*electronics*

MDPI

*Article*

# Orchestrating Isolated Network Slices in 5G Networks

**Ali Esmaeily * and Katina Kralevska**

Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway; katinak@ntnu.no
* Correspondence: alies@alumni.ntnu.no

**Abstract:** Sharing resources through network slicing in a physical infrastructure facilitates service delivery to various sectors and industries. Nevertheless, ensuring security of the slices remains a significant hurdle. In this paper, we investigate the utilization of State-of-the-Art (SoA) Virtual Private Network (VPN) solutions in 5G networks to enhance security and performance when isolating slices. We deploy and orchestrate cloud-native network functions to create multiple scenarios that emulate real-life cellular networks. We evaluate the performance of the WireGuard, IPSec, and OpenVPN solutions while ensuring confidentiality and data protection within 5G network slices. The proposed architecture provides secure communication tunnels and performance isolation. Evaluation results demonstrate that WireGuard provides slice isolation in the control and data planes with higher throughput for enhanced Mobile Broadband (eMBB) and lower latency for Ultra-Reliable Low-Latency Communications (URLLC) slices compared to IPSec and OpenVPN. Our developments show the potential of implementing WireGuard isolation, as a promising solution, for providing secure and efficient network slicing, which fulfills the 5G key performance indicator values.

**Keywords:** B5G; eMBB; URLLC; network slice isolation; VPN; security; WireGuard; orchestration



**Citation:** Esmaeily, A.; Kralevska, K. Orchestrating Isolated Network Slices in 5G Networks. *Electronics* **2024**, *13*, 1548. https://doi.org/10.3390/electronics13081548

Academic Editors: Christos J. Bouras and Dimitris Kanellopoulos

Received: 29 February 2024
Revised: 12 April 2024
Accepted: 17 April 2024
Published: 18 April 2024



## 1. Introduction

The introduction of Network Function Virtualization (NFV), Software-Defined Networking (SDN) and Multi-Access Edge Computing (MEC) has enabled a flexible 5G network that can deliver services with diverse Quality-of-Service (QoS) requirements [1]. NFV allows the network operator to deploy and manage applications in a flexible manner. To achieve this, NFV Management and Orchestration (MANO) associates with a hypervisor running on an NFV Infrastructure (NFVI) and Virtual Infrastructure Manager (VIM). This approach enables the network operator to deploy Virtual Machines (VMs) and containers to the VIM. In the 5G network, VMs and containers with their functionalities are known as Virtual Network Functions (VNFs). In other words, the NFV MANO is responsible for configuring and managing various applications merged in VNFs, Network Services (NSs) and Network Slice Instances (NSIs) that operate on one or multiple VIMs. Another crucial aspect of the VIM and MANO relationship pertains to the network setup. The MANO must establish the network between VNFs. Networking could be internal within a VNF or between VNFs on the same or different VIMs. As application configuration needs to be prompt, the networking aspect must be adaptable.

Verticals require computation in the cloud or through rented or shared hypervisors and shared networks to deliver End-to-End (E2E) services. Rentable or shareable cloud infrastructures are needed for efficient resource utilization, financial benefits and load distribution. However, sharing infrastructure introduces new security challenges. Keeping application data secure during transfer over shared networks and between VNFs is a typical example of such challenges. Network slice isolation is an important concept supporting multiple verticals running on shared infrastructures, such as cloud infrastructure, in an isolated manner. In the 5G network, multiple isolation technologies will be available rather

than a single technology. Therefore, it is essential to incorporate and oversee various isolation mechanisms at different levels.

There are various degrees of slice isolation, including traffic, bandwidth, processing and storage isolation [2,3]. In this article, we expand the traffic isolation concept to include security isolation, which refers to preventing an unauthorized party outside the slice, such as another user of the same infrastructure but from a different slice, from modifying or intercepting the slice's traffic flow [4]. This also safeguards the confidentiality and integrity of the tenant's traffic, even from the Mobile Network Operator (MNO). For example, if a tenant requires a network suitable for a specific use case, it may rent a slice from the infrastructure provider to deliver network services to end-users. The only way to ensure confidentiality is by encrypting the data and operating VPN between VNFs. There are several proposed potential slice isolation technologies, such as tag-based slice isolation, VLAN-based network slice isolation, VPN-based slice isolation with IPsec, Secure Socket Layer/Transport Layer Security (SSL/TLS), Secure Socket Tunneling Protocol (SSTP), SSH (Secure Shell) and SDN-based isolation [2]. However, establishing VPN tunnels introduces additional overhead. The additional overhead may affect service performance, such as increasing the latency and decreasing the throughput. The NFV MANO can provide traffic isolation for VNFs in NSs by deploying VPN tunneling between VNFs and interconnecting them. In this way, the secure tunneling isolates NSIs and the provided NSs via the NSIs. Nevertheless, this approach is only feasible if the VPN does not introduce significant overhead violating QoS requirements.

Encrypting the data between chained VNFs can lead to telecommunication operators being more open towards multi-vendor NFV solutions [5]. There are two main reasons for telecommunication operators being hesitant with respect to multi-vendor NFV solutions. First, the NFV implementations are not steered by operators but rather by the vendors and integrators that perform the original root cause analysis of the problems. Here, the risk lies in the fact that most vendors' businesses rely on proprietary boxes, which could be negatively impacted by moving to a pure NFV platform. Second, operators are used to managing contracts with a handful of large vendors that provide the design, engineering and network deployment. Thus, the implementation of a multi-vendor NFV solution brings unwanted complexity for operators with regard to whom to blame for security issues. For this reason, most operators are adopting a vertical NFV solution, where a vendor delivers the full NFV deployment and retains the responsibility. On the other hand, a horizontal NFV model will significantly improve automation and contribute to higher flexibility. In this type of deployment, multiple NFV vendor products are managed and orchestrated on a single commodity NFVI by a single NFV MANO entity. This architecture is the telecom equivalent of cloud computing. The deployment of VPN between VNFs in an automatic mode to provide security isolation between slices and the effect of the introduced overhead on the performance isolation among slices in a shared environment are still open research questions.

Figure 1 illustrates the central contribution of this work, representing two distinct network slices tailored to specific use cases in the 5G landscape. On the one hand, it showcases entertainment applications like VR, necessitating mainly high throughput, effectively supported by the eMBB slice. On the other hand, automotive V2X applications, such as health care and vehicular communications, require ultra-reliable low-latency connectivity, which is fulfilled by the URLLC slice. To guarantee optimal functionality of the applications, it is imperative to maintain strict isolation between the eMBB and URLLC slices. This isolation ensures that each slice operates independently, free from interference or congestion caused by other slices. This approach safeguards the high throughput demand of VR applications within the eMBB slice while concurrently preserving the ultra-reliable and low-latency requirements of automotive V2X applications within the URLLC slice. Figure 1 demonstrates the essence of this research, highlighting the targeted optimization of network resources for diverse application requirements in the 5G ecosystem.

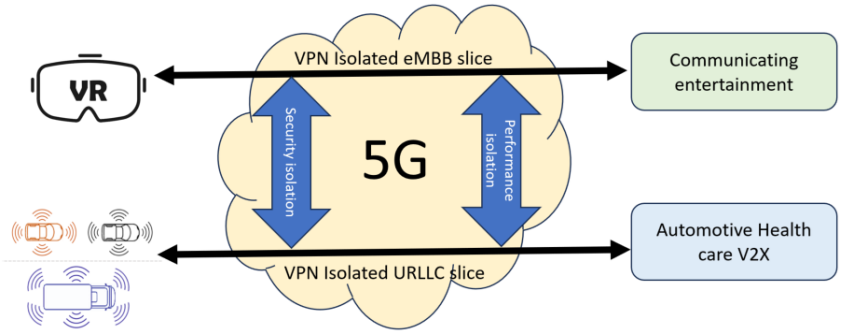


**Figure 1.** VR and automotive V2X applications operate over one eMBB slice and one URLLC slice, respectively, in a slice-isolated 5G network in which performance and security isolation are guaranteed.

We summarize our contributions as follows:

- We propose an integrated XVPN-OSM architecture that provides XVPN-as-a-Service (XVPNaaS) with (1) security isolation that results in secure communication between the involved VNFs of NSs, and NSIs, and (2) performance isolation between the slices.
- Since VNFs are instantiated in multiple domains architecture, a tenant may need or ask for the end-to-end confidentiality of the traffic passing the VNFs; thus, the presented architecture provides a point-to-point VPN solution.
- We implement a Proof-of-Concept (PoC) and evaluate the performance of different VPN solutions for providing slice isolation in a 5G SA environment.
- Our PoC involves an automated service instantiation incorporating WireGuard, OpenVPN and IPSec as VPN solutions. We use Open-Source MANO (OSM) [6] to orchestrate NSs and NSIs and establish different VPN tunnels between the VNFs. The OSM Northbound Interface (OSM-NBI) and juju proxy charms enable us to manage the different VPN solutions with our API easily.
- The performance evaluation shows that the presented XVPN-OSM architecture meets the required KPI values, including high throughput for eMBB slices and low latency for URLLC slices.

The remainder of this paper is organized as follows. Section 2 provides a background of the isolation concept in network slicing and related works. Section 3 explains the proposed XVPN-OSM system architecture. In Section 4, we summarize the required steps to implement the architecture. Section 5 illustrates the performance evaluation results of the XVPN-OSM architecture over some of the important interfaces in the Control Plane (CP) and Data Plane (DP). Section 6 concludes the paper. Abbreviations section presents a list of the acronyms used in this paper.

## 2. Background

### *2.1. Isolation Concept*

Isolation is one of the major concerns in separating parallel slices operating on the top of a physical network with shared resources. Isolation can eliminate indirect or direct connections between slices or other entities entangled inside or outside the NSI, including VNFs, end-users and network interfaces.

As summarized in Figure 2, network slice isolation can be considered and analyzed in terms of [7–9]:

- Isolation dimensions: security, dependability and performance;
- Isolation degrees: hardware, physical and logical resources;
- Isolation domains: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN); and
- Management level isolation.

Security isolation demands independent mechanisms preventing any unauthorized admission to modify the slice configuration or management between multiple NSIs, known

as inter-slice isolation, or within one NSI itself, known as intra-slice isolation. Inter-slice isolation involves fully isolating the hardware resources to stop sharing them between slices. It protects the network slices against numerous attacks by precluding the spread of the attack between slices. Intra-slice isolation splits the hardware resources between the involved components inside each slice. It delivers security against attacks by decreasing the influence of the attacks with low recovery time and a high level of resource availability [10]. Other slices should not be influenced in terms of performance and dependability, which extends the dependability dimension. Performance isolation occurs when the required performance by a slice is achieved regardless of the performance degradation and the security issues on the other slices. For instance, the workload, resource number and hardware or software malfunction of one NSI must not degrade the performance of an NF in a different NS or NSI.

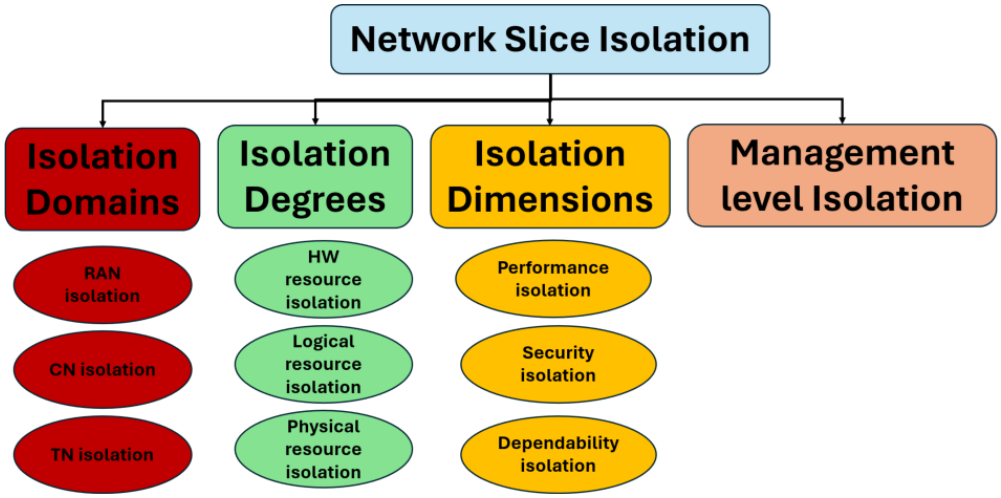


**Figure 2.** Classification of network slice isolation.

Isolation degrees refer to separating physical infrastructure and resources, such as hardware, firewalls, gateways and operating systems. However, hardware and physical isolation are costly and often infeasible, resulting in logical isolation as a straightforward and more affordable separation. Logical isolation can be provided via virtual machines, cloud-based instances, diverse trust zones and hypervisors.

When providing E2E NSIs, slice isolation has to be solved in the RAN, TN and CN domains [11]. In the RAN domain, a specific number/amount of available resources are allocated to diverse network slices and each network slice must not employ any resources allocated to other slices. Some of the RAN resources are antennas, available frequency band(s), subcarriers and physical resource blocks. In the TN domain, traffic flow from a distinct network slice must not be transmitted to another slice. Some of the TN resources include optical fibers, cables and routers. Finally, in the CN domain, all the slices may utilize the same resources; nevertheless, appropriate strategies are needed to perform independent data packet processing in different slices.

Management isolation seeks to oversee and control each slice independently by considering it as an individual and solitary network. Isolation characteristics must be enforced among slices at the virtualization level, creation phase and orchestration stage. Thus, to achieve each slice's QoS requirements, a number of policies and rules, including the isolation essentials, need to be defined and appropriately implemented in order to maintain end-user services' requirements.

In this paper, we study slice isolation in terms of security and performance, over the CN domain and in logical resource degree.

### *2.2. Related Work*

The work in [12] proposes a novel mutual authentication and key establishment protocol utilizing proxy re-encryption. The protocol grants specific authentication between

components of a network slice to enable secure connection and protected key establishment among component pairs for slice security isolation.

Reference [13] offers a two-layer scheduler for efficient and dynamic RAN slicing. The work presents different trade-offs between performance, isolation and priority in dynamic radio resource allocation between RAN slices. The work suggests a strategy based on specific UEs' QoS requirements and the number of demanded PRBs for each UE to establish a RAN slice. Nevertheless, the scheme's complexity grows linearly with the number of created slices pointing to inter-slice scheduling and with the number of UEs within a particular slice referring to intra-slice scheduling.

This work [14] presents a solution to proactively reduce Distributed Denial-of-Service (DDoS) attacks in the CN domain by applying slice isolation. The authors propose implementing a mathematical model to afford on-demand slice isolation and ensure delay for CN slices. The results confirm a reduction in DDoS attacks and also an increment in the availability of CN slices.

In [15], various security challenges of 5G networks are explored and classified. Ensuring isolation of logical resources is crucial to prevent the introduction of new risks. Eavesdropping and tampering with data are among the vectors that an attacker could exploit to undermine security if the application data is not appropriately encrypted. The authors of [16] recommend that encrypted tunnels should be used by operators to establish trust between Service Functions (SFs) to ensure packet integrity and prevent bypassing of policies.

Reference [17] presents a security-aware VNF sharing model for 5G networks, optimizing resource utilization while satisfying service requirements and enhancing slice security. By incorporating novel security constraints, the model allows organized decision-making for VNF sharing. The work in [18] achieves performance optimization via requirement-based slicing using optimal resource allocation. Their proposed approach (slice creation by machine learning, slice isolation through resource allocation and slice management through resource transfer) dynamically allocates resources based on network service requests. Paper [19] shows an architecture for managing secured network slices and focuses on secure service level agreement requirements. The architecture's components, deployment processes and monitoring mechanisms are validated through a DoS attack scenario. Authors in [20] introduce a method to maintain performance and ensure security simultaneously in software-defined networks. By implementing the virtual queue concept over a priority queuing-based pipeline in P4 switches, the study suits Industry 4.0 use cases efficiently. An experimental setup demonstrates the effectiveness of the approach over mixed data paths.

References [21,22] discuss the deployment of VPN using OSM. In reference [21], the authors demonstrate how WireGuard can be incorporated into VNFs and compare the performance of WireGuard and OpenVPN. The PoC is carried out in a single NS with two VNFs and manual configuration of peer connectivity in WireGuard. On the other hand, Ref. [22] proposes the use of IPsec as a VPN solution to establish link-layer connectivity for multi-site deployments. In that work, multiple NSs are deployed using OSM and each NFV Infrastructure (NFVI) is connected through one VNF. These VNFs are responsible for handling the link-layer abstraction of other VNFs. IPsec is used to secure the connection between the VNFs providing the link layer. The operator supplies keys and connection parameters during the instantiation of the NSI.

To the best of our knowledge, none of the SoA works proposes an automated secure service provisioning employing complex and real-life VNFs. The lack of a comprehensive PoC in this field has motivated us to integrate benchmark VPN solutions with the OSM orchestrator granting secure communication between VNFs for automated and realistic network services. Table 1 summarizes the comparison of the related work and our proposed work.

**Table 1.** Comparison of related work and the proposed work in this paper, where ✓denotes that the corresponding work covers the topic and ✗ denotes that the corresponding work does not cover the topic.

| Related Work | 5G SA PoC | Cloud-Native | Isolation Type | SoA VPN Solutions |
|---|---|---|---|---|
| [12] | ✗ | ✗ | Security | ✗ |
| [13] | ✗ | ✗ | Performance | ✓ |
| [14] | ✗ | ✗ | Security | ✗ |
| [15] | ✗ | ✗ | Security | ✓ |
| [16] | ✗ | ✗ | ✗ | ✗ |
| [17] | ✗ | ✗ | Security & performance | ✗ |
| [18] | ✗ | ✗ | Security & performance | ✓ |
| [19] | ✗ | ✗ | Security | ✗ |
| [20] | ✗ | ✗ | Performance | ✗ |
| [21] | ✗ | ✗ | Security & performance | ✓ |
| [22] | ✗ | ✗ | Performance | ✓ |
| [23] | ✗ | ✗ | Security & performance | ✓ |
| **This work** | ✓ | ✓ | Security & performance | ✓ |

## 3. XVPN-OSM System Architecture

In the proposed XVPN-OSM architecture, we use OSM as an orchestrater. In the context of OSM, the terms Day-0, Day-1 and Day-2 operations refer to the three stages of Life-Cycle Management (LCM) of VNFs. These stages are related to the actions needed while providing service automation by utilizing the involved VNFs. Such operations are known as the VNF onboarding process. In this process,

- The Day-0 phase pertains to the instantiation of the VNF, which involves creating and editing charms, developing and validating VNF/NS/NSI Descriptors (VNFD/NSD/NSID) and VNF packaging and emulating;
- The Day-1 phase is focused on initializing the service, which comprises testing, releasing and deploying it;
- And finally, the Day-2 phase deals with the runtime actions of the VNFs, which includes operating and monitoring them.

OSM has three inbuilt supporting applications for LCM, which are Cloud-init, Juju charms and Helm charts. For the Day-0 phase, Cloud-init manages the initial operations, such as setting usernames and passwords. For the Day-1 procedure, Helm charts or Juju charms are used. Day-2 processes are also possible with Juju. Helm charts are used exclusively for Kubernetes-based VNFs (KNFs), while Juju is also functional at the NS level and for VNFs that are not K8s based [24,25]. Juju delivers two operation modes—native charms and proxy charms. Native charms operate processes instantly inside a VNF, while proxy charms employ a centrally placed controller called VNF Configuration and Abstraction (VCA) to handle Day-1 and Day-2 actions. The VCA connects to the VNFs through their management interface and instructs them. The VCA-VNF connection utilizes the SSH protocol by default. We use cloud-init, Helm charts and Juju proxy charms with VCA for OSM onboarding in our implementation. In particular, we employ proxy charms with a VCA installed and integrated with OSM. Accordingly, both OSM and VCA can access the VNFs management interface to perform their particular actions.

Juju allows users to build custom actions using Python scripts. The OSM instance is connected through description files of the VNFs and NSs, along with Juju configuration files that describe metadata and available Day-1 and Day-2 actions. The *charms.osm.sshproxy* library is available in OSM to facilitate the integration of proxy charms. This library handles tasks such as setting up the basic Juju proxy peer. Juju is able to (1) run actions in VNFs and (2) create relations between Juju units for managing and scaling VNFs and handling dependencies between them. In the context of this work, Juju relations are used to

facilitate the transfer of the XVPN peer information between VNFs. This allows for secure communication between different VNFs in the network. We use proxy charms and relations in Juju to create a bridge for transferring information between different VNFs or different involved components of a VNF. Proxy charms are also employed to execute primitives on Helm KNFs. In the VNFDs, we can specify the list of services exposed by the Helm charts and the information of the services is passed to the Proxy charms.

In the context of VPN tunnel establishment, manual key distribution can be a time-consuming process, especially in dynamic environments with multiple interfaces that need to be secured. In addition, the initialization of VPN tunnels can also be delayed until the tenant manager completes the configuration process. Alternatively, the work in [22] proposes an approach that allows the application to send data right after Day-1 actions by inputting the necessary information, including keys. Another strategy is to use a Key Management System (KMS); however, OSM does not support this functionality. Therefore, extra functionality outside the OSM framework must be added in order to use the KMS approach.

In the previous work [23], we implemented the 5G NSA architecture to perform key management using Juju relations and proxy charms for the involved VNFs. Our approach comprises creating unique keys for each new interface deployment using Juju relations. This technique allows the NS to be used instantly after completing Day-1 jobs. With our approach, the private keys are stored only inside the VNFs, ensuring better security. The necessary information for the peer setup, including the public key, is automatically transferred to the peer. In this work, we extend our method to include 5G SA architecture while at the same time providing secure services with three SoA VPN solutions (IPSec, OpenVPN and WireGuard). Our approach helps to customize security for the executed services, as each of the mentioned VPN solutions has its own pros and cons. Thus, depending on the QoS requirements of a service and the available resources, the infrastructure provider can create a tailored service for a specific tenant.

OSM operation relies on its comprehensive information model that aligns with ETSI-NFV. This information model serves as the foundation for automating the complete LCM of NFs (including VNFs/KNFs), network services and network slices. Figure 3 illustrates creating and establishing a network service based on three deployed XVPN KNFs. In our proposed XVPN-OSM framework, the OSM-NBI, in combination with Juju proxy charms, manages the XVPN KNFs running on NFVI via OSM actions. The use of proxy charms with the OSM-NBI allows script configurations on the NFs for a particular VPN solution.

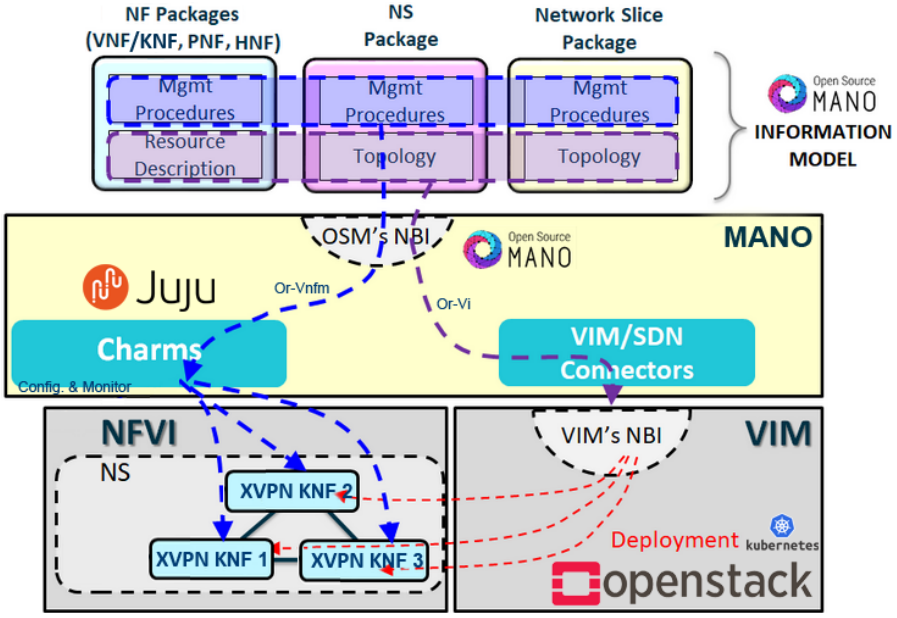


**Figure 3.** The proposed XVPN-OSM framework, where the OSM-NBI and Proxy charms are used to configure different types of NFs deployed on the NFVI [26].

## 4. XVPN-OSM Implementation

In this section, first we give details about the implementation environment, second we explain the development steps and third we summarize the actions for establishing automatic peering.

In order to create a realistic 5G environment, our platform employs open-source software packages. In particular, free5GC [27] is used to set up NSs with 5G SA functionalities. Then, depending on the required QoS, we establish VPN connections on various interfaces to enable XVPN-capable KNFs. In addition, our platform uses OSM [6] to communicate with a multi-site infrastructure consisting of two OpenStack platforms located on different campuses, each serving as a separate VIM [28] (https://github.com/aes5001/5GIIK-testbed,accessedon16April2024).The VIMs host different KNFs, create virtual networks and handle routing for outbound traffic from the KNFs. The VPN tunnels between the KNFs on the NS interfaces are automatically created. In order to overcome the extra low latency challenge for URLLC services, we follow the idea of having MEC by pushing the UPF component of the DP close to the gNB. In our implementation, we deploy gNB and UPF on the same VIM.

We follow these steps to for the deployment phase of XVPNaaS: (1) compose virtualized 5G SA services; (2) set up a mechanism for automatic XVPN peering; (3) structure NSs into NSID, also known as Network Slice Template (NST) and lastly, (4) test the established XVPN connectivities in a multi-site deployment. We further describe the development phase with more details for creating the descriptors In the following:

1. *Composing virtualized 5G SA services:* The system architecture is given in Figure 4. For the sake of simplicity in the illustration, Figure 4 illustrates only one of the possibilities in which an E2E eMBB NSI with WG isolation has been established and the connection points and virtual links are not drawn. We implement the 5G network's main components according to the free5GC solution, which includes:
   - Virtualized version of UE generating real-life traffic along with gNB in the RAN domain;
   - User Plane Function (UPF) in the Edge domain;
   - Authentication Server Function (AUSF), Network Slice Selection Function (NSSF), Policy Control Function (PCF), Network Repository Function (NRF), Unified Data Repository (UDR), Access and Mobility Management Function (AMF), Session Management Function (SMF), Unified Data Management (UDM) and MongoDB in the CN domain.

   When connecting the UEs to the gNB, the air interface is simulated, resembling the real wireless link. Then, we verify that the diverse Edge and CN components operate as anticipated and deliver service to the UEs. The UEs then connect to an external network through the UPF and via the gNB. In this first implementation step, we still need to include a VPN solution between the involved components. Our architecture builds the NS by spreading the service components into separate KNFs. This approach allows splitting the KNFs into two separate Kubernetes clusters called RAN/Edge cluster and Core cluster. These two Kubernetes clusters are deployed on two OpenStack VIMs. Such a multi-VIM environment allows us to emulate cases in which the Edge component (UPF) can be deployed closer to the end-users. This policy enables us to mimic the MEC technology in order to overcome the latency requirement in URLLC use cases. The KNFs distributed to the remote site can communicate with the CN securely with the help of an established VPN solution.
2. *Automatic XVPN Peering*: Manually setting up VPN tunnels between several interfaces can be time-consuming. Thus, we use Juju relations for automatic peering, with no extra information given to the other end of the peer at the time of instantiating the NS. The first step in the automatic peering is the establishment of relationships between KNFs on both sides. Then the paired KNFs retrieve information like a public key, endpoint and listening port to communicate with each other. To establish

XVPN connectivity on all interfaces, we changed the IP address configuration in the components. Changing the interface addresses is necessary to route application data over the XVPN tunnel and, at the same time to ensure that applications inside the KNF have been installed and started correctly, even when waiting for the tunnel establishment. Besides, to verify that the NS runs a specific VPN solution, we connect UEs and observe that they connect to the Data Network (an external network such as the Internet) and they get service.

3. *NST creation*: After having a working NS with XVPN connectivity between the interfaces, we include it in two NSTs, for eMBB and for URLLC services, to observe if and how the performance is affected by providing security with each VPN solution. The NSTs have different values of quality indicators corresponding to different 5G QoS Identifiers (5QIs) [29]. The QoS parameters correspond to eMBB and URLLC use cases, respectively. Further, the NSTs are prepared with only the management interfaces of the KNFs. The management interfaces are attached to the external connection points in the NSTs.
4. *Multi-VIM deployment*: When using OpenStack, the external floating IP address is, by default, unknown inside a VM or a container. However, the VCA can retrieve the management IP address to perform its actions. To find the floating IP addresses of the KNFs, we use the same function that Juju employs for its *proxypeer* connection between a Juju unit at the VCA and the Kubernetes Deployment Unit (KDU) in the KNF. After the endpoint IP address is found, the 5G SA components on separate VIMs connect automatically with the created VPN connectivity. A requirement for Multi-VIM connectivity is to deploy a port opened in the firewalls.

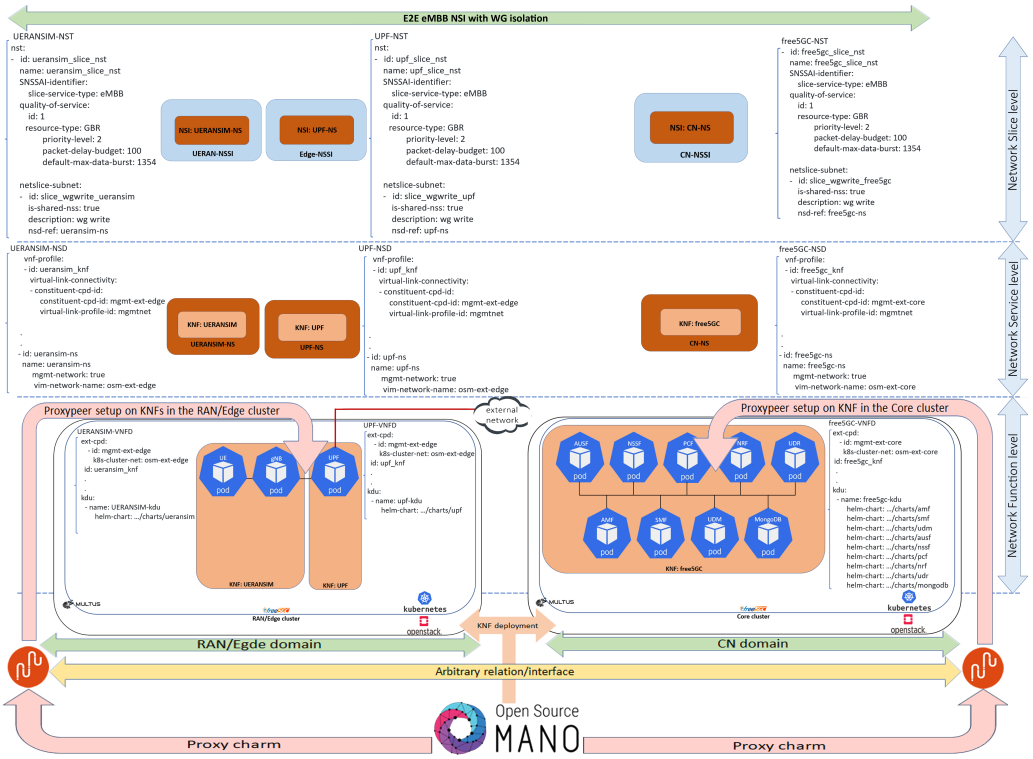


**Figure 4.** An E2E eMBB NSI with WG isolation created on XVPN-OSM architecture.

We next summarize the required actions to implement XVPNaaS with automatic peering:

1. Append installation of a specific VPN solution (IPSec, OpenVPN or WireGuard) in cloud-init.
2. Include name and parameters for Day-1 and Day-2 actions in the actions.yaml file.
3. Implement relations between VNFs in the metadata.yaml file.

4. Include the Python code to append the charm script. The name of the relationship must correspond to the name used in metadata.yaml and the listener in the *__init__* function of the Python script.
5. Implement the actions from actions.yaml into Day-1 and Day-2 operations in the VNFDs. The relevant actions need to be implemented in the *initial-config-primitive* section in the VNFDs in order to create the XVPN tunnel as a Day-1 operation. Day-2 actions are positioned in the *config-primitive* section.
6. Employ Day-2 actions for additional configuration and maintenance (establishing a new connection to a VNF) while the default implementation sets up the XVPN.

## 5. Results

In this section, we evaluate the performance of delivering eMBB and URLLC services and discuss the observed results. We execute measurement tests in both the CP and DP. We employ both arbitrary data and the UEs generate traffic in the network. We monitor the impact of integrating secure communication with SoA VPN solutions on the performance metrics that should be aligned with the 5G KPI [30].

While creating arbitrary data for high network load, we measure throughput and latency in the CP and DP. In particular, the following tasks are accomplished to test the performance of eMBB and URLLC NSIs:

- Measuring throughput and latency in the DP over the N3 interface (which connects gNB and UPF);
- Measuring throughput and latency between components in the CP over the N2 interface (which connects gNB and AMF), N8 interface (which connects AMF and UDM) and N12 interface (which connects AMF and AUSF).

According to the 3GPP Release 18 [31], the E2E latency requirements in the DP for the URLLC and eMBB use cases are 1 ms and 4 ms, respectively, while the E2E latency for both use cases in the CP is 10 ms. The E2E latency consists of the processing time on the transmitter/receiver side, air transmission delay on both sides and network processing time [32,33]. It is also considered that the air latency is about 10–20% of E2E latency for both use cases [34]. We measure the network processing time.

### *5.1. Lab Environment*

The primary VIM is an HP server running OpenStack with resources of 56 vCPUs, 126 GB RAM and 915 GB storage. The second VIM, used for multi-site deployment, also runs OpenStack with the same resources. Kubernetes is then installed and established over the VIMs. For the KNFs to communicate securely across the VIMs, an XVPN tunnel is established between the NFVIs. A nested XVPN tunnel is used on the N2 interface for the multi-VIM deployment. The internal throughput of the NFVIs, where the primary VIM runs, is 20 Gbps. The following information gives a summary of our lab environment follows the deployment guidelines proposed by free5GC [35]:

- Kubernetes cluster (RAN/Edge cluster) for the DP components with all worker nodes operating kernel 5.0.0-23-generic, with gtp5g kernel module;
- Kubernetes cluster (Core cluster) for the CP components with a persistent volume provisioner and SCTP support;
- Multus-Container Network Interface (Multus-CNI) deployed on each cluster;
- Helm3 to communicate with each cluster;
- A physical network interface on each Kubernetes node on the two clusters; and
- A physical network interface on each Kubernetes node on the DP cluster to connect the UPF to the Data Network.

It is worth noting that the KNFs' features demand operative Kubernetes clusters. In our architecture, the Kubernetes clusters combined with the OSM enable the deployment of microservices in a cloud-native fashion. The Kubernetes clusters must be associated with the VIMs (located inside the OpenStacks) in order to manage the networks and facilitate

the connection to the infrastructure. As requirements, both Kubernetes clusters need to have (1) a LoadBalancer to expose the KNFs to the network and (2) a default storage class to support persistent volumes [36]. Besides, in the presented architecture, we deploy the case in which Proxy charm and Helm chart are not located in the same Kubernetes cluster. Thus, with the help of the LoadBalancer, the Proxy charm can reach the services. This can be done via the *osm-libs Charm* library.

### *5.2. Discussion*

The following results are based on running 40 tests (20 Iperf3 tests, 20 ICMP tests), each running for approximately 30 min. The figures show the results of the mean value of the metrics with a confidence interval of 95%.

A comparison of the latency measurement for URLLC NSIs over the interfaces in the DP and CP is demonstrated in Figure 5. The horizontal red line represents the 1 ms E2E latency requirement of the DP for URLLC NSIs in 5G. As it can be seen from the figure, the latency of the *N*3 interface in the DP increases from 0.4 ms in the case of no isolation to 0.51 ms when we add the WireGuard (WG) isolation to the URLLC NSI. The WG isolation introduces a visible overhead. Nevertheless, as mentioned earlier, we still have a reasonable safe margin to the red boundary line, confirming that the WG isolation can be a promising solution in the URLLC use case. However, IPSec and OpenVPN isolation results over the same interface, *N*3, show that the corresponding latency values cross the horizontal red line by reaching 1.8 ms and 3.6 ms, respectively. Accordingly, IPSec and OpenVPN isolation are not the best solution, at least not for the URLLC DP.

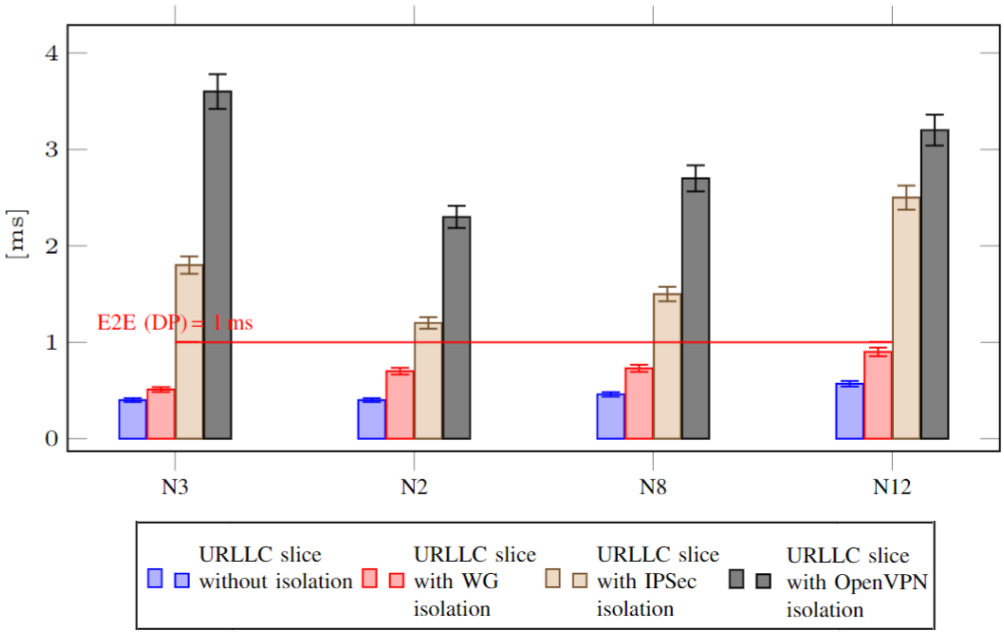


**Figure 5.** Latency comparison at different interfaces in DP and CP while operating an individual URLLC NSI with 2 URLLC UEs per slice.

Regarding the *N*2, *N*8 and *N*12 in the CP, there is no strict requirement of having less than 1 ms, but less than 10 ms as a more loose latency requirement compared to the DP. The observed results also verify that the latency increases over the mentioned interfaces in the CP, but still, the introduced overheads do not result in a latency higher than 10 ms. Again WG isolation outperforms IPSec and OpenVPN isolations in terms of providing less latency in both DP and CP.

The latency measurement for URLLC and eMBB NSIs over the interfaces in the DP and CP is presented in Figure 6. The objective here is to evaluate the system performance in terms of scalability when diverse slices operate simultaneously. In particular, we establish multiple non-similar NSIs (eMBB and URLLC) where each slice is serving two UEs.

The horizontal red line still shows the 1 ms E2E latency requirement of the DP for URLLC NSIs. As the figure illustrates, the latency of the *N*3 interface in the DP of URLLC increases from 0.57 ms for WG isolation to 4.3 ms and 5.8 ms for IPSec and OpenVPN isolations, respectively. In the case of the *N*3 interface in the DP of eMBB, the latency again rises from 1.6 ms in WG isolation to 6.6 ms in IPSec isolation and 7.7 ms in OpenVPN isolation. As observed from the results in the DP of URLLC and eMBB, WG isolation still outshines the other competitors. WG isolation can deliver both eMBB and URLLC with a delay value less than the E2E delay requirements of the use cases. Moving on to the CP, although it is not crucial, for the URLLC use case, WG exhibits a delay of 0.6 ms, 0.75 ms and 0.78 ms for *N*2, *N*8 and *N*12, respectively.

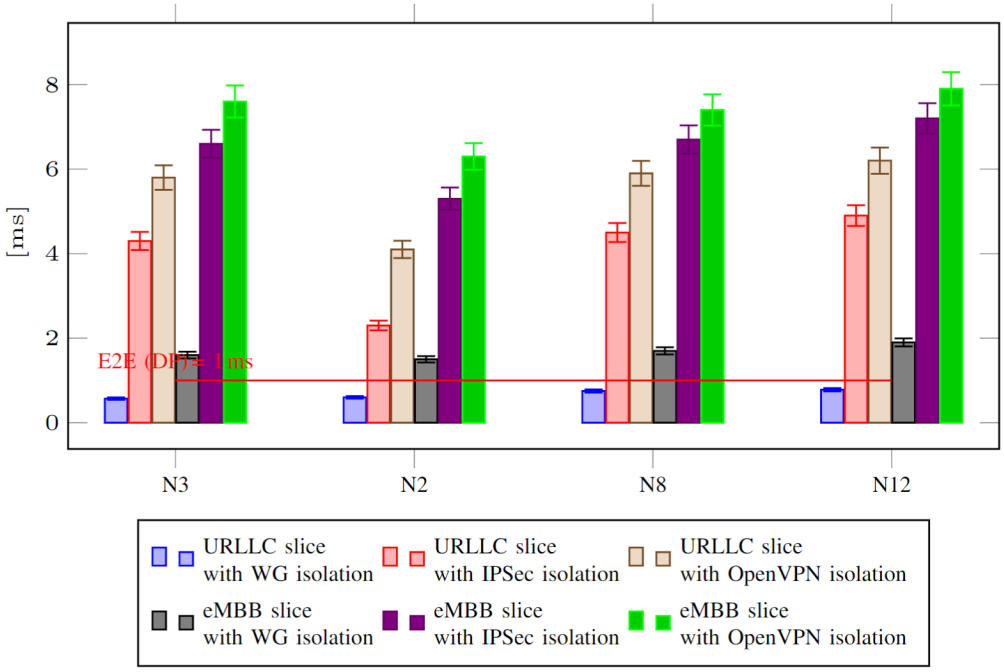


**Figure 6.** Latency comparison of different interfaces in DP and CP while operating two simultaneous URLLC and eMBB NSIs with 2 URLLC UEs and 2 eMBB UEs per slice.

Figure 7 compares the throughput between the involved interfaces in the DP and CP for URLLC and eMBB NSIs with different VPN solutions. The slices are operating simultaneously and each slice serves two UEs. The red line represents the 100 Mbps downlink user data rate KPI in 5G. Regarding the *N*3 interface in the DP, we can see that the throughput with the WG isolation can reach up to 1050 Mbps, which is a very promising throughput value for the eMMB use case. Although the URLLC use case does not need as high throughput as eMBB, we can still see the result of the URLLC slice with WG isolation reaching up to 740 Mbps. IPSec and OpenVPN hold second and third place in achieving high throughput for both use cases in the DP. All solutions reach higher throughput than the bottom red line.

Regarding the *N*2, *N*8 and *N*12 in the CP, the observed results show that WG and IPSec isolations are the main competitors in providing higher throughput values. However, WG isolation is a better choice. For example, in the case of *N*8 for the eMBB use case, we can see a throughput of 610 Mbps and 533 Mbps for the WG and IPSec isolations, respectively. In all cases, the OpenVPN option is the least promising choice.

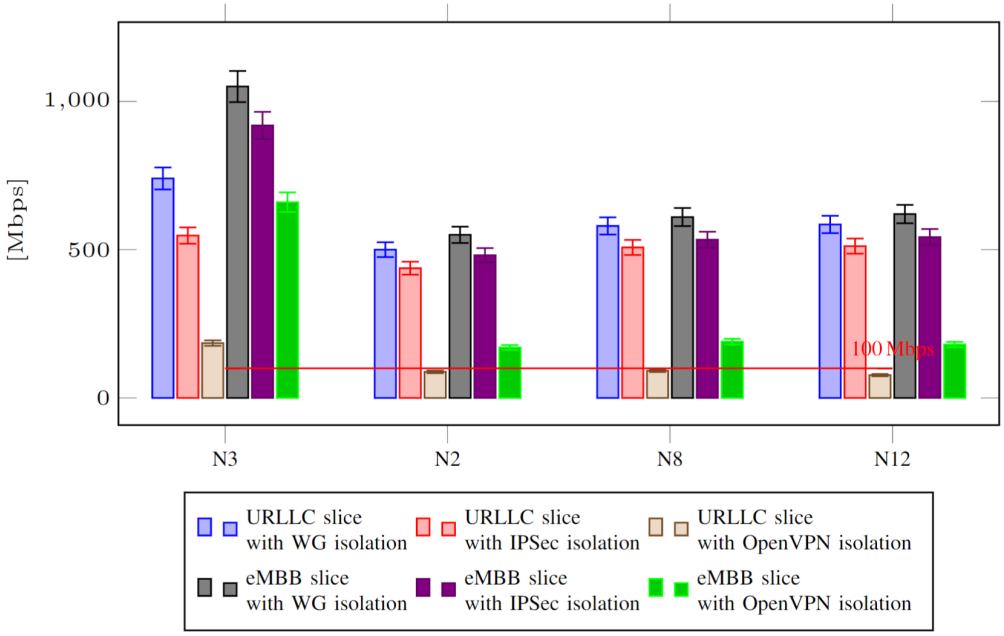


**Figure 7.** Throughput comparison of different interfaces in DP and CP while operating two simultaneous URLLC and eMBB NSIs with two UEs per slice.

Figure 8 compares the throughput between the involved interfaces in the DP and CP for one URLLC and two eMBB NSIs with different VPN solutions. We establish one Guaranteed Bit Rate (GBR) eMBB and one Non-GBR eMBB. The slices are operating simultaneously and each slice serves 5 UEs. The GBR eMBB is a fixed NSI that delivers the exact requested throughput. The Non-GBR eMBB operates as a best-effort NSI, meaning that depending on the available resources of the slice, they are allocated to the corresponding users in the slice. In the implementation, the URLLC has the highest priority, the GBR eMBB has the second priority and the Non-GBR eMBB gets the best-effort slices. Regarding the *N*3 interface in the DP, we notice that, firstly, the system starts serving the URLLC UEs as the highest priority. Since they usually generate short packets, they are served immediately and the throughput reaches up to 34 Mbps for the URLLC slice with WG isolation. Secondly, as the challenging part, the system needs to satisfy the GBR eMBB users to obtain the demanded resources while also serving other users. As it can be seen, the throughput of the *N*3 interface with the WG isolation can attain up to 520 Mbps for the GBR eMMB use case, which is still a promising value and higher than the red line. IPSec solution can achieve up to 260 Mbps for the GBR eMMB use case. WG and IPSec solutions can obtain 90 Mbps and 66 Mbps, respectively, for the Non-GBR eMBB use case. Here, again, OpenVPN shows the worst performance.

The reported results indicate that WG and IPSec isolations are the leading solutions for the *N*2, *N*8 and *N*12 in the CP.

Overall, according to the observed results and considering the scalability of providing the demanded services, WG is seen as the most promising solution in providing performance isolation on eMBB and URLLC use cases over the DP and CP.

Before introducing the XVPN tunnels, we were able to capture connection information such as the International Mobile Subscriber Identity (IMSI), network realms and hostnames at the VIMs. However, after establishing the XVPN connectivity, the only observable information at the VIMs is employing each specific VPN protocol and link-layer discovery messages. Hence, all data except link layer discovery ARP messages are encapsulated and transferred inside the established XVPN tunnel which results in security isolation.

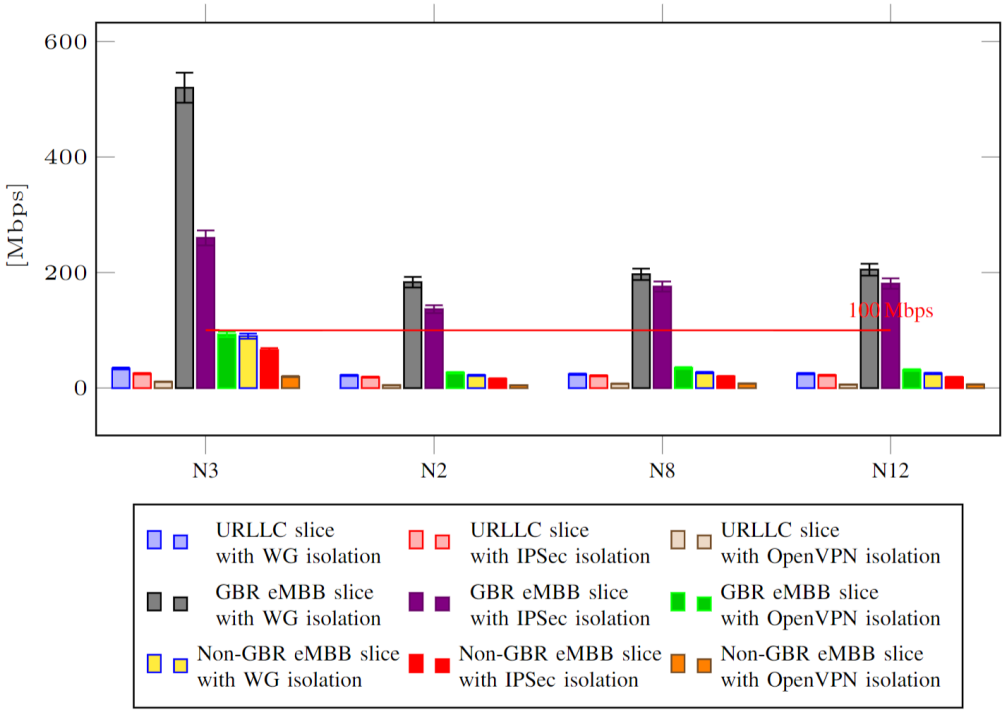


**Figure 8.** Throughput comparison of different interfaces in DP and CP while operating three simultaneous; one URLLC and two eMBB NSIs with 5 UEs per slice.

## 6. Conclusions

We presented an implementation of XVPNaaS within the context of network slicing in 5G networks and addressed the critical challenge of ensuring security and performance isolation. By deploying SoA VPN solutions such as WireGuard, IPSec and OpenVPN, we investigated their efficacy in enhancing security and performance when isolating slices within a physical infrastructure. Our findings indicate that WireGuard, with its automatic peer setup and efficient encryption mechanisms, outperforms IPSec and OpenVPN in terms of both throughput and latency, particularly for eMBB and URLLC slices, respectively. This spotlights WireGuard's potential as a promising solution for providing secure and efficient network slicing in accordance with 5G KPIs. Moreover, the implementation of XVPNaaS using Juju relations offers a practical and orchestrated platform for achieving isolation in network slicing. Beyond the technical distinctions, the practical implications of our research extend to various sectors and industries whose operations highly rely on isolated slices in 5G networks. Extra low latency and very high throughput operations are critical characteristics for industries such as healthcare and autonomous vehicles. Consequently, such industries can significantly benefit from the isolation offered by XVPNaaS in network slicing while simultaneously satisfying their low latency and high throughput requirements. Nevertheless, in the practical implementation, consideration must be given to hardware limitations and potential challenges under real-world conditions where certain constraints may still arise. Firstly, while resources may seem ample for initial PoC testing and small-scale deployments, they may demonstrate insufficient when scaling up to fulfill the demands of larger networks or high-traffic scenarios. Hence, elements such as network latency, bandwidth limitations and hardware failures in large-scale networks could impact performance and reliability. Furthermore, the complexity of orchestrating VNFs across distributed environments and ensuring seamless communication between the involved components of VNFs may introduce operational challenges. Accordingly, further research and experimentation are needed to address these practical constraints for robust implementation in real-world 5G networks.

There are several directions for future work. Firstly, further optimization and fine-tuning of the XVPNaaS framework can be followed to maximize its efficiency and scalability

in large-scale 5G deployments. Additionally, investigating more use cases and industry-specific applications of XVPNaaS can deliver valuable insights into its broader applicability and potential benefits. Furthermore, as the landscape of telecommunications resumes to evolve, future research directions may involve integrating emerging technologies such as edge computing and artificial intelligence with slice isolation. Investigating the implications of XVPNaaS in MEC architectures and network slicing orchestration frameworks can contribute to a more in-depth knowledge of its role in shaping the future of 5G networks.

In conclusion, this study highlights the significance of XVPNaaS as a pivotal component in ensuring the security and performance of network slicing in 5G networks. By leveraging advancements in VPN technology and orchestration frameworks, we pave the way for a more secure, efficient and adaptable telecommunications infrastructure that can satisfy the various requirements of industries and sectors in the era of 5G and Beyond 5G connectivity.

**Author Contributions:** Conceptualization, A.E. and K.K.; methodology, A.E.; software, A.E.; validation, A.E.; formal analysis, A.E.; investigation, A.E.; resources, A.E.; data curation, A.E.; writing—original draft preparation, A.E.; writing—review and editing, A.E. and K.K.; visualization, A.E.; supervision, K.K.; project administration, A.E. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** All data underlying the results are available as part of the article and no additional source data are required.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

List of the used acronyms in this paper.

| Abb. | Definition | Abb. | Definition |
|---|---|---|---|
| 5G | fifth-generation | API | Application Programming Interface |
| AMF | Access and Mobility Management Function | AUSF | Authentication Server Function |
| CN | Core Network | CP | Control Plane |
| CNI | Container Network Interface | DDoS | Distributed Denial-of-Service |
| DP | Data Plane | E2E | End-to-End |
| GBR | Guaranteed Bit Rate | IMSI | International Mobile Subscriber Identity |
| KDU | Kubernetes Deployment Unit | KMS | Key Management System |
| KNF | Kubernetes-based VNF | KPI | Key Performance Indicator |
| LCM | Life-Cycle Management | MANO | Management and Orchestration |
| MEC | Multi-access Edge Computing | MNO | Mobile Network Operator |
| NBI | Northbound Interface | NFV | Network Function Virtualization |
| NFVI | NFV Infrastructure | NS | Network Service |
| NSD | NS Descriptor | NSI | Network Slice Instance |
| NSID | NSI Descriptor | NSSF | Network Slice Selection Function |
| NRF | Network Repository Function | PCF | Policy Control Function |
| PoC | Proof-of-Concept | QoS | Quality-of-Service |
| QI | QoS Identifier | RAN | Radio Access Network |
| SA | Standalone | SF | Service Function |
| SDN | Software-Defined Networking | SMF | Session Management Function |
| SSH | Secure Shell | SSL | Secure Socket Layer |
| SSTP | Secure Socket Tunneling Protocol | TN | Transport Network |
| TLS | Transport Layer Security | UPF | User Plane Function |
| URLLC | Ultra-Reliable Low-Latency Communications | VCA | VNF Configuration and Abstraction |
| VM | Virtual Machine | VNFD | VNF Descriptor |
| VIM | Virtualized Infrastructure Manager | VPN | Virtual Private Network |
| VPNaaS | VPN-as-a-Service | WG | WireGuard |

## References


1. Blanco, B.; Fajardo, J.O.; Giannoulakis, I.; Kafetzakis, E.; Peng, S.; Pérez-Romero, J.; Trajkovska, I.; Sayyad Khodashenas, P.; Goratti, L.; Paolino, M.; et al. Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN. *Comput. Stand. Interfaces* **2017**, *54*, 216–228. [CrossRef]
2. Kotulski, Z.; Nowak, T.; Sepczuk, M.; Tunia, M.; Artych, R.; Bocianiak, K.; Osko, T.; Wary, J. On end-to-end approach for slice isolation in 5G networks. Fundamental challenges. In Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 3–6 September 2017; pp. 783–792.
3. Kotulski, Z.; Nowak, T.W.; Sepczuk, M.; Tunia, M.A. 5G networks: Types of isolation and their parameters in RAN and CN slices. *Comput. Netw.* **2020**, *171*, 107135. [CrossRef]
4. Schneider, P.; Mannweiler, C.; Kerboeuf, S. Providing strong 5G mobile network slice isolation for highly sensitive third-party services. In Proceedings of the IEEE Wireless Comm. and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6.
5. Whitestack. *Open Multi-Vendor NFV Showcase*, 1st ed.; Technical Report; ETSI: Sophia Antipolis, France, 2019.
6. ETSI OSM. Deploying a 5G Network. Available online: https://osm.etsi.org/ (accessed on 3 July 2023).
7. Gonzalez, A.J.; Ordonez-Lucena, J.; Helvik, B.E.; Nencioni, G.; Xie, M.; Lopez, D.R.; Grønsund, P. The isolation concept in the 5G network slicing. In Proceedings of the 2020 European Conference on Networks and Communications (EuCNC), Dubrovnik, Croatia, 15–18 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 12–16.
8. Kotulski, Z.; Nowak, T.W.; Sepczuk, M.; Tunia, M.; Artych, R.; Bocianiak, K.; Osko, T.; Wary, J.P. Towards constructive approach to end-to-end slice isolation in 5G networks. *EURASIP J. Inf. Secur.* **2018**, *2018*, 2. [CrossRef]
9. Cominardi, L.; Deiss, T.; Filippou, M.; Sciancalepore, V.; Giust, F.; Sabella, D. MEC support for network slicing: Status and limitations from a standardization viewpoint. *IEEE Commun. Stand. Mag.* **2020**, *4*, 22–30. [CrossRef]
10. Salahdine, F.; Liu, Q.; Han, T. Towards secure and intelligent network slicing for 5g networks. *IEEE Open J. Comput. Soc.* **2022**, *3*, 23–38. [CrossRef]
11. Gligoroski, D.; Kralevska, K. Expanded Combinatorial Designs as Tool to Model Network Slicing in 5G. *IEEE Access* **2019**, *7*, 54879–54887. [CrossRef]
12. Sathi, V.N.; Srinivasan, M.; Thiruvasagam, P.K.; Chebiyyam, S.R.M. A Novel Protocol for Securing Network Slice Component Association and Slice Isolation in 5G Networks. In Proceedings of the Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWIM '18, Montreal, QC, Canada, 28 October–2 November 2018; pp. 249–253.
13. Marabissi, D.; Fantacci, R. Highly Flexible RAN Slicing Approach to Manage Isolation, Priority, Efficiency. *IEEE Access* **2019**, *7*, 97130–97142. [CrossRef]
14. Sattar, D.; Matrawy, A. Towards Secure Slicing: Using Slice Isolation to Mitigate DDoS Attacks on 5G Core Network Slices. In Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019; pp. 82–90.
15. Kim, H. 5G core network security issues and attack classification from network protocol perspective. *J. Internet Serv. Inf. Secur.* **2020**, *10*, 1–15.
16. Hantouti, H.; Benamar, N.; Taleb, T. Service Function Chaining in 5G amp; Beyond Networks: Challenges and Open Research Issues. *IEEE Netw.* **2020**, *34*, 320–327. [CrossRef]
17. Mahyoub, M.; Abdulghaffar, A.; Alalade, E.; Matrawy, A. A Security-aware Network Function Sharing Model for 5G Slicing. *arXiv* **2023**, arXiv:2303.03492.
18. Srinivasan, T.; Venkatapathy, S.; Jo, H.G.; Ra, I.H. VNF-Enabled 5G Network Orchestration Framework for Slice Creation, Isolation and Management. *J. Sens. Actuator Netw.* **2023**, *12*, 65. [CrossRef]
19. Alemany, P.; Molina, A.; Dangerville, C.; Asensio, R.; Ayed, D.; Muñoz, R.; Casellas, R.; Martínez, R.; Skarmeta, A.; Vilalta, R. Management and enforcement of secured E2E network slices across transport domains. *Opt. Fiber Technol.* **2022**, *73*, 103010. [CrossRef]
20. Chang, C.Y.; Ruiz, T.G.; Paolucci, F.; Jiménez, M.A.; Sacido, J.; Papagianni, C.; Ubaldi, F.; Scano, D.; Gharbaoui, M.; Giorgetti, A.; et al. Performance isolation for network slices in industry 4.0: The 5growth approach. *IEEE Access* **2021**, *9*, 166990–167003. [CrossRef]
21. Haga, S.; Esmaeily, A.; Kralevska, K.; Gligoroski, D. 5G Network Slice Isolation with WireGuard and Open Source MANO: A VPNaaS Proof-of-Concept. In Proceedings of the 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Leganes, Spain, 10–12 November 2020; pp. 181–187. [CrossRef]
22. Vidal, I.; Nogales, B.; Lopez, D.; Rodríguez, J.; Valera, F.; Azcorra, A. A Secure Link-Layer Connectivity Platform for Multi-Site NFV Services. *Electronics* **2021**, *10*, 1868. [CrossRef]
23. Kielland, S.; Esmaeily, A.; Kralevska, K.; Gligoroski, D. Secure Service Implementation with Slice Isolation and WireGuard. In Proceedings of the 2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, Greece, 5–8 September 2022; pp. 148–153. [CrossRef]
24. ETSI OSM. ETSI-NFV-NSD. Available online: http://osm-download.etsi.org/repository/osm/debian/ReleaseTEN/docs/osm-im/osm_im_trees/etsi-nfv-nsd.html (accessed on 3 July 2023).
25. ETSI OSM. ETSI-NFV-VNFD. Available online: https://tinyurl.com/2p9yp7cr (accessed on 3 July 2023).

26. ETSI Community. *Osm Scope, Functionality, Operation and Integration Guidelines;* Standard Issue 1; European Telecommunications Standards Institute: Sophia Antipolis, France, 2019.
27. Technical Team. Deploying a 5G Network. Available online: https://free5gc.org/ (accessed on 3 July 2023).
28. Esmaeily, A.; Kralevska, K.; Gligoroski, D. A Cloud-based SDN/NFV Testbed for End-to-End Network Slicing in 4G/5G. In Proceedings of the 2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, 29 June–3 July 2020; pp. 29–35. [CrossRef]
29. ETSI. *System architecture for the 5G System (5GS)*; Technical Report TS 123 501 V16.6.0; ETSI: Sophia Antipolis, France, 2020.
30. ETSI. Why Do We Need 5G? Available online: https://www.etsi.org/technologies/mobile/5g (accessed on 3 July 2023).
31. 3GPP. *Study on Scenarios and Requirements for Next Generation Access Technologies*; V18.0.0; 3GPP: Sophia Antipolis, France, 2024.
32. Fettweis, G.; Boche, H.; Wiegand, T.; Zielinski, E.; Schotten, H.; Merz, P.; Hirche, S.; Festag, A.; Häffner, W.; Meyer, M.; et al. *The Tactile Internet-ITU-T Technology Watch Report*; International Telecommunication Union (ITU): Geneva, Switzerland, 2014.
33. Mohammed, N.A.; Mansoor, A.M.; Ahmad, R.B. Mission-critical machine-type communication: An overview and perspectives towards 5G. *IEEE Access* **2019**, *7*, 127198–127216. [CrossRef]
34. 5Gamericas. New Services and Applications With 5G Ultra-Reliable Low Latency Communications. Available online: https://www.5gamericas.org/wp-content/uploads/2019/07/5G_Americas_URLLLC_White_Paper_Final__updateJW.pdf (accessed on 16 April 2024).
35. Khichane, A. Setup free5GC on Multiple Clusters. Available online: https://github.com/free5gc/free5gc. (accessed on 3 July 2023).
36. ETSI OSM. OSM Usage. Available online: https://osm.etsi.org/docs/user-guide/latest/05-osm-usage.html#osm-usage. (accessed on 3 July 2023).