-cloud environments, characterized by the use of services from multiple cloud providers, have become a strategic choice for enterprises looking to optimize performance, mitigate risks, and avoid vendor lock-in. However, the heterogeneous nature of these environments introduces complexity in terms of orchestration, security, and performance management. For SFC, this complexity is further compounded by the need to coordinate network functions across different platforms with varying networking models, APIs, and security protocols. As such, implementing SFC in a multi-cloud setup requires robust frameworks that can handle these discrepancies while maintaining consistent service performance [6].

Network Function Virtualization (NFV) and Software-Defined Networking (SDN) are critical technologies that underpin SFC. NFV allows for the virtualization of network functions, decoupling them from dedicated hardware, which enables their deployment as software on standard servers. This approach not only reduces hardware costs but also provides the flexibility to deploy VNFs anywhere within the multi-cloud ecosystem. SDN, on the other hand, provides centralized control over network traffic, allowing data to be dynamically steered through

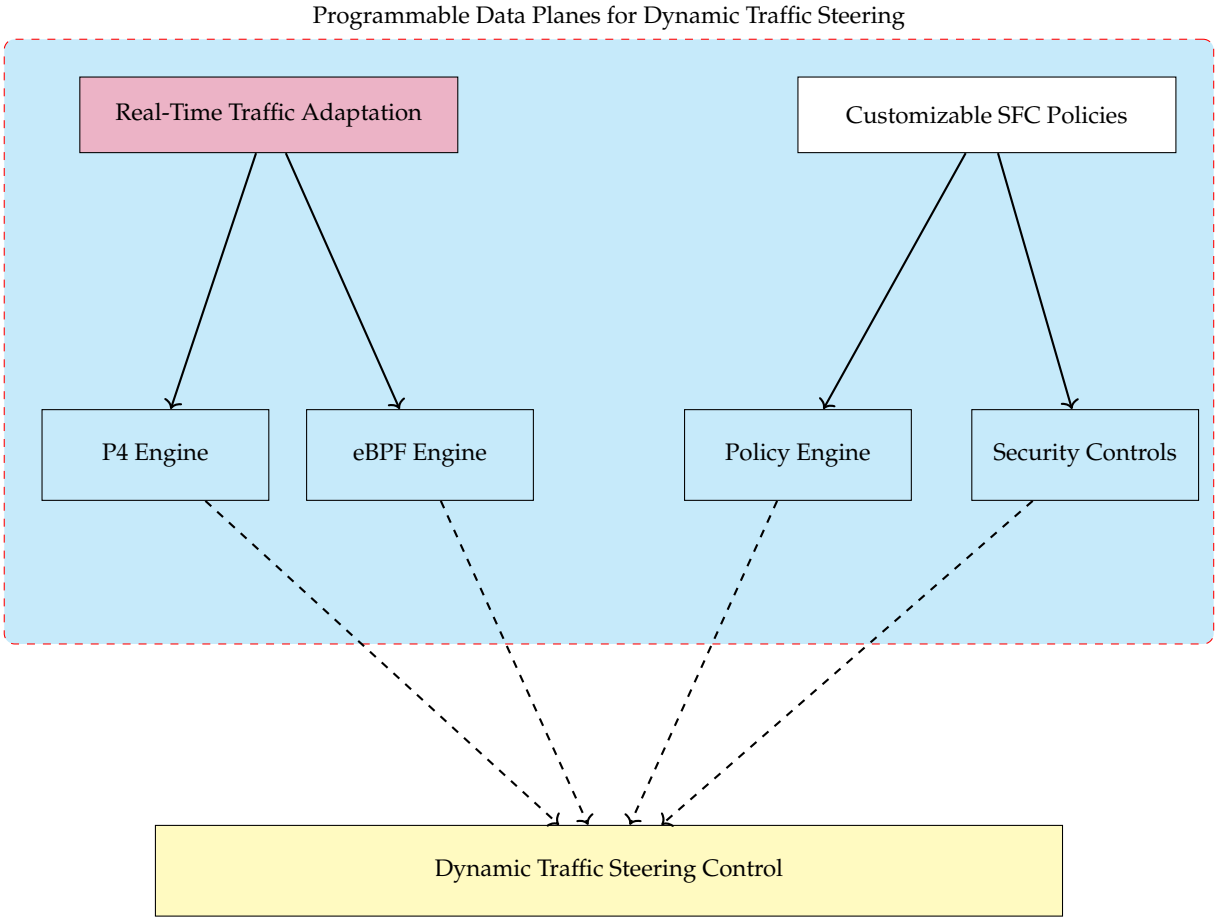

Fig. 10. Programmable Data Planes for Dynamic Traffic Steering: The diagram groups Real-Time Traffic Adaptation and Customizable SFC Policies within an encompassing node, illustrating their integration and influence on Dynamic Traffic Steering Control for optimized, adaptive traffic management.

the desired VNFs as defined by the SFC. The combination of NFV and SDN forms a powerful foundation for automating and optimizing SFC in multi-cloud environments.

A crucial component of modern SFC implementations is the use of cloud-native technologies like Kubernetes, which facilitates the orchestration of containerized applications and services. Kubernetes' inherent support for service meshes, network policies, and multi-cluster management makes it an ideal platform for managing VNFs in a multi-cloud setup. Additionally, the microservices architecture, which breaks down monolithic applications into smaller, independently manageable services, aligns well with the modular nature of VNFs. This approach enhances scalability and flexibility, allowing each VNF to be independently deployed, managed, and scaled according to the demands of the network.

Policy-driven orchestration plays a vital role in managing SFC across diverse cloud environments. By defining high-level intents or policies, network administrators can abstract the underlying complexity of multi-cloud configurations. Intent-based networking, for instance, allows administrators to specify desired outcomes, such as latency thresholds or security requirements, without having to manually configure each network element. Tools like Ansible, Terraform, and Helm further automate the deployment and configuration of SFC components, ensuring consistency and reducing the potential for human error.

Security remains a paramount concern in SFC deployments, especially in multi-cloud environments where data traverses different cloud infrastructures. Implementing end-to-end security policies, including encryption, access controls, and continuous monitoring, is essential to safeguarding data as it moves through the SFC. Additionally, compliance management tools help ensure that SFC implementations adhere to varying regulatory standards across different jurisdictions, an important consideration in industries like finance and healthcare where data governance is tightly regulated.

Scalability and resilience are also critical aspects of SFC in multi-cloud environments. Utilizing elastic scaling capabilities inherent in cloud-native platforms, VNFs can be automatically scaled up or down in response to real-time traffic demands, ensuring that resources are used efficiently. Load balancing further enhances resilience by distributing traffic across multiple instances of a VNF, reducing the impact of potential failures and improving overall service reliability. These strategies are essential for maintaining high performance and availability in dynamic, multi-cloud settings.

Optimizing the deployment, orchestration, and scalability of SFC in multi-cloud environments requires a strategic approach that considers the unique challenges of these architectures. One effective strategy is to implement a decoupled architecture where VNFs are independently managed, reducing

dependencies and enabling rapid updates. A unified control plane that spans multiple clouds can further simplify SFC management, providing a single interface for controlling network functions across diverse infrastructures.

Orchestration enhancements, such as multi-cluster management using platforms like Kubernetes Federation, facilitate the consistent management of SFC across clusters deployed in different clouds. Service meshes, like Istio, extend the capabilities of Kubernetes by offering advanced features such as observability, traffic management, and security, which are critical for orchestrating SFC at scale. By integrating these tools, organizations can achieve a more streamlined and automated approach to managing their SFC deployments.

Scalability mechanisms, such as horizontal and vertical scaling, allow VNFs to adjust based on current performance needs. Horizontal scaling involves adding more instances of a VNF, while vertical scaling increases the resources allocated to existing instances. Both approaches are essential for optimizing the performance of SFC in multi-cloud environments, especially when dealing with fluctuating workloads. Additionally, cross-cloud traffic optimization techniques, such as traffic engineering and path optimization, can reduce latency, further enhancing the efficiency of SFC.

One promising addition to SFC architecture in multi-cloud environments is cross-layer coordination, which integrates the management of application, network, and infrastructure layers. This approach addresses the disjointed nature of traditional management practices, where each layer operates independently, often leading to suboptimal performance. Integrated orchestration frameworks enable seamless communication between SDN controllers, NFV orchestrators, and application-level managers, allowing for real-time adaptation of SFC paths based on current network conditions.

Programmable data planes offer another innovative solution for enhancing SFC in multi-cloud environments. By utilizing technologies like P4 and eBPF, network operators can achieve dynamic traffic steering, which adjusts the routing of data in real time based on network conditions. This flexibility is valuable in multi-cloud setups, where traffic patterns can be highly variable. Programmable data planes also support customizable SFC policies, enabling fine-tuned traffic management to meet specific performance, security, or compliance requirements.

One of the primary limitations of the research is the inherent complexity involved in orchestrating and managing Service Function Chaining (SFC) across multi-cloud environments. Each cloud provider has distinct APIs, networking models, security protocols, and performance characteristics, which can significantly complicate the integration of Virtual Network Functions (VNFs) across different platforms. This heterogeneity creates challenges in maintaining consistent performance, security policies, and operational efficiency. The research assumes the availability of advanced orchestration tools capable of bridging these gaps; however, the reality is that current technologies often lack seamless integration capabilities across disparate cloud platforms. Moreover, the manual intervention required to troubleshoot, configure, and optimize SFC paths across multiple clouds remains a significant barrier, undermining the full potential of automated orchestration and increasing the risk of misconfigurations and security vulnerabilities.

Scalability and latency issues pose another significant limitation of the research in distributed multi-cloud setups where VNFs are deployed across geographically dispersed data centers. While the research proposes solutions like elastic scaling, load balancing, and cross-cloud traffic optimization, these methods may not fully address the inherent latency introduced by the physical distances between cloud regions. As data flows traverse multiple cloud environments, the latency can accumulate, negatively impacting the performance of the chained services. Additionally, scaling VNFs in real-time to accommodate dynamic workloads can be resource-intensive and may not always align perfectly with the performance requirements of the network functions, especially when there is a need to scale across clouds with different performance profiles. The reliance on cloud-native technologies like Kubernetes and service meshes provides some mitigation, but the research does not fully account for the overhead introduced by these tools, which can further exacerbate latency and scaling inefficiencies.

Security and compliance are critical concerns when implementing SFC in multi-cloud environments when data crosses international borders with varying regulatory requirements. The research emphasizes the need for end-to-end security policies and automated compliance management; however, it does not fully address the complexities associated with multi-jurisdictional data flows. Differences in data sovereignty laws, privacy regulations, and security standards across regions can complicate the deployment of a unified security framework. Additionally, the distributed nature of SFC in multi-cloud settings increases the attack surface, making it more challenging to ensure comprehensive security coverage and consistent compliance enforcement. The automated tools suggested in the research, such as AI/ML-based monitoring and anomaly detection, may provide some level of proactive security, but they are not foolproof and may struggle to adapt to the constantly evolving regulatory landscape. This limitation underscores the need for ongoing research into more robust security and compliance solutions that can dynamically adjust to the unique requirements of multi-cloud SFC deployments.

REFERENCES

1. J. Zhang, Z. Wang, N. Ma, *et al.*, "Enabling efficient service function chaining by integrating nfv and sdn: Architecture, challenges and opportunities," IEEE Netw. **32**, 152–159 (2018).
2. D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," J. Netw. Comput. Appl. **75**, 138–155 (2016).
3. D. Borsatti, G. Davoli, W. Cerroni, *et al.*, "Performance of service function chaining on the openstack cloud platform," in *2018 14th International Conference on Network and Service Management (CNSM),* (IEEE, 2018), pp. 432–437.
4. M. S. Castanho, C. K. Dominicini, R. S. Villaça, *et al.*, "Phantomsfc: A fully virtualized and agnostic service function chaining architecture," in *2018 IEEE Symposium on Computers and Communications (ISCC),* (IEEE, 2018), pp. 354–359.
5. Q. Duan, "Modeling and performance analysis for service function chaining in the sdn/nfv architecture," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft),* (IEEE, 2018), pp. 476–481.
6. M. Di Mauro, M. Longo, F. Postiglione, *et al.*, "Service function chaining deployed in an nfv environment: An availability modeling," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN),* (IEEE, 2017), pp. 42–47.
7. B. Yi, X. Wang, and M. Huang, "Design and evaluation of schemes for provisioning service function chain with function scalability," J. Netw. Comput. Appl. **93**, 197–214 (2017).
8. R. A. Eichelberger, T. Ferreto, S. Tandel, and P. A. P. Duarte, "Sfc path tracer: a troubleshooting tool for service function chaining," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM),* (IEEE, 2017), pp. 568–571.
9. Y. Xu and V. P. Kafle, "Reliable service function chain provisioning in software-defined networking," in *2017 13th International Conference on Network and Service Management (CNSM),* (IEEE, 2017), pp. 1–4.
10. Y. Xie, Z. Liu, S. Wang, and Y. Wang, "Service function chaining resource allocation: A survey," arXiv preprint arXiv:1608.00095 (2016).
11. B. Tschaen, Y. Zhang, T. Benson, *et al.*, "Sfc-checker: Checking the correct forwarding behavior of service function chaining," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN),* (IEEE, 2016), pp. 134–140.
12. A. Farrel, "Recent developments in service function chaining (sfc) and network slicing in backhaul and metro networks in support of 5g," in *2018 20th International Conference on Transparent Optical Networks (ICTON),* (IEEE, 2018), pp. 1–4.
13. A. Tomassilli, N. Huin, F. Giroire, and B. Jaumard, "Resource requirements for reliable service function chaining," in *2018 IEEE International Conference on Communications (ICC),* (IEEE, 2018), pp. 1–7.
14. M. Ghaznavi, N. Shahriar, R. Ahmed, and R. Boutaba, "Service function chaining simplified," arXiv preprint arXiv:1601.00751 (2016).
15. G. Sun, Z. Xu, H. Yu, *et al.*, "Low-latency and resource-efficient service function chaining orchestration in network function virtualization," IEEE Internet Things J. **7**, 5760–5772 (2019).
16. J. Halpern and C. Pignataro, "Service function chaining (sfc) architecture," Tech. rep. (2015).
17. I. J. Sanz, D. M. F. Mattos, and O. C. M. B. Duarte, "Sfcperf: An automatic performance evaluation framework for service function chaining," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium,* (IEEE, 2018), pp. 1–9.
18. G. Mirjalily and Z. Luo, "Optimal network function virtualization and service function chaining: A survey," Chin. J. Electron. **27**, 704–717 (2018).