IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 28, 2020, accepted January 4, 2021, date of publication January 8, 2021, date of current version January 20, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3050155*

# Cloud-Native Network Slicing Using Software Defined Networking Based Multi-Access Edge Computing: A Survey

**SYED DANIAL ALI SHAH, MARK A. GREGORY, (Senior Member, IEEE), AND SHUO LI, (Member, IEEE)**
School of Engineering, RMIT University, Melbourne, VIC 3000, Australia

Corresponding author: Mark A. Gregory (mark.gregory@rmit.edu.au)

This work was supported by Royal Melbourne Institute of Technology (RMIT) University Research Stipend Scholarship (RRSS).

**ABSTRACT** Fifth-Generation (5G) mobile cellular networks provide a promising platform for new, innovative and diverse IoT applications, such as ultra-reliable and low latency communication, real-time and dynamic data processing, intensive computation, and massive device connectivity. End-to-End (E2E) network slicing candidates present a promising approach to resource allocation and distribution that permit operators to flexibly provide scalable virtualized and dedicated logical networks over common physical infrastructure. Though network slicing promises the provision of services on demand, many of its use cases, such as self-driving cars and Google's Stadia, would require the integration of a Multi-Access Edge Computing (MEC) platform in 5G networks. Edge Computing is envisioned as one of the key drivers for 5G and Sixth-Generation (6G) mobile cellular networks, but its role in network slicing remains to be fully explored. We investigate MEC and network slicing for the provision of 5G service focused use cases. Recently, changes to the cloud-native 5G core are a focus with MEC use cases providing network scalability, elasticity, flexibility, and automation. A cloud-native microservices architecture, along with its potential use cases for 5G network slicing, is envisioned. This paper also elaborates on the recent advances made in enabling E2E network slicing, its enabling technologies, solutions, and current standardization efforts. Finally, this paper identifies open research issues and challenges and provides possible solutions and recommendations.

**INDEX TERMS** Network slicing, software defined networking, multi-access edge computing, cloud native, ultra-reliable, and low latency communication.

## I. INTRODUCTION

AS envisioned by the network operators, Fifth Generation (5G) mobile cellular networking takes communications closer to the vision of the Internet of Everything (IoE) [1], [2]. 5G networks are envisioned to support not only the Internet of Things (IoT) but also the emerging vertical industries [2]. IoT demands support for a diverse set of services such as smart cities, eHealth, smart buildings, Internet of Vehicles (IoV), and so on. The rapid growth of IoT alone means that billions of devices will be connected to the network over the next decade.

The requirements for IoT-enabled smart cities are diverse. Services, such as smart grids, intelligent traffic light management, smart households, and smart agriculture, will require a massive number of connections, and therefore, high transit bandwidth will be needed for the aggregated traffic. Mission-critical services such as autonomous vehicles, Vehicle-to-Vehicle (V2V) cooperative driving, remote health monitoring, and industrial control will require Ultra-Reliable and Low Latency Communication (URLLC). The heterogeneous and diverse requirements for future smart cities indicate that current network designs based on the conventional approach of "one-size-fits-all" will no longer be appropriate and 5G network design should reflect the need for scalable and flexible network designs. 5G network architectures need to evolve to provide service diversity, guaranteed performance, and a short time to market to ensure that there is support for the deployment of new services, resource allocation, reduced Capital Expenditure (CAPEX), services automation, and convergence of fixed and mobile access.

The associate editor coordinating the review of this manuscript and approving it for publication was Noor Zaman.

**TABLE 1. A list of commonly used acronyms in this paper.**

| Abb. | Defination | Abb. | Defination | Abb. | Defination | Abb. | Defination |
|---|---|---|---|---|---|---|---|
| **NFV** | Network Function Virtualization | **VNFs** | Virtual Network Functions | **NSIs** | Network Slice Instances | **CUPS** | Control and User Plane Separation |
| **NFs** | Network Functions | **AMF** | Access Management Function | **SMF** | Session Management Function | **UPF** | User Plane Function |
| **PCF** | Policy Control Function | **UDM** | Unified Data Management | **AUSF** | Authentication Server Function | **NSSF** | Network Slice Selection Function |
| **NRF** | Network Resource Functions | **MEC** | Multi-Access Edge Computing | **MEP** | MEC Platform | **NST** | Network Slice Template |
| **TSCI** | Transport Slice Connectivity Interface | **API** | Application Programming Interface | **RNIS** | Radio Network Information Service | **ITU** | International Telecommunication Union |
| **NEF** | Network Exposure Function | **MEO** | MEC Orchestrator | **MEH** | MEC Hosts | **AF** | Application Function |
| **V2X** | Vehicle to Everything | **PNFs** | Physical Network Functions | **SLAs** | Service Level Agreements | **RSVP** | Resource Reservation Protocol |
| **DCs** | Data Centers | **SR** | Segment Routing | **TS** | Transport Slices | **BR** | Border Routers |
| **SBA** | Service Based Architecture | **DiffServ** | Differentiated Services | **CNFs** | Cloud-Native Network Functions | **E2E** | End-to-End |
| **vRAN** | Virtual Radio Access Network | **NFVI** | NFV Infrastructure | **UAVs** | Unmanned Aerial Vehicles | **VM** | Virtual Machine |
| **URLLC** | Ultra-Reliable Low-Latency Communication | **3GPP** | 3rd Generation Partnership Project | **5GPP** | 5G Infrastructure Public Private Partnership | **ETSI** | European Telecommunication Standards Institute |

The diversity and flexibility expectations for 5G networks raises an important challenge to provide service flexibility whilst enabling network technology diversity. To overcome the challenges, End-To-End (E2E) network slicing is a potential key enabler technology that supports customized network services through provisioning of on demand Network Slice Instances (NSIs).

The network slicing concept emerged as a result of recent advancements in cloud computing and Network Function Virtualization (NFV). Network slicing is the slicing of physical network infrastructure resources into dedicated logical networks, thus facilitating vertical segmentation of networks, services and applications [3]. The logical or dedicated networks can be used to provision tailored solutions for distinct service types and application scenarios.

NFV is a key enabling technology for 5G network slicing as it permits the creation and instantiation of isolated or partially shared NSIs by abstracting the virtual and physical infrastructure resources, and offering customized configurations and policy to dedicated logical resources. The logical resources or networks are then assigned to a vertical application, that may include providing Virtualized Network Functions (VNFs). The VNFs are interconnected through well defined logical or virtual links to form fully-fledged NSI.

Software Defined Networking (SDN) is also a key enabler for E2E network slicing. It is a networking paradigm that separates the control and data planes. SDN controllers provide centralized management and a global network topology view that increases the efficiency of network traffic flow related decision making. SDN supports flexible programmatic operation of the control plane, including rapid deployment of new and updated network applications, traffic steering, mobility management for wireless mobile stations, and traffic rerouting for congestion avoidance. It allows efficient connectivity and traffic steering among different VNFs forming an NSI by providing dynamic service chaining [4]. SDN controllers maintain knowledge of the network topology by exchanging information with adjacent controllers and domain gateways.

Network slicing, through its enabling technologies, e.g., SDN/NFV, aims to satisfy new vertical use cases. It is envisioned that to provide new customized services on demand, the service providers need to automate the operations and deployment of the 5G mobile core. The disaggregation of the 5G mobile core, i.e., Control and User Plane Separation (CUPS), facilitates 5G use cases including Multi-Access Edge Computing (MEC) and network slicing. The 5G use cases require that the 5G mobile core Network Functions (NFs) be deployable in the core and at the Mobile Edge (ME) utilizing private and public clouds. Service providers will be able to realize new vertical solutions that achieve URLLC, Massive Machine Type Communication (mMTC), and Enhanced Mobile Broadband (eMBB). The cloud-native 5G core and its enabling technologies such as containers, microservices, container orchestration engine, and Continuous Integration and Deployment (CI/CD), will allow the 5G mobile core to be fully automated, flexible and scalable. The role of the cloud-native architecture in realizing MEC enabled network slicing remains an area of current research. This paper provides insights and use cases on how the cloud-native architecture and modern software development paradigms, such as a microservice architecture can facilitate the 5G use cases.

Unlike the traditional client-server application development model, the emergence of MEC, which introduces an intermediate entity at the network edge, results in a new three-layer application development model, e.g., client, near server and far server. This raises challenges for application developers, to identify the application features that require low-latency and real-time responses, so that those application features can be deployed in the near server, i.e., edge, whereas the application features that do not demand real-time response and require high compute power can be deployed at the far server, i.e., cloud. To deal with the challenges and adapt to this new application development paradigm, developers are adopting virtualization based application design, e.g., microservices and container based architectures. The recent white paper released by the ETSI MEC group [11] emphasizes the importance of a microservices based cloud-native architecture for MEC. In this paper, a cloud-native 5G microservices architecture for MEC enabled network slicing is envisioned. Its potential use cases in the context of MEC enabled network slicing are also introduced. The main contribution of this article is summarized as follows:

- An in-depth and comprehensive review of recent advances made in enabling E2E network slicing across multiple technologies and administrative domains.
- The role of MEC, as one of the key drivers of 5G and 6G, is explored. Potential use cases considering network slicing are discussed.
- The cloud-native 5G core and its design principles are investigated.
- We described the limitations of the traditional network virtualization techniques used to create network slices. A cloud-native 5G microservices architecture is envisioned along with its potential use cases in supporting MEC enabled network slicing.
- Open issues and research challenges are identified related to E2E network slicing and MEC integration.
- A preliminary evaluation was carried out of an envisioned cloud-native 5G architecture that supports network slice mobility and services migration across cloud-native edge clouds deployed at different zonal locations in southeast Australia.

### A. EXISTING WORKS

Network slicing has recently been the focus of different standardization bodies, including 3GPP, ETSI, and ITU-T [12]–[14]. This paper provides a detailed review of E2E network slicing considering both the single and multiple administrative and technology domains. This review covers the recent progress made and provides the emerging technology vision for E2E network slicing, including the recently proposed 5G transport slice connectivity interface [15], the role of MEC and the vision for the cloud-native 5G mobile core. There are a few related surveys available on the topic such as [5]–[7]. The authors in [7] present a detailed review and analysis on the topic of network slicing. However, they do not consider, in detail, enabling slice federation among multiple administrative domains, new transport-layer mechanisms, MEC integration, and the cloud-native solutions for the automation of the 5G mobile core. In [9], authors present a detailed survey on 5G network slicing, including the architectures, recent advancements, and future challenges. However, unlike [9], our work focuses on the new challenges introduced by the cloud-native transformation of 5G networks and mobility management of MEC-enabled 5G networks. A brief overview of the state of the art network architecture for 5G network slicing is provided in [6], [16]. The work in [5] deals with the resource allocation problem in network slices that only takes into account network slicing across a single technology domain, i.e., Radio Access Network (RAN) slicing. In [17], the authors proposed an on-demand RAN-slicing approach that jointly considers both the network slicing and spectrum sharing to realize the spectrum-aware slicing across all the RAN resources. A cloud-native approach for network slicing is introduced in [18], but the paper does not include a discussion of the latest key enabling technologies for the 5G cloud-native microservices architecture, e.g., dockers, containers, Kubernetes, and also the integration of edge computing, one of the fundamental motivations for a 5G cloud-native architecture. The authors in [19] consider the transport network architecture based on SDN/NFV. Other relevant articles related to network slicing and its enablers, i.e., SDN/NFV, [20]–[24] also, do not consider the latest trends and progress made in this field. Table 2 indicates a summary and comparison of the recent related survey papers on 5G network slicing.

### B. RELATED APPROACHES TO NETWORK SLICING

Techniques and solutions similar to the virtualized 5G core were proposed for 4G, including Dedicated Core (DECOR), Enhanced Dedicated Core (eDECOR), and RAN sharing. DECOR allows operators to deploy multiple Dedicated Core Networks (DCN), with each DCN dedicated to a specific

**TABLE 2.** Summary of recent surveys and tutorials with their primary focus.

| Reference | Cloud-Native 5G Core | E2E 5G Network Slicing | Slicing at the Edge | Network Slice Mobility | Remarks |
|---|---|---|---|---|---|
| Richart et al. [5] [2016] | ✗ | ✗ | ✗ | ✗ | The authors present a survey and detailed analysis of network slicing focusing on resource slicing across radio access networks. |
| Foukas et al. [6] [2017] | ✗ | ✓ | ✗ | ✗ | The authors provide a survey on the state-of-the-art in 5G network slicing, present a holistic network slicing framework, and identifies research gaps. |
| Afolabi et al. [7] [2018] | ✗ | ✓ | ✗ | ✗ | The authors present a detailed review and analysis on network slicing, mainly focusing on key enabling technologies, and 5G network requirements, and applications. |
| Kaloxylos et al. [8] [2018] | ✗ | ✓ | ✗ | ✗ | The authors provide a comprehensive overview of recent advances and solutions, focusing on 3GPP standardization activities of network slicing. |
| Barakabitze et al. [9] [2020] | ✗ | ✓ | ✗ | ✗ | A detailed survey on 5G network slicing, including the architectures, recent advancements, and future challenges. The authors mainly focus on the adaption of SDN and NFV in realizing the 5G network slicing. |
| Khan et al. [10] [2020] | ✗ | ✓ | ✗ | ✗ | The authors present recent advances, taxonomy, and network slicing requirements to focus on IoT applications. |
| Our Survey | ✓ | ✓ | ✓ | ✓ | Our survey presents recent advances on network slicing, focusing on the adaption of cloud-native 5G core and MEC to realize diverse 5G requirements and applications. |

customer or application. DECOR enables the customization of the control and user plane functions for the DCNs. The solution introduces a new subscription information parameter called a "UE usage type" that is stored in the Home Subscriber Server (HSS). This parameter is used by the operators to configure the required type of service; in other words, this value enables the selection of the DCN. The parameter values can reflect different types of services such as general Machine Type Communications (MTC) or low latency services for autonomous cars. In comparison to E2E network slicing that spans across multiple technological domains such as radio, edge, transport, and core, this scheme only deals with core network slicing.

eDECOR was also designed to achieve the same functionalities as DECOR, but with slight enhancements specified by the 3GPP [25]. In contrast to DECOR, where the DCN selection is made by enodeB (enB), in Edecor the UE assists in the selection of DCN by providing two parameters, i.e., DCN selection assistance parameter and Network-Attached Storage (NAS) type. This approach also only takes into account core network slicing.

Another similar slicing approach, RAN sharing, considers network slicing in the RAN. RAN sharing involves sharing network infrastructure, e.g., antenna and backhaul equipment [26]. However, this approach doesn't include the softwarization and virtualization needed to provide flexibility and scalability. Also, it doesn't consider E2E network slicing.

### C. PAPER ORGANIZATION

The rest of the paper is organized as follows. We briefly introduce the MEC integration into 5G Service Based Architectures (SBAs) in Section II. In Section III, a detailed review of recent progress related to E2E network slicing across multiple technologies and administrative domains is provided. In Section IV, we explore mobility management solutions for network slicing. In Section V, the key enabling technologies for network slicing are provided, and new features for these technologies, such as the SDN meter table, are discussed. In Section VI and VII, we discuss cloud-native 5G core for network slicing, along with its enabling technologies and potential use cases. In Section VIII, research issues and challenges are identified, and future research directions are provided. The conclusion is provided in Section IX. To better understand the structure and organization of this survey, we refer the reader to Fig. 1. Table 1 provides a list of commonly used acronyms in the survey.

## II. 5G SBA AND MEC DEPLOYMENT

5G SBAs include the separation of the control and user planes to provide scalability and flexibility [27]. The control plane functions are connected to each other via service-based interfaces. The Access Management Function (AMF) and Session Management Function (SMF) are connected to the user plane nodes via N1, N2, and N4 interfaces as shown in Fig. 2. AMF and SMF are used to manage subscriber attachment, mobility, and sessions. A brief summary of the 5G SBA NFs includes:

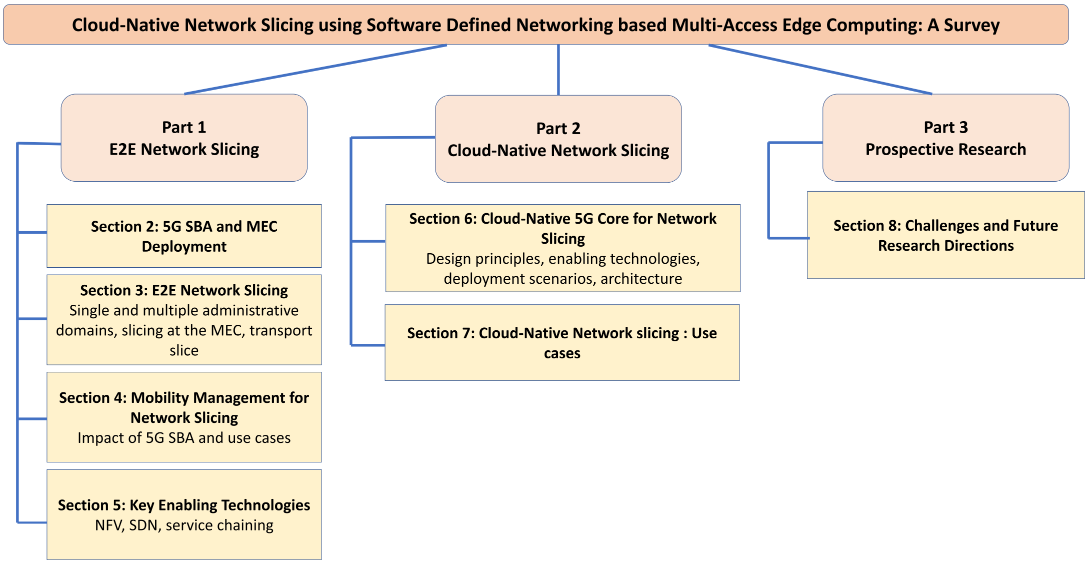

Cloud-Native Network Slicing using Software Defined Networking based Multi-Access Edge Computing: A Survey
Part 1
E2E Network Slicing
Part 2
Cloud-Native Network Slicing
Part 3
Prospective Research
Section 2: 5G SBA and MEC Deployment
Section 3: E2E Network Slicing
Single and multiple administrative domains, slicing at the MEC, transport slice
Section 4: Mobility Management for Network Slicing
Impact of 5G SBA and use cases
Section 5: Key Enabling Technologies
NFV, SDN, service chaining
Section 6: Cloud-Native 5G Core for Network Slicing
Design principles, enabling technologies, deployment scenarios, architecture
Section 7: Cloud-Native Network slicing : Use cases
Section 8: Challenges and Future Research Directions

**FIGURE 1.** Structure and organization of the paper.

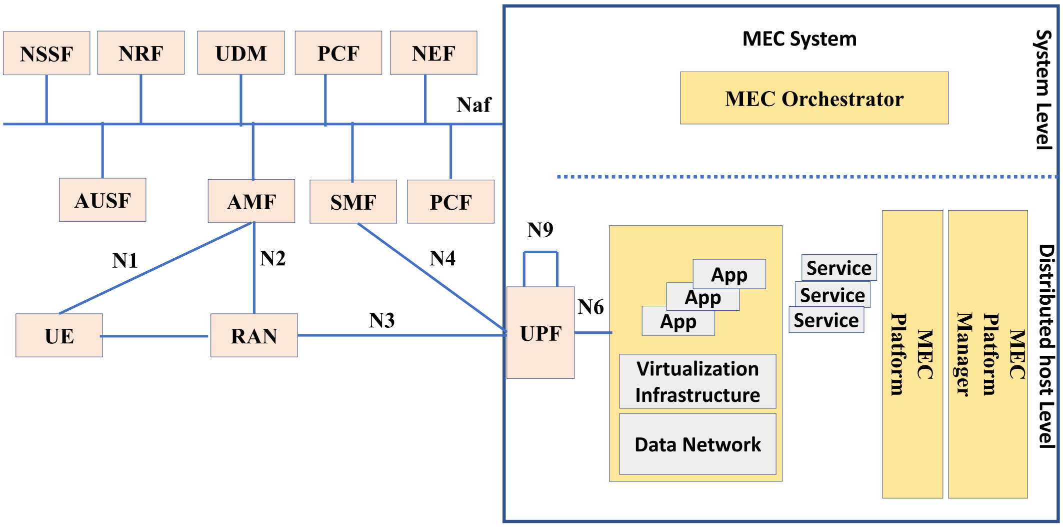


**FIGURE 2.** MEC integration into 5G [28].

- **Access and Mobility Management Function (AMF)** is used to manage access control and mobility. It also provides reachability and communication services for other NFs.
- **Session Management Function (SMF)** is required to manage and create sessions according to the defined network policy. Some of its functionalities include IP address allocation and selection, traffic rules configuration of user plane function, and roaming support.
- **User Plane Function (UPF)** is a centralized entity that plays a key role in traffic routing towards required network functions and applications. This function can be deployed in various locations or configurations, depending upon the type of service required.
- **Policy Control Function (PCF)** is used to provide the policy framework incorporating mobility management, network slicing, and roaming.
- **Unified Data Management (UDM)** is used to store the subscribers' data and profiles. It is similar to HSS in 4G.
- **Authentication Server Function (AUSF)** is used to perform the authentication function of 4G HSS, e.g., it implements Extensible Authentication Protocol (EAP) and stores keys for UE authentication.
- **Network Resource Function (NRF)** is a new and most important function that is incorporated in the 5G SBA. It allows network functions discovery functionality so that the network functions can discover and communicate with each other via APIs. NRF maintains the profile

of all the NF instances and their supported services such as capacity information, IP addresses. In other words, NRF is responsible for discovering all the available services.

- **Network Exposure Function (NEF)** is a centralized point that is responsible for exposing the capability information and services offered by the 5G core network functions to external entities. This function authorizes external users such as partner operators, to monitor and enforce application policy.
- **Network Slice Selection Function (NSSF)** assists the selection of suitable NSIs, and allocation of required AMFs to the users depending upon the service requirements.

NFs may incorporate other NFs that are reusable, independent of each other, and often referred to as microservices. An NF can act as either producer or consumer of these services; for example, a consumer NF can request subscriber policy information from a producer NF [29].

CUPS is one of the most important 5G core design principles. It facilitates flexible service deployment at centralized or distributed locations, i.e., edge [27]. The modular function design of the 5G core enables E2E network slices for different service requirements and concurrent access to both local and centralized services, e.g., to support low-latency mission-critical communication. In this case, low-latency applications or services can be deployed in the local data center or the ME by using the MEC Platform (MEP). In most cases, the user plane functions such as UPF are deployed in the edge or local data center, whereas, the control plane is centralized. In some cases, control plane network functions e.g. NEF, can also be hosted in a distributed manner such as in the edge to support mission-critical communication services [27].

### A. MEC INTEGRATION INTO 5G SBA

The deployment of an MEC system in 5G SBA is shown in Fig. 2, as defined by the European Telecommunications Standards Institute (ETSI) [28]. The MEC system consists of an MEC Orchestrator (MEO) at the system level that acts as an Application Function (AF) and interacts with the 5G core NEF. The MEO maintains an overall view of the MEC system, i.e., available resources, offered services, deployed MEC hosts (MEH). It is also responsible for the selection of appropriate MEH for the application instantiation and application relocation if needed [30]. The key components that facilitate the integration of MEC with the 5G core are the ability of MEC to act as an AF and influence the routing of edge application traffic by interacting with the 5G core NEF; and its ability to receive event notifications such as a mobility event that initiates application relocation.

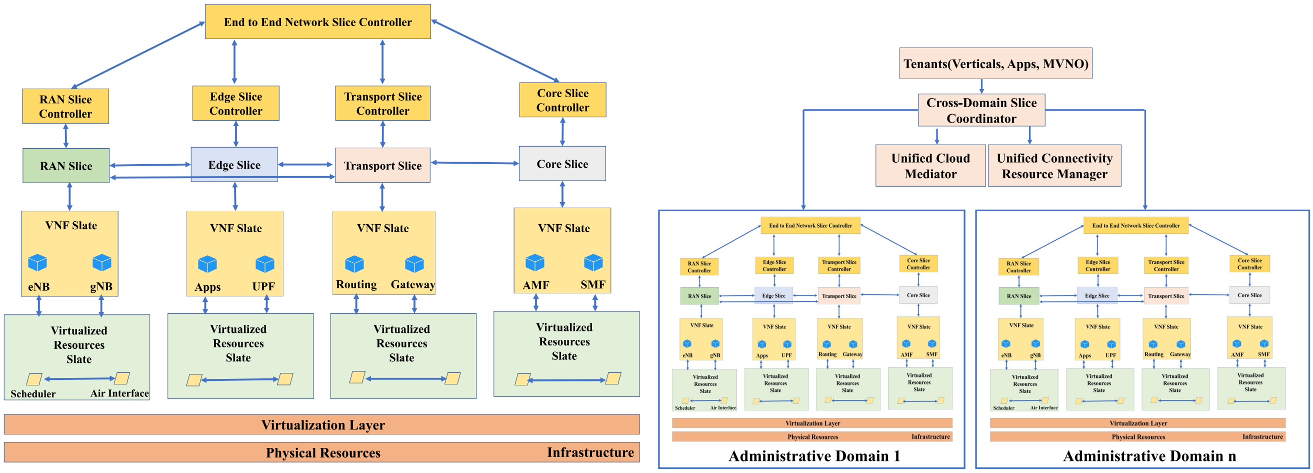
The MEC host-level consists of the MEP and the virtualization infrastructure that provides resources, i.e., compute, storage, and networking, to the MEC applications. The MEP offers an environment where MEC applications can discover, advertise and offer MEC services via platforms in the same or different MEC system. The MEP receives traffic rules from the MEC platform manager, applications, or services and configures the data plane accordingly.

By flexibly locating the UPF, the MEC can be deployed in a data network that is external to the 5G system. The distributed MEH deployed in a data network can accommodate MEC applications, e.g., computational offloading applications or MEC services, e.g., message broker. The applications running in the MEC system, can produce a wide range of services such as Vehicle to Everything (V2X), and mobile virtual reality. More details on the MEC integration with the 5G network can be found in [28].

## III. E2E NETWORK SLICING

Network slicing is the integration of a set of technologies to create customizable and specialized dedicated logical Networks as a Service (NaaS) in order to meet diverse and heterogeneous requirements from vertical industries. It involves efficient virtualization and isolation mechanisms, customized and flexible functions design, and Operation and Maintenance (O&M) tools to provide dedicated logical networks upon a shared infrastructure [2]. The components of E2E network slicing are briefly described below, followed by a detailed description in later sections:

- **Network Slice Instance (NSI)** is the most important concept in E2E network slicing. It is described as an E2E logical network that consists of various virtual NFs, resources, and connectivity relationships. NSI differentiates the E2E 5G network slicing concept from the existing approaches as it covers multiple technical domains, such as terminal, Radio Access Network (RAN), Edge Network (EN), Transport Network (TN), and Core Network (CN). Additionally, it also involves the Data Center (DC) domains to host third party applications from different vertical industries. Different NSIs may consist of different VNFs and allocated resources. However, NSI can also share VNFs and resources to reduce CAPEX.
- **Network Slice Type:** Three broad usage scenarios and service categories of 5G, as defined by ITU-R, are eMBB, mMTC, and URLLC. Each of these categories has its own demands and requirements that are highly distinct to each other. Network slice types are used to represent high-level categories in order to define the NSIs.
- **Network Slice Template (NST):** NST design is different from the operation of NSI and is used in the slice designing phase. NST is generated based on the network capabilities of each technical domain and specific requirements of tenants. NSI instantiation depends upon the NST output, which also includes VNF configuration and deployment and resources in multiple technical domains.
- **Network Slice Subnet Instances (NSSIs):** An NSI typically consists of multiple NSSI that integrate to form



**(a)** NSI in a single administrative domain domain

**(b)** Federated NSIs among multiple administrative domains

**FIGURE 3.** E2E network slicing architecture.

a fully-fledged NSI. NSSIs represent a group of VNF instances.

- **RAN Slice**, also known as RAN Sub-Slice, consists of customized and independent RAN network functions such as eNB and Next-Generation node B (gNB) for each E2E network slice.
- **Edge Slice**, also known as Edge Sub-Slice is used to host various RAN and mobile core components, user service-based functions, and applications to provide URLLC services.
- **Transport Slice**, also known as transport sub-slice, is a set of connections between various VNFs or PNFs with deterministic Service Level Agreements (SLAs). This type of slice can be realized by various technologies and transport such as IP, optics, microwave, and Resource Reservation Protocol (RSVP), segment routing, SDN meter tables, respectively.
- **Core Slice**, also known as Core Sub-Slice, consists of customized and independent core, virtual NFs, such as UPF, AMF, and SMF.
- **E2E Network Slice** is defined as a virtual network capable of supporting a specific vertical or service, functional and performance requirements. It is provided by the slice provided after certain agreements with the slice buyer, such as slice lasting time.

### A. E2E NETWORK SLICING IN A SINGLE ADMINISTRATIVE DOMAIN

As defined by 3GPP [15], each E2E network slice comprises a multitude of RAN, core, and transport slices, each having its own controller. Considering the dynamic nature of the E2E network slice, the life cycle of each network slice might be a few hours, days or months. Therefore, various controllers, i.e., a controller in each respective domain, are needed to perform the life cycle management of network slices in their domain. Additionally, to achieve automation and optimization of network slices, an E2E network also consists of an E2E network slice controller that interacts with and manages the individual controller of each technical domain, e.g., RAN, Transport, and core Slices controller [15]. An E2E network slice controller is also required to enable the control and coordination of network slices.

In accordance with the 3GPP definition of E2E network slice, Fig. 3a represents the E2E network slice architecture in a single administrative domain. Considering the use case study of MEC for the provision of low-latency 5G services in [31], a MEC layer is proposed for the E2E network slice architecture. The MEC approach can be utilized to place the VNFs, and user functions closer to the end-users by distributing network data centers closer to the network edge. This approach will help to improve the overall service experience of end-users, such as the provision of URLLC services. The addition of an MEC layer would also result in a reduced load on the transport infrastructure by providing the cloud computing facilities within or close to the transport network as shown in Fig. 3a.

The MEC data centers host both the virtualized RAN components such as the Centralized Unit (CU) and also the mobile core components such as the UPF, depending upon the network design requirements, as shown in Fig. 3a. Additionally, user service-based functions and applications related to the provision of URLLC services can also be placed in the MEC.

Similar architectures are proposed for the provision of a V2X URLLC slice in our previous work [32] and in another work [33]. The authors proposed to host the SDN mobility management application in the MEH, to provide up-to-date topology information, the position and trajectory of each vehicle, thus ensuring low-latency operation and triggering seamless UP/CP functions migration, handovers, and reconfiguration of network resources in active slices. The recent state-of-the-art approaches for network slicing across single administrative and technical domains are summarized in Table 3.

**TABLE 3.** Summary of state-of-the-art network slicing approaches.

| References | Multiple Admin Domain | Technical Domains | | | | Cloud-Native Microservices Architecture | Contribution |
|---|---|---|---|---|---|---|---|
| | | RAN | Edge | Transport | Core | | |
| Zhang et al. [16] [2019] | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | This paper presents an overview of the working process of network slicing in context of 5G SBA |
| Taleb et al. [3] [2019] | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | This paper proposes a four-layered network slicing architecture to enable network slices federation across multiple administrative and technology domains |
| Chartsias et al. [63] [2017] | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | This paper proposes a light-weight network slice management framework that allows slicing a physical network into multiple virtual networks |
| Wang et al. [64] [2018] | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | This paper presents the ongoing European project called SliceNet that advances the state of the art architecture (SOTA) in enabling an E2E multi-domain network management paradigm |
| Chien et al. [65] [2019] | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | This paper proposes a joint edge and core (central) slicing framework to realize E2E slicing |
| Mayoral et al. [66] [2016] | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | This work presents a network and experimental design for network slicing architecture with distributed clouds |
| Afolabi et al. [67] [2019] | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | The authors propose an E2E multi-domain network slicing orchestrator system and dynamic auto-scaling algorithm to enable scalable orchestration of federated network slices |
| Afolabi et al. [68] [2017] | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | This work envisions a 5G architecture and workflow for E2E network slicing over multiple domains |
| Li et al. [69] [2016] | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | The concept of horizontal and vertical network slicing is introduced |
| Zhang et al. [70] [2017] | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | A logical architecture of 5G network slicing is proposed, that focuses on managing mobility between access networks (AN) |
| Li et al. [71] [2017] | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | The technology background is introduced and a three-layer framework for network slicing is proposed |
| Katsalis et al. [72] [2017] | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | A network slicing architecture is proposed for integrated 5G communications |
| Rost at al. [73] [2017] | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | It provides an overview of network slicing for 5G and its potentials to network operators |
| Vincenzi et al. [74] [2017] | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | An E2E architecture for dynamic spectrum management through multi-tenant slicing is proposed |
| Rostami et al. [75] [2017] | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | A design and implementation concept of modular and programmable architecture across RAN and Transport network is presented |
| Tseliou et al. [76] [2019] | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | The base station agnostic framework for network slicing (NetSliC) is proposed that targets the creation of wireless slices across heterogeneous RAN deployments |
| Boubendir et al. [77] [2018] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | A layered architecture is proposed that compromises of brokering layer and multiple functions allowing third-party operators to federate cross-domain IT and networking resources |
| Addad et al. [78] [2020] | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | A cost-optimal cross-domain network slice deployment model is proposed that guarantees each network slice the required bandwidth and delay, and reduces the OPEX |
| Ordonez-Lucena et al. [79] [2020] | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | Common standard interfaces are proposed required to facilitate the cross-domain orchestration procedure and interoperability among multiple vendors |

### B. LOGICAL FLOW OF CREATION AND AUTOMATION OF E2E NETWORK SLICE

The logical flow process to achieve E2E network slice automation for different smart city services such as URLLC, eMBB, mMTC is discussed in this section.

The customer requests the slice provider or operator to create an E2E network slice for a service. Based on an SLA agreement, this request is forwarded to an E2E network slice controller, which uses its predefined NST and creates an NSI. The NSI contains information about the NFs in RAN, core, and edge that will be part of this E2E network slice. It then requests the RAN and edge slice controller to create a RAN and edge slice, respectively. Both of the domain controllers trigger the creation of virtual NFs in their respective domains by using the NFV interface, known as the ETSI interface os-Ma-nfvo. The details of NFV and this interface are provided later. The Network Function Virtualization Orchestrator (NFVO) performs the life cycle management of the virtual NFs. The NFs are then programmed by their respective domain controllers.

The same process as mentioned above applies to the creation of a core slice. To provide connectivity between various NFs, multiple transport slices, i.e. various connections, will be needed, e.g., transport slices between RAN, edge and core slices. A transport slice also triggers the creation of VNFs in its domain, such as a firewall and security gateway, if required. After all the respective domain slices are created, the E2E network slice controller will associate all of them together to form a single E2E network slice instance for the specific service type. A unique network slice id, i.e., Network Slice Selection Assistance Information (NSSAI) is also allocated to the new network slice. The UEs will then be able to request access to this network slice by using signalling procedures [15]. The logical flow for the provisioning of an E2E network slice is shown in Fig. 4. The number mark represents the order of actions in which the network slice is provisioned.

The interface connecting the E2E network slice controller with the RAN and core slice controller has been defined in technical specifications released by the 3GPP [15]. However, the literature available on the transport slice interface is limited [15]. The transport slice and its connectivity interface are summarized later.

### C. E2E NETWORK SLICING IN MULTIPLE ADMINISTRATIVE DOMAINS

An E2E network slice can belong to one or more administrative domains that may be distributed between DCs. Thus, to deal with this deployment challenge, the E2E network slicing architecture needs to be overhauled for the multiple administrative domain scenarios.

A multi administrative domain federated NSI combines two or more NSIs that belong to different administrative domains, to form a slicing federation. In order to facilitate E2E network slicing across multiple administrative domains, there is a need for an additional control layer to be added to the single administrative domain architecture. This layer should be able to map the service requirements to the capability of the infrastructure domain by identifying the domains with the required resources, i.e., computing, storage, and networking resources, thus ensuring efficient E2E network slicing federation.

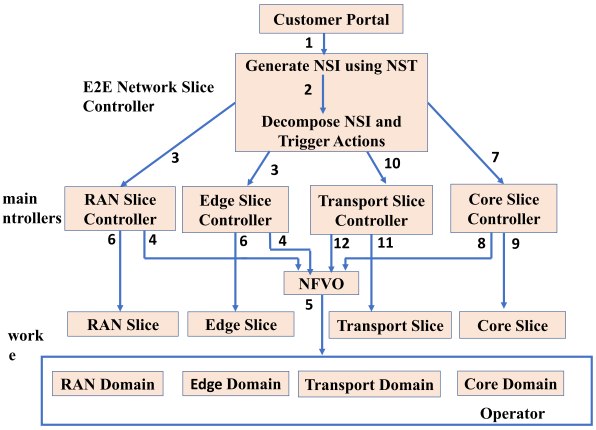


**FIGURE 4.** Logical flow of automation and creation of E2E network slice.

After identifying the infrastructure domains, the layer, i.e., cross-domain slice coordinator, should request the E2E network slice controller of the administrative domains to instantiate an NSI instance within their respective domains. This NSI instantiation within a single administrative domain follows the same steps as mentioned in the logical flow process above. This cross-domain slice coordinator should then be able to join the NSIs within the administrative domains to form a federated NSI.

The cross-domain slice coordinator is responsible for the management, control, and monitoring of the resources related to a federated NSI. It should also ensure secure and reliable connectivity between administrative domains.

The authors in [3] recommended using a cross-domain slice coordinator to perform federated resource allocation, i.e., compute storage and network resources. To perform this federated resource allocation, two architectural entities are required to assist the cross-domain slice coordinator with federated resource allocation: unified cloud mediator and unified connectivity resource manager. The unified cloud mediator contains the performance capability description of the infrastructure resources, and the connectivity resource manager negotiates cross-domain connectivity. Fig. 3b depicts the E2E network slice federation among administrative domains integrating the two additional entities as proposed by the authors in [3]. The recent state-of-the-art approaches for network slicing across multiple administrative and technical domains are summarized in Table 3.

### D. SLICING AT THE MEC

Several 5G use cases are expected to rely on the MEC paradigm to support a new generation of services, e.g., mMTC, URLLC, and eMBB [34]–[36]. Though network slicing and MEC are two key 5G enablers, they are evolving in parallel as defined by two different standardization bodies, i.e., 3GPP and ETSI. Therefore integration of network slicing and MEC is a research challenge that should be addressed to provide enhanced slicing capabilities at the network edge. This section reviews selected recent proposals for orchestration and management platforms that integrate MEC and network slicing.

### 1) ARCHITECTURE

The authors in [34] proposed a novel architecture compliant with ETSI and 3GPP that integrates MEC as a sub-slice. The work proposed a multi-tenancy and in-slice deployment model to support MEC network slicing. In this proposal MEP is deployed as a VNF at the edge NFV Infrastructure (NFVI) and is shared among the slices or deployed inside the slice, respectively. The MEO is responsible for instantiating the applications at the edge NFVI and communicating the MEP application IP address to enforce traffic steering. Cominardi *et al.* in [37] proposed solutions to evolve the MEC framework towards integration with 5G network slicing, enabling multi-tenancy support. The authors emphasize the need for interaction between MEC, NFV, and 3GPP systems to facilitate the slice-aware MEC app allocation on MEC facilities. The authors also propose a communication channel to support MEC inter-slice communication.

### 2) RESOURCE ALLOCATION

D'Oro *et al.* in [38] proposed a unified MEC slicing framework that optimizes the resource allocation in the strictly-constrained MEC computing and storage resources and supports instantiating MEC slices without suffering resource over-provisioning. Xiang *et al.* in [39] proposed a mathematical model that integrates MEC and network slicing focusing on addressing the stringent latency requirements of critical services. The optimization problem deals with joint allocation of RAN and edge computing resources to the MEC sub-slice. Liu *et al.* in [40] proposed a decentralized resource orchestration system that supports dynamic network slicing in edge computing networks. The proposed work leverages deep reinforcement learning techniques to learn optimal policies, e.g., the resource demands of E2E slices, and dynamically allocates the resources accordingly. Jošilo *et al.* in [41] proposed an optimization algorithm that aims to meet the latency-sensitive computational task requests by jointly assigning the tasks to the most suitable MEC sub-slice, and dynamically managing the radio resources within the slices.

### 3) OPERATOR COSTS

Feng *et al.* in [42] proposed a novel framework that jointly optimizes the slice-admission request and resource-allocation in MEC to maximize the operators average revenue. The proposed optimization algorithm achieves a balance between the average delay and the average operator revenue by making dynamic and effective slice request admission decisions. Sanguanpuak *et al.* in [43] proposed an infrastructure cost minimization algorithm where a network operator could deploy MEC slices by efficiently using the edge infrastructure resources to meet latency requirements. Xiang *et al.* in [44] proposed a model that jointly considers the computational resources available at the MEC servers, slicing of the RAN and edge computing resources, and routing diverse traffic requests towards the optimal MEC slices. The authors aim to achieve a flexible balance between the network operator costs and the user-perceived latency by making effective use of the available computing resources.

## E. 5G TRANSPORT SLICE

Transport slice is a distinct set of connections between multiple virtual or physical NFs, each with its own specific SLA. It is implemented in the network by using IP and tunnels, e.g., IP, and Segment Routing (SR).

### 1) TRANSPORT SLICES IN CLOUD RADIO ACCESS NETWORKS (C-RAN)

The RAN consists of two functional units known as the Baseband Unit (BBU) and the Radio Unit (RU), which is also known as the Remote Radio Head (RRH) [45]. The RU is responsible for the transmission and reception of radio waves over the air interface to the User Equipment (UE) and is connected to the BBU through the Fronthaul Network (FN). The BBU has signal processing capabilities and is connected to the core network through the transport network. The MEC system, equipped with MEO, acts as an AF, that interacts with the NEF of 5G SBA. The MEC system also consists of a distributed cloud, that can be used to host different applications belonging to one or more network slice instances.

In this architecture, a single E2E network slice involves four Transport Slices (TS): TS1, TS2, TS3 and TS4. TS1 connects the RAN to the core, TS2 connects the RRHs to the centralized BBU, TS3 that connects the MEC to Core and TS4 that connects the RAN to the MEC as shown in Fig. 5.

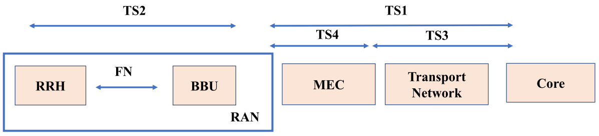


FIGURE 5. Transport slices connecting multiple domains.

### 2) TRANSPORT SLICE CONNECTIVITY INTERFACE

The Transport Slice Connectivity Interface (TSCI) is an interface between the E2E network slice controller and the transport slice controller. It provides association and binding between RAN to transport and core to transport slices. The transport slice controller receives the request for the required connections between various NFs in the RAN and core by using a TSCI. The connections are then implemented by the transport slice controller by using various IETF models. It is important to note that this new TSCI provides information about the required connections and not the services. For example, an E2E network slice controller requests a transport slice controller to create a transport slice with multiple connections between the RAN and core NFs. The E2E network slice controller can make use of the TSCI interface to inform the transport slice controller of the required connections between the RAN and core NFs, e.g., between RAN1 and UPF1, to serve a particular customer, tenant, service type or SLA.

To implement the required connections, the transport slice controller finds the endpoints and best path to create a service between these endpoints, e.g., to provide the connection between the RAN1 and UPF1, the controller first finds the best available Border Routers (BR) for each NF, then finds the best path available between them. Finally, it creates a service between these endpoints.

## IV. MOBILITY MANAGEMENT FOR NETWORK SLICING

User mobility from one domain to another may cause a degradation in QoS or disconnection of the ongoing mobile communication and service session. This may happen when a new network slice with the same characteristics, e.g., computing, storage, and networking resources, is to be instantiated at the destination network. Therefore, network slicing should support mobility of the slice computing, storage, and network resources [46]. In addition to mobility support, the network slice should also be capable of dynamically adjusting and adapting the resource allocation, e.g., freeing the unused resources or adding more resources depending upon the service requirements and resource availability. Recent research proposals deal with the various aspects of mobility management in network slicing.

Addad *et al.* in [46] proposed slice mobility patterns, e.g., full slice and partial slice mobility patterns, to efficiently manage and migrate resources. The authors also introduced the concept of slice breathing and scaling to dynamically adapt to the varying slice resources demand, i.e., sudden increase or decrease in the service demands causing over-consumed or under-consumed slice resources, respectively. Shah *et al.* in [32] leveraged SDN to track the user mobility patterns across different mobile networks and trigger network slice mobility action towards the most optimal destination network. In addition the SDN contoller, dynamically allocates the required resources to the relocated network slice. De Vita *et al.* in [47] proposed a deep reinforcement learning algorithm that learns optimal policies to relocate the network slice across different MEC servers without any explicit knowledge of the underlying processes. Yousaf *et al.* in [48] proposed the inclusion of specialized mobility management NFs within a network slice capable of selecting the mobility management scheme depending upon the service mobility requirements. Meneses *et al.* in [49] proposed SDN based mechanisms to increase the acceptance rate of incoming slice handover requests by conserving the slice resources after the user handover. Mouawad *et al.* in [50] proposed an SDN based network slice management solution where an SDN overlay application performs network slice selection functionalities in the case of inter-slice V2X handover. The authors proposed a utility function that considers the slice resource availability to determine the optimal target slice.

### A. IMPACT OF 5G SBA ON NETWORK SLICING MOBILITY MANAGEMENT

The 5G SBA plays an important role in realizing the network slicing concept by providing NF or service reuse and enables customization across slices. Various NFs combine to form an E2E slice to support different service requirements. For example, consider the deployment of an AMF NF in two different types of slice with different service requirements, i.e. URLLC and IoT. The URLLC slice is often required to support mission-critical communications such as autonomous driving and demands high mobility. The IoT slice, on the other hand, enables massive device connectivity with relatively low mobility requirements. The AMF in the URLLC slice will require more instances of AMF services because of its high mobility requirements as compared to the IoT slice. Additionally, the UPF can also be deployed in the edge using the MEC paradigm to support low-latency communications in a URLLC slice as shown in Fig. 6.

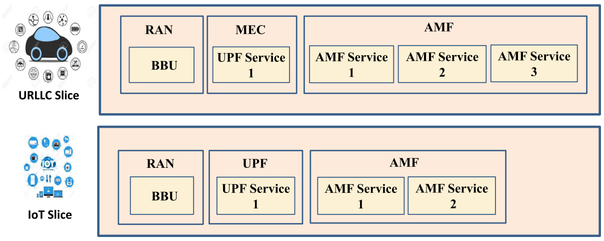


**FIGURE 6. 5G SBA scalability and mobility support.**

#### 1) V2X SlICE USE CASE SUPPORTED BY 5G-MEC INTEGRATION

Mobility support is an essential feature in V2X communications [51]. V2X applications have diverse Quality of Service (QoS) requirements; for example, the autonomous driving application requires ultra-low-latency and reliability and very high availability. Infotainment applications have very high throughput requirements. As the vehicle exhibits very high mobility characteristics and travels across multiple cells, it is essential for V2X applications to guarantee the QoS requirements and maintain service continuity.

As V2X applications have very demanding QoS requirements, the applications are often offloaded to the MEH closer to the network edge. As the vehicle moves from the service area of one cell to another, the source MEH may no longer be appropriate to provide V2X services to the vehicle. The MEO should identify a target MEH by acting as an AF and subscribing to the 5G AMF and SMF for vehicle mobility-related events and user plane management events, respectively. Based on location updates, the MEO can identify a new target MEH, and the application instance

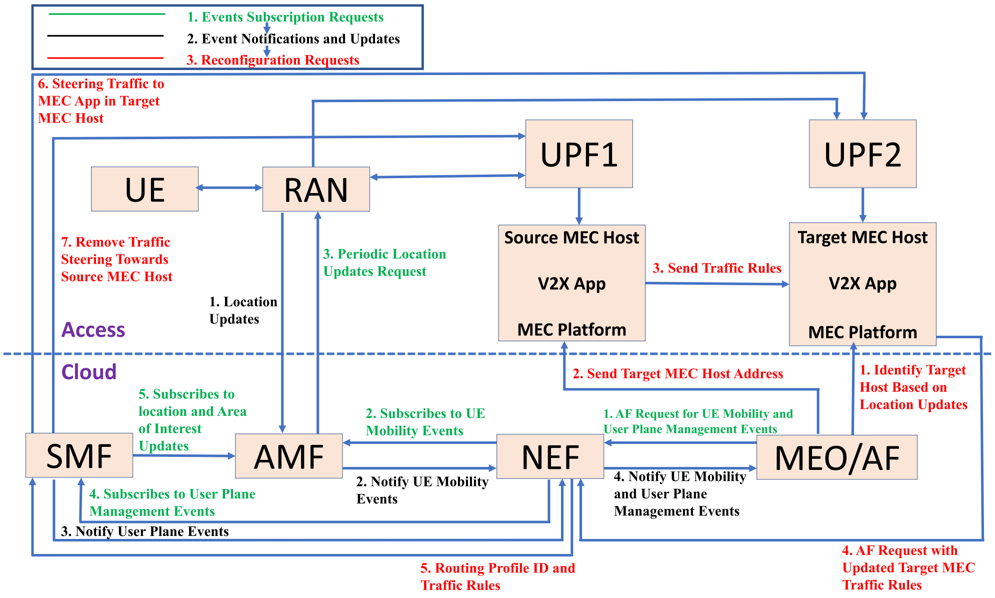

1. Events Subscription Requests
2. Event Notifications and Updates
3. Reconfiguration Requests
6. Steering Traffic to MEC App in Target MEC Host
UE
RAN
UPF1
UPF2
Source MEC Host
V2X App
MEC Platform
Target MEC Host
V2X App
MEC Platform
7. Remove Traffic Steering Towards Source MEC Host
1. Location Updates
3. Periodic Location Updates Request
3. Send Traffic Rules
Access
Cloud
2. Send Target MEC Host Address
1. Identify Target Host Based on Location Updates
5. Subscribes to location and Area of Interest Updates
2. Subscribes to UE Mobility Events
1. AF Request for UE Mobility and User Plane Management Events
SMF
AMF
NEF
MEO/AF
2. Notify UE Mobility Events
4. Notify UE Mobility and User Plane Management Events
4. Subscribes to User Plane Management Events
3. Notify User Plane Events
5. Routing Profile ID and Traffic Rules
4. AF Request with Updated Target MEC Traffic Rules

**FIGURE 7.** V2X slice mobility use case.

can be relocated to the target MEH. The MEP of the target MEH then acts as an AF and interacts with the NEF of 5G SBA, specifying a new target MEH and traffic routing profile. This new request is passed to the PCF, which triggers the updated rules and passes this to the SMF. The SMF will reconfigure the traffic rules and insert a new Uplink (UL) classifier in the UPF to steer the traffic/packets coming from the vehicle towards the target MEH. Various transport slices facilitate the interaction between these NFs instantiated in the V2X slice by providing connectivity from the edge enabled access layer to the cloud. This process can be seen in Fig. 7.

## V. KEY ENABLING TECHNOLOGIES

### *A. NETWORK FUNCTION VIRTUALIZATION*

Network Function Virtualization (NFV) is a key enabling technology for 5G network slicing. It allows flexible creation of network slices on shared physical resources and removes the dependencies on dedicated hardware by providing an efficient resource abstraction layer. NFV allows the network services to run in virtual machines (VMs) or containers on the edge on cloud infrastructure. This allows each VM to perform independent network operations, e.g., load balancing or firewall. The main NFV components are summarized below:

#### 1) VIRTUAL NETWORK FUNCTION

VNF is the virtualization of NFs to enable their independent operation. Each VNF can be further divided into sub-functions called VNF components.

#### 2) NFV INFRASTRUCTURE

NFVI defines the software and hardware required to deploy, monitor, and operate VNFs. It provides abstraction of hardware resources such as computing, storage, and networking through a virtualization layer enabling each VNF to work independently from hardware resources. This virtualization layer can be a server such as Xen and VMware and network such as OpenFlow or Virtual Extensible LAN (VXLAN).

#### 3) NFV MANAGEMENT AND ORCHESTRATION

NFV Management and Orchestration (NFV MANO) consists of three main components including the Virtualized Infrastructure Manager (VIM), VNF Manager (VNFM), and NFV Orchestrator (NFVO). VIM is responsible for managing and controlling VNF interactions with physical resources such as resource allocation and deallocation. VNFM performs VNF life-cycle management, i.e., initialization and termination of VNFs and NFVO are responsible for implementation of different network services on the NFVI.

Another component of the NFV framework is Operation Systems and Business Support (OSS/BSS) that assists NFV MANO in executing networking policies.

### *B. SDN AND SERVICE CHAINING*

SDN is another key enabler of network slicing. The SDN controller can be used to provide effective network slice management by applying independent rules for each network slice as defined by the corresponding network policy or slice provider. It introduces programmability in the network by decoupling the control plane from the data plane. SDN provides centralized network intelligence that can be leveraged to instantiate new services by dynamically chaining NFs, i.e., PNFs or VNFs, depending upon the network conditions and user requirements.

Service chaining is a network capability that allows application-driven networking through the ordered connection of NFs [16], [52]. It allows flexible chaining of both control and data plane functions. It also enables application-driven traffic steering, i.e., traffic of certain application, service, or users traverses a particular set of NFs as defined by the service chaining policies [53]. This dynamic service chaining process enables the network operators to flexibly create, scale and remove NFs of a specific network slice depending upon the varying demands or service requirements.

Several research papers have focused on providing network slicing through NFV and SDN integration [54]–[56]. In summary, network slicing allows the mobile operators to add virtual networks and services to create mobile virtual networks on the same physical network. In this scenario, NFV provides the ability to create mobile network services through VNFs, and, in turn, the SDN framework is used for NF connectivity.

#### 1) SDN ENABLED TRAFFIC STEERING

VNFs or virtual function instances may reside at different locations, e.g., different technical domains or administrative domains, and the VNFs are chained to form a service. Traffic steering through the VNFs that span multiple technical and administrative domains is a very challenging task. In traditional networks, traffic is directed to the desired NFs using manual device configuration. However, in the case of network slicing that requires the dynamic and real-time deployment of VNFs to create services on demand, this traditional approach can not be imported. Because network slicing consists of dynamic allocation of resources to VNFs, there is a need for autonomous traffic steering capabilities. SDN, because of its centralized architecture, offers intelligence and flexible control and enables efficient traffic steering towards VNFs. Several studies show that by extending SDN capabilities, i.e., Layer 2 (L2) and Layer 3 (L3) forwarding functions, it can allow efficient and dynamic traffic steering through VNFs [57]–[59]. The authors in [59] propose an algorithm that finds the best path through a set of VNFs.

#### 2) SDN METER TABLES

SDN, through its central management platform and intelligent control, improves network programmability and allows dynamic control of the routing elements. OpenFlow is the protocol used to pass flow messages between controllers and the data forwarding devices, to make network flow decisions and carry out other network control related and monitoring functions. The control plane interacts with the data plane to specify forwarding instructions based on flow entries.

SDN network can also be used for efficient traffic splitting for Differentiated Services (DiffServ) by using the recently released OpenFlow 1.3 [60], [61], i.e., OpenFlow 1.3 enables the use of a new feature called a meter table. It consists of meter entries that are used to define per-flow meters, where a meter performs QoS operations such as rate-limiting and DiffServ. The most important element of meter entry is the meter band that specifies band type, rate, and type-specific arguments to define the way the packet should be processed, rate limitations, and Differentiated Services Code Point (DSCP) numbers, respectively. DSCP is a 6-bit field included in the IP header that is used to identify the service level of the packet, e.g., DSCP value 0 and 46 are used to define the best effort and high priority class, respectively.

This feature provided by SDN OpenFlow 1.3 [62] can be used for efficient traffic splitting to provide DiffServ to the customers. SDN and OpenFlow meter table features as described can be adapted to realize the vision of network slicing, i.e., to provide diverse service requirements.

## VI. CLOUD-NATIVE 5G CORE FOR NETWORK SLICING

5G network slicing presents network operators with opportunities to achieve a significant revenue boost by providing new enterprise use cases beyond the enhanced mobile broadband services. To provide the new use cases, evolved 5G core technologies are required. Cloud-native 5G core software design can facilitate network slicing by providing services on demand.

A key motivation for 5G, compared to the previous mobile generations, is that it is service-focused, e.g., URLLC, mMTC, and eMBB, to support a wide range of vertical sectors. To provide the diverse services with a different set of requirements, operators should be able to deploy 5G core components across public, local, or private DCs and in any geographical location dynamically, depending upon the application demands. Thus, the 5G core design needs to be flexible and portable, which can be achieved by adapting the cloud-native software design of the 5G core and transitioning VNFs to Cloud-Native Network Functions (CNFs), i.e., allowing 5G core components (NFs) to run on containers enabling automation across any cloud environment. Cloud-native principles can be applied to control and user plane functions of the 5G core, AMF, SMF, and UPF respectively, to achieve flexibility, scalability and performance efficiency. Operators can make full use of the cloud-native approach in the UPF, thus eliminating the need for dedicated hardware for core network routing and switching.

Design principles and some of the major components that will enable the cloud-native approach in the context of 5G network slicing are described in the following sections.

### A. CLOUD-NATIVE DESIGN PRINCIPLES FOR NETWORK SLICING

#### 1) AGNOSTICITY

Cloud-native is essential for the 5G NFs. For customized services on demand, e.g., URLLC, the 5G network applications should not be built for specific infrastructure. The cloud-native network applications should be able to run on any Kubernetes enabled infrastructure, as these cloud-native applications can be deployed in a distributed manner in the edge, core, or public cloud depending on the service requirements.

### 2) DECOMPOSED SOFTWARE

The 5G network applications should be designed so that the applications can be decomposed into microservices. This will enable independent life cycle management and scalability.

### 3) ORCHESTRATION AND AUTOMATION

To manage the complexity of 5G applications and the service-based use cases, it is necessary to utilize and build out the orchestration needed to manage cloud-native applications and infrastructure.

## *B. CLOUD-NATIVE ENABLING TECHNOLOGIES FOR NETWORK SLICING*

### 1) CONTAINERS

Containers are a lightweight virtualization alternative to VMs [80] that leverage two Linux kernel features: namespace and cgroups. The namespace is used to enable application isolation by providing it with a limited view of the underlying operating system environment, i.e., network resources (IP addresses, routing tables and interfaces). The cgroups provides the capability to enforce limitations and prioritization of system resources (CPU and memory) [32].

The 5G SBA and its use cases, such as MEC and network slicing, rely heavily on virtualization techniques; VM based virtualization adds complexity and overhead to the system as the VMs require packaging of the entire Operating System (OS) along with the hosted applications or functions. When compared to VMs, containers can reduce overhead by packaging only the application or function and the application-specific OS dependencies [32], [81]. One of the most widely used containerization technologies is Docker because of its ability to provide portability and scalability [82]. The advantage of using containers rather than VMs for network slicing can be seen in Fig. 8.

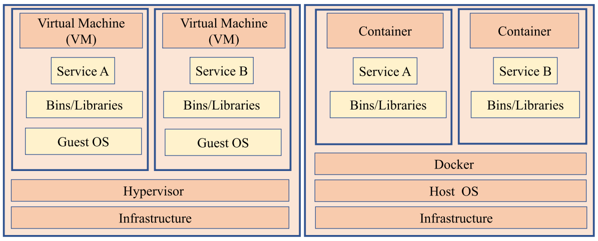


**FIGURE 8.** **Containers vs VMs.**

Containers allow the efficient deployment of microservices, where each service part can be decoupled into separate containers and wrapped into pods where they can communicate with each other. This permits modular development, efficient scaling, and deployment models.

Docker containers are considered to be an integral part of the NFV framework [83] that is the key enabling technology for network slicing. Software developed by the SONATA project called ''vim-emu: A NFV multi-pop emulation platform'' [84] allows VNFs to be provisioned using Docker containers. This emulation platform fully integrates the MANO solutions, e.g., Open Source Mano (OSM) [85], that delivers a MANO stack fully aligned with ETSI NFV information models.

### 2) KUBERNETES

Docker deals with the packaging and distribution of applications or functions, whereas the Kubernetes platform is used to scale, run and monitor the applications or functions [86]. Kubernetes is also referred to as a container orchestrator that provides deployment automation, scheduling, scaling, and coordination of containerized applications [87]. Kubernetes does not run the containers directly, instead, one or more containers are wrapped into a high-level architecture called pods. The containers in the same pod share the resources and network, and communicate with each other. The general architecture of the Kubernetes cluster consists of a master and nodes, as can be seen in Fig. 9. The master is responsible for exposing the Application Program Interface (API) to the developers, scheduling the cluster deployments, e.g. pods, nodes. The nodes contain the container runtime, e.g., one or more Docker containers running inside pods, and an element called kubelet that is responsible for communication between the node and the master.

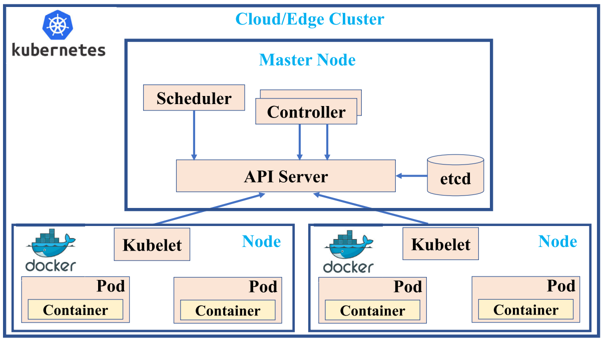


**FIGURE 9.** **Kubernetes edge/cloud cluster.**

### 3) OPENSTACK

OpenStack because of its flexible and modular nature, is also considered as one of the ideal candidates for enabling 5G edge computing use cases [88]. It is open-source software used to build private and public clouds and provides robust support to virtualization and container technologies. OpenStack is also referred to as a cloud OS that manages and controls large pools of resources, e.g., computing, networking, and storage in the DCs [89]. OpenStack provides a highly distributed infrastructure software platform, and it is used in thousands of DCs around the world today. Recently, it has been adopted by the telecommunications industry to advance the edge computing use cases.

## *C. DEPLOYMENT SCENARIOS*

There are multiple deployment scenarios in which the cloud-native approach, e.g., Docker, Kubernetes, and OpenStack, can be used to provision the service-focused use cases of MEC enabled network slicing. For example, the whole Kubernetes cluster can be wrapped in a lightweight package and deployed at the edge, where it can host different vertical applications and cloud-native 5G core NFs as containers running inside pods. Another approach proposed is Huawei's IoT edge platform [90]. Instead of deploying the whole Kubernetes cluster at the edge, the control plane residing in the cloud manages the containers and pods running at the edge nodes. This approach is also proposed by the open source system called KubeEdge [90].

### *D. CLOUD-NATIVE ARCHITECTURE AND ITS ADVANTAGES*

The self-management and scalability capability of cloud-native VNFs differentiates them from the conventional VNFs. Following are some of the key benefits of the cloud-native VNFs that overcomes the limitations of conventional VNFs.

- Automated installation and configuration of VNFs.
- Automated and dynamic scaling of network resources and VNFs depending upon the workload.
- Self-healing and fault-tolerant where the cloud-native orchestration platform automatically restarts the failing VNFs.
- Automated performance monitoring of VNFs for analysis of bottlenecks, for improved overall performance.
- Simplified and softwarized management enabling reduced energy consumption.
- High reusability and portability enabled by light-weight containerization platform.

Previous works on network slicing, identified in a literature search, were found to not fully consider a cloud-native 5G core and microservices architecture. Also, most of the existing research does not present the benefits of using container-based virtualization of VNFs; VM and Hypervisor are often used for virtualization as shown in the Table 3. Based on the design principles identified in this paper, an architecture for MEC enabled cloud-native network slicing is envisioned that exploits the benefits of enabling new features for NFV with cloud-native technologies, e.g., Docker containers and Kubernetes, as shown in Fig. 10. The logical flow of the provisioning of E2E network slices follows the same process as shown in Fig. 4, but the cloud-native technologies are leveraged to envision a cloud-native NFV stack instead of the traditional virtualization orchestrator(s).

### *E. SERVICE CHAINING IN CLOUD-NATIVE ENVIRONMENT*

In a cloud-native environment, services are offered by instantiating service containers or pods to dynamically apply single or multiple services to traffic from one endpoint to another. To create a service chain, SDN can facilitate the creation of tunnels across the underlay network spanning through all services in the chain. Fig. 11 shows two compute nodes deployed in a Kubernetes cluster, each with one service instance and traffic going through all the services to and from one endpoint to the other.

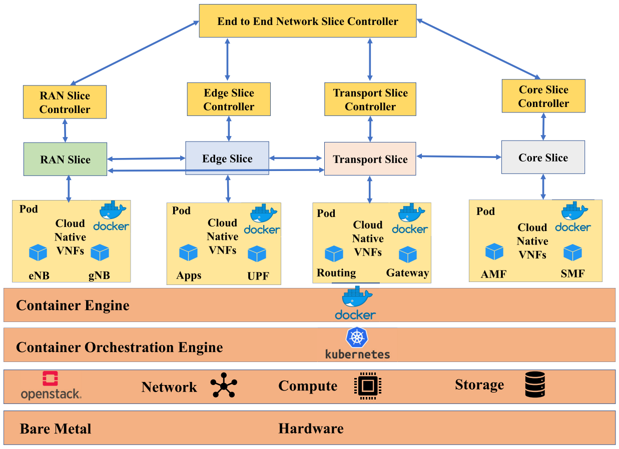


**FIGURE 10.** Cloud-native microservices architecture for network slicing.

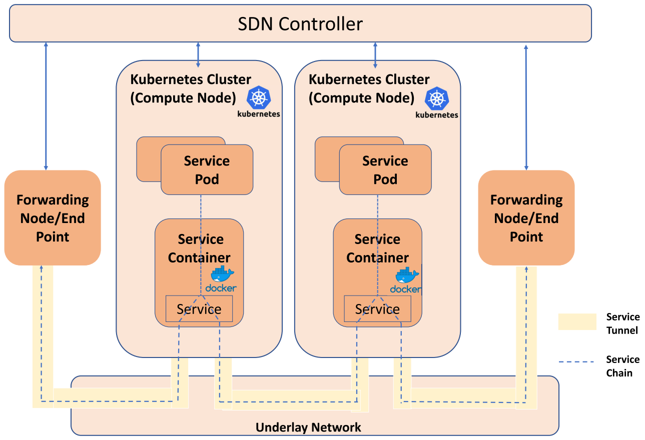


**FIGURE 11.** Service chaining in cloud-native environment.

## VII. CLOUD-NATIVE NETWORK SLICING: USE CASES

### *A. 5G E-hEalth (MISSION-CRITICAL Communication)*

The 5G e-health connected ambulance is one of the most important use cases of network slicing as envisioned by the SliceNet project [64]. The use case intends to improve ambulance services and provide real-time health care and first-aid to patients. The vision of this use case is to enable a connected ambulance to serve as a mobile edge (connection hub) for the emergency medical equipment or wearables, enabling real-time and dynamic streaming and storing of patient health data to the emergency team awaiting at the destination hospital. Through real-time video feeds and provision of patient insights, the emergency team will be able to support the paramedics attending the patients with intelligent decision support.

This use case demands the deployment of URLLC slices on-demand to enable intensive and real-time patient data (video feeds) communication between the paramedics and the emergency team waiting at the destination hospital. This use case can be supported by different scenarios of the envisioned cloud-native 5G microservices architecture for network slicing. For example, consider the case where the ambulance has its own mini DC, with the capability to host VNFs. As the connected ambulance case requires the provision of URLLC, the control plane VNFs can be located in the central DC whereas the user plane VNFs as well as Virtual RAN (vRAN) and connected ambulance applications can be located in the ambulance mini DC as shown in Fig. 12.

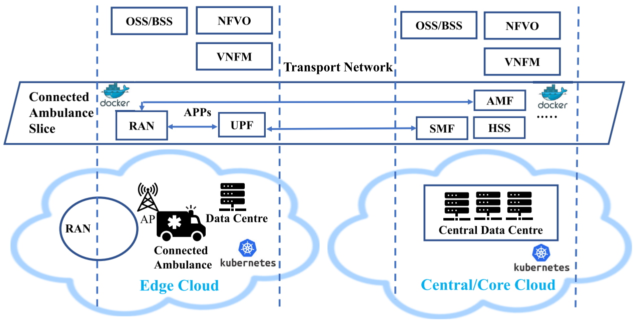


**FIGURE 12.** 5G e-health use case.

As the URLLC slice is requested, the E2E slicing orchestrator or E2E network slice controller designs and instantiates NST based on the network capabilities and specific requirements of tenants. NSI is instantiated as an output of NST, i.e., 5G core CNFs are instantiated and deployed in a flexible and distributed way across the cloud and edge. These CNFs register with the NRF of the 5G SBA. The cloud-native 5G microservices architecture, as described above, facilitates the real-time and distributed deployment of the 5G core CNFs across the cloud and edge. By using a containers based solution, the 5G core CNFs and RAN components can be deployed as containers, whereas the SDN transport network solution can facilitate the service chaining and connectivity among different 5G core CNFs. The Layer 3 Virtual Private Network (L3VPN) is configured over the transport network to create an E2E slice. In this use case, two Kubernetes clusters can be deployed with one at the core cloud and the other at the edge cloud to host control and user plane 5G core CNFs as containers, respectively. The RAN components can also be deployed as containers on the edge cluster. Before or during this operation, the service providers will be able to deploy and scale different functions on demand. The cloud-native microservices solution based on containerization technology enables the real-time deployment of the service components.

Similar use cases of cloud-native design, e.g., Kubernetes for 5G network slicing and vertical services, are provided by the 5G-PPP in [91]. The 5G-PPP software network working group emphasizes on the importance of cloud-native design to meet emerging customers' demands for new services in [91]. The 5G-PPP group also recommends the network operators adopt the cloud-native microservices architecture.

### B. V2X USE CASE FOR URLLC SERVICES

V2X communication requires the provision of high bandwidth and URLLC services. Certain V2X applications such as advanced safety applications, e.g., machine learning applications for pattern detection, classification, and cooperative driving for V2V communication, require low latency communication. The European METIS project envisions the E2E latency to less than five milliseconds and reliability of 99.99 percent [92].

MEC holds significant promise here by offering the cloud computing resources closer to the end-users at the edge of the network. MEH provides the compute, storage and networking resources to host different V2X applications and ME services. In response to the high mobility requirements of the V2X use case, applications and services should seamlessly migrate from one MEH to the adjacent MEH, while ensuring reduced latency and service interruptions.

This use case can be supported by the containerized based approach for network slicing. The lightweight Docker containers support portability and real-time deployment of distributed applications. The features provided by the Docker container-based solutions make it a key enabler for the MEC [93], [94].

For the high mobility scenario, as the vehicle moves from the service area of one MEH to another, the MEO by the use of the Radio Network Information Service (RNIS), tracks the trajectory of the vehicle and finds out the appropriate target MEH. By taking full advantage of the modularity offered by the Docker containerized solution, applications and services can be migrated to the target MEH in real-time, i.e. service and application instances are replicated in the target MEH. The SDN controller reconfigures the traffic rules and installs new flows to reroute the traffic from the vehicle towards the target MEH. This use case can be seen in Fig. 13.

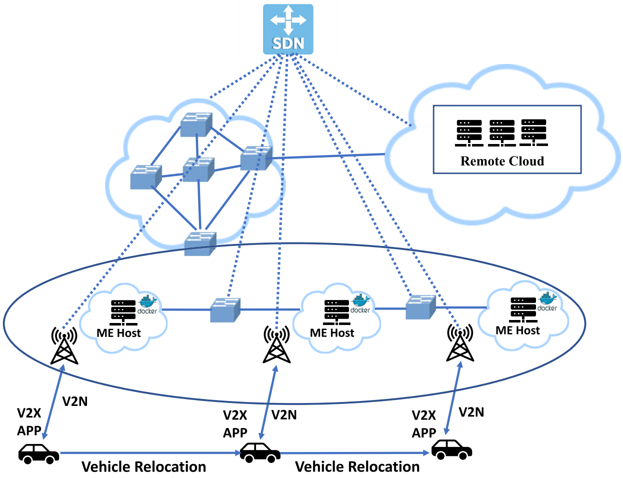


**FIGURE 13.** Docker containerization solution for V2X slice.

A similar approach for real-time service migration is proposed in [95] and our previous work [32]. In [32], an SDN enhanced edge computing architecture is proposed that integrates the containerization engine for the provision of V2X URLLC slice in a high mobility scenario. Results provided show that by using the Docker containerized approach for service migration, the service downtime values are reduced.

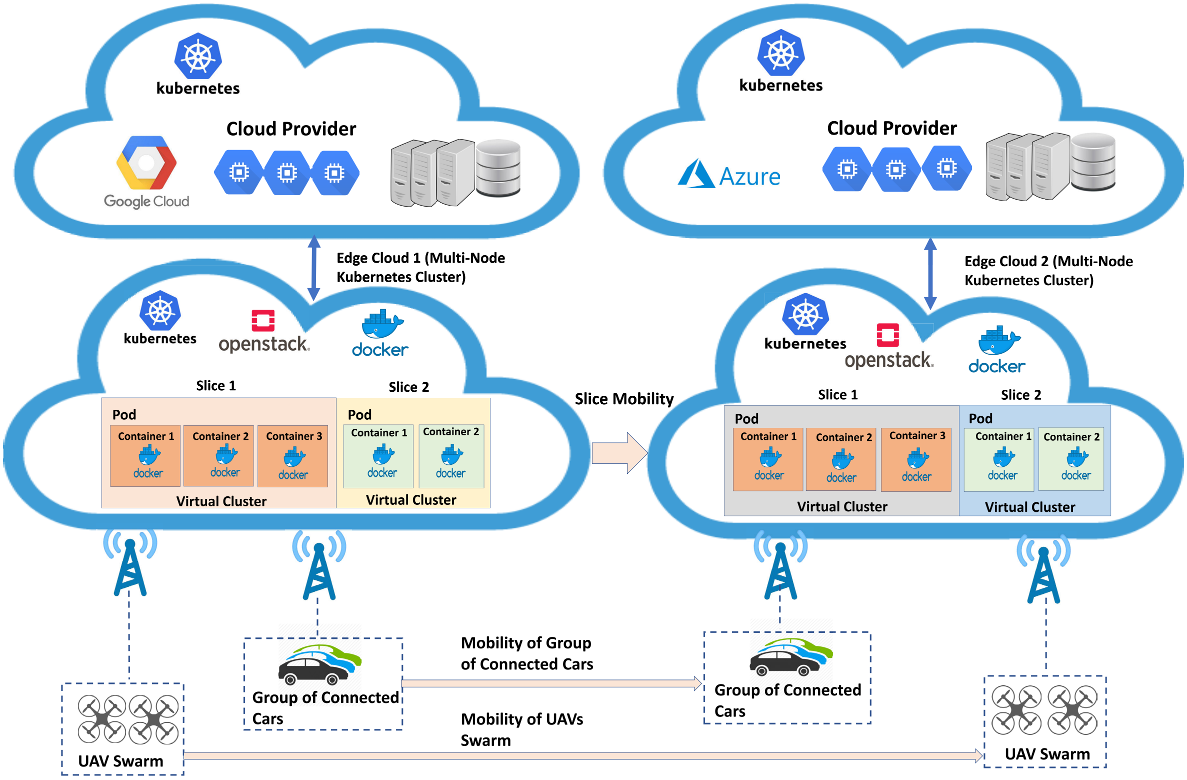


**FIGURE 14.** Virtual cluster based network slice migration in cloud-native edge clouds.

### C. NETWORK SLICE MOBILITY AND SERVICE MIGRATION IN CLOUD-NATIVE EDGE CLOUDS

5G supports new vertical use cases that require new mobility approaches beyond the conventional device-centric approaches. For example, the mobility of low-latency communication services deployed at the edge and shared by a group of mobile users, e.g., a group of connected cars or Unmanned Aerial vehicles (UAVs). To ensure service continuity, as the users move from the service area of one edge cloud to another, the service configuration files and slice resources should also move in real-time to the destination edge cloud. The cloud-native approach can be used to support service portability as it provides the capacity to move the edge services from one edge cloud to another in real-time by making use of container advanced features.

We conducted a preliminary evaluation of the cloud-native approach for supporting real-time migration of communication services shared by a group of mobile users across different edge clouds. Fig. 14 portrays the real experimental testbed set up to emulate the slice mobility across different edge clouds deployed as multi-node Kubernetes clusters in different regions in the Google Cloud Platform. The Kubernetes clusters are allocated 3 CPUs and 11.25 GB of memory. We assume that a group of mobile users, e.g., a group of UAV and a group of connected cars, are being served by slice 1 and slice 2, respectively. The slices consist of multiple services running as containers in their respective pods. The slices are isolated by forming virtual clusters by defining different namespaces within the Kubernetes cluster. The virtual clusters allow seamless and parallel migration of all the services defined within the namespace.

Conventionally, Kubernetes doesn't support service migration between clusters. We made use of the open-source project '*Velero*' [96] that permits the backup of cluster resources and persistent volumes and migration of the backup to another cluster. We considered the different sizes of network slices by defining pods having 2, 3, and 4 service instances running as containers. The size of these service containers is taken as 114MB, 127MB, 107MB, and 197 MB. We fixed random sizes of service instances to represent the services with varying data demands. We assumed that the group of connected UAVs and cars, being served by slice 1 and 2, moves out of the service area of the source edge cloud towards the destination edge cloud, and the service configuration files as well as the cluster resources are migrated to the new destination edge cloud. It takes approximately 20 seconds to migrate the service configuration files and 23, 30, and 39 seconds to restore the pods having 2, 3 and 4 service instances, respectively, as seen in Fig. 15. It was noted that the size of the service containers does not impact the latency induced by the migration of service configurations; however, the pods having service instances of larger container sizes take longer service restoration time as seen in Fig. 15. This is because the pods containing more service instances

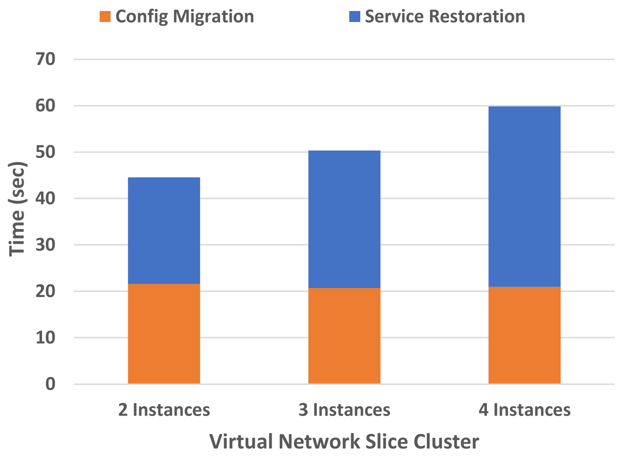


**FIGURE 15.** Network slice migration in cloud-native edge clouds.

of higher data sizes induce high computing and processing demands.

The preliminary results show that because of the OS-independent virtualization and portability offered by the cloud-native environment, it takes less than 60 seconds to migrate and restore multiple services in the destination edge cloud. The results highlight that with an accurate prediction of the time instant at which the service pre-relocation should start, service continuity can be maintained for the group of mobile users. The results also indicate that the time taken is substantial, being seconds rather than the preferred micro-seconds for real-time service or application continuity.

## VIII. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

### A. CLOUD-NATIVE 5G CORE ADAPTABILITY

Cloud-native 5G core adaptability is necessary to take full advantage of cloud-native functionalities, e.g., service automation, dynamic application and NF scaling, and efficient use of storage and computing capabilities. To fully adopt the cloud-native architecture, the core NFs should be designed in a way that they are fully compatible with the cloud-native microservices architecture. For example, the role of UPF is to handle the traffic coming from the end-user devices and to perform several operations such as managing sessions and routing traffic to the edge. To make UPF cloud-native, the challenge is to design a packet processing solution that is fully compatible with containers and can be scaled elastically. In addition, it should also be cost-effective i.e., reduced CPU requirement.

### B. HYBRIDIZATION OF CONTAINERS AND VMs

It might not be possible to make NFs cloud-native and to adapt to the stateless microservices approach because not all applications might benefit from the cloud-native approach, e.g., LANs and WANs. Coexistence of both technologies might be the way forward. Therefore, more research efforts are needed to develop an orchestration platform that would interconnect two different types of workloads, e.g., VMs orchestrated by OpenStack and pods orchestrated by Kubernetes. It's an area with limited research carried out so far.

### C. NETWORK SLICE ISOLATION FOR THIRD-PARTY SERVICES IN A CLOUD-NATIVE ENVIRONMENT

Future 5G mobile networks are expected to support the creation of network slices that might be made available to third-party organizations, e.g., enterprises. Traditional cloud-native orchestration platform(s), e.g., Kubernetes, provide a flat networking model wherein resources created, e.g., pods, can talk to each other. Therefore, a prerequisite for supporting highly sensitive services in a network slice is to develop an effective policy to provide isolation between different pods and services.

### D. DYNAMIC SERVICE CHAINING IN MODULAR SOFTWARE ARCHITECTURE OF 5G

The 5G SBA enables decoupled network functionalities, e.g., CUPS. When compared to the traditional network entities that are closely-coupled to each other, the 5G SBA is envisioned to contain loosely-coupled, and modular NFs and services. This improves the network programmability, and each service can be realized by a set of specific functionalities depending upon the service type and requirements. Therefore, each service can be updated or scaled independently of others, thus enabling a highly flexible and scalable architecture. The self-contained smaller and modular NFs are connected and flexibly chained to realize an E2E network slice for a dedicated service [16]. However, because of the dynamic and modular 5G software architecture, chaining and connection of the dynamic NFs components is a very challenging task due to the number of connectivity interfaces involved in the process that remain need to be designed.

### E. TRUST MANAGEMENT AMONG MULTIPLE ADMINISTRATIVE DOMAINS

Slicing federation among multiple administrative domains is an important network slicing challenge [64]. In a high mobility scenario when URLLC slicing is requested, e.g., V2X communications there is a need for slicing federation. The security and trust management between different vendors across different administrative domains that share physical resources to realize slicing federation needs to be investigated.

### F. MOVING INTELLIGENCE AND USER-SPECIFIC COMPUTATION TO EDGE

6G envisions taking the intelligence and user-specific computation to the edge [97]. Edge computing will permit the computation-intensive and low-latency applications to run at the edge. The increasing number of smart devices and smart city applications generate a massive amount of local data that is transmitted to the centralized cloud for processing, resulting in latency and computational complexity at the cloud. To deal with these challenges, the 6G vision is to facilitate artificial intelligence (AI) use cases such as self-learning networks, e.g., Deep Reinforcement Learning (DRL) over wireless links, at the edge cloud. Shifting the AI and computation capability to the edge will enable new use cases and services such as self-driving cars, mobile virtual reality, and mixed reality applications. The challenge is to design efficient and new neural network architectures over the wireless link at the network edge to realize the new services.

### G. CONVERGENCE OF JOINT COMMUNICATION, COMPUTATION, CACHING, AND CONTROL RESOURCES

One of the 6G initiatives is the convergence of joint communication, computation, caching, and control (4C) resources [98], [99] at the edge. MEC has been standardized to augment cloud computing by reducing the network delay and computational load on the cloud servers. However, an edge server has minimal network resources, and when operated independently, it can't handle the computational load and big data demands generated by IoT applications. Therefore, new mechanisms and algorithms need to be developed to facilitate the cross-domain federation of MEC resources enabling 4C at the edge. The existing works on 4C in MEC do not consider the challenges introduced by a mobile environment, e.g., vehicular communications, where federation of MEC servers becomes essential for maximum bandwidth savings and reduced service migration costs.

### H. NETWORK SLICE MOBILITY AND DYNAMIC SERVICE MIGRATION

The provision of network slices on-demand in mobile cellular networks requires dynamic migration of NFs and services from one edge cloud to another for service continuity. The preliminary results, as provided in the research shown in Fig. 15, exhibits that migration of multiple services used by a group of mobile users takes seconds rather than the preferred microseconds as E2E latency envisioned for the URLLC slice [92]. Therefore, to ensure service continuity, it is essential to develop self-learning networks, e.g., DRL, for accurate prediction of the user mobility patterns and early initiation of the service migration process. Research on service migration generally considers a single service and network scenario [100]–[102]. However, a group of users demanding multiple-services simultaneously with different service requirements is becoming a norm now. Service migration as users move across the network can be cost-inefficient, consume limited bandwidth, and the target edge cloud may not have the resources needed to support service continuity. Therefore, services should be migrated depending upon the service requirements, e.g., latency and bandwidth. Efficient algorithms should be designed to answer the following challenges: when to migrate, what to migrate, and where to migrate, in particular for multiple services and multiple mobile network operators scenarios.

### I. INTEGRATION OF MACHINE LEARNING FOR EFFICIENT RESOURCE UTILIZATION

The role of machine learning algorithms, e.g., Support Vector Machine (SVM) and DRL, in network slicing is yet to be fully exploited. For example, SVM can be used as an efficient tool for service requirements classification assisting in network slice selection and DRL can be used in dynamic workload dependant resource allocation problems such as efficient and dynamic allocation of resources to each logical network (slice resource allocation) over a common shared physical network.

### J. CONTROLLER PLACEMENT SOLUTIONS

Network slicing supports the realization of E2E services on demand, each with its own specific requirements, e.g., latency, bandwidth, and availability. Depending upon the service requirements, 5G core NFs can be deployed across public, local, or private DCs in any geographical location. Thus, the challenge is to find the optimal placement strategy for the E2E network slice controller that will enable slice management considering the specific service requirements. Also, determining the optimal number of controllers required per slice is an open issue to be investigated.

### K. CONTROL PLANE ISOLATION AND INTERACTIVITY

Business verticals may have different service requirements, such as the automotive industry may require a control application, i.e., mobility management application to fulfill the high mobility service requirements. The challenges are to provide an isolated and customized control plane for different vertical customers instead of a common control plane. This will permit the vertical business to provide a customized control application to satisfy service requirements. Also, new interfaces and definitions are required that will facilitate the interaction of the SDN control plane with the network slices. The SDN control plane interactivity with the network slices is an open issue to be investigated.

## IX. CONCLUSION

Network slicing based on MEC, SDN, NFV and a cloud-native 5G core is emerging as a key enabling technology for 5G network operators and service providers to achieve new revenue opportunities and provide new and innovative customized services on demand. However, to fully achieve the service-focused goals of 5G, there are multiple technical issues and challenges remaining, such as slicing federation among multiple administrative domains, cloud-native 5G core adaptability to support the MEC use cases, dynamic service chaining, and controller design and placement. This paper investigates the recent efforts and progress made in realizing E2E network slicing, its key enabling technologies such as NFV for virtualization support, MEC for URLLC services, cloud-native 5G core for service automation and SDN for dynamic service chaining and VNF management. As the NFV and 5G use cases have started to shift to a cloud-native platform, and adoption of cloud-native applications and NFs is accelerating among network operators for services scalability, agnosticity, portability, and automation. An SDN and MEC enabled cloud-native architecture for 5G network slicing is envisioned in this article, along with some of its potential use cases. The recent progress made by industry standardization and research into 5G network slicing is discussed, and selected open issues and future research directions are identified, to realize the future vision of 5G network slicing.

## REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[2] Huawei, "5G service-guaranteed network slicing white paper," China Mobile Commun. Corp., Beijing, China, Tech. Rep., Feb. 2017.

[3] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf, "On multi-domain network slicing orchestration architecture and federated resource control," *IEEE Netw.*, vol. 33, no. 5, pp. 242–252, Sep. 2019.

[4] M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Comput. Surv.*, vol. 51, no. 6, p. 114, Feb. 2019.

[5] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 462–476, Sep. 2016.

[6] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.

[7] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018.

[8] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Commun. Standards Mag.*, vol. 2, no. 1, pp. 60–65, Mar. 2018.

[9] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984.

[10] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36009–36028, Feb. 2020.

[11] D. Sabella, V. Sukhomlinov, L. Trang, S. Kekki, P. Paglierani, R. Rossbach, X. Li, Y. Fang, D. Druta, F. Giust, and L. Cominardi, "Developing software for multi-access edge computing," ETSI White Paper 20, Feb. 2019.

[12] S. Redana, Ö. Bulakci, A. Zafeiropoulos, A. Gavras, A. Tzanakaki, A. Albanese, A. Kousaridas, A. Weit, B. Sayadi, B. T. Jou, and C. J. Bernardos, "5G PPP architecture working group: View on 5G architecture," Eur. Commission, Brussels, Belgium, Tech. Rep., Jun. 2019.

[13] MEC ETSI ISG, "Multi-access edge computing (MEC); support for network slicing," ETSI, Sophia-Antipolis, France, Tech. Rep. GR MEC 024, Nov. 2019.

[14] *Framework for the Support of Network Slicing in the IMT-2020 Network*, document ITU-T Y.3112, Dec. 2018.

[15] R. Rokui, S. Homma, D. R. Lopez, X. de Foy, L. M. Contreras-Murillo, J. J. Ordonez-Lucena, P. Martinez-Julia, M. Boucadair, P. Eardley, K. Makhijani, and H. Flinck, "5G transport slice connectivity interface," IETF, Fremont, CA, USA, Tech. Rep., Jul. 2019.

[16] S. Zhang, "An overview of network slicing for 5G," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 111–117, Jun. 2019.

[17] X. Li, K. Jiao, F. Jiang, J. Wang, and M. Pan, "A service-oriented spectrum-aware RAN-slicing trading scheme under spectrum sharing," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11303–11317, Nov. 2020.

[18] S. Sharma, R. Miller, and A. Francini, "A cloud-native approach to 5G network slicing," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 120–127, Aug. 2017.

[19] X. Costa-Perez, A. Garcia-Saavedra, X. Li, T. Deiss, A. de la Oliva, A. di Giglio, P. Iovanna, and A. Moored, "5G-crosshaul: An SDN/NFV integrated fronthaul/backhaul transport network architecture," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 38–45, Feb. 2017.

[20] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[21] J. Gil Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[22] V.-G. Nguyen, A. Brunstrom, K.-J. Grinnemo, and J. Taheri, "SDN/NFV-based mobile packet core network architectures: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1567–1602, 3rd Quart., 2017.

[23] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 655–685, 1st Quart., 2016.

[24] I. Afolabi, M. Bagaa, T. Taleb, and H. Flinck, "End-to-end network slicing enabled through network function virtualization," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2017, pp. 30–35.

[25] X. De Foy and A. Rahman, "Network slicing—3GPP use case," Inter-Digital Commun., LLC, Wilmington, DE, USA, Tech. Rep., Oct. 2017.

[26] X. Costa-Perez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 27–35, Jul. 2013.

[27] G. Brown, "Service-based architecture for 5G core networks," Huawei White Paper 1, Nov. 2017.

[28] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, and K. W. Wen, "MEC in 5G networks," ETSI White Paper 28, Jun. 2018, pp. 1–28.

[29] D. Chandramouli and G. Gkellas, "5G-service-based architecture," in *Proc. Wiley 5G Ref: The Essential 5G Reference Online*, May 2019, pp. 1–15.

[30] MEC ETSI ISG, "Multi-access edge computing (MEC); framework and reference architecture," ETSI, Sophia-Antipolis, France, Tech. Rep. GS MEC 003, Jan. 2019.

[31] *Cisco Converged 5G xHaul Transport, White Paper*, Cisco Systems Inc, San Francisco, CA, USA, 2018.

[32] S. D. A. Shah, M. A. Gregory, S. Li, and R. D. R. Fontes, "SDN enhanced multi-access edge computing (MEC) for E2E mobility and QoS management," *IEEE Access*, vol. 8, pp. 77459–77469, Apr. 2020.

[33] C. Campolo, R. Fontes, A. Molinaro, C. E. Rothenberg, and A. Iera, "Slicing on the road: Enabling the automotive vertical through 5G network softwarization," *Sensors*, vol. 18, no. 12, p. 4435, Dec. 2018.

[34] A. Ksentini and P. A. Frangoudis, "Toward slicing-enabled multi-access edge computing in 5G," *IEEE Netw.*, vol. 34, no. 2, pp. 99–105, Mar. 2020.

[35] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-access edge computing: A survey," *IEEE Access*, vol. 8, pp. 197017–197046, Oct. 2020.

[36] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, Jun. 2020.

[37] L. Cominardi, T. Deiss, M. Filippou, V. Sciancalepore, F. Giust, and D. Sabella, "MEC support for network slicing: Status and limitations from a standardization viewpoint," *IEEE Commun. Standards Mag.*, vol. 4, no. 2, pp. 22–30, Jun. 2020.

[38] S. D'Oro, L. Bonati, F. Restuccia, M. Polese, M. Zorzi, and T. Melodia, "Sl-EDGE: Network slicing at the edge," 2020, *arXiv:2005.00886*. [Online]. Available: http://arxiv.org/abs/2005.00886

[39] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint network slicing and mobile edge computing in 5G networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[40] Q. Liu, T. Han, and E. Moges, "EdgeSlice: Slicing wireless edge computing network with decentralized deep reinforcement learning," 2020, *arXiv:2003.12911*. [Online]. Available: http://arxiv.org/abs/2003.12911

[41] S. Jošilo and G. Dán, "Joint wireless and edge computing resource management with dynamic network slice selection," 2020, *arXiv:2001.07964*. [Online]. Available: http://arxiv.org/abs/2001.07964

[42] J. Feng, Q. Pei, F. R. Yu, X. Chu, J. Du, and L. Zhu, "Dynamic network slicing and resource allocation in mobile edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7863–7878, Jul. 2020.

[43] T. Sanguanpuak, N. Rajatheva, D. Niyato, and M. Latva-aho, "Network slicing with mobile edge computing for micro-operator networks in beyond 5G," in *Proc. 21st Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Nov. 2018, pp. 352–357.

[44] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint planning of network slicing and mobile edge computing in 5G networks," 2020, *arXiv:2005.07301*. [Online]. Available: http://arxiv.org/abs/2005.07301

[45] S. D. A. Shah, D. Kim, P. Khan, H. Kim, and S. Han, "A two step multi-carrier proportional fair scheduling scheme for cloud radio access networks," *Int. J. Interdiscipl. Telecommun. Netw.*, vol. 10, no. 1, pp. 49–62, Jan. 2018.

[46] R. A. Addad, T. Taleb, H. Flinck, M. Bagaa, and D. Dutra, "Network slice mobility in next generation mobile systems: Challenges and potential solutions," *IEEE Netw.*, vol. 34, no. 1, pp. 84–93, Jan. 2020.

[47] F. De Vita, G. Nardini, A. Virdis, D. Bruneo, A. Puliafito, and G. Stea, "Using deep reinforcement learning for application relocation in multi-access edge computing," *IEEE Commun. Standards Mag.*, vol. 3, no. 3, pp. 71–78, Sep. 2019.

[48] F. Z. Yousaf, M. Gramaglia, V. Friderikos, B. Gajic, D. von Hugo, B. Sayadi, V. Sciancalepore, and M. R. Crippa, "Network slicing with flexible mobility and QoS/QoE support for 5G networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 1195–1201.

[49] F. Meneses, R. Silva, D. Corujo, A. Neto, and R. L. Aguiar, "Dynamic network slice resources reconfiguration in heterogeneous mobility environments," *Internet Technol. Lett.*, vol. 2, no. 4, p. e107, May 2019.

[50] N. Mouawad, R. Naja, and S. Tohme, "Inter-slice mobility management solution in V2X environment," in *Proc. Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2019, pp. 1–6.

[51] MEC ETSI ISG, "Multi-access edge computing (MEC); study on MEC support for V2X use cases," ETSI, Sophia-Antipolis, France, Tech. Rep. GR MEC 022, Sep. 2018.

[52] P. Suengyoung, D. Saikia, and K. Seokhwan, "SDN-based service chaining system," U.S. Patent 9 654 395, May 16, 2017.

[53] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an SDN-enabled NFV architecture," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 187–193, Apr. 2015.

[54] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.

[55] I. F. Akyildiz, S.-C. Lin, and P. Wang, "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation," *Comput. Netw.*, vol. 93, pp. 66–79, Dec. 2015.

[56] X. Li, R. Casellas, G. Landi, A. de la Oliva, X. Costa-Perez, A. Garcia-Saavedra, T. Deiss, L. Cominardi, and R. Vilalta, "5G-crosshaul network slicing: Enabling multi-tenancy in mobile transport networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 128–137, Aug. 2017.

[57] S. K. Fayazbakhsh, V. Sekar, M. Yu, and J. C. Mogul, "FlowTags: Enforcing network-wide policies in the presence of dynamic middlebox actions," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 19–24.

[58] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, "SIMPLE-fying middlebox policy enforcement using SDN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 27–38, Sep. 2013.

[59] N. Akhtar, I. Matta, A. Raza, L. Goratti, T. Braun, and F. Esposito, "Virtual function placement and traffic steering over 5G multi-technology networks," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 114–122.

[60] J. Wang and M. Luo, "Packet prioritization in a software-defined network implementing OpenFlow," U.S. Patent 9 923 831, Mar. 20, 2018.

[61] N. Kitsuwan and E. Oki, "Implementation of traffic splitting using meter table in software-defined networking," *J. Eng.*, vol. 2017, no. 12, pp. 662–665, Dec. 2017.

[62] M. K. Jaiswal, "Introduction to OpenFlow," in *Innovations in Software-Defined Networking and Network Functions Virtualization*. Hershey, PA, USA: IGI Global, 2018, pp. 52–71.

[63] P. K. Chartsias, A. Amiras, I. Plevrakis, I. Samaras, K. Katsaros, D. Kritharidis, E. Trouva, I. Angelopoulos, A. Kourtis, M. S. Siddiqui, A. Vines, and E. Escalona, "SDN/NFV-based end to end network slicing for 5G multi-tenant networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–5.

[64] Q. Wang *et al.*, "SliceNet: End-to-end cognitive network slicing and slice management framework in virtualised multi-domain, multi-tenant 5G networks," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2018, pp. 1–5.

[65] H.-T. Chien, Y.-D. Lin, C.-L. Lai, and C.-T. Wang, "End-to-end slicing as a service with computing and communication resource allocation for multi-tenant 5G systems," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 104–112, Oct. 2019.

[66] A. Mayoral, R. Vilalta, R. Casellas, R. Martinez, and R. Munoz, "Multi-tenant 5G network slicing architecture with dynamic deployment of virtualized tenant management and orchestration (MANO) instances," in *Proc. 42nd Eur. Conf. Opt. Commun. (ECOC)*, vol. 2016, pp. 1–3.

[67] I. Afolabi, J. Prados-Garzon, M. Bagaa, T. Taleb, and P. Ameigeiras, "Dynamic resource provisioning of a scalable E2E network slicing orchestration system," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2594–2608, Nov. 2020.

[68] I. Afolabi, A. Ksentini, M. Bagaa, T. Taleb, M. Corici, and A. Nakao, "Towards 5G network slicing over multiple-domains," *IEICE Trans. Commun.*, vol. E100.B, no. 11, pp. 1992–2006, 2017.

[69] Q. Li, G. Wu, A. Papathanassiou, and U. Mukherjee, "An end-to-end network slicing framework for 5G wireless communication systems," 2016, *arXiv:1608.00572*. [Online]. Available: http://arxiv.org/abs/1608.00572

[70] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[71] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5G: Challenges and opportunities," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 20–27, Sep. 2017.

[72] K. Katsalis, N. Nikaein, E. Schiller, A. Ksentini, and T. Braun, "Network slices toward 5G communications: Slicing the LTE network," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 146–154, Aug. 2017.

[73] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72–79, May 2017.

[74] M. Vincenzi, A. Antonopoulos, E. Kartsakli, J. Vardakas, L. Alonso, and C. Verikoukis, "Multi-tenant slicing for spectrum management on the road to 5G," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 118–125, Oct. 2017.

[75] A. Rostami, P. Öhlén, M. A. S. Santos, and A. Vidal, "Multi-domain orchestration across RAN and transport for 5G," in *Proc. ACM SIGCOMM Conf.*, Aug. 2016, pp. 613–614.

[76] G. Tseliou, F. Adelantado, and C. Verikoukis, "A base station agnostic network slicing framework for 5G," *IEEE Netw.*, vol. 33, no. 4, pp. 82–88, Jul. 2019.

[77] A. Boubendir, F. Guillemin, C. Le Toquin, M.-L. Alberi-Morel, F. Faucheux, S. Kerboeuf, J.-L. Lafragette, and B. Orlandi, "Federation of cross-domain edge resources: A brokering architecture for network slicing," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 415–423.

[78] R. A. Addad, M. Bagaa, T. Taleb, D. L. C. Dutra, and H. Flinck, "Optimization model for cross-domain network slices in 5G networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 5, pp. 1156–1169, May 2020.

[79] J. Ordonez-Lucena, C. Tranoris, J. Rodrigues, and L. M. Contreras, "Cross-domain slice orchestration for advanced vertical trials in a multi-vendor 5G facility," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2020, pp. 40–45.

[80] K. Gillani and J.-H. Lee, "Comparison of linux virtual machines and containers for a service migration in 5G multi-access edge computing," *ICT Exp.*, vol. 6, no. 1, pp. 1–2, Mar. 2020.

[81] M. Chae, H. Lee, and K. Lee, "A performance comparison of Linux containers and virtual machines using docker and KVM," *Cluster Comput.*, vol. 22, no. S1, pp. 1765–1775, Jan. 2019.

[82] S. K. Guru, M. V. T. Patil, and A. Dhus, "Survey on docker," *Nat. J. Comput. Appl. Sci.*, vol. 2, no. 3, pp. 5–9, Oct. 2019.

[83] G. A. Carella, M. Pauls, T. Magedanz, M. Cilloni, P. Bellavista, and L. Foschini, "Prototyping NFV-based multi-access edge computing in 5G ready networks with open baton," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–4.

[84] M. Peuster, H. Karl, and S. van Rossem, ''MeDICINE: Rapid prototyping of production-ready network services in multi-PoP environments,'' in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 148–153.

[85] A. Israel, A. T. Sepulveda, A. Reid, F. Vicens, F. J. R. Salguero, G. G. de Blas, G. Lavado, M. Shuttleworth, M. Harper, and M. Marchetti ''OSM release FIVE technical overview,'' ETSI White Paper, Jan. 2019.

[86] J. Shah and D. Dubaria, ''Building modern clouds: Using docker, kubernetes & Google cloud platform,'' in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 0184–0189.

[87] E. Casalicchio, ''Container orchestration: A survey,'' in *Proc. Syst. Modeling, Methodol. Tools*, 2019, pp. 221–235.

[88] G. Merlino, R. Dautov, S. Distefano, and D. Bruneo, ''Enabling workload engineering in edge, fog, and cloud computing through openstack-based middleware,'' *ACM Trans. Internet Tech.*, vol. 19, no. 2, pp. 1–22, 2019.

[89] J. Hao, K. Ye, and C.-Z. Xu, ''Live migration of virtual machines in OpenStack: A perspective from reliability evaluation,'' in *Proc. Int. Conf. Cloud Comput.*, 2019, pp. 99–113.

[90] Y. Xiong, Y. Sun, L. Xing, and Y. Huang, ''Extend cloud to edge with KubeEdge,'' in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 373–377.

[91] *Cloud-Native and Verticals' Services*, 5G-PPP, 5G-PPP Software Network Working Group, Aug. 2019.

[92] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, ''A survey on low latency towards 5G: RAN, core network and caching solutions,'' *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, 4th Quart., 2018.

[93] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, ''Consolidate IoT edge computing with lightweight virtualization,'' *IEEE Netw.*, vol. 32, no. 1, pp. 102–111, Jan. 2018.

[94] G. Avino, M. Malinverno, F. Malandrino, C. Casetti, and C. F. Chiasserini, ''Characterizing docker overhead in mobile edge computing scenarios,'' in *Proc. Workshop Hot Topics Container Netw. Networked Syst.*, Aug. 2017, pp. 30–35.

[95] C. Campolo, A. Iera, A. Molinaro, and G. Ruggeri, ''MEC support for 5G-V2X use cases through docker containers,'' in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[96] S. Kriss, A. Amarnath, C. Campos, N. Brubaker, T. Hinderliter, and S. Bauman. (2019). *Backup and Migrate Kubernetes Applications and Their Persistent Volumes*. [Online]. Available: https://github.com/vmware-tanzu/velero

[97] M. Katz, M. Matinmikko-Blue, and M. Latva-Aho, ''6Genesis flagship program: Building the bridges towards 6G-enabled wireless smart society and ecosystem,'' in *Proc. IEEE 10th Latin-Amer. Conf. Commun. (LATINCOM)*, Nov. 2018, pp. 1–9.

[98] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, W. Saad, D. Niyato, and C. S. Hong, ''Joint communication, computation, caching, and control in big data multi-access edge computing,'' *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1359–1374, Jun. 2020.

[99] Y. Zhou, L. Liu, L. Wang, N. Hui, X. Cui, J. Wu, Y. Peng, Y. Qi, and C. Xing, ''Service-aware 6G: An intelligent and open network based on the convergence of communication, computing and caching,'' *Digit. Commun. Netw.*, vol. 6, no. 3, pp. 253–260, Aug. 2020.

[100] P. Bellavista, A. Corradi, L. Foschini, and D. Scotece, ''Differentiated service/data migration for edge services leveraging container characteristics,'' *IEEE Access*, vol. 7, pp. 139746–139758, Sep. 2019.

[101] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, ''Live service migration in mobile edge clouds,'' *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 140–147, Feb. 2018.

[102] F. De Vita, G. Nardini, A. Virdis, D. Bruneo, A. Puliafito, and G. Stea, ''Using deep reinforcement learning for application relocation in multi-access edge computing,'' *IEEE Commun. Standards Mag.*, vol. 3, no. 3, pp. 71–78, Sep. 2019.

**SYED DANIAL ALI SHAH** received the B.S. degree from the University of Engineering and Technology, Taxila, Pakistan, in 2016, and the master's degrees in research from Incheon National University (INU), South Korea, in 2018. He is currently pursuing the Ph.D. degree with the School of Engineering, RMIT University, Melbourne, Australia. He worked as a Research Assistant with INU for two years. He is currently an Academic Staff with the School of Engineering, RMIT University. His research interests include 5G/6G, cloud/edge computing, software-defined wireless networks, vehicular networks, and network design.

**MARK A. GREGORY** (Senior Member, IEEE) received the Ph.D. degree from RMIT University, Melbourne, Australia, in 2008. He is currently an Associate Professor with the School of Engineering, RMIT University. In 2009, he received an Australian Learning and Teaching Council Citation for an outstanding contribution to teaching and learning. His research interests include telecommunications, network design, 5G/6G, and technical risk management. He is a fellow of the Institute of Engineers Australia. He is also the Managing Editor of two international journals, such as the *Australian Journal of Telecommunications and the Digital Economy* (AJTDE) and *International Journal of Information, Communication Technology and Applications* (IJICTA) and the General Co-Chair of ITNAC.

**SHUO LI** (Member, IEEE) received the B.S. and Ph.D. degrees from the City University of Hong Kong, in 2009 and 2014, respectively. She worked as a Lecturer with Tianjin University, China, from 2014 to 2017. She is currently a Lecturer with the School of Engineering, RMIT University, Australia. Her research interests include telecommunications, analysis and design of optical networks, 5G/6G core networks, and underwater sensor networks.