Cloud-Native Plinth: A Platform to Support Containerized 5G Core Networks

Shen Gao, Shuo Quan, and Jie Wu
Cloud Computing Research Institute
China Telecom, Beijing, P.R.China
{*gaos13, quansh, wujie*}*@chinatelecom.cn*

***Abstract*—With the proliferation of 5G Core (5GC) network workloads, the cloudification of 5GC has emerged as a predominant trend. Cloud-native technologies enhance the carrier cloud of 5GC by providing capabilities for rapid recovery, agility, elasticity, and scalable service access. Beyond the management and orchestration systems, it is imperative to deploy several other common services to fully realize the advantages of cloud-native environments. In this paper, we propose Cloud-Native Plinth (CNP), a collection of cloud-native services built on the Platform as a Service (PaaS) layer to support the deployment of containerized 5GC. Within the architecture of CNP, specific services essential for the operation of containerized Virtual Network Functions (VNFs) are delineated. These services are implemented through Cloud-Native Computing Foundation (CNCF) projects and are tailored to the unique requirements of 5GC. Furthermore, the introduction of CNP not only facilitates the deployment of containerized 5GC but also enhances its high availability, performance, and reliability.**

***Index Terms*—5G Core (5GC), cloud-native, Platform as a Service (PaaS), Virtualized Network Function (VNF)**

I. Introduction

The 5G Core (5GC) is a crucial component of the fifth-generation mobile networks (5G), responsible for establishing highly available and reliable connectivity for User Equipment (UE) and data networks. The anticipated data transmission rates for 5G are projected to range from 10 to 50 Gbps [5], representing an increase of over 100 times compared to previous generations. Concurrent with the evolution of 5G technology, there will be a significant rise in both the data traffic from UEs and the operational demands on the 5GC. Given its capacity to handle substantial data traffic and dynamically allocate network resources, the migration of network infrastructure and services to cloud environments has emerged as a predominant trend among mobile operators [13]. However, the successful cloudification of 5GC faces several critical challenges: (i) ensuring that 5GC provides highly available, reliable, and secure network services, despite the inherently less reliable and secure nature of cloud environments; (ii) the refactoring and scaling of 5GC network functions within the carrier cloud; (iii) resource management predicated on effective observability; and (iv) the orchestration of Virtual Machine (VM) migration.

With the burgeoning development of cloud-native technologies, the cloudification of the 5GC has advanced into a new phase [8]. Containers, a cornerstone technology in the cloud-native framework, are recognized for their lightweight nature and ease of migration. Consequently, the deployment strategy for 5GC Virtualized Network Functions (VNFs) is progressively transitioning from VMs to containers. Utilizing Kubernetes to support containerized 5GC has become a prevalent implementation approach. This method addresses several challenges associated with the cloudification of 5GC: (i) Kubernetes offers robust security features, such as access control, which bolster the reliability of 5GC; (ii) Kubernetes provides comprehensive vertical and horizontal scaling capabilities that facilitate the refactoring and effective scaling of 5GC network functions [11]; and (iii) Containers managed by Kubernetes are readily migratable, simplifying deployment processes. Despite these advantages, certain challenges are still unaddressed [7], [11]. (i) Kubernetes' scaling capabilities are inadequate for stateful services. Scaling these services directly, without deploying proper load balancing strategies, may disrupt network states, including session continuity; (ii) Kubernetes lacks inherent observability functionalities. While other cloud-native systems, such as Prometheus, offer observability services, a holistic monitoring solution specifically tailored for the unique demands of 5GC has yet to be developed; (iii) Kubernetes supports basic storage solutions like Persistent Volume Claims (PVCs), but it falls short in providing advanced data backup methods essential for recovery post-system breakdowns; and (iv) Many 5GC network functions require interactions across multiple network interfaces with various components of 5GC. However, Kubernetes traditionally supports only a single network plane, which may hinder complex network configurations.

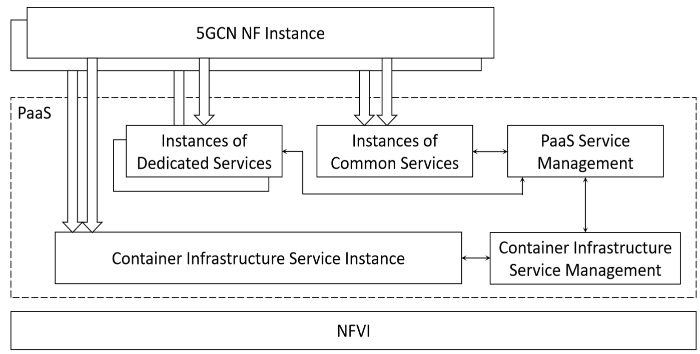


Fig. 1. Common services on PaaS for 5GC VNF.

This paper proposes a novel architecture for common services in the Platform as a Service (PaaS) layer of cloud-native 5GC. The European Telecommunications Standards Institute (ETSI) has issued a report on enhancing Network Function Virtualization (NFV) architecture for cloud-native applications and PaaS. This report outlines several use cases for common services in PaaS that support the operation of VNFs. The proposed PaaS architecture, which provides common services to 5GC VNFs, is illustrated in Fig. 1. These common services offer functionalities that can be utilized by a variety of network functions. This paper specifically addresses common services within the PaaS of containerized 5GC, termed as the Cloud-Native Plinth (CNP). With the support of CNP, 5GC can be implemented in a cloud-native environment, ensuring high availability, performance, and reliability. We have integrated the characteristics of cloud-native VNFs as defined in ETSI NFV-EVE011, along with the specific features of 5GC, to develop a viable architecture for 5GC CNP. Moreover, it addresses non-functional characteristics crucial for maintaining the reliability of 5GC, such as resiliency, scalability, state handling, and load balancing. Additionally, the architecture facilitates the support of multiple network interfaces per container, inter-cluster node networking, and connectivity with external network functions. Our implementation demonstrates that CNP can effectively support 5GC operations in a cloud-native setting and integrate the benefits of cloud-native technologies into 5GC infrastructure. The primary contributions of this paper are summarized as follows:

- The architecture of CNP is proposed, encompasses various components: scheduling and orchestration, image repository, network connectivity, scaling management, observability, support for multiple network planes, storage management, network acceleration, and reliability assurance.
- The services inside CNP are clarified and implemented, which includes essential common services that encourage the encapsulation of 5GC network functions into containers for operation within a cloud-native environment.
- The functions, performance and reliability of 5GC based on CNP are evaluated.

The remainder of this paper is organized as follows. Section II reviews existing literature and developments related to cloud-native 5GC, providing a foundation for the proposed enhancements. Section III presents details the design and implementation of the CNP architecture, describing its components and their interactions. Section IV focuses on the testing and evaluation of the CNP, specifically examining the availability, performance, and reliability of 5GC when deployed using this architecture. Section V concludes the paper by demonstrating the efficacy of the proposed CNP in facilitating the deployment of containerized 5GC, highlighting the main achievements and potential areas for future research.

II. Related Works

The telecommunication industry is making a major shift towards cloud-native NFV [9]. Traditional deployment methods of virtual network functions are based on virtual machines that have been conceived prior to the cloud-native era. Thus, they inherit many traits, which impede pivoting to cloud-native [3]. As one of the most important network functions, efficient deployment of 5GC on cloud-native platforms has caused widespread discussion. Since Kubernetes is the corner stone of cloud-native technology and 5GC has some customized requirements, almost all works are carried out by enhancing the kubernetes.

Arouk et al. [1] indicate that Kubernetes is not sufficient to fully support the deployment of cloud-native 5GC because there are some complex network services in 5GC environment. Considering the scenario of network function scaling, Kubernetes native scaling functions cannot handle stateful network services and may cause the interruption of user sessions. Arouk et al. [2] also propose a realizable agile service platform kube5G to achieve the coexistence of physical and virtual functions, (near) realtime resource provisioning, service continuity, and strict latency and data rates. Essentially, it customizes an operator based on k8s to achieve automated lifecycle management operations. Vazquez et al. [14] focus on automating the network service lifecycle, supporting real time restrictions and the coexistence of physical and virtual functions, and presented a 5G cloud-native platform using open source components. In addition to automated lifecycle management, Zhao et al. [15] present a framework using Private Wireless Experience Kit (PWEK) of Smart Edge, a highly integrated open-source Edge services platform, for optimizing infrastructure performance and accelerate. Regarding research work about architecture design of cloud-native 5GC, Imadali et al. [6] propose the 5G as a service (5GaaS) software platform which can support multi-radio technologies through VNFs. They also mentioned cloud-native principles which include a middleware platform and cloud-native tools of CNCF.

From previous work, the following points can be concluded:

- PaaS, which includes several cloud-native tools, needs to be a middleware for containerized 5GC,
- The cloud-native tools can be selected from CNCF projects,
- Merely Kubernetes cannot fully support the deployment of cloud 5GC.

We recognize that the deployment of 5GC on cloud-native platforms constitutes a systematic endeavor, necessitating thorough consideration of several critical aspects, including network connectivity, scaling management, storage management, observability, acceleration, and security. Existing studies typically address only one or a limited number of these aspects. Consequently, we propose a comprehensive platform designed to support the deployment of containerized 5GC.

III. System Architecture

Fig. 2 illustrates the comprehensive architecture of cloud-native 5G Core (5GC) as proposed in this paper, which is structured into four distinct layers. The foundational layer, Infrastructure as a Service (IaaS), virtualizes computing, storage, and network resources from physical servers. Positioned above the IaaS layer is Container as a Service (CaaS), primarily

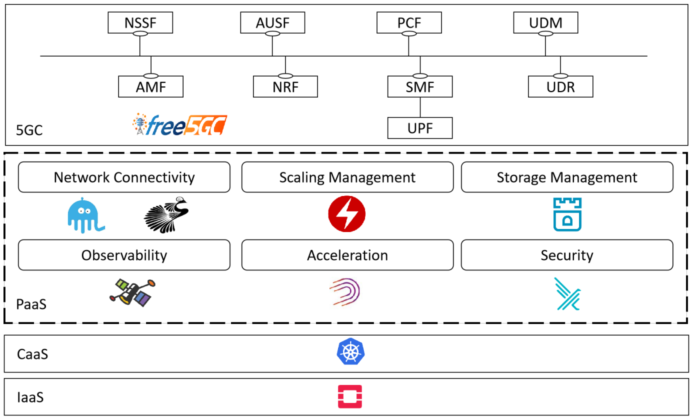


Fig. 2. Architecture of cloud-native 5GC.

TABLE I
SELECTED CNCF OPEN-SOURCE PROJECTS

Name	Description
Kube-OVN	Kube-OVN introduces OVN functions of Openstack into Kubernetes which provides more network abilities, such as static IP addresses, to cloud-native clusters.
Multus-CNI	Multus CNI enables switching on multiple network interfaces in pods of Kubernetes.
Operator Framework	Operator Framework is a toolkit to manage Kubernetes applications, especially complex stateful applications.
Rook	Rook is a cloud-native storage orchestration for Kubernetes which provides self-managing, self-scaling and self-healing storage services.
Hubble	Hubble is built on top of Cilium and eBPF to enable deep observability.
DPDK	DPDK, designed to run on x86, POWER and ARM processers, is to accelerate package processing workloads.
Falco	Falco is a security system which can detect and alert on Linux system calls.

composed of Kubernetes, which facilitates container management. The third layer, PaaS, consists of various middleware common services. These services are crucial for enhancing the functionality of the cloud-native environment and include network connectivity, scaling management, storage management, observability, acceleration, and security services. Together, these services ensure that 5GC operates with high availability, performance, and reliability. The topmost layer encompasses containerized 5GC network functions and applications, which deliver the requisite network services.

The CNP of containerized 5GC integrates CaaS with PaaS within the system architecture. The functionalities of CaaS are primarily facilitated through Kubernetes. However, the selection of tools for the PaaS components requires further consideration. Table I presents the technical selections for cloud-native projects.

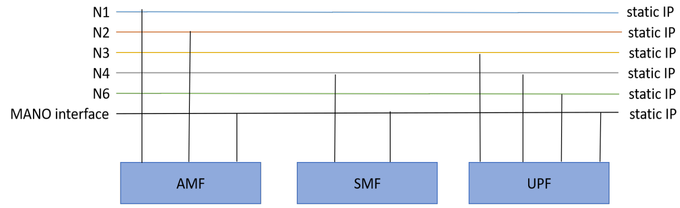


Fig. 3. Network connectivity of 5GC.

A. Network Connectivity

Fig. 3 illustrates an example of network connectivity within cloud-native 5GC. Communication among 5GC network functions relies on specific interfaces, which require static IP addresses to ensure stable network connectivity across different functions. Additionally, certain network functions necessitate mounting on multiple interfaces. Consequently, the technological requirements for network connectivity in cloud-native 5GC include the provision of static IP addresses for communication interfaces and the support of multiple network planes for containerized network functions.

Kube-OVN, in conjunction with Multus CNI, meets the technological requirements for network connectivity of cloud-native 5GC. Multus CNI enables the addition of multiple network interface controllers (NICs) from different networks to a single pod. Furthermore, Kube-OVN provides IP Address Management (IPAM) capabilities for the CNI network, while other networks can leverage the subnet and fixed IP functionalities within Kube-OVN. The implementation workflow for network connectivity in cloud-native 5GC is outlined as follows:

- Create NetworkAttachmentDefinition to provide IPAM for Multus CNI,
- Claim multiple NICs are all Kube-OVN type NICs,
- Generate Kube-OVN subnet and pod with multiple NICs,
- Fix the IP address of pod by adding NetworkAttachment.

B. Scaling Management

The efficiency of scaling is crucial for ensuring the availability of network services on demand and the reliability of fault tolerance mechanisms. Consequently, it is essential to incorporate a scaling management service within the CNP. Kubernetes offers capabilities for both vertical and horizontal pod autoscaling, which facilitate the scaling management for a subset of 5GC network functions and applications. However, Kubernetes does not adequately address the scaling management needs of stateful services.

The Operator Framework provides a Software Development Kit (SDK) that enables the customization and extension of Kubernetes functionalities. This framework facilitates the creation of Custom Resource Definitions (CRDs) within the Kubernetes environment. Upon the creation of a CRD, a controller is subsequently developed to execute the specific functions attributed to the CRD. The scaling management controller comprises a scaling algorithm and a stateful handler. The scaling algorithm is based on the work of Ren et al.

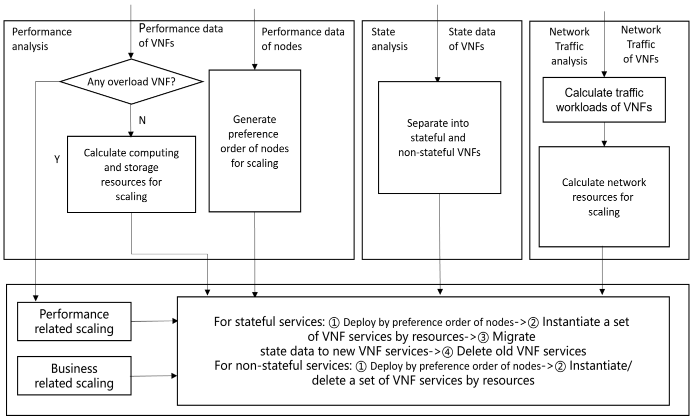


Fig. 4. Scaling controller.

[10], who proposed an adaptive VNF scaling algorithm for 5G networks. Fig. 4 depicts the logic of the scaling controller as implemented by the Operator Framework.

C. Storage Management

Storage management encompasses not only the judicious allocation of storage resources but also the creation of a highly reliable storage architecture through essential redundancy measures. Within the realm of cloud-native projects, Rook stands out as a particularly effective solution for fulfilling these storage management goals. Operating atop Kubernetes, Rook delivers a storage service that is self-managing, self-scaling, and self-healing.

In this paper, Rook was integrated into the CNP. The deployment process began with the establishment of a CRD via the Operator Framework. This research then selected block storage for the configuration of Rook's storage classes. To ensure enhanced resilience, the replication factor was set to three, with replicas distributed across disparate clusters. Additionally, to improve fault tolerance and recovery capabilities, snapshots were configured within each storage class.

D. Observability

Observability is a critical component in the operation of cloud-native 5GC networks. Effective resource management and fault response are heavily reliant on observability outcomes. However, the highly dynamic nature of Kubernetes diminishes the effectiveness of traditional visibility tools. Cilium, leveraging eBPF—a technology originating from the Linux kernel—extracts fine-grained observability data to provide an advanced observability service in a cloud-native environment. Building upon Cilium, Hubble enables detailed visibility into network behavior, network policy compliance, HTTP rates and latency, as well as DNS rates and errors.

Despite these capabilities, Hubble's observability functions do not fully extend to the specialized requirements of 5GC. In this study, we developed a data analysis function to enhance the observability capabilities of Cilium. This data analysis module achieves business-level observability for 5GC by filtering and analyzing data collected by the cilium-agent. The augmented observability capabilities introduced by this module include monitoring the session workload of the Session Management Function (SMF) and User Plane Function (UPF), and real-time parallel access numbers for the Access and Mobility Management Function (AMF), among others.

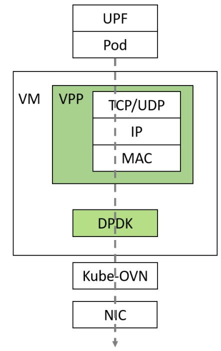


Fig. 5. Acceleration model.

E. Acceleration

Acceleration is a critical module for enhancing the performance of cloud-native 5GC networks. The I/O channel, responsible for forwarding the traffic of network functions, comprises physical NICs, virtual switches, and virtual NICs. The transfer of network traffic involves multiple steps, including traversing physical NIC drivers, host kernel network protocol stacks, the virtual switching layer of the kernel, and virtual NIC drivers. This process encompasses numerous operations such as system interrupts, memory copying, and virtualization encapsulation. To enhance the performance of 5GC, it is necessary to bypass some of these steps.

Fig. 5 illustrates the acceleration method employed in cloud-native 5GC. The forwarding of user data is expedited by the Data Plane Development Kit (DPDK), which facilitates network acceleration by bypassing the operating system's kernel. However, DPDK lacks the functionality to process TCP/IP protocols. Consequently, the Vector Packet Processing (VPP) platform is utilized to manage the transfer of user protocol data, leveraging its capability to process TCP/IP protocols.

F. Security

Security represents a crucial dimension in ensuring the reliability of cloud-native 5GC networks. While Cilium is capable of enforcing network security policies from Layers 3 to 7, the security of containers remains a critical requirement. Falco, an effective cloud-native security system, enhances security capabilities by offering container runtime threat detection, protection for cloud environments, and system event monitoring. The Falco image can be directly retrieved and its security services can be encapsulated within containers using Docker.

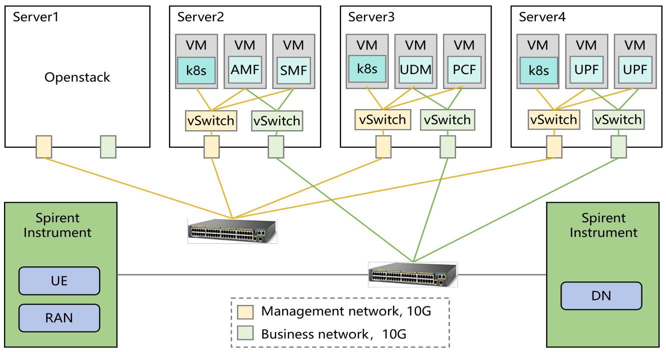


Fig. 6. Deployment schematic diagram.

IV. Experiment Evaluation

A. *Experiment Environment*

We tested procedures of 5GC based on for CNP environment. The testing equipment is Spirent 5G Core Automation Platform. Our experiment equipment is composed of four DELL PowerEdge R350 Rack servers with:

- 4 x Intel Xeon E-2324G 3.1GHz CPUs,
- 256GB RAM,
- 1.2TB Hard Disk,
- 2 x 10G network interface card,

We set up OpenStack on the four-server infrastructure and applied KVM hypervisor for virtualization. 15 VMs are instantiated for the propose of creating Kubernetes cluster. The deployment strategy is delineated in Fig. 6, with one server allocated for the deployment of OpenStack management components, and three servers designated for the deployment of Kubernetes Master, among which two are designated for the deployment of 5GC control plane network elements such as AMF, SMF, and PCF, while one is dedicated to deploying the data plane UPF.

B. *Analysis of Capabilities and Performance Results*

In order to validate the effectiveness of the solution, we conducted tests from both capabilities and performance perspectives. The test cases are shown in TABLE III. It presents that 5GC supported by CNP can accomplish the basic services. In the scaling testing experiment, CNP can keep the PDU session state by 97.2%, which is better than the Kubernetes horizontal scaling which cannot keep PDU session. Thus, CNP can fulfill the availability of cloud-native 5GC.

CNP can improve the availability and reliability of cloud-native 5GC by enhancing the capabilities of VNF and storage scaling, observability, storage recovery and security. Although DPDK is introduced for acceleration, the cost of other modules may influence the performance of cloud-native 5GC. So evaluation of performance is still required.

We conducted a comparative analysis of network performance under two different configurations: (i) CNP, and (ii) Kubernetes integrated with OpenvSwitch (OVS). The results, as depicted in Fig. 7, pertain to the data plane performance metrics of 5GC. Fig. 7(a) illustrates that both upload and

TABLE II
AVAILABILITY TEST

Test Case	Test Result
UE Registration	The procedure of UE context transferring by AMF, authentication by AUSF, subscription data storage by UDM, policy selection by PCF and network slice allocation by NSSF can be accomplished successfully.
PDU session establishment	The procedure of SM context creation by SMF, SM policy association establishment by PCF and N4 session modification by SMF can be accomplished successfully.
User profile management	The procedure of subscriber data update notification, session management subscriber data update notification and purge of subscriber data can be accomplished successfully.

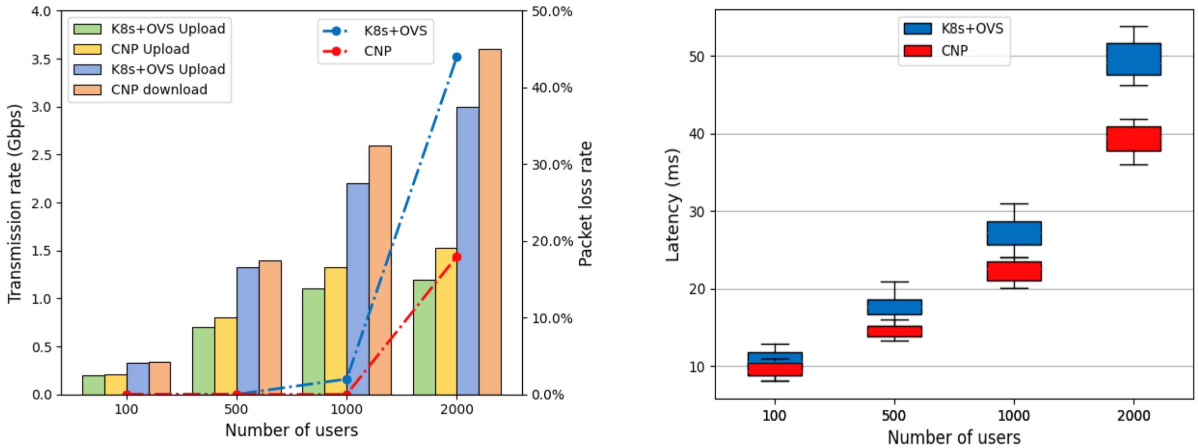
download speeds in the data plane increase with the rise in user numbers, underscoring that data traffic is proportional to user count. Notably, the throughput observed in the CNP configuration exceeds that of Kubernetes combined with OVS, indicating that CNP can provide higher average speeds to users. Similarly, as shown in Fig. 7(b), latency escalates with an increase in user numbers, with CNP exhibiting superior performance compared to the Kubernetes and OVS combination. This suggests that CNP can enhance the performance of cloud-native 5GC to a significant extent.

This is because DPDK can reduce the context switching between user space and kernel space, and minimize memory copy. So it allows for faster packet processing and reduces the overhead associated with traditional kernel-based networking. Additionally, DPDK is highly optimized for modern multicore processors, enabling parallel processing of packets and improving overall throughput. These factors combined contribute to DPDK's ability to achieve superior performance in data plane applications.

Additionally, reliability tests were performed. According to Fig. 7(a), CNP demonstrates a lower packet loss rate, which supports the assertion that CNP can more effectively manage increasing data traffic in 5GC networks. A simulated power failure was induced by shutting down one of the control plane nodes, resulting in approximately 33% network disconnections and data loss in the absence of CNP. Conversely, with CNP, there was only a 4% network disconnection and a marginal 0.3% data loss. Through features such as load balancing and data backup, CNP substantially enhances the reliability of 5GC.

C. *Issue Discussion*

Although CNP can bring many benefits, in practice, we have found that it has certain limitations, mainly in terms of the requirements for 5GC architecture and cloud infrastructure. In order to fully leverage the advantages of cloud-native, there needs to be a deep decoupling within the network elements, early V-based NFV designs may need to be upgraded to better benefit from new trends, including declarative APIs, stateless services [4], and network element microservices [12]; Cloud



(a) Packet transmission and loss

(b) Latency

Fig. 7. Performance of 5GC data plane.

infrastructure usually shields the underlying physical network from the cloud-native platform, which does not meet the networking requirements of the 5GC network. For example, network elements need to implement sub-interface multi-network planes in virtual machines, network elements need to configure ECMP/BFD on the physical network, and these requirements bring even greater challenges when deployed in public clouds.

V. Conclusion

This paper introduces the CNP, a supportive layer designed for the deployment of containerized 5GC, which leverages the foundational principles of cloud-native architecture to enhance the scalability, flexibility, and reliability of 5GC systems.CNP is not merely an augmentation but a transformative approach to deploying 5GC in a cloud-native environment. It introduces a novel architecture that incorporates a suite of common services designed to optimize the operational efficiency of 5GC. While Kubernetes serves as the orchestration backbone, providing basic management of containerized services, CNP extends its capabilities by integrating specialized cloud-native functions tailored for 5GC applications. This integration includes advanced networking functionalities, security enhancements, and performance optimization tools that are not inherently part of Kubernetes. In conclusion, CNP represents a significant advancement in the deployment of cloud-native 5GC, providing a higher level of performance, reliability, and scalability than traditional Kubernetes-based solutions.

References

[1] Osama Arouk and Navid Nikaein. 5g cloud-native: Network management & automation. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–2. IEEE, 2020.

[2] Osama Arouk and Navid Nikaein. Kube5g: A cloud-native 5g service platform. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[3] David Breitgand, Vadim Eisenberg, Nir Naaman, Nir Rozenbaum, and Avi Weit. Toward true cloud native nfv mano. In *2021 12th International Conference on Network of the Future (NoF)*, pages 1–5. IEEE, 2021.

[4] Keliang Du, Luhan Wang, Xiangming Wen, Yu Liu, Haiwen Niu, and Shaoxin Huang. Ml-sld: A message-level stateless design for cloud-native 5g core network. *Digital Communications and Networks*, 9(3):743–756, 2023.

[5] Akhil Gupta and Rakesh Kumar Jha. A survey of 5g network: Architecture and emerging technologies. *IEEE access*, 3:1206–1232, 2015.

[6] Sofiane Imadali and Ayoub Bousselmi. Cloud native 5g virtual network functions: Design principles and use cases. In *2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2)*, pages 91–96. IEEE, 2018.

[7] Raul Martin, Ivan Vidal, and Francisco Valera. A software-defined connectivity service for multi-cluster cloud native applications. *Computer Networks*, 248:110479, 2024.

[8] Alireza Mohammadi and Navid Nikaein. Athena: An intelligent multi-x cloud native network operator. *IEEE Journal on Selected Areas in Communications*, 2023.

[9] Sridhar KN Rao, Federica Paganelli, and Al Morton. Benchmarking kubernetes container-networking for telco usecases. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2021.

[10] Yi Ren, Tuan Phung-Duc, Yi-Kuan Liu, Jyh-Cheng Chen, and Yi-Hao Lin. Asa: Adaptive vnf scaling algorithm for 5g mobile networks. In *2018 IEEE 7th international conference on cloud networking (CloudNet)*, pages 1–4. IEEE, 2018.

[11] Syed Danial Ali Shah, Mark A Gregory, and Shuo Li. Toward network slicing enabled edge computing: A cloud-native approach for slice mobility. *IEEE Internet of Things Journal*, 2023.

[12] Pengpeng Song, Hui Peng, and Xiaowen Zhang. A micro-service approach to cloud native ran for 5g and beyond. *IEEE Access*, 11:130257–130271, 2023.

[13] Tarik Taleb. Toward carrier cloud: Potential, challenges, and solutions. *IEEE Wireless Communications*, 21(3):80–91, 2014.

[14] Álvaro Vázquez-Rodríguez, Carlos Giraldo-Rodríguez, and David Chaves-Diéguez. A cloud-native platform for 5g experimentation. In *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 60–64. IEEE, 2022.

[15] Qianli Zhao, Sunku Ranganath, Shaohe Feng, Gang Li, Simon J Li, Zhongbao Shi, Baoguo Ding, and Jiming Gao. Customizable cloud-native infrastructure for private 5g. In *2023 26th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 50–57. IEEE, 2023.