CSI Transactions on ICT manuscript No.
(will be inserted by the editor)

ONVM-5G: A Framework for Realization of 5G Core in a Box using DPDK

Madhura A · Mohit Kumar Singh · Bheemarjuna Reddy Tamma

the date of receipt and acceptance should be inserted later

Abstract Software Defined Networking (SDN) and Network Functions Virtualization (NFV) are the two fundamental paradigms that introduce concealed flexibility in communication systems. 5G has adopted SDN/NFV to make Network Functions (NFs) as virtualized software modules which run on top of commodity hardware. 5G aims to expand its reach to various business verticals and serve diverse users using network slicing which creates multiple logical networks on top of a shared infrastructure. This paper proposes a novel OpenNetVM based 5G framework (ONVM-5G) which uses Data Plane Development Kit (DPDK) for the communication between the core NFs and thus, is a candidate solution for implementing 5G Core (5GC) in a box. The proposed framework based deployment of network slices enhances the user experience by reducing the latency involved in performing various User Equipment (UE) activities and increasing data plane throughput. Moreover, ONVM-5G also helps in achieving the 5G goals by allowing the network operators to efficiently serve the users in small dense cells and make service level isolation among the diverse users through network slicing by providing a network in a box like deployment of 5GC.

This work has been partially supported by the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) and the project "Converged Cloud Communication Technologies", Meity, Govt. of India.

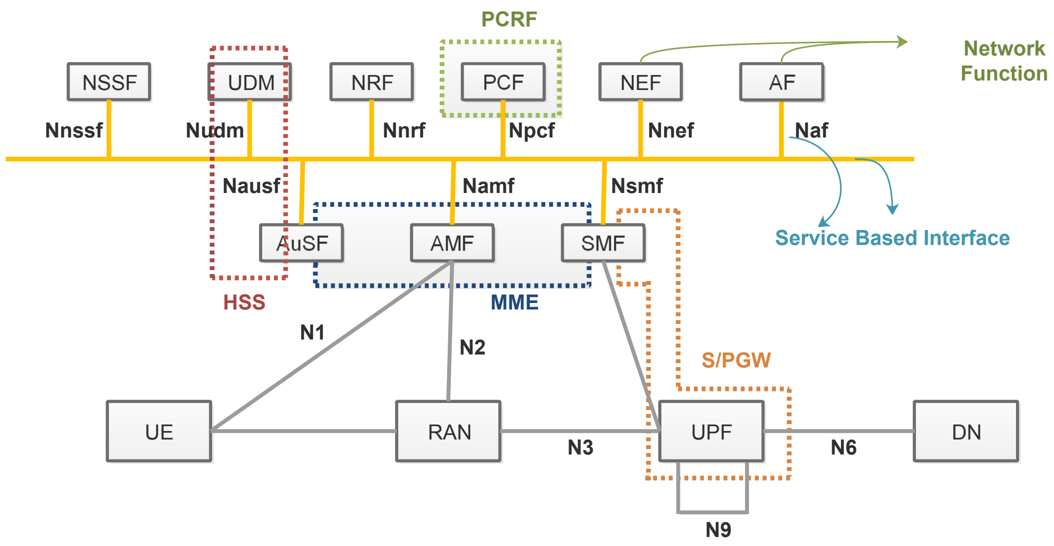
Madhura A, Mohit Kumar Singh, and Bheemarjuna Reddy Tamma
Dept. of Computer Science and Engineering
IIT Hyderabad, India, 502285
E-mail: (cs17mtech11013,cs17mtech11015,tbr)@iith.ac.in

Keywords 5G · Data Plane Development Kit (DPDK) · Software Defined Networking (SDN) · Network Functions Virtualization (NFV)

1 Introduction

Network Functions Virtualization (NFV) enables Network Functions (NFs) running on propriety and dedicated hardware to be replaced by virtualized software modules and makes them deployable over commodity hardware. NFV being a key enabler for the realization of 5G architecture involves virtualization of various network function services constituting Control Plane (CP) and User Plane (UP) of 5G Core (5GC). Network slicing is a crucial part of the 5G architecture that takes the benefits of NFV for adopting new services with a wide range of requirements over the same physical infrastructure. 5G architecture is designed to provide support for three different generic services [1], namely, enhanced Mobile BroadBand (eMBB), massive Machine-Type Communications (mMTC), and Ultra-Reliable Low-Latency Communications (URLLC), each offering distinct Quality of Services (QoS) to their users. These services can be accommodated in different network slices of 5G system deployed where a specific slice can handle the user requests of certain kind. Thus, using network slicing, 5G achieves service level isolation among the users with the aim of efficient network utilization. Also, 5G aims to provide better coverage for various types of cells that differ in area and number of users, while serving the users with high throughput and low latency.

5G has evolved from the existing 4G Long Term Evolution (LTE) by decomposing CP of LTE (Long-Term Evolution) EPC-CUPS (Evolved Packet Core-



AuSF: Authentication Server Function, **AMF:** Access and Mobility Management Function, **UDM:** Unified Data Management, **NEF:** Network Exposure Function, **NRF:** Network Repository Function, **NSSF:** Network Slice Selection Function, **PCF:** Policy Control Function, **SMF:** Session Management Function, **DN:** Data Network, **UPF:** User Plane Function, **AF:** Application Function, **UE:** User Equipment, **RAN:** Radio Access Network, **MME:** Mobility Management Entity, **SGW:** Serving Gateway, **PGW:** Packet Data Network Gateway, **PCRF:** Policy and Charging Rules Function, **HSS:** Home Subscriber Server

Fig. 1: **Service Based Architecture of 5G, evolving from the existing LTE EPC Architecture.**

Control and User Plane Separation) architecture consisting of MME (Mobility Management Entity), HSS (Home Subscriber Server), SGW-C (Serving Gateway-Control
plane), PGW-C (Packet Gateway-Control pane), and PCRF (Policy and Charging Rules Function) NFs into AMF (Access and Mobility Management Function), SMF (Session Management Function), AuSF (Authentication Server Function), UDM (Unified Data Management), and PCF (Policy Control Function) NFs forming the 5GC CP. The UP of 5G architecture consists of a single NF, namely, UPF (User Plane Function), which is derived from SGW-U (Serving Gateway-User plane) and PGW-U (Packet Gateway-User plane) NFs of LTE EPC-CUPS as shown in Fig. 1. Thus, 5G helps in achieving scalability, performance, and flexibility by converting the existing monolithic network architecture into micro-service based view of network composition to support flexible management capabilities, lightweight NFs with improved deployability and interoperability.

OpenNetVM [2] is a scalable and efficient packet processing framework that helps to steer packets between the NFs utilizing the principle of **zero packet copying**. OpenNetVM leverages Data Plane Development Kit (DPDK) [3] for high performance I/O and uses its concept of establishing zero packet copy based communication using the shared memory among the NFs [4] to achieve low latency and high throughput, thereby making it a candidate solution for implementing the 5GC. Our proposed DPDK based 5G implementation (ONVM-5G) uses OpenNetVM for implementing the 5G architecture [5], using shared memory based communication among the 5G core NFs. We have shown that the DPDK based 5GC helps in achieving the various use cases of 5G, providing different deployment scenarios for the network slices with different QoS. Proposed implementation of 5GC helps in achieving lower latency and higher throughput application-specific network slices. Also, we have shown ONVM-5G based deployment will be beneficial in setting up network-in-a-box like architecture and providing the 5G services with much ease through network slicing.

Rest of the paper is organised as follows. Section 2 describes background knowledge on 5G architecture and explains OpenNetVM framework that uses DPDK as the underlying principle for communication between NFs. Section 3 depicts the scenarios considered for deploying the different network slices in 5G using ONVM-5G. Section 4 illustrates the experimental setup for deploying the network slices using 5G Reference Point Architecture, Service Based Architecture, and ONVM-5G and compares the latency incurred in performing UE related tasks in different deployment scenarios. Section 5 provides various use cases of ONVM-5G where network operators can provide the users with guaranteed QoS in dense cells and can achieve service level isolation by building network slices using ONVM-5G constituting core network of 5G architecture.

2 Background: 5GC Architecture and OpenNetVM Framework

2.1 5G Architecture

According to 3GPP release 15, the architecture of 5G consists of CP and UP where CP handles the processing of UE control messages with the help of NFs, namely, NSSF, NEF, NRF, PCF, UDM, AF, AUSF, and SMF and UP entity, namely, UPF handles the forwarding of data messages between the UE and DN. Based on the way of communication between the NFs, two kinds of 5G architectures are possible, to be specific Reference Point architecture and Service Based Architecture (SBA). In Reference Point architecture, between each pair of communicating NFs, a dedicated interface is created, and NFs communicate with each other through these interfaces. However, in SBA, each NF registers its service with Network Repository Function (NRF). NRF exposes the services registered by the NFs to all the authorized NFs forming Service Based Interfaces (SBIs). So in SBA, NFs communicate with each other through SBIs for exposing their services as well as utilizing the services offered by other NFs [6].

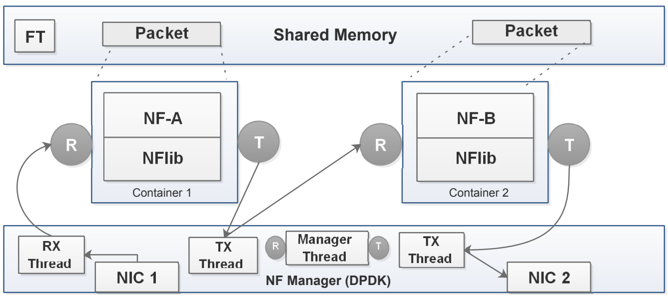


Fig. 2: **OpenNetVM Framework [2].**

2.2 OpenNetVM Framework

OpenNetVM is a highly efficient packet processing framework that significantly simplifies the development of NFs. As shown in Fig. 2, NF manager is the key component in the OpenNetVM framework performing the main task of receiving packets from Network Interface Card (NIC), configuring shared memory for the NFs, and communicating with NFs through an API called NFlib. Each NF has a transmit (T) ring and a receive (R) ring for holding the packets. NF manager maintains a flow table, performs NIC interaction for handling the incoming and outgoing packets from NIC and handles communication between the NFs by passing packet reference between the transmit and receive rings of the NFs. In OpenNetVM, all the NFs run either as a process or in a docker container on a single physical machine.

OpenNetVM uses DPDK library to directly capture the incoming packets from NIC bypassing the kernel and stores the packets in the allocated memory in a huge page. OpenNetVM uses the concept of receive ring and transmit ring attached with each NF which are monitored by Tx threads of the NF Manager for passing the packet reference between the NFs. Rx thread processes a batch of packets from the receive queue of the NIC and examines them to route to the destination NF.

3 Deployment of 5G Core (5GC) using ONVM-5G

This section describes two deployment scenarios, namely, centralized CP-UP architecture and distributed CP-UP architecture as shown in Fig. 3 and Fig. 4, respectively. Fig. 3 shows one such deployment suited for URLLC slice having all the NFs running on the same machine while Fig. 4 shows the other scenario well suited for bandwidth-thirsty services like eMBB having the data plane running exclusively in a dedicated machine.

In both deployments, 5GC NFs AMF, SMF, AuSF, and UDM forming the CP and UPF forming the UP are implemented along with the UE related activities like registration, session creation, modify session, sending uplink/downlink data to DN and de-registration procedure. All the implemented NFs are derived from LTE EPC architecture [7]. These deployment scenarios are realized with OpenNetVM framework for 5GC a.k.a. ONVM-5G.

In OpenNetVM based implementation of 5GC, predetermined service IDs are given to each NF, so that depending upon the type of packet, an NF can decide the next action and next destination. Furthermore, an NF knows service IDs of other NFs with which it communicates as well as whether destination NF is located in the same machine or not. If the destination NF is situated in a different machine, source NF knows the link to which the packet needs to be forwarded to. For example, since AMF communicates with AuSF and SMF, it knows their service IDs. If an initial registration packet arrives at AMF, the destination would be the service ID of AuSF, and if a session creation request needs to be forwarded, the destination would be the service ID of SMF. In Fig. 3, the data packet received by NF Manager of machine-2 is sent to the receive ring of UPF by the Rx thread directly. In Fig. 4, the data packet is directly sent from machine-1 to the receive ring of the UPF by the NF Manager running in machine-3.

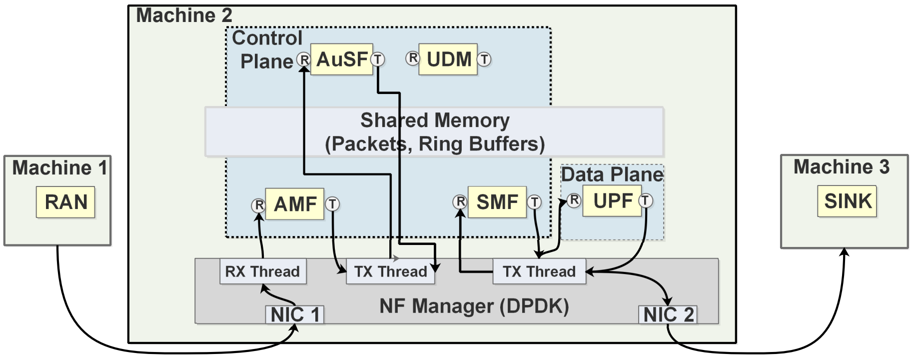


Fig. 3: **Centralized CP-UP Implementation of 5GC.**

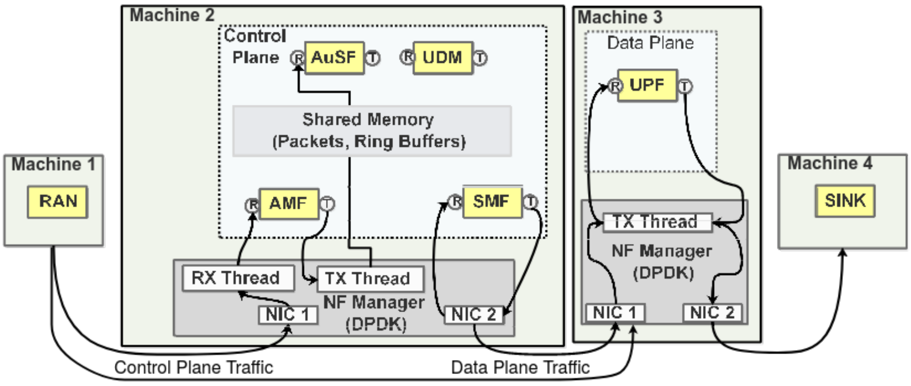


Fig. 4: **Distributed CP-UP Implementation of 5GC.**

The following UE activities are performed over ONVM-5G involving the CP and UP NFs, each having a dedicated transmit ring and a receive ring.

1. Initial Registration: This procedure emulates activities indulged in the attachment of a UE to the network including the initial exchange of UE information, authentication, security setup, and location update as shown in Fig. 5.
2. Session Creation: This procedure emulates UE performing session creation and modifying session procedures as shown in Fig. 6 for UE.
3. De-registration: De-registration is responsible for deletion of session created and complete detachment of UE from the network.

In OpenNetVM based implementation of 5GC, instead of point-to-point interfaces between the NFs, an NF relies on the ONVM-manager for the transmission of packets to the destination. OpenNetVM leverages DPDK's support for creating a shared memory region using huge pages, allowing zero-copy access to the packet data. Without DPDK, kernel-based sockets are used for communication, where a packet first traverses to the kernel and then kernel sends it to the user space. In ONVM-5G based deployment packets directly arrive at the user space bypassing the kernel, which can further reduce the latency and increase the throughput.

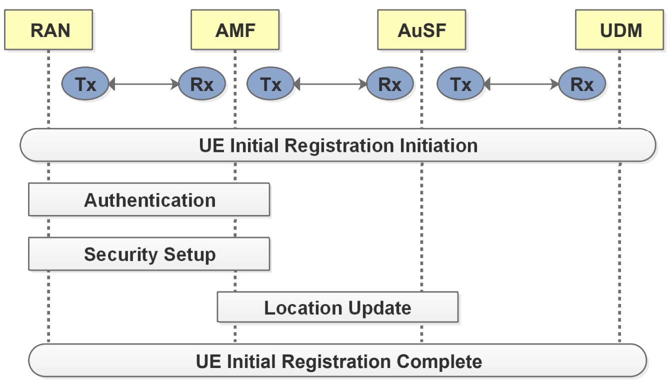


Fig. 5: **UE Activity for Initial Registration in ONVM-5G.**

3.1 Centralized and Distributed CP-UP Deployment of 5GC

Centralized CP-UP deployment of 5GC consists of CP NFs and UP NF which run in the docker containers in privileged mode on the same physical machine using OpenNetVM. The motivation for running the 5GC NFs in the same physical machine is to get the advantage of zero-copy packet transfers between NFs, which can reduce the communication latency drastically. This kind of centralized implementation is more suitable for devices performing mission-critical tasks in which there is a requirement of minimal latency, forming the URLLC services.

Distributed CP-UP deployment of 5GC is similar to that of centralized CP-UP deployment with the only difference in CP and UP separation as shown in Fig. 4. The main inspiration for separating UPF from the physical machine on which CP is running is to show a proper isolation of CP and UP so that 5GC is used as "network in a box" with respect to CP. This implementation provides flexibility to the operators in the placement of UPF depending upon the service requirements and thus providing eMBB devices with high data plane throughput.

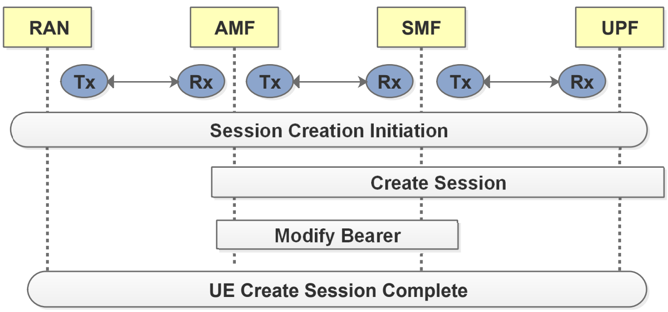


Fig. 6: **UE Activity for Session Creation in ONVM-5G**

4 Experimental Setup and Results

In this section, we present experimental setup used to evaluate the performance of proposed ONVM-5G framework and compare it with other implementations that do not rely on DPDK. We study the performance of ONVM-5G deployment scenarios which use DPDK with the Reference Point and SBA implementations of 5G without DPDK and compare the performance in terms of latency involved in performing the UE centric activities and data plane throughput.

Reference Point based deployment uses point-to-point interfaces between the NFs as described by 3GPP. And for realizing SBIs between the NFs in SBA-based deployment, nghttp2 API library [8] is used which uses HTTP2 for establishing REST-based communication between the NFs. Consul [9] is used for implementing NRF NF in SBA-based deployment, where each NF registers and communicates for fetching the information of the other NFs present in the network. Consul is a tool for service discovery and configuration. Consul is distributed, highly available, and extremely scalable. Consul also provides several key features like service discovery, health checking, key/value storage, multi data centre configuration, and secure communication.

Each machine used in the testbed setup is equipped with Intel Xeon Gold 6126 CPU (2.60GHz, 48 cores) and two 1G NICs with Ubuntu 16.04. In ONVM-5G based implementation of 5GC, each NF runs inside a Docker container on a dedicated physical CPU core in the privileged mode to ensure that all the containers have access to host machine resources like network bandwidth and shared memory. The centralized and distributed ONVM-5G implementations are evaluated using a testbed setup with three and four machines, respectively having shared memory based communication between the NFs as shown in Fig. 3 and Fig. 4, respectively. The Reference Point based 5G architecture and SBA are evaluated in a testbed setup having seven machines where each of the NFs is running on a separate machine having socket-based communication between the NFs without DPDK.

CP traffic contains initial registration traffic, create session traffic, and de-registration traffic. CP traffic is generated by the RAN simulator which simulates various UE activities. For a UE, first a registration request is created at the RAN simulator which flows as described in Fig. 5. Once initial registration is completed, create session is performed as specified in Fig. 6 followed by data transfer. After the data transfer, deregistration procedure is carried out. To mimic realistic scenarios, control traffic is generated continuously which simulates concurrent users in the system, each performing its activities. To vary the load on the core NFs, we have varied the number of concurrent users performing their activities from 1 to 100. To create the UP traffic, ONVM-5G based implementations use pktgen, while Reference Point and SBA implementations use iperf3. Pktgen is a DPDK based traffic generator that transmits and receives traffic to measure the achievable throughput. For evaluating the performance of deployment scenarios, machine-1 is used for running RAN simulator which generates CP traffic and DP traffic using Pktgen for UEs. In centralized CP-UP

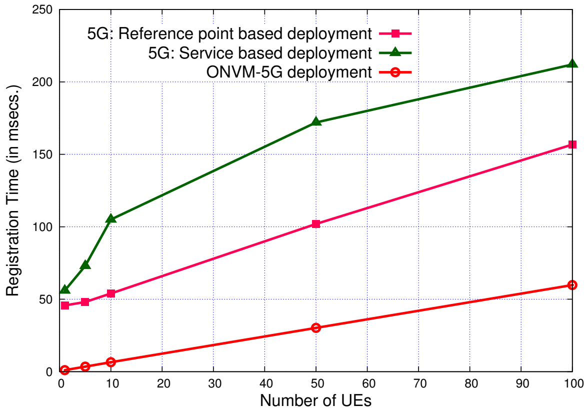


Fig. 7: **UE Registration Time vs. Number of UEs.**

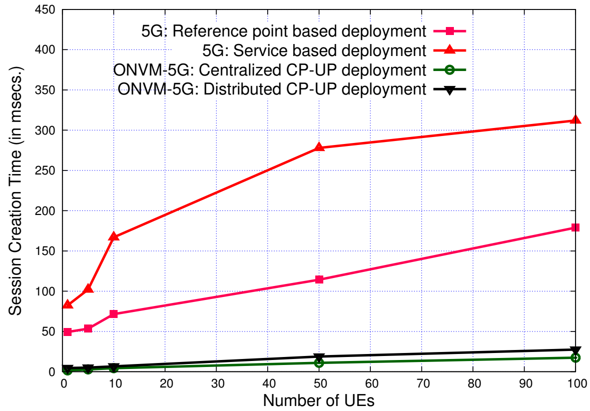


Fig. 8: **Session Creation Time vs Number of UEs.**

scenario, traffic passes through machine-2, running the core network, while machine-3 is used for running the SINK node with pktgen for calculating DP throughput for UE. In case of distributed CP-UP scenario, machine-2 runs CP NFs, machine-3 runs UPF while machine-4 runs SINK node. To vary data traffic load, we varied the number of concurrent users in the network as well as the rate at which traffic generators generate traffic from CP to UP. All the deployment scenarios are evaluated based on two metrics: (i) Data Plane Throughput and (ii) Latency incurred in performing UE activities.

We have evaluated the performance of ONVM-5G based deployment scenarios with the following goals:

1. Demonstrate that ONVM-5G based deployments have minimal latency for performing initial registration, create session, and de-registration procedures by varying number of users.
2. Demonstrate that ONVM-5G based deployment gives high data plane throughput per user.
3. Study data plane throughput improvement in ONVM-5G based implementation with varying transmission rates.

4.1 Result Analysis

In this section, we present the performance evaluation of ONVM-5G, Reference Point, and SBA based 5GC while performing various UE activities.

4.1.1 Improvement in latency overhead

Fig. 7 shows the average latency incurred in performing initial registration procedure for a UE in ONVM-5G, Reference Point, and SBA based 5GC by varying number of users from 1 to 100. The average latency incurred in performing session creation

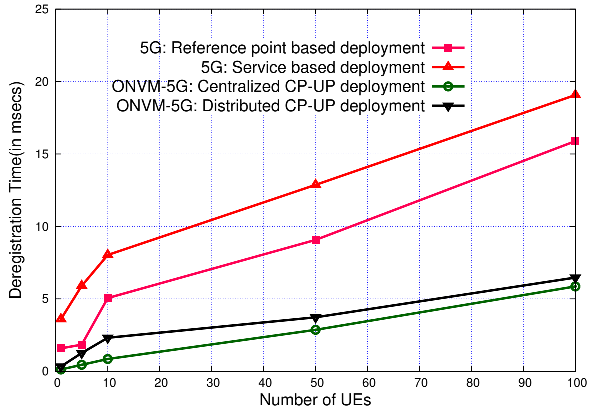


Fig. 9: **De-registration Time vs Number of UEs.**

and de-registration procedure are shown in Fig. 8 and Fig. 9, respectively. Results demonstrate that ONVM-5G based 5GC incurs low latency in performing UE activities. Average latency incurred in performing initial registration for 100 users in case of ONVM-5G deployment is approximately 58% less than that of Reference Point implementation and approximately 68% less than that of SBA. In case of create session procedure, average latency incurred in ONVM-5G deployment for 100 users is approximately 72% less than that of Reference Point, and approximately 83% less than SBA. For performing UE de-registration procedure for 100 users, the average latency of ONVM-5G is approximately 56% less than that of Reference Point, and approximately 63% less than that of SBA. The latency incurred in completing the initial registration in centralized CP-UP and distributed CP-UP scenarios remain the same as there is no change in the positioning of the NFs engaged in performing the UE registration. Distributed CP-UP based scenario has slightly higher create session latency as UPF is running in a different machine, adding a round trip time between machine-2 and machine-3 to the create session latency.

In scenarios having ONVM-5G based 5GC, NFs communicate with each other using shared memory created in the huge page. When a packet arrives at the NIC, the packet is stored in the shared memory. The sender NF sends only the reference of the packet, unlike the complete physical packet in kernel socket-based communication in case of Reference Point and SBA based 5GC. Communication overhead is more in Reference Point based implementation, while in ONVM-5G based deployment scenarios, NFs identify each other through service IDs assigned by the NF manager and use the shared memory for communication. When a packet arrives at the NIC, it is given directly to the user space bypassing the kernel in ONVM-5G based implementations, thereby reducing the latency further. Thus, ONVM-5G based network slice will be able to serve the delay critical services provided by 5G like V2X,Voice services requiring lower latency overheads.

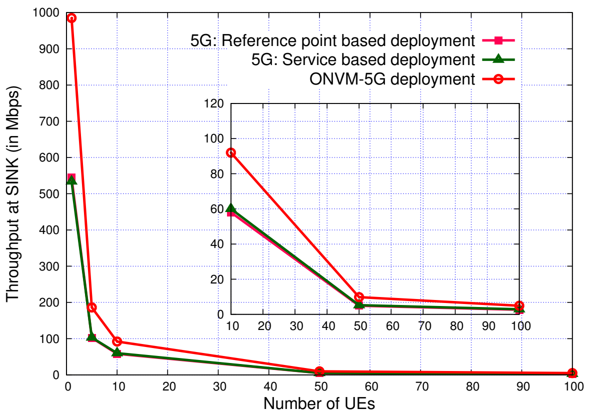


Fig. 10: **Data Plane Throughput vs Number of UEs.**

4.1.2 Data plane throughput

For this experiment, UP traffic is emulated using Pktgen in ONVM based 5GC and iperf3 in Reference Point and SBA based 5GC. The transmission rate of both Pktgen and iperf3 is set to 1 Gbps, which is the maximum transmission capacity of NICs of the machines used. In centralized CP-UP based implementation of 5GC, pktgen, running in machine-1 transfers data packets to the core network in machine-2 where the NF manager receives those packets from the NIC and stores them in the shared memory. Later, the NF manager passes the packet references to UPF, and then UPF forwards the original packets to the SINK node running with Pktgen in machine-3 for calculating the data plane throughput as shown in Fig. 3.

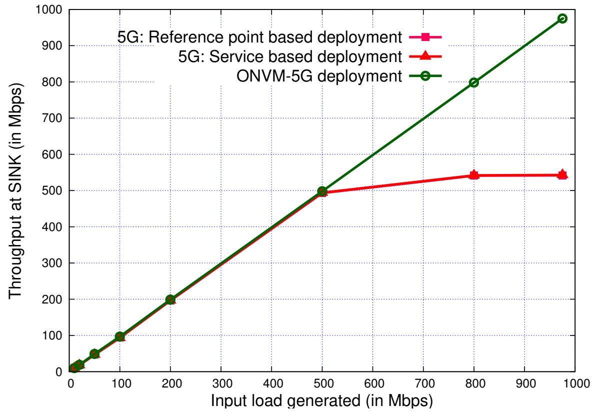


Fig. 11: **Aggregate Data Plane Throughput vs. Transmission Rate of Traffic Generator.**

In distributed CP-UP based implementation of 5GC, the only difference lies in sending the data packet directly to the machine-3 which runs UPF as shown in Fig. 4. The DP throughput is measured in both cases by varying number of concurrent UEs from 1 to 100 as shown in Fig. 10. We find that ONVM-5G based deployment scenarios are capable of processing the packets at a very high rate and thus achieving high data plane throughput. Fig. 10 shows the data plane throughput per UE when multiple UEs are connected to the same 5G network. When the number of UEs connected to the 5G network is increased, the throughput achieved by a UE decreases as the network capacity is shared among more number of UEs.

When the traffic generator is sending packets at a rate of 1 Gbps, the UP NF, namely, UPF in 5G Reference Point and SBA-based deployments reach their maximum processing capacity, i.e., they could only serve at a maximum rate of 540 Mbps. Given the similar CPU provisioning, ONVM-5G based deployment scenario can handle the traffic at nearly 1 Gbps with some real-time variations as shown in Fig. 11. In Reference Point and SBA based 5GC, UPF listens to the socket for a packet over N3 interface and then forwards it to SINK through N6 interface, thus incurring high latency in processing the data plane traffic. However, in ONVM-5G based implementation, data plane traffic forwarding is handled by NF manager by just specifying the service ID of the destination NF in packet descriptor and sending only the reference of the packet to UPF, which then sends actual data packet through NIC to

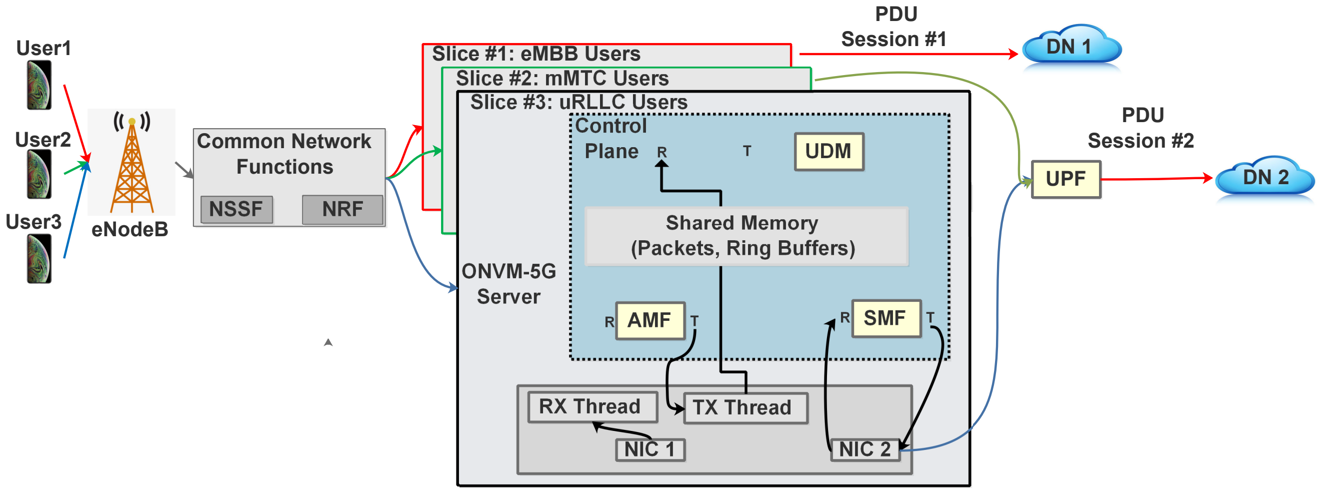


Fig. 12: **Network Slicing using ONVM-5G Architecture.**

the destined Pktgen. This processing of the data plane traffic in ONVM-5G bypasses the kernel, hence results in low latency while forwarding the data plane packets and therefore results in high data plane throughput. In our experiments, we never be able to find any bottleneck for ONVM-5G based 5GC, as we were limited by the transmission rate of 1 Gbps NIC. Hence, we believe that given a NIC with high capacity, we can deploy a ONVM-5G based network slice to meet the generic services provided by 5G like eMBB and URLLC, requiring high data plane throughput.

5 Application and Use cases for ONVM-5G

ONVM-5G, being a candidate solution for implementing 5GC architecture providing ultra-low latency and high data plane throughput is helpful in realizing various use cases of 5G. Here, we present some of the use cases.

– **Network Slicing:** 5G is designed to support the needs of various types of users where each type requiring different QoS. ONVM-5G is capable of providing the support for creating different network slices [10] as shown in Fig. 12, each supporting a specific type of users with certain QoS. ONVM-5G serves best for the devices requiring very low network latency and high data plane throughput in performing their activities. This includes devices performing mission critical tasks forming URLLC services, Internet of Things (IoT) devices forming mMTC services, and AR/VR (Augmented/Virtual Reality) devices and mobile edge applications forming eMBB services.
  URLLC devices require the type of networks offering very low latency in performing their tasks, while AR/VR and mobile edge devices run bandwidth-hungry applications. Thus, such devices can be connected to the specific slice, with ONVM-5G forming the core network of the system and performing all the UE tasks with low latency and high throughput, meanwhile achieving service level isolation between the users.

Dense Deployment of Small Cells: The 5G architecture is designed to support small cells confined to small areas and serving small number of users. Here, we can have a Base Station to which all the users are connected and forwarding the CP traffic to ONVM-5G. Each such cell can have a single dedicated server running the ONVM-5G with all the NFs forming a network-in-a-box like system as shown in Fig. 13. Thus, users, despite being present in the dense cell scenario, will achieve low latency in performing their activities and enjoy high data plane throughput from the network.

6 Conclusions and Future Work

In this work, we have successfully studied various deployment of 5GC. It is observed that ONVM-5G based deployment outperforms Reference Point and SBA-based deployment in terms of network latency and data plane throughput. Also, we have discussed how ONVM-5G based 5GC helps in achieving some of the use cases of 5G by running the entire core network in a single server, giving rise to network-in-a-box like deployment cases. Thus, ONVM-5G is a prominent solution for serving users requiring generic services like URLLC, mMTC, and eMBB and also providing better network coverage in dense deployment of small cells.

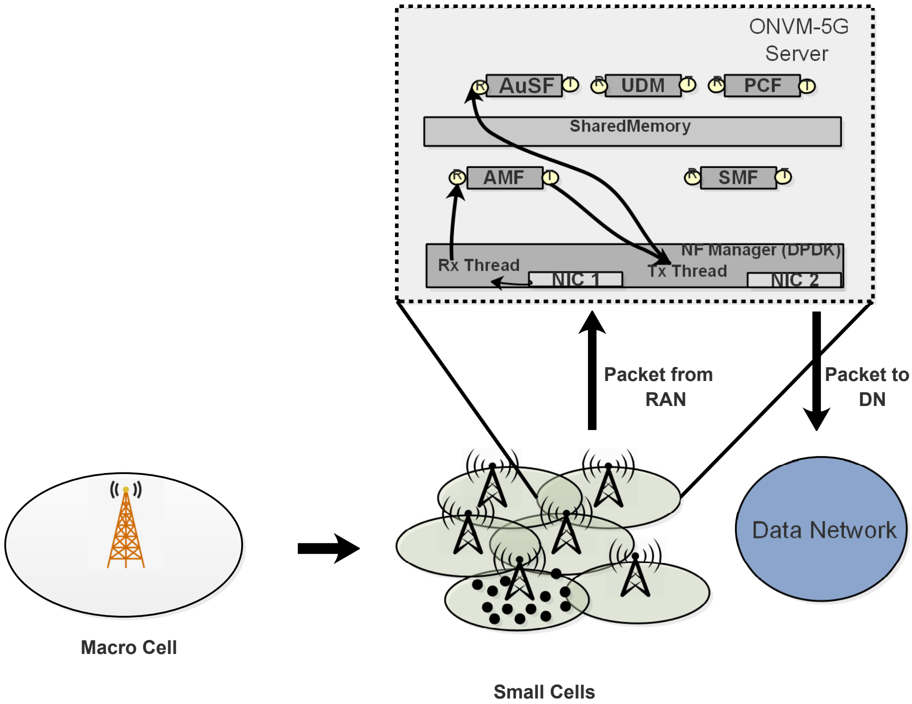


Fig. 13: **ONVM-5G Architecture in Dense Deployment of Small Cells.**

References

1. P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5g wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view", *IEEE Access*, vol. 6, pp. 55765–55779, September 2018.
2. Wei Zhang, Guyue Liu, Wenhui Zhang, Neel Shah, Phillip Lopreiato, Gregoire Todeschi, K.K. Ramakrishnan, and Timothy Wood, "OpenNetVM: A Platform for High Performance Network Service Chains", in *Proc. of ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization.* August 2016, pp. 26–31, ACM.
3. "Data plane development kit", https://www.dpdk.org/.
4. Guyue Liu, K. K. Ramakrishnan, Mike Schlansker, Jean Tourrilhes, and Timothy Wood, "Design challenges for high performance, scalable nfv interconnects", in *Proc. of Workshop on Kernel-Bypass Networks.* August 2017, IEEE.
5. 3GPP, "System architecture for the 5g system", Tech. Rep. TS 23.501, section 6.2, 2018.
6. Cheng Zhang, Xiangming Wen, Luhan Wang, Zhaoming Lu, and Lu Ma, "Performance evaluation of candidate protocol stack for service-based interfaces in 5g core network", in *Proc. of ICC Workshops.* May 2018, IEEE.
7. A. Jain, S. N. S, S. K. Lohani, and M. Vutukuru, "A comparison of sdn and nfv for re-designing the lte packet core", in *Proc. of NFV-SDN.* November 2016, IEEE.
8. "Nghttp2: HTTP/2 C Library", https://nghttp2.org/.
9. "Consul", https://www.consul.io.
10. Y. Choi and N. Park, "Slice architecture for 5g core network", in *Proc. of 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2017, pp. 571–575.