Rapid Prototyping of Network Slice Analytics Architectural and Federation Framework in 5G Core

Shwetha Vittal
Department of CSE
IIT Hyderabad, India
cs19resch01001@iith.ac.in

Ganesh Vernekar
Department of CSE
IIT Hyderabad, India
cs15btech11018@iith.ac.in

Mohit Kumar Singh
Department of CSE
IIT Hyderabad, India
cs17mtech11015@iith.ac.in

Antony Franklin A
Department of CSE
IIT Hyderabad, India
antony.franklin@iith.ac.in

***Abstract*—As the demand for network slicing is growing in 5G, the operators are looking into various technologies and enablers to cater the needs of users in three fundamental and promising deployment scenarios defined by 3GPP, namely enhanced Mobile Broadband (eMBB), ultra Reliable Low Latency Communication (uRLLC), and massive Machine Type Communication (mMTC). Technology enablers like Network Functions Virtualization (NFV) and Software Defined Networking (SDN) in cloud infrastructure pave the way to deploy network slicing in various forms in 5G networks, by reducing Capital Expenditure (CAPEX) and Operating Expenditure (OPEX). However, this presents various challenges to the mobile operators in life cycle management, performance management, and fault management of network slices. In this work, we focus on monitoring of different network slices with respect to performance management in 5G Core Network. We propose a performance management monitoring and federation framework for network slices by building our own prototype model of 5G Core Service Based Architecture (SBA) and orchestration with Open Source MANO along with open source tools such as Prometheus TSDB, Consul, and cAdvisor. Our experiments on federated and non federated setup help operators to design an end to end framework with Network Data Analytic Functions (NWDAF)s and Performance Management Data Analytic Functions (PMDAF)s in the 5G end to end slice life cycle and performance management system at various levels of building slice monitoring and analytical frameworks with closed loop automation.**

I. Introduction

5G networks have the capability to support a wide variety of services and requirements using network slicing [1]. Operators could customize their network for different applications and customers using slices. Slices furnish the flexibility in providing services as they differ in functionality (e.g., priority, and policy control), in performance requirements (e.g. latency, data rates, and availability), or in serving only specific kind of users (e.g., public safety users, industrial users, corporate users). A network slice can provide the functionality of a complete network, including radio access network and core network functions. One network can support one or several network slices. Additionally, the provision of the services by slices is boosted by Network Function Virtualization (NFV) [2], for the 5G architecture, involving the virtualization of various network function services constituting control plane and data plane of 5G Core (5GC) and thereby forming the key enabler here. 3GPP [3] defines Network Data Analytic Function (NWDAF) for data collection and data analytics in centralized manner, making it a critical entity in the 5GC. 3GPP in [4] specifies how an NWDAF may be used for analytics as potential solutions to address various key issues on one or more network slices.

The remainder of this article is arranged as follows. In section III and IV we detail the needed theoretical background and significance of the data analytics, performance measurement, and Key Performance Indicators (KPI) for an end to end network slice in alignment with ETSI Management and Orchestration (MANO) framework by focusing on 5G core slice subnet and related functions. Then we describe our work of implementing the complete framework and federation setup, in section V and section VI. In section VII we demonstrate how various system metrics like CPU and memory along with 5G core KPIs [5] of accessibility, utilization, integrity KPIs of a network slice instance in 5G core could be monitored and presented to different data analytics functions using our frameworks. In conclusion portion, we summarize our overall work and highlight the future work we consider in this domain of 5G.

II. Motivation and Related Work

In [6], authors detail on an integrated analytics architecture with respect to Radio Access Network (RAN) and core network architectures individually, by listing various use cases, the enhancements of the current architecture of the 3GPP 5G System (5GS) and key design directions where it could be useful. Authors in [7] discuss the applicability of exploiting data analytics for supporting the operation of the Radio Resource Management (RRM) algorithms embedded within the Next Generation-RAN (NG-RAN) nodes by conceiving RAN Data Analytic Function (RANDAF) as an execution platform for Data Analytics (DA) applications. In [8] authors propose a context based framework towards RRM using three mechanisms: Compass, Context Extraction and Profiling Engine (CEPE) and Context Information Processing (CIP) for optimizing the RAT selection, traffic steering and switching operations in 5G network environments along with evaluating one of these to target the minimization of the information collected and used by the data analytics engine using NWDAF. All these works definitely help realize the significance of big data analytics across various domains of 5GS. However, it is worth mentioning that little attention has been placed to date to detail the actual implementation on the proposed framework or architecture involving actual components of it to solve the different issues faced by operator in building network slice supportive self organizing network. Also the open source monitoring tools used in our frameworks are widely being used in various NFV MANOs like Open Source MANO (OSM) to report metrics mainly CPU and memory of individual VNFs and Network Services (NS)s, as part of NFV MANO NS performance management. However monitoring and reporting the 5G core subnet slice and end to end slice specific KPIs through NFV MANO adds to the complexity of overall slice performance management, as MANO is transparent to 5G system and core network function specific metrics. Hence in this work, we focus on building a full prototype integrated framework of network slice monitoring and analytics capturing both 5G network slice and network function specific metrics to help achieve self optimization in an end to end 5G network slicing.

III. Network Slice Management Data Analytics Service

As defined by 3GPP, a Management Data Analytics Service (MDAS) [9] provides data analytics for the network. MDAS can be deployed at different levels, for example, at domain level (e.g., Radio Access Network (RAN), Core Network (CN), Network Slice Subnet Instance (NSSI)) or in a centralized manner (e.g., in a PLMN level). A domain-level MDAS provides domain specific analytics, e.g., resource usage prediction in a CN or failure prediction in a NSSI, etc. A centralized MDAS can provide end-to-end or cross-domain analytics service, e.g., resource usage or failure prediction in an NSI, optimal CN node placement for ensuring lowest latency.

5G management system can therefore benefit from management data analytics services for improving performance of the network and efficiency of network slices to accommodate and support the diversity of services and requirements. The management data analytics utilize the network management data collected from the network including e.g. service, slicing and/or network functions related data and make the corresponding analytics based on the collected information. For example, the information provided by Performance Management (PM) Data Analytics Function (PMDAF) can be used to optimize the network performance.

IV. Key Performance Indicators of Network Slice

5G system needs to support stringent KPIs for latency, reliability, throughput, etc. Enhancements in the core network contribute to meeting these KPIs using network slicing, in-network caching, scalable assignment of network resources and hosting services closer to the end points. Therefore, measuring the KPIs is very crucial to cater the various requirements of 5G Core (5GC) enhancements, flexibility, and optimization.

3GPP defines KPIs through the measurement of key parameters of input and output of internal network system. KPIs are considered to be primary metrics to evaluate process performance as indicators of quantitative management. As service serveability performance is a significant factor here, it falls into one of the three related categories: service accessibility, retain ability and integrity performance. Pertaining to these KPIs, we focus on 5G core performance monitoring and quality bench-marking. Table I lists a few set of KPIs in 5G system for various KPI categories specified by 3GPP.

TABLE I: KPI categories

KPI Category	Example KPI
Accessibility	Registered Subscribers of Network and Network Slice Instance through AMF
Integrity	End-to-end Latency of 5G Network
Retainability	Quality of Service (QoS) flow Retainability
Utilization	Mean number of PDU sessions of network and network slice instance

V. Detailed Description on Implementation

In this section, we detail on the implementation of our work on the end-to-end network slice performance monitoring and analytics framework.

A. The Developed Framework

The developed framework consists of 5G Service Based Architecture (SBA) [10] as the System Under Test, orchestrated using the NFV MANO functions provided by OSM [11] Rel.5. On its northbound interface there is a Network Slice Management Function (NSMF) taking the role of OSS/BSS, communicating with Network Slice Subnet Management Function (NSSMF) and Performance Measurement Data Analytics Function (PMDAF). Here PMDAF has Prometheus Time Series Database (TSDB) [12] plugged in it. Prometheus TSDB collects the various slice specific KPIs and Virtual Network Function (VNF) specific metrics from cAdvisor[13] and node_exporter[14] running in 5G SBA system respectively.

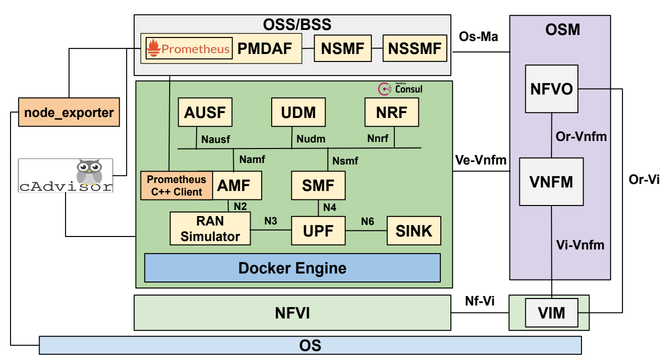

Fig. 1: Developed framework for monitoring network slice.

The developed prototype model of the 5G SBA comprised of 5G Core control plane and data plane functions. Control plane functions listed as Access and Mobility Management Function (AMF), Authentication Selection Function (AUSF), Unified Data Management (UDM) Entity, and Session Management Function (SMF) implementing REST API message interaction model using HTTP2 library API provided by nghttp2. The Network Resource Function (NRF) runs as a Consul [15] server providing service registration and discovery services to other network functions in the framework which run Consul client inside them. The data plane functions listed as User Plane Function (UPF), Data Network Sink entity on the reference point architecture for N3 and N6 interfaces. In our testbed, we have developed a light weight RAN with embedded User Equipment (UE)'s Non Access Stratum (NAS) function terming it as RAN + UE Emulator. Fig. 1 shows our complete framework.

All these network functions including RAN + UE Emulator are developed as virtualized docker containers [16] each intended to provide micro service functionalities such as UE registration, deregistration, and end-to-end uplink and downlink data exchange over different network slices.

NSMF and NSSMF make use of the NFV MANO functions provided by OSM on its North Bound Interface. OSM provides the NFV Orchestration (NFVO) and Virtual Network Function Management (VNFM) functionalities that supports communicating with different Virtual Infrastructure Management(s). We have picked a light weight VIM-Emulator [17] which emulates the Openstack functions for VIM named as vim-emu. Vim-emu allows the execution of real network functions packaged as Docker containers in emulated network topologies running locally on the developer's machine.

TABLE II: Metrics exposed by AMF

Metric Name	Description
ue_registration_requests_total	Total number of registrations requested by all UE
ue_registrations_success_total	Total number of successful registrations
ue_session_creation_total	Total number of sessions created
ue_deregistration_requests_total	Total number of deregistrations requested by all UE
ue_deregistrations_success_total	Total number of successful deregistrations
active_ue_total	Number of active UE

B. Monitoring Technologies

1) Prometheus: Prometheus [12] is an open source, pull-based, service monitoring system. It collects metrics from configured targets at given intervals, evaluates rule expressions, displays the results, and can trigger alerts if some condition is observed to be true.

2) cAdvisor (container Advisor): cAdvisor [13] is an OSS which is a running daemon that collects, aggregates, processes, and exports information about running containers. We use this to get CPU utilization, memory usage, and network usage of the VNFs which runs as docker containers. cAdvisor exposes these metrics at a HTTP endpoint which can be scraped by Prometheus.

3) node_exporter: node_exporter [14] is a Prometheus exporter for hardware and OS metrics exposed by *NIX kernels. We use this to get CPU utilization, memory usage, and other OS metrics of the actual hardware as opposed to individual containers from cAdvisor.

4) Prometheus C++ Client: Prometheus C++ Client [18] is a C++ open source library, which we use to instrument our 5G SBA code. It exposes custom defined metrics at a HTTP endpoint to be scraped by Prometheus. We use this in AMF to expose the metrics listed in table II.

5) Grafana: Grafana [19] is a leading open source dashboard and graph editor which helps us in visualizing the metrics from Prometheus and creating dashboard with required graphs.

C. Monitoring using Prometheus

We use Prometheus as our main monitoring system. It supports multi-dimensional data model with data center name, slice ID, VNF ID, and VNF name as labels of the metrics pertaining to the VNFs. Using Prometheus's powerful query language, these labels help us aggregate the monitoring data for each data center, slice, and VNF type.

1) Registering Metric Endpoints: Upon bringing up the 5G network slice using OSM, cAdvisor and node_exporter is run on the system as daemon. The IP and port of the metric endpoint of both the daemon is registered in NRF, which is Consul in our case. cAdvisor exposes the data center, slice ID and VNF ID combined into a single label in all metrics.

When a VNF is started, it registers its own metrics endpoint into the consul. In our implementation, AMF registers its exporter IP and port in the Consul. Similarly, any VNF here can make use of Prometheus C++ Client library to expose metrics and register it in Consul. We directly expose data center, slice ID and VNF ID as separate labels.

2) Service Discovery and Monitoring: Prometheus allows us to automatically discover the services from Consul which it monitors. We use this feature to discover the targets that we registered in the Consul. The discovery is refreshed at regular intervals, hence it allows us to dynamically monitor targets that are joining and leaving the network without any change of configuration or downtime in monitoring.

3) Relabelling: This concept in Prometheus allows us to modify the labels of the metrics and drop unwanted metrics when Prometheus is scraping the metric endpoints. Using this we pick only relevant metrics from cAdvisor (CPU, memory, and network usage), while dropping all the unwanted metrics from it. It also helps us drop unwanted labels from all metrics. We also use this concept to break the combined label containing data center, slice ID and VNF ID from cAdvisor metrics into their own labels.

TABLE III: Federated System Configuration

Architecture	Intel(R) Xeon(R) CPU E5-2640 v4
Total Number of CPUs	40
Thread(s) per core	2
Clock Speed	1199 MHz, with max capacity 3400MHz

VI. Network Slice Monitoring in Federation

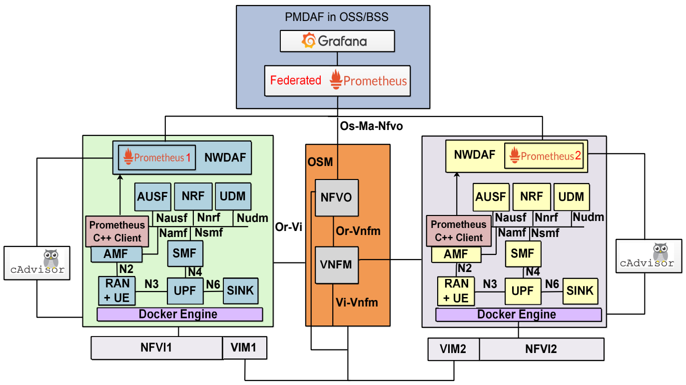

Fig. 2: Federated testbed setup for monitoring network slices.

As shown in Fig. 2, we setup the federated test framework to monitor multiple end to end network slices across different sites with our single MANO. Here we deploy two different end to end slices on site1 and site2 managed by single MANO.

TABLE IV: Site1 and Site2 System Configuration

Architecture	Intel(R) Xeon(R) Gold 6126
Total Number of CPUs	48
Thread(s) per core	2
Clock Speed	1499 MHz, with max capacity 3400MHz

We used the system configuration shown in Table III for running OSM (MANO), PMDAF, NSMF and the one shown in Table IV for $site1$ and $site2$.

We use Prometheus in federated mode. While there is a non federated Prometheus which runs at Network Data Analytic Function (NWDAF) in each of the sites along with slice's respective network functions, federated Prometheus runs on a different system along with OSM. Each of the non federated Prometheus at their respective NWDAF collect the KPIs of respective slice and VNF specific metrics.

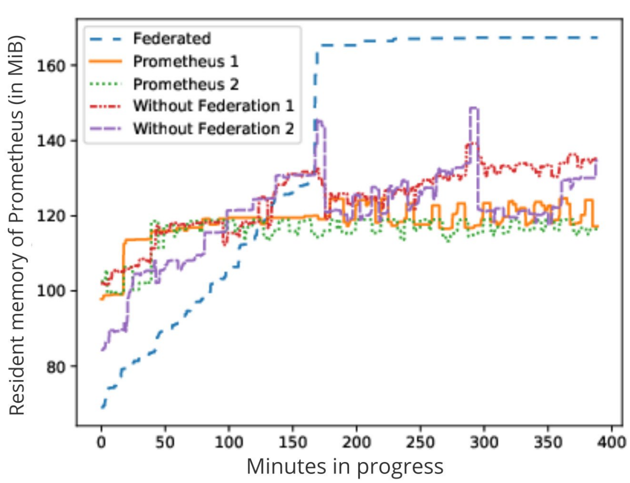

Fig. 3: Resident memory trend with and without federation.

Final aggregation of these metrics is done at PMDAF with federated Prometheus. We use Grafana as our main visualization tool which queries the federated Prometheus instead of the Prometheus at the individual sites. This mode of functioning of Prometheus helps to scrape selected metrics (with or without aggregating) from another Prometheus server. Fig. 3 shows the resident memory trend of federation from our tests. Following are some of the key reasons along with observations as to why we rely on using federated framework:

1) Global View: Prometheus supports hierarchical federation. Hence by federating the Prometheus of all 5G data centers (it can span upto multiple levels of federation depending on the scale), we can get a global view of the 5G network, i.e. view of all slices at one place.
2) Reduction in monitoring resources: Prometheus has both in-memory database for new data coupled with storing old data on disk. We disable the retention on disk completely on the 5G data centers as we visualize the federated Prometheus, and hence requiring no storage space. We also reduce the retention time of in-memory database as the new data is federated as soon as it is generated, hence capping the max memory requirement for monitoring at data centers. It is also apparent from Fig. 3 that resident memory of Prometheus 1 and 2 capped after some time, while Prometheus without federation had their resident memory increase with time. On the contrary, we store less data at the federated Prometheus compared to without federation 1 and 2 together by carefully selecting the required metrics. As federated Prometheus has longer in-memory and disk retention like a normal deployment, the trend in Fig. 3 is as expected.
3) Efficient 5G functioning: As we have federated required metrics from the data centers, we query Prometheus for the graphs and dashboards. Hence relieving data centers from the memory and CPU requirements for the queries.
4) Reliability: We can have more than one federated Prometheus and form a federated Prometheus cluster, along with redundancy. Therefore it achieves high availability, to prevent any data loss.

It can be inferred from Fig. 3 that at a much larger scale (>100x the monitoring requirement) and a much wider deployment (>2 data centers), the reduction in resource requirement for monitoring with federation would be very significant.

VII. Detailed Description on Testing & Measurement

In this section we detail on how various system/VNF specific and network slice specific metrics are captured for an active end to end network slice on our framework, during various phases of testing the slice life cycle.

For testing purpose, we deployed an example eMBB end to end slice on this framework. This slice consisted of three slice subnets namely 1. Core network slice subnet with AMF, AUSF, UDM, SMF, and UPF functions, 2. Data network slice

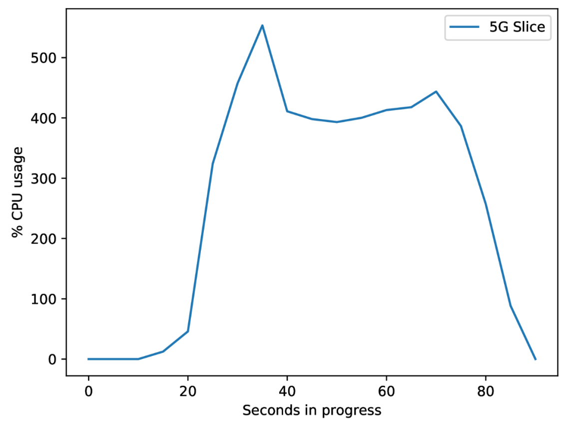

Fig. 4: CPU consumption of slice, with 1 thread at core functions.

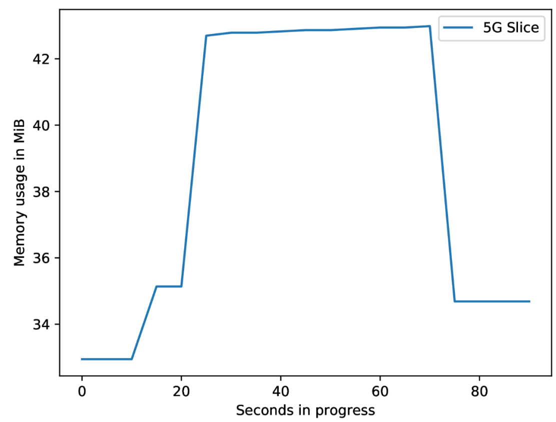

Fig. 5: Memory consumption of slice with 1 thread at core functions.

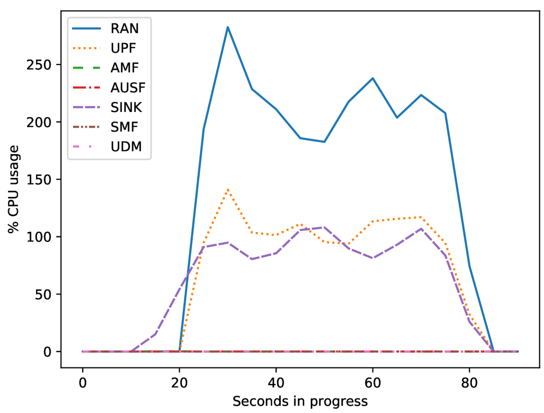

Fig. 6: CPU consumption of each VNF with 1 thread at core functions.

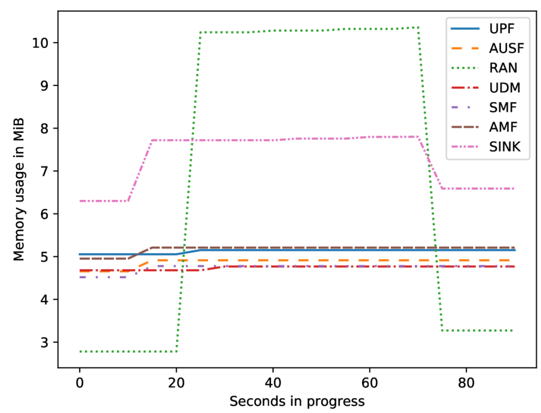

Fig. 7: Memory consumption of each VNF with 1 thread at core functions.

subnet with Data Sink entity, and 3. RAN slice subnet with RAN+UE emulator entity.

A. Slice Preparation & Commissioning Phase

NSMF on boot up, on boards the slice template consisting of data network slice subnet NSD with data sink Virtual Network Function Descriptor (VNFD), common core network slice subnet NSD with AMF, AUSF, UDM, SMF, and UPF VNFDs, RAN slice subnet with RAN+UE emulator VNFD. Once the preparation is done, this slice is instantiated to move into active state for operation phase. Upon activation, each VNF in the respective slice subnets registers itself at NRF. It is now ready to serve the traffic of 5GC control plane and data plane at their respective Service Based Interface (SBI).

B. Slice Operation Phase

1) UE Registration, Data Exchange and UE De Registration: Once the RAN+UE emulator function gets active, the UE starts registering to the 5GC with default PDU session establishment on this eMBB slice. Here, AMF is treated as the default AMF. Further, AMF contacts the NRF to find the matching SMF providing this slice service. After this, rest of the transactions on the PDU session establishment continues with SMF and UPF before the UE registration is completed successfully.

Figures Fig. 4 and Fig. 5 show the Grafana snapshots of CPU and memory metrics, consumed by the slice, when 10 UEs perform UE registration, end to end uplink and downlink data exchange and UE deregistration when served by a *single thread* at each of the core network functions respectively.

Similarly figures Fig. 8 and Fig. 9 show the Grafana snapshots of CPU and memory metrics, consumed by the slice, when 10 UEs perform UE registration, end to end uplink and downlink data exchange, followed by UE deregistration, when served by 10 *threads* at each of the core network functions respectively.

Figures Fig. 10 and Fig. 11 indicate the CPU and memory consumed at each of the network function associated with slice.

When investigating the root cause of high CPU of slice from figures Fig. 4 and Fig. 8 in the operation phase of its life cycle, we could identify the high CPU at RAN, UPF and

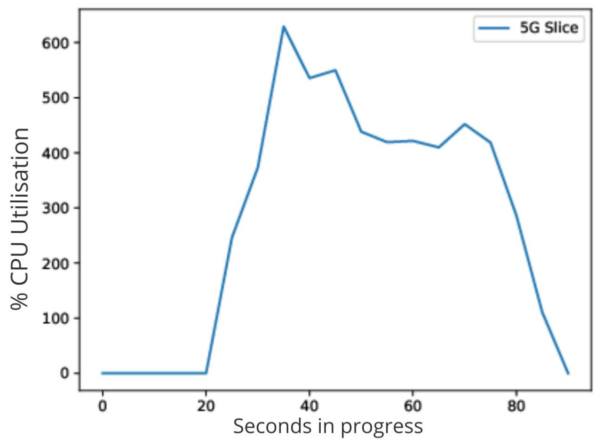

Fig. 8: CPU consumption of slice, with 10 threads at core functions.

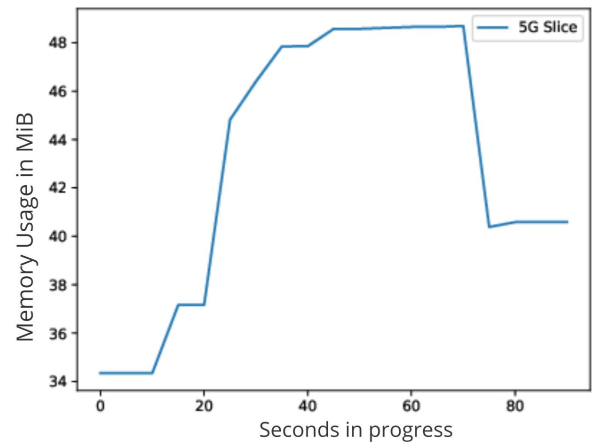

Fig. 9: Memory consumption of slice, with 10 threads at core functions.

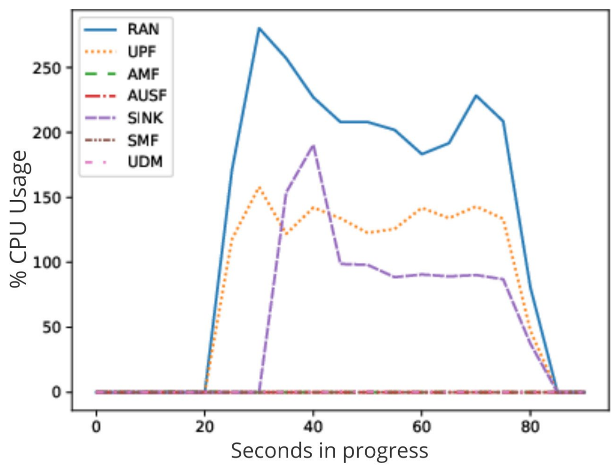

Fig. 10: CPU Consumption of each VNF with 10 thread at core functions.

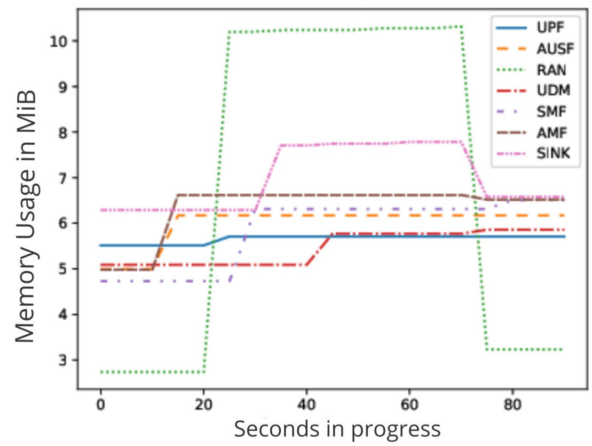

Fig. 11: Memory consumed by each VNF, with 10 threads at core functions.

SINK VNFs using figures Fig. 6 and Fig. 10 where multiple UEs try to register, exchange data in parallel with one or ten threads serving them. Similar analogy is done on memory consumption of slice at figures Fig. 5 and Fig. 9 to infer that RAN VNF contributes highest using figures Fig. 7 and Fig. 11.

In addition to system metrics of network functions, we captured 5G control plane Accessibility KPIs from AMF and data plane throughput from UPF. Accessibility KPIs involved number of registered UEs and de-registered UEs at AMF, exported by Prometheus Client which is plugged into AMF.

Figures Fig. 12 and Fig. 13 depict the data plane throughput measured at UPF, on its egress and ingress ports with respect to uplink and downlink data exchanged on slice with 10 users.

VIII. Applicability

Our implementation work here is useful in placing NWDAFs and PMDAFs at various levels of building slice monitoring and analytical frameworks with closed loop automation and hence provides the flexibility in plugging them to satisfy various use cases of slice monitoring and analytics. The information provided by NWDAFs and PM data analytic services can be used to carefully evaluate and monitor various network slice parameters like virtualised resource utilization, UE location, system load, throughput in data plane, accessibility of end to end network slice during the different phases of its life cycle. And then use them to optimize the network performance in various situations of slice service assurance like detecting bottlenecks, realising resource and elasticity profiling, or evaluate if a slice ensures the expected QoS. The instant availability of various system and 5G slice metrics helps an operator visualize the performance of slice and network functions together giving a holistic view of the end to end slice. Also, the availability of these raw data directly from different network functions in 5G core, would help in faster and effective slice management and assurance at NSMF and Network Slice Selection Function (NSSF). The NSMF could use the aggregated analytic data from PMDAF

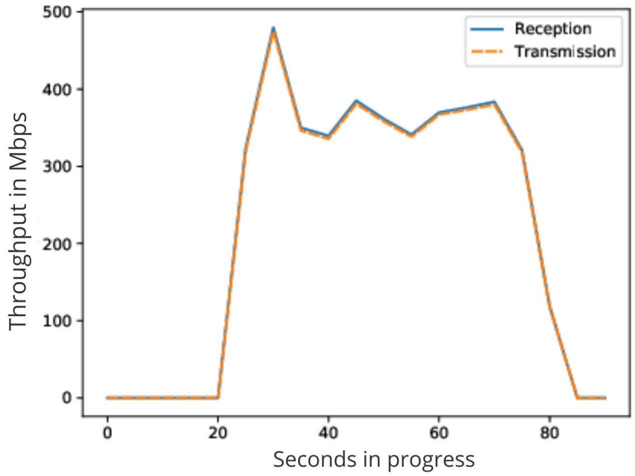

Fig. 12: Throughput of UPF, with 1 thread at core functions.

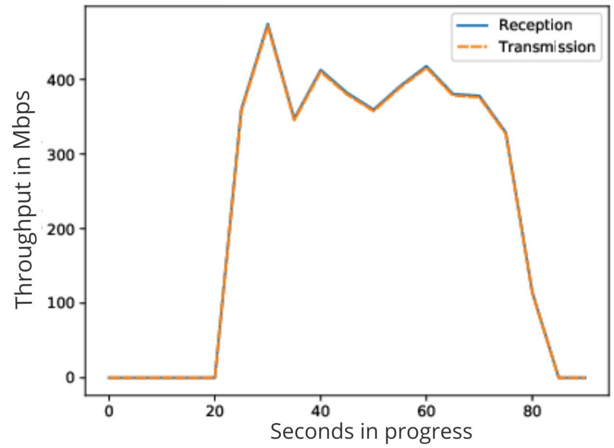

Fig. 13: Throughput of UPF, with 10 threads at core functions.

in slice admission control, NSSF could use the instant slice load information from NWDAF for slice selection.

IX. Conclusion and Future Work

We have built an end-to-end network slice performance monitoring and analytic framework on our own prototype model of 5G SBA using REST API based HTTP2, by using Prometheus, cAdvisor, Grafana in the orchestrated environment provided by ETSI NFV MANO aligned OSM Rel.5. In this context, we have measured various slice specific and VNF parameters along with the application specific KPIs like number of users registered and deregistered at 5GC through AMF. We also setup the federation test framework, for monitoring multiple slices deployed at various sites managed by single MANO and compared the results of using this mode against the non federated mode. The experimental results in these setups show that our proposed framework is flexible and extensible in supporting various areas of slice analytics towards achieving self optimization, in Self Organizing Networks (SON) of 5G Core Network. Our future work encompasses using these KPIs captured effectively to support the diversity of services and requirements for network slicing in data analytic domain and life cycle management of slices in 5G core networks.

References

[1] A. Kaloxylos, "A Survey and an Analysis of Network Slicing in 5G networks", *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 60–65, MARCH 2018.

[2] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An SDN/NFV based Framework for Management and Deployment of Service Based 5G Core Network", *China Communications*, vol. 15, no. 10, pp. 86–98, Oct 2018.

[3] 3GPP, "Architecture enhancements for 5G System (5GS) to support network data analytics services", Tech. Rep. TS 23.288, 3GPP, 2019.

[4] 3GPP, "Study of Enablers for Network Automation for 5G", Tech. Rep. TS 23.791, 3GPP, 2019.

[5] 3GPP, "5G end to end Key Performance Indicators (KPI)", Tech. Rep. TS 28.554, 3GPP, 2019.

[6] E. Pateromichelakis, F. Moggio, C. Mannweiler, P. Arnold, M. Shariat, M. Einhaus, Q. Wei, Ö. Bulakci, and A. De Domenico, "End-to-end data analytics framework for 5g architecture", *IEEE Access*, vol. 7, pp. 40295–40312, 2019.

[7] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "Data analytics in the 5g radio access network and its applicability to fixed wireless access", in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, April 2019, pp. 1–6.

[8] Sokratis Barmpounakis, Alexandros Kaloxylos, Panagiotis Spapis, Chan Zhou, Panagis Magdalinos, and Nancy Alonistioti, "Data analytics for 5g networks: A complete framework for network access selection and traffic steering", 11 2018.

[9] 3GPP, "Management and 0rchestration; Architecture Framework", Tech. Rep. TS 28.533, 3GPP, 2018.

[10] 3GPP, "System Architecture for the 5G System", Tech. Rep., 3GPP.

[11] "OSM", https://osm.etsi.org/wikipub/index.php/OSM_Release_FIVE.

[12] "Prometheus", http://prometheus.io.

[13] "cAdvisor", https://github.com/google/cadvisor.

[14] "node_exporter", https://github.com/prometheus/node_exporter.

[15] "Consul", https://www.consul.io.

[16] "Docker", https://www.docker.com.

[17] H. Karl M. Peuster and S. v. Rossem, "Medicine: Rapid Prototyping of Production-Ready Network Services in Multi-PoP Environments.", in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. 2016, IEEE.

[18] "Prometheus C++ Client", https://github.com/jupp0r/prometheus-cpp.

[19] "Grafana", https://grafana.com/.