EXPLORING THE POTENTIAL OF DYNAMIC QUALITY OF SERVICE CLOUD-BASED NETWORK SLICING IN 5G AND NEXT GENERATION VIRTUALIZED NETWORKS: A SIMULATION-BASED STUDY

Poovendren Govender[1*], Kingsley A. Ogudo[1], Chabalala Chabalala[2]
[1]University of Johannesburg, South Africa, [2]University of Witwatersrand, South Africa

NOTE: Corresponding author: Poovendren Govender, pgovender@ieee.org (*main author)

Abstract *– With global mobile traffic doubling every 18 months along with the massive increase in unique data users across digital platforms during the Covid-19 pandemic, the world's demand for high-speed connectivity continues to increase every second. The Covid-19 pandemic has transformed the global workforce with remote working arrangements, embracing new platforms, subsequently increasing the requirement for reliable, quality, ubiquitous high speed data connectedness everywhere. 5th Generation (5G) mobile communication technology provides an effective solution, modernizing network infrastructure, efficiently providing reliable high bandwidth capacity for massive data growth and ultra-low latency, making delays virtually impossible to perceive. This means elevating reality even further with fiber-like speeds for everyone, everywhere. 5G network slicing makes it possible for services to own dedicated portions of the 5G network with guaranteed performance for their particular need. These virtual networks allow for a service like IoT, at industrial scale, making society more sustainable and increasing resource efficiency. Our paper proposes a model for 5G network slice creation that meets industry verticals service QoS requirements which are critical to service adoption. A software emulation is utilized to guide the design and evaluation of our proposed model. Simulated results show QoS guarantees are met for the reviewed service requirements in varying network conditions.*

Keywords – 5G, AWS cloud computing, network functions virtualization, network slicing, OpenDaylight SDN controller, QoS

1. INTRODUCTION

The growth of mobile data demand and varying use cases have presented interesting challenges for mobile operators. In the past, the inflexible architecture of mobile networks could not satisfy the evolving requirements of services, and customers contended for similar resources in a system optimized to accommodate basic broadband data activity. This limitation could not satisfy the performance requirements or scalability and diversity of evolving services.

With the introduction of 5G networks, industry verticals require networks with deterministic service level guarantees through network performance assurances on metrics like packet loss rate, bandwidth, jitter, and latency. 5G networks provide society and industry transformation with a hyper-connected customer experience, new business opportunities, agile and lean network operations across the Core Network (CN), Transport Network (TN), Radio Access Network (RAN), and the cloud [1]. To support various Internet of Things (IoT) type device connectivity with varied services, 5G allows for varied business models, including 3P services, Communication Services Provider (CSP) services, business to business, business vs. consumer, and Mobile Virtual Network Operator (MVNO). Resource contention is not a consideration for customers of the different 5G service offerings and applications. Network slicing is introduced for 5G to solve the challenges faced by mobile operators. The process of network slicing involves dividing the entire physical network to create multiple virtual slices, each representing a distinct logical network. These networks can be of different structures and size and are run on shared infrastructure for varied industry services and applications. The 5G network offers three standardized network slices for different service categories with varying performance requirements [2]. The enhanced Mobile Broadband (eMBB) slice which forms the focus of our study supports high-peak data rates for services. 5G network slicing provides flexibility to operators for the effective management of resources for varied services available to all customer types, inclusive of

3rd party and roaming ones. It enables effective dynamic network slicing and resource management for each service offering and application. Each slice can be managed independently, potentially with dedicated resources with no sharing of resources among slices, where a slice is optimized individually to meet its associated 5G performance requirements. 5G architecture provides greater reliability, connectivity, flexibility, and scalability with the aim of providing seamless customer experience across verticals. The software capability within Network Function Virtualization (NFV) and Software Defined Networking (SDN) in 5G networks provides a flexible and agile architecture extensible in the cloud with programmable control, resource management enabling dynamic partitioning for slice creation with support in meeting verticals' performance requirements hence our cloud-based simulation approach. Typically, different verticals present varied QoS requirements that the 5G network must meet. 5G network slicing is also considered successful when we can guarantee QoS for slices sharing common resources.

SDN decouples control planes and network data. SDN enables the programmability of the control function and centralized control logic, facilitating greater network control through dynamic network slicing and resource management. SDN and NFV enable network partitioning for specific use cases, allowing for the creation of logical networks which form the basis of our research. In 5G, network slicing is expected to be widespread, enabling high data rates and robust performance. Our proposed research objectives include development and evaluation for our framework for dynamic 5G QoS-driven network slicing. Network slicing is investigated in relation to SDN and NFV, which are key ingredients of an end-to-end 5G architecture. Our goal is to achieve optimal 5G network slicing that satisfies verticals QoS requirements. A 5G slicing architecture is considered ideal when industry verticals QoS requirements are considered for the dynamic creation, deletion, and modification of slices [1]. A network slicing model that is considered optimal must consider QoS and build network slices whose performance matches QoS profiles of the different verticals, which improves Quality of Experience (QoE) [1]. Our research work studies network slicing in 5G and proposes a network slice selection mechanism that builds network slices dynamically based on QoS requirements presented by applications in different verticals. Recent network design research focus in 5G has been on fulfilling customers' different service quality requirements. Various collaborative industrial and academic 5G network slicing research projects utilize NFV and SDN as their enabling technologies. Key to network slicing in 5G are NFV and SDN enabling the flexibility to partition the 5G network into on-demand and scalable virtual networks that fulfill specific purposes. The results from various research effort models show while QoS can be managed to an extent with varying techniques, there is still an open research opportunity to further improve. Optimal slicing in a dynamic manner is still limited by the various techniques isolated focus, limited templating rule base, short-sighted node placement strategies, limited routing approaches, limited orchestration capabilities, limited slice maintenance and limited controller logic. There is still significant interest and research opportunity for 5G and virtualized next generation networks, in particular for QoS-based slice selection [1].

To ensure that these virtual networks (slices) deliver the required QoS performance, our work seeks to solve this problem through a QoS-based slice selection model that is dynamic. The model considers new and existing verticals' service applications needs through SLA criteria, performance measures and the status of resources on the network path. Our dynamic slice selection algorithm in Fig. 1 deployed on the SDN controller at the OSS leverages the Dijkstra's shortest path algorithm, as well as resource optimization to determine and maintain optimal end-to-end paths with characteristics that match the QoS requirements. Our effort aims at optimal lifecycle management of network slices. Our algorithm creates the optimal forwarding rules to compose the network services that meet QoS requirements. Our model's logic is configured within the SDN controller allowing for dynamic optimal node selection and orchestration ensuring QoS-enabled network slices for specific application requirements. Our algorithm considers resource management of data forwarding devices as part of lifecycle management. Dynamic network changes impacting the SLA, triggers reruns of our algorithm which determines if a configuration or chaining change is necessary to meet the QoS criteria for a service. We manage resources through the controller and nodes to adapt to dynamic network changes. We further utilize optimization techniques for effective path configuration and resource management to meet SLA requirements. We are thus able to manage the lifecycle of the slices ensuring QoS.

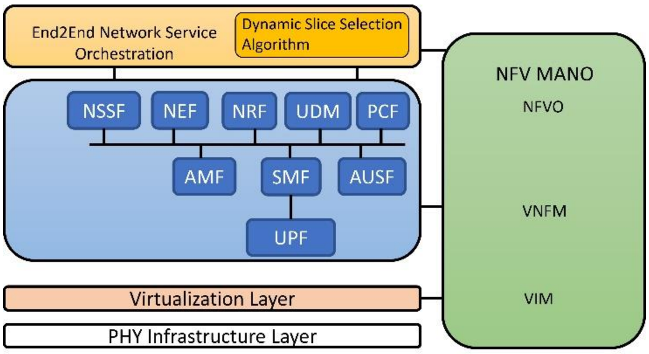

Fig. 1 – 5G-MANO mapping [1]

The NFV Management and Orchestration (NFV-MANO) framework [3] is managed by the European Telecommunications Standards Institute (ETSI) for the management and operation of services and VNFs. The architecture of NFV-MANO includes the 5G CN components detailed in Fig. 1 and allows for better standardization of operations and deployment of various services and VNFs. ETSI [4] has defined how network slicing is supported by the ETSI NFV architecture where a slice is considered connected network functions forming a graph with explicit capabilities, as well as requirements forming a network service for a specific application scenario across the network infrastructure. The NFV Orchestrator (NFVO) coordinates management and allocation of virtualized resources to support network functions. Virtual resources are created from physical resources and the Virtual Infrastructure Manager (VIM) plays a crucial role in resource control and management. The VNF Manager (VNFM) handles the VNF lifecycle management. VNFs can offer services through their deployed virtualized resources, i.e., Network Exposure Function (NEF) through which we expose our eMBB services of interest, Network Slice Selection Function (NSSF), a User Plane Function (UPF) along with the relevant Policy Control Function (PCF). We define an architectural framework within the MANO architecture for our model, as well as a performance evaluation strategy and novel cloud-based infrastructure setup. The MANO framework through our algorithm facilitates lifecycle management on network services which are formed by end-to-end chaining of VNFs. The work in this research proposes to enhance the MANO network service lifecycle management to be network slice aware and therefore manage the lifecycle of network slices accordingly. This is achieved through our algorithm deployed in the controller which in effect manages the VNFM and VIM. Our algorithm utilizes the monitoring inputs against the SLA for decision making. Fig. 2 depicts a typical lifecycle that a network slice goes through. The cycle begins with the preparation of a network slice, which entails the selection of suitable VNFs and how the VNFs are chained thus forming a network slice. The VNFs in our deployment model are represented by the switches and when stitched together form a network slice. We manage the entire process through our algorithm deployed on the SDN controller. Dynamic slice selection is a crucial aspect of 5G slicing since it efficiently facilitates the allocation of resources between slices to meet QoS demands for various applications.

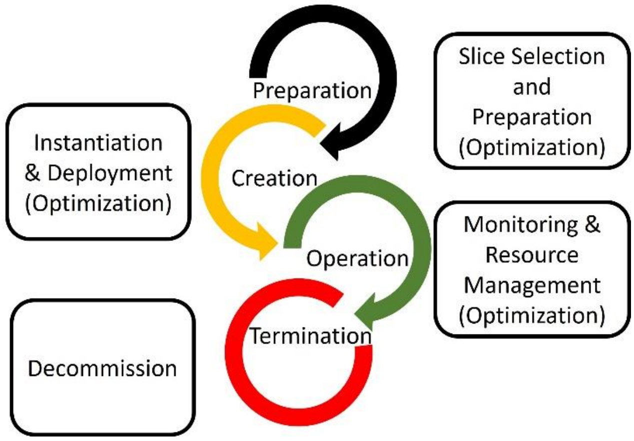

Fig. 2 – Network slice lifecycle management

This enables network provision of exceptional quality and reliable services to different types of applications, such as low-latency, high-bandwidth applications, including real-time applications, i.e., streaming video and gaming. Additionally, dynamic slice selection enables the network to enhance resource utilization and boost the network's overall performance and efficiency. Our work focuses on video and gaming use case performance optimization for greater customer experience. The research further delves into how the allocation of resources in a network slice is impacted by factors such as delay and packet loss. This paper presents our proposed approach for the dynamic composition and management along with optimization techniques of 5G network slices to guarantee QoS. Our performance model is evaluated in an AWS cloud environment. Our paper structure is detailed accordingly. In Section 2 a comprehensive literature study was performed. Relevant insights are used to guide our effort. Section 3 details the proposed slice selection algorithm and simulation methodology. Section 4 details our experimental evaluation where we first validate our algorithms performance against a default slice selection algorithm. We then evaluate two eMBB use cases i.e., video and gaming. Section 5 concludes our work where we also provide our insights on future work.

2. RELATED WORK

Standardizing 5G and next generation mobile networks has seen input from various standards bodies, associations, alliances, projects, and industry. Much of the network design research focus in 5G has been on fulfilling customers' different service quality requirements. Various collaborative industrial and academic 5G slicing research projects utilize NFV as well as SDN as their enabling technologies. A summary of these projects are provided in [5] and we further review relevant projects in this section with a QoE focus, in the context of their application and architecture.

2.1 5G Network slicing research projects

Project Slicenet [6] focused on a cognitive, agile QoE management-based 5G network slicing framework for different verticals' service assurance. Slicenet utilizes cognitive techniques for control, orchestration and management of end-to-end slicing with infrastructure sharing support across multioperator domains within the ambit of NFV/SDN-enabled 5G networks. The use cases for three verticals include ehealth, smart grid and smart city. The 5GTANGO project produced an NFV DevOps capability through SONATA which is an NFV service platform for greater NFV adoption for 5G service verticals. The modular platforms' Software Development Kit (SDK) for VNFs and network slices supports testing and validation. The MANO's automation and slicing support manages the lifecycle of VNFs and network slices. SONATA [7] allows for agile deployment and efficient orchestration of services thus supporting commercial and operational objectives. The agnostic framework allows for multivendor support. The multiorganizational design allows for greater collaboration and support. SONATA's contribution includes quicker go to market through its SDK, a DevOps model for virtual services, resource utilization optimization, deployment and operation cost reduction while accelerating the integration of NFV. The 5GTANGO project's key features include a policy framework, SLA management, and slicing. The policy engine introduces intelligence-driven orchestration mechanisms in both operational considerations and deployment of optimal network services. The framework for the management of SLAs consists of a template generator which interfaces with a policy manager resulting in templates that are customized for service provider requirements. A mapping mechanism ensures the policies from the policy manager, as well as service provider and user requirements are mapped to SLAs where the mapping results are monitored and optimized. The entire lifecycle management of the slice is managed through the slice lifecycle manager. Work in [8] focused on efficiencies in the DevOps test workflows in support of validation and verification (V&V) for network services, as well as VNFs. Through feature advancements i.e., verification and package-based testing allows for the certification of reliable V&V operators. Developers are thus able to test and verify network services and VNFs from trusted sources [9]. The catalogue's continuous adaptive optimization and decision support features allow for optimal service delivery. The 5GNORMA architecture depicted in Fig. 3 supports network slicing in a multiservice, multitenancy environment over shared infrastructure while providing context aware adaptability of network functions in a resource efficient manner for greater cost effectiveness and QoE/QoS service fulfillment. The layered architecture consists of the subsequent layers respectively: data, control, mano, and service.

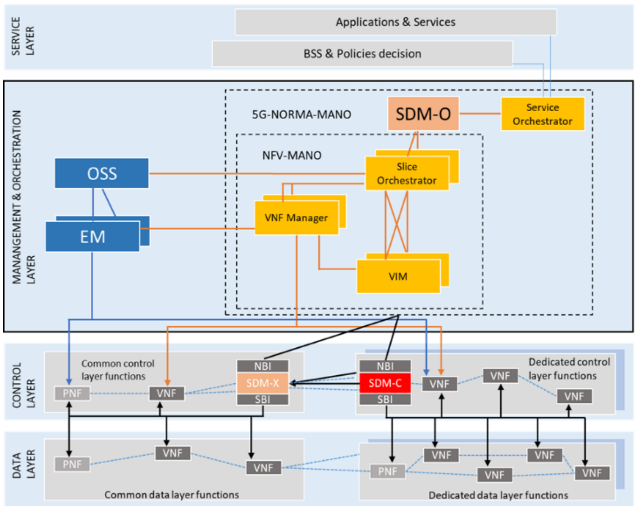

Fig. 3 – 5G NORMA functional reference architecture [10]

Software Defined Mobile Network orchestration (SDM-O) manages various requests for slices and maps business requirements with infrastructure requirement fulfillment mapping functions based on templates. 5G NORMA's QoS/QoE framework accommodates multitenant, multiservice requirements and are triggered through Software Defined Mobile Network Control (SDM-C). When rearrangement of resources is required, this is enabled through Software Defined Mobile Network

Coordinator (SDM-X). SDM-C features within slices assigned by the SDM-O and manages network slice resources ultimately setting up and adapting forwarding paths for the SFC with due consideration of the SDM-O's service requirements i.e., policy, QoS and legal requirements. The adaptation through decomposition of network functions along with the optimal placement of the NFs allows for greater operational efficiencies. The SDM-X manages resource control as well as network functions (VNFs and PNFs) across various shared slices and their associated SDM-Cs for interslice control. Policy management at the SDM-O for multitenant SLA fulfillment is key for the effective coordination of the business and infrastructure areas. 5G NORMA networks are secure due to the security approaches embraced through SDN, NFV, network slicing and multitenancy [10].

5G-MONARCH [11] focused on an adaptable, programmable and flexible 5G network architecture for various verticals' use cases i.e., media, industrial, entertainment and smart city. The architecture with its end-to-end support includes key components i.e., slice specific/common functions, multitenancy, interslice resource management, RAN control application integration and features which include service-based interfaces and capability across layers, slice blueprints and an analytics framework for network automation. The architecture consists of a security and cloud resilience capability where cognitive network functions are utilized for fault detection and controllers with security enhancements are responsible for various threat and anomaly detection. Through analytics, AI/ML elastic virtual functions allow for optimal cloud resource elasticity thereby improving resource efficiencies. The project's testbeds and validation framework demonstrate the realistic viability in meeting performance targets for the use cases. Suitable KPIs from a technical and commercial/economic perspective are used to validate the work. The project has been successful through its standardization contribution.

2.1.1 *Related research work*

Various research efforts continue to focus on solving different aspects of 5G and next generation network slicing [1]. Our research effort is relevant and gives us an opportunity to contribute towards standards improvement with a view to realizing optimal deployment capability for realistic use cases. This provides a guided approach to the development of our model and optimization methodology which we present in the subsequent section to guarantee QoS for service verticals. The following review provides an overview of related works.

G. Zhao et al. [12] investigate optimal end-to-end slice selection by considering service requirements and network constraints when matching users, access points and slices. The proposed slice selection aims to increase satisfaction degree, and improve network resource utilization as well as guaranteeing QoS of users. The presented network slice selection scheme is NP-hard. A Genetic Algorithm (GA) is used to attain optimal user satisfaction. The slice selection GA is compared to the greedy and RSS access point algorithms. The GA outperforms both the access slice algorithms and guarantees users QoS whilst improving network resource utilization. However, dynamic slice sharing and revocation are not directly addressed.

W. Guan et al. [13] develop a deployment model for end-to-end network slices. The authors propose an infrastructure network model, a network slice request implementation algorithm, along with deployment algorithms for different verticals, i.e., uRLLC, eMBB and mMTC slices which must efficiently utilize infrastructure resources. The slices service demands define associated objectives.

The uRLLC slice has strict requirements for latency, throughput, and availability with a latency QoS guarantee. The eMBB slice requires maximizing resources and the mMTC slice requires maximizing bandwidth on physical links. The deployment algorithms are compared against the simulated annealing (SA), VNF placement algorithm (GLL) and the VNE algorithm (LAVA). The simulation results show that the developed algorithms outperform the other three algorithms in terms of acceptance ratio and resource efficiency.

Various end-to-end isolation approaches with security considerations are discussed by Z. Kotulski et al. [14]. Isolation attributes assist in determining end-to-end isolation levels as security policies are adjusted for the service requirements. The attributes allow for both the verification and definition of the isolation levels within slices. Dynamic isolation mechanisms also support the creation of isolated resources and the use of virtual resources from different slices in a secure manner. Even though network slice isolation is required to isolate traffic for the different verticals, the authors do not address QoS guarantees for the verticals.

A. Kammoun et al. [15] propose an algorithm where overload cost and requirement considerations determine optimal slice selection for user requests based on a combined criterion consisting of availability, computational performance, reliability and network resource latency. The aim of their work is to improve user QoS as user requests are mapped to an optimal slice instance. The proposed algorithm's performance results, based on a mathematical model, demonstrate optimal slice selection and minimized overloading cost.

M. Gragmalia et al [16] provide a view of the 5G Norma architecture aiming at achieving flexible heterogeneous service support. The proposed use cases are intended to highlight a collaborative systems design approach with respect to the Norma architecture. The authors focus on the controller aspect and discuss mobility management, QoS/QoE and orchestration methods with the aim of optimizing the respective CN and RAN functions. The research contribution is a modular architecture which allows for a shared resource capability within a multitenancy heterogenous service environment.

G. A. Carella et al. [17] propose a network slicer which is an extension to the Open Baton platform to guarantee QoS for different network service requirements. The proposed network slice engine manages the allocation of VNFs with dedicated resources based on requirements per slice. The network slice engine functions between the NFV MANO domain and SDN controller, utilizing traffic shaping mechanisms to limit bandwidth rates on virtual links over several VNFs. The network slice manager's northbound interface utilizes events from the Open Baton NFVO and the south bound interface that communicates with the Connectivity Management Agent (CMA) in order to manage QoS for network services. The CMA manages the requests and QoS enforcement. The CMA configures QoS parameters on the appropriate instances. Network services on customized slices with explicit requirements were evaluated with respect to latency, bandwidth and reliability in the Open5GCore. The results showed the different services on different slices were not impacted by each other. The latency and bandwidth were not impacted by the consumption heavy Internet Performance Working Group (iPerf) service.

The related work presented here provided important learnings to realize an optimal slicing solution. The ideal 5G network slicing architecture would consist of the dynamic creation, modification, and deletion of network slices with due consideration of various verticals and their applications QoS requirements. Our contribution to solving this problem is detailed in the following sections with our appropriate model, use cases, analysis, and recommendations.

3. SYSTEM MODEL

This section describes the proposed solution enabling slicing within the 5G core, taking into consideration network path parameters that make up network slices within the network. The proposed solution's goal ensures when slices are formed to serve a specific application, such applications QoS requirements are taken into consideration and the network elements are chained on the network paths that have the performance characteristics that can meet the application's QoS requirements. When the path characteristics change, the proposed solution shall adapt and reconfigure network slices accordingly.

3.1 Network model for the proposed network slicing strategy

The architecture of 5G networks consists of data and control planes. The control plane's responsibility includes providing forwarding decisions of the data plane along with rules for routing. The data plane's responsibility includes user traffic forwarding from source to destination through the routing rules. Several connected nodes which form end-to-end paths represent the 5G architecture. These nodes, forming end-to-end paths can be considered as the data plane which bears user traffic. The nodes and their links can be modeled as an undirected graph defined as $\mathbf{G} = (\mathbf{N}, \mathbf{L})$ where 5G network nodes are represented by vertices set $\mathbf{N}$. The links responsible for node connectivity forming paths between source to destination are represented by edges set $\mathbf{L}$. Our model conceptualizes the 5G core as an undirected graph ($\mathbf{G}$) in Fig. 4.

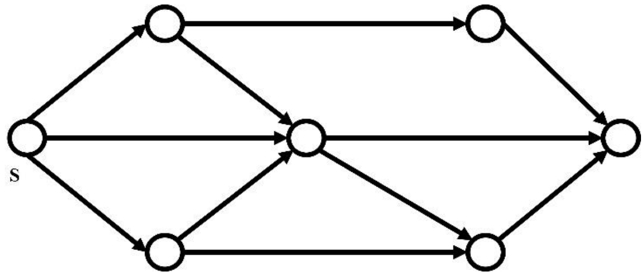

Fig. 4 – Core network model as an undirected graph [1]

Any two nodes, ni, $nj \in \boldsymbol{N}$, in the model connected with link $lij \in \boldsymbol{L}$ where every link ($l \in \boldsymbol{L}$) consists of a link cost represented by weight $w(l)$. This link weight cost represents bandwidth, loss, and delay. A 5G network slice is thus created through chaining from source to destination of several nodes. In order to fulfill varied verticals' applications QoS requirements, the specific link costs $w(l)$ has to optimized to fulfill a network SLA and policy. A network node where, $n \in \boldsymbol{N}$ represents a VNF responsible for the control plane forwarding rules. Through our system model, our proposed methodology seeks to find optimal network paths by linking suitable network nodes that result in paths which satisfy or optimize applications' bandwidth, latency, or loss rate requirements. Network slices are thus created once optimal paths are determined. The data plane utilizes control plane forwarding rules to perform application traffic path routing that satisfies a particular QoS. In essence through our model, we form a network slice by linking several nodes or VNFs between source and destination. We optimize link costs to essentially fulfill specific verticals' applications requirements in terms of QoS. To this end, we utilize Dijkstra's shortest path algorithm [18] to determine network paths that can be used, resulting in network slices capable of offering the desired path attributes for applications running within certain verticals. Dijkstra's algorithm can be used to solve other parameters in a 5G network where distance could represent bandwidth, packet loss or latency for links between nodes. As mentioned earlier, the VNFs in the 5G network that's implemented according to the ETSI NFV MANO framework are continually monitored and the performance information is reported to the VNF manager. Using this information our algorithm deployed in the SDN controller in the OSS dynamically determines optimal VNF chaining to create network slices that meet QoS requirements by different applications. The NSSF selects the optimal slice for service requests presented by our algorithm through the SDN controller. This is done by a policy alignment on the NSSF. Algorithm **I** illustrates how to determine candidate VNF graphs that can be used as network slices with performance characteristics that match the QoS required by different applications. Our algorithm is built on the shortest path algorithm by Dijkstra.

Our work seeks to determine optimal network paths which serve to facilitate QoS parameter optimization. The link weights we consider for optimization include bandwidth, latency, and packet loss. Our algorithm considers the input made up of the nodes $\boldsymbol{N}$, links $\boldsymbol{L}$ and link weight $w(l)$. Through the algorithm we determine the set of network paths $\boldsymbol{N'}$. Through the algorithm we can find the path with the least cost for any node u from the source node and updates output $\boldsymbol{N'}$. This iteration continues for all nodes in $\boldsymbol{N}$. The output $\boldsymbol{N'}$ forms the least cost network path tree in terms of forwarding rules. The subsequent forwarding rules are utilized by the control plane for the configuration and chaining of VNFs for the creation of slices that are QoS-enabled. The result from executing Dijkstra's algorithm on a set of 5G VNFs is a graph with the minimum path costs to all nodes in the network from a given source VNF. Path **P** can be set up between source ($\mathbf{s}$) to destination ($\mathbf{v}$) in terms of the least cost path tree for network slice creation which satisfies a verticals application requirement. Therefore, path **P** between source and destination forms the desired slice if end-to-end cost (packet loss, bandwidth, latency) for path **P** meets QoS to serve a vertical's application. A resulting network slice from Path **P** comprises VNFs that must be stitched suitably with supporting configuration for traffic steering. NFV enables the configuration of the network slices which allows for optimization efficiencies. Algorithm I details the iterative approach to create the desired path tree with the least cost. This forms a basis for the slice selection proposal that is QoS-enabled. Algorithm **I** is a section of our dynamic slice selection algorithm, and we discuss the resource and lifecycle management components along with the optimization techniques to enhance performance in the following section.

Algorithm I Dijkstra-based network slice selection

1 : *Input: set of nodes, **N**; source node **s**; $w(l)$, cost of link $l \in L$*

2 : *Output: **N'** | **P***(**s**,**v**) represents a path from **s** to **v** that is optimal and **c**(**P***(**s**,**v**)) represents optimal path cost*

3 : ***For Each*** *network node v*

4 : $c[v] = \infty$ // initial cost of the nodes other than the source

5 : ***EndFor***

6 : $c[s] = 0$ // cost source node

7 : $\boldsymbol{N}$: = {set of all network nodes}

: ***N′: =*** {optimal sequences of network nodes}
: ***While N*** $\neq \phi$
:   *u ← node with minimum distance in N*
:   *remove u from N*
:   *enqueue u into N′*
:   ***For Each*** *node v adjacent to u*
:   ***If*** $(c[u] + c[u, v]) < c[v]$ ***then***
: $c[v] := c[u] + c[u, v]$
:   ***EndIf***
:   ***EndFor***
: ***EndWhile***
: *Return* ***N′***

While various methods of chaining VNFs exist in the path tree with the least cost, the IETF's SFC method [19] is the most prominent. Our proposal utilizes SFC for the node configuration and linking which forms network slices that are QoS-enabled. Our dynamic approach of the selecting of slices operates at the control layer whilst providing data layer directives for traffic forwarding. The goal is to meet industry verticals varied applications specific performance requirements which are contracted on an SLA tenant agreement with operators [1]. Control layer network monitoring is imperative to be implemented within the architecture to ensure that network performance variations are considered for algorithm reruns. We propose the use of both network slice and infrastructure monitoring in the following sections which form the inputs considered by our model.

3.2 Proposed architectural performance evaluation framework

We define a QoS framework within NFV MANO architecture in Fig. 5. Our controller is deployed as part of the OSS. Through templating via the NSSF tenants can contract with operators to manage their service performance requirements. QoS thresholds for Latency = Q_thres1, Bandwidth = Q_thres2 and Loss rate = Q_thres3 are defined and contracted between the operator and tenant. This is managed at the SLA manager and enforced via the SDN controller through our algorithm. Our algorithm provides the orchestration through the controller by configuring and optimizing the VNFs forming forwarding paths through chaining with orchestration layers QoS and policy insight which is received from the SLA manager.

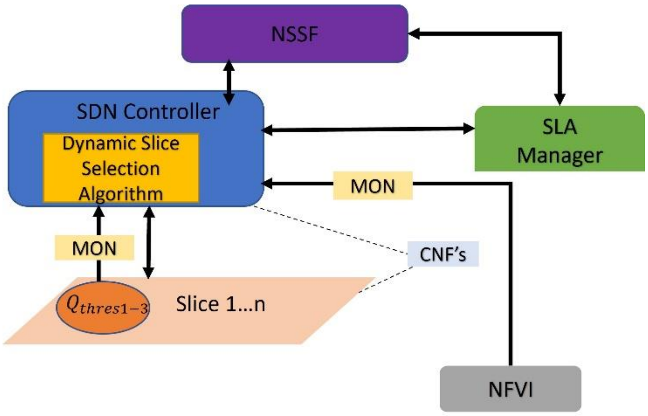

Fig. 5 – Proposed 5G-MANO QoS architecture

3.2.1 *Slicing pseudocode inclusive of resource management*

In addition to the algorithm's resource management, we employ a streaming API, NETCONF with caching, multiprocessing and multithreading process modules to enhance performance.

For each Open vSwitch connected to the SDN controller*:*

a. Enable all available QoS tools on the switch

b. For each port in the switch:

i. Create a default queue

ii. Create n additional queues

iii. Assign a specific bandwidth to each queue

iv. End the loop for all ports

c. End the loop for all Open vSwitches

For each defined QoS rule*:*

a. Retrieve information for ip source, ip destination, port, queue, and protocol

b. Check for an optimal link between the ip source and ip destination using Dijkstra's shortest path algorithm

c. If a link exists:

i. Create a virtual path between the ip source and ip destination

ii. Create a rule using the information retrieved in step 2a

iii. Add the rule to the policy

d. If a link does not exist:

i. Print an error message indicating that a circuit could not be created

e. End the loop for all QoS rules

***For situations where QoS is negatively impacted for a slice (SLA degradation) (either at setup or during operation)**:*

a. Check if there is a resource surplus

i. If so, determine the necessary resource requirements

ii. Make a request to the VNFM and VIM to scale in/down the VNFs

iii. Opendaylight Controller to switches via NETCONF

b. Check if there is a resource shortfall

i. If there is no shortfall, update the software or perform a node reset

ii. If there is a shortfall, make a request to the VNFM and VIM to scale up or out the VNFs if resources are available

iii. Opendaylight Controller to switches via NETCONF

iv. If resources are not available, change the VNFs

c. Reconfirm if the path that was upgraded is the best path to meet QoS

i. If not, select the path with Dijkstra's shortest path algorithm which would support the required QoS

d. End the loop for all QoS rules

3.2.2 Slicing lifecycle management

We propose utilizing performance monitoring capabilities where our algorithm utilizes the network slice’s QoS threshold reporting and infrastructure network performance monitoring including the VNF’s resource monitoring to manage the slice performance appropriately. The network slice’s VNF provides performance reporting to the VNFM. The VIM provides infrastructure resource performance reporting to the VNFM which in turn forwards this to the NFVO and which ultimately provides the application specific performance reporting to our controller within the OSS/BSS. The SDN controller is also updated by the slices contracted QoS thresholds through the SLA manager. Our algorithm thus has full oversight of not only the contracted QoS SLA criteria, but slice performance and network performance. Our algorithm can therefore make optimal path selection decisions dynamically. In the case where reorchestration and rearrangement are required, our algorithm provides optimal path reselection. This is where our effort improves on the NFV MANO standard in [20], whereas some automated VNF contraction is triggered by performance metrics, QoS performance is not guaranteed as the optimal path of optimal chained VNF's is not set up. Our effort dynamically sets up the most cost-effective path in term of QoS taking all conditions into consideration. This is a performance differentiator which will drive service monetization for operators as tenants will be able to provide optimal service guarantees that will suit customer requirements and experience ultimately driving the net promoter score. Changes in network or slice performance trigger a rerun of our algorithm. The changes are picked up from monitoring through the control layer at an application slice and infrastructure level. This is important as the dynamic nature of our algorithm depends on accurate and timeous reporting to make decisions. Our model decides suitable actions to meet a particular service’s applications QoS requirements. This could include VNF rechaining or reconfiguration and in some cases resource sharing of CNFs. Reconfiguration refers to continuous resource management to meet the QoS of an existing or new path. This is necessary to address the dynamic 5G network. We are not only interested in finding the optimal path but also maintaining an optimal performance to satisfy QoS as detailed in Fig. 6 and Fig. 7 through optimization where possible. In the case of a path setup our dynamic slice selection algorithm builds a VNFFG using SLA QoS thresholds from the NSSF and SLA manager to determine the optimal path and directs the NFVO to execute the path instructions. The NFVO instructs the VIM to allocate relevant resources for the VNFFG. The VIM allocates the resources and confirms resource allocation to our algorithm via the NFVO. For an updated path, our algorithm sends VNFFG update instructions for the path to the NFVO which instructs the VIM to allocate resources. The VIM allocates the resources and updates the NFVO. The NFVO sends our algorithms path update request to the VIM. The VIM does a path configuration update and confirms the update to the NFVO. The NFVO updates path records and confirms the VNFFG update to our algorithm. In the case of a path deletion, our algorithm sends the VNFFG deletion instructions to the NFVO which forwards this to the VIM. The VIM deallocates resources and confirms to our algorithm of the VNFFG deletion for the path via the NFVO.

In the case where our algorithm detects QoS degradation from the slices’ VNFs, our algorithm would recommend reconfiguration of resources which could mean scaling up or scaling out and where there is a resource surplus, scaling in or down. This could mean updating existing VNFs or selecting different VNFs as per the algorithm's recommendation. Our algorithm triggers the specific scaling action towards the NFVO. The NFVO validates the request and instructs the VNFM to start the scaling preparation. Once the VNFM is complete, it requests a resource change in the form of resource update, allocation, or scaling. The NFVO passes the resource change request towards the VIM which updates the connectivity of the internal network. Compute and storage resources are started and new VMs are attached to the internal network. The VIM then confirms the resource changes to the NFVO which confirms the same to the VNFM. The VNFM sets up the scaled VNFs and confirms the completion of scaling to the NFVO. The NFVO updates our algorithm with the completion of scaling. Through our algorithm we are able to manage resource efficiency in a dynamic network, along the path including changes of VNFs with path reconfiguration such that the QoS is realized. Through our algorithm we are able to manage resource efficiency in a dynamic network, along the path including changes of VNFs with path reconfiguration such that the QoS is realized.

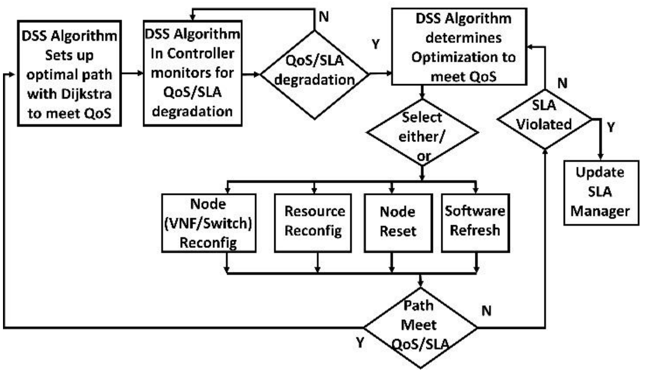

Fig. 6 – Optimization flow

Our algorithm in the controller considers not just a lack of resources impacting SLAs but cases where resources could be hanging and a node, in our case a switch, would require a restart or software update. In the case where resources are available then the nodes are scaled up, out or new nodes are selected where optimal path selection with Dijkstra's algorithm is used. Resource management optimization detailed in Fig. 7 starts at the initial step where the optimal path is selected by our algorithm and continues for any new or existing paths such that QoS is realized for the optimal path else the SLA breach is triggered, depending on the violation duration.

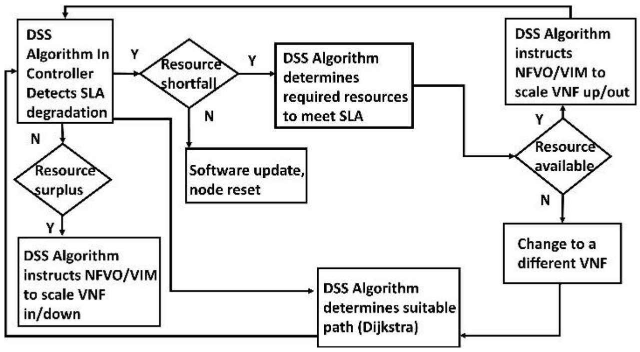

Fig. 7 – Resource optimization management

In the case where a different VNF is selected, the Dijkstra component for the optimal path is still triggered as we are interested in the optimal path that will satisfy the SLA QoS at all times.

3.3 Performance evaluation strategy

We present the evaluation of our network slice selection model (Dynamic Slice Selection). We verify our model's performance through a comparison with a static slice selection model (Default Slice Selection). A network slice is made up of network paths and the paths' dynamic nature is not a consideration for the default slice selection model. Through our comparison we review QoS performance measures which include throughput, packet loss and latency. Our goal is to determine if our dynamic slice selection model adapts to changing slice performance characteristics and honors the required QoS requirements. We then proceed to review eMBB use cases to confirm if our model satisfies the associated QoS requirements. Some funded 5G slicing projects discussed earlier were able to build testbeds and evaluate their models over a period. Low cost or open source testbed infrastructure for 5G remains wanting. This is one of the challenges researchers face in 5G service model performance evaluations.

As a result of this limitation our approved research approach is a software emulation. To this end we utilize a Mininet network emulator [21] and OpenDaylight SDN controller [22]. For the performance evaluation, our algorithm is deployed on the OpenDaylight SDN controller at the control layer and Mininet OpenFlow switches function at the data layer. OpenDaylight has been successfully utilized to perform control layer functionality in other 5G initiatives and therefore suitable for our evaluation. The OpenFlow switches are comparable to data forwarding elements in 5G and therefore an appropriate means to emulating the data layer in 5G while the control layer is emulated by the SDN controller. As we utilize the OpenDaylight SDN controller to deploy our algorithm, subsequent rules for forwarding that are generated are sent to the data layer for Mininet OpenFlow switches to execute [1].

3.3.1 Infrastructure setup

We evaluate the model's performance through the cloud infrastructure configuration in Fig. 8. Our system configuration runs within an AWS cloud environment. The cloud environment compute resources consist of AWS EC2 compute instances. The cloud environment provides a realistic deployment state for 5G core networks whilst cloud compute benefits include low ownership costs and greater flexibility. AWS is selected over other cloud providers such as Google Cloud and Azure because the researcher has experience working on AWS. The AWS EC2 compute instances have Ubuntu Linux distribution with OpenDaylight controller and Mininet deployed. Our default slice selection algorithm is easily deployed on the OpenDaylight controller through new features as OpenDaylight is open source and modular [1].

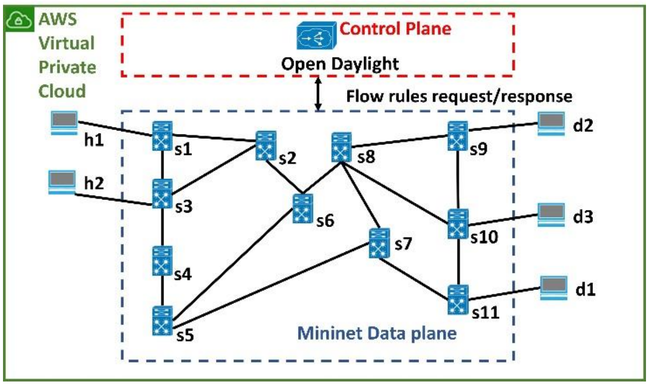

Fig. 8 – Evaluation infrastructure configuration in AWS

4. EXPERIMENTAL EVALUATION

This section presents our 5G network slicing system model's performance evaluation. We have confirmed our algorithms performance in our previous work [1] in a simulated environment. Our model was compared and validated against a network slice selection method. In this work we proceed to evaluate eMBB vertical use cases which include HD mobile cloud gaming and HD video type traffic.

4.1 Performance review

We utilized Distributed Internet Traffic Generator (D-ITG) [23] for video, gaming and background traffic generation in Mininet. Wireshark and iPerf are used for our analysis. D-ITG supports various application protocol traffic and allows for QoS performance metric evaluation. D-ITG emulates the gaming and video streaming service by generating the service traffic over the User Datagram Protocol (UDP) which is ingested in Mininet. UDP type traffic allows for the dynamic application of our emulation especially where we need to simulate congestion scenarios. We utilize two hosts for our services and a common destination where 100 Mbps is set uniformly for link bandwidth as detailed in Fig. 8. Our QoS service performance evaluation considers the metrics defined by 3GPP [24] and 5G slicing association [25] detailed in Table 1. We utilize this minimum QoS criteria to review our algorithm's performance for the service type as we utilize a simulation environment. We establish queue priority in order of preference, minimum delay, bandwidth, and best effort. We utilize the same max packet MTU size and vary the packet rate for gaming due to its interactive nature and higher priority. The interactive nature of gaming and the need to send and receive high-resolution graphics and data for multiplayer games results in a higher packet rate and bandwidth requirement than video streaming.

We use Python scripts to set up the network configuration and with the help of DITG, we establish the flow of data from Table 2 and Table 3 that needs to be transferred from the source to the destination with a ratio of 1:3:6. Our model is implemented in the OpenDaylight SDN controller with performance monitoring. Our SDN controller logic uses a process of monitoring, analysis, decision-making, configuration, and verification to reconfigure the network slice as needed.

Table 1 – 5G QoS characteristics mapping [24] [25]

Service Type	Bandwidth	Latency	Packet loss rate
HD Video	>15 Mbps	<100 ms	$<10^{-3}$ (1%)
HD Gaming	>25 Mbps	<100 ms	$<10^{-3}$ (1%)

Table 2 – Traffic characteristics

Flow	Traffic (UDP)	Packet Size (Bytes)	Packet rate (Packets/s)	Priority
h1-d1	HD Video	1 500	3 000	2
h2-d1	HD Gaming	1 500	3 500	1

We implement a novel streaming API to collect real-time stats from the OpenFlow switches instead of requesting periodic updates which impacts rerouting time, as the controller needs to make decisions timeously to be effective at guaranteeing QoS. Based on the analysis, our SDN controller model makes decisions on how to reconfigure the network slice to optimize its performance.

Table 3 – Background traffic characteristics

Flow	Traffic (UDP)	Packet Size (Bytes)	Packet rate (Packets/s)	Priority
h1-d2	VoIP	Random	Random	3
h1-d3	Best effort	1 000	1 000	5
h2-d3	Best effort	1 000	3 000	4
h2-d2	VoIP	Random	Random	3

This may include adjusting the allocation of resources, modifying the traffic flow, resets, software updates or making changes to parameters for QoS. Our controller model sends instructions for network devices within a slice to configure the network according to the decisions made. The SDN controller model then verifies that the network slice has been reconfigured correctly and that the desired performance has been achieved. We utilize novel NETCONF scripts with caching, multithreading and multiprocessing to achieve this between the OpenDaylight controller and OpenFlow switches.

4.1.1 Experimental results

The following are simulated results from the emulation where 50 simulated results provided the average detailed in the tables below.

Table 4 – HD video stream performance comparison

Traffic Class	SLA	Default Slice Selection	Dynamic Slice Selection
HD Video	Average Loss Rate %	15.12	0
HD Video	Average Bandwidth (Mbps)	7.3	19.6
HD Video	Average Latency (ms)	282	87

Simulated results from Table 4 and Table 5 show the default slice selection model does not meet the QoS criteria in Table 1.

Table 5 – HD gaming stream performance comparison

Traffic Class	SLA	Default Slice Selection	Dynamic Slice Selection
HD Gaming	Average Loss Rate %	15.64	0
HD Gaming	Average Bandwidth (Mbps)	22.6	33.1
HD Gaming	Average Latency (ms)	247	63

The bandwidth is close to optimal for the HD gaming use case however the 74% difference in latency when compared to the dynamic slice model would certainly impact user experience. This coupled with the loss rate would render the service adoption negatively. For video and gaming latency can cause delays in the display of images and movement, making the experience less responsive and less enjoyable for the user.

Packet loss can cause video to be choppy, audio to be garbled, and in-game actions to be delayed or not executed at all. Low latency and low packet loss are crucial for providing a smooth and high-quality experience for video and gaming services. The dynamic slice selection model's results show full compliance with Table 1 requirements and confirms optimal path selection and reconfiguration.

4.1.2 Congestion control performance

In this section we extend our study to include practical deployment considerations in a mobile network. Generally, Operators would dimension their network for acceptable performance up to 10~15% congestion before rate limiting and redundancy techniques come into effect.

This provides an opportunity to review both models serving the eMBB use cases and showcase the relevance of our model. The results are presented in Table 6 and Table 7 respectively.

The focus remains on QoS metrics while QoE evaluations can be considered as part of future work. Congestion control support is critical to managing network resources and ensuring that the QoS requirements of video and gaming services are met, by managing bandwidth, reducing latency, and providing fairness among different services.

Table 6 – HD video stream congestion performance comparison

Traffic Class	SLA	Default Slice Selection	Dynamic Slice Selection
Network Congestion at 10%			
HD Video	Average Loss Rate %	19.5	0.03
HD Video	Average Bandwidth (Mbps)	6.9	19.4
HD Video	Average Latency (ms)	352	89
Network Congestion at 15%			
HD Video	Average Loss Rate %	21.4	0.05
HD Video	Average Bandwidth (Mbps)	6.2	19.1
HD Video	Average Latency (ms)	417	93

Table 7 – HD gaming stream congestion performance comparison

Traffic Class	SLA	Default Slice Selection	Dynamic Slice Selection
Network Congestion at 10%			
HD Gaming	Average Loss Rate %	15.64	0.06
HD Gaming	Average Bandwidth (Mbps)	18.3	32.1
HD Gaming	Average Latency (ms)	343	67
Network Congestion at 15%			
HD Gaming	Average Loss Rate %	23.7	0.09
HD Gaming	Average Bandwidth (Mbps)	11.3	31
HD Gaming	Average Latency (ms)	459	75

Overall, simulating 10% and 15% congestion on the network model for HD gaming and video services is important for understanding how the network slicing will perform under realistic network conditions and for optimizing the network slicing configuration to guarantee that the QoS specifications are satisfied. The simulations gave us insights into ways to enhance the configuration and code to achieve better results. Response times are critical to dynamic slice selection. By caching results of Dijkstra's shortest path algorithm, we optimize performance by reducing the duration necessary to run the algorithm. The program will not need to recalculate the shortest path for the same graph and source node multiple times. The controller logic has to interrogate the monitoring information and determine the necessary actions to implement in order to fulfill the QoS requirements. If there are any inaccuracies with the reporting data or lack of synchronization causing race conditions and concurrency issues, then this would impact the optimal path setup. This impacts the service meeting the SLA and results in poor customer experience. We utilize a multiprocessing module in Python, which allows us to create and manage different processes. For example, we are able to create separate processes to handle the add_bandwidth and remove_bandwidth functions so that they run concurrently with the rest of the code. We also leverage a threading module through Python, which allows us to create and manage threads. For example, we are able to create a separate thread to handle the get_slice_performance function so that it runs concurrently.

4.1.3 Default slice selection performance analysis

The results of this study confirm the flexibility of SDN architecture. The physical layer remained unchanged for the default slice selection model while multiple configurations were implemented in the logical infrastructure. This methodology influenced the results presented. The default slice selection model was unable to adapt to the congestion. In this study, network latency measurements were obtained and analyzed. The results of the analysis indicate a direct correlation between the assigned bandwidth and the observed delays in the network for the default slice selection model. Simulated results for throughput in the case of default slice selection indicates that the total accumulated bandwidth across various path configurations is consistent, regardless of the specific bandwidth allocation used. As congestion increased, we can see a correlation between the packet loss rate and the varying bandwidth. This correlation for latency is clear and concerning from a user perspective especially with interactive gaming. There is an almost exponential decline in latency for increased congestion. The results for both congestion levels show the default slice selection model's inability to satisfy the criteria detailed in Table 1.

4.1.4 Dynamic slice selection performance analysis

Our dynamic slice selection model allows for optimal path selection and optimal reconfiguration to meet the slices' QoS requirements. As congestion is increased, while the optimal path is found, further optimization is carried out. Our model configures required resources where possible on the VNF to meet QoS. From the results we can observe that as the packet loss rate is reduced, the links' reliability improves, which results in better utilization of the assigned bandwidth. Our model ensures consistent throughput for both use cases even as congestion increases. As the available resources increase by our algorithm, the packet loss rate decreases. We did observe a slight increase in loss rate to 0.09% for the HD gaming case at 15% congestion but this is still within the acceptable QoS performance metric. Latency remains stable with increasing congestion as resource capacity is added.

This study's outcomes reveal that bandwidth availability has a direct impact on delay and packet loss. Our model's effect can be understood for HD video latency at 10% congestion with a 74% improvement while there is 77% improvement at 15% congestion. In the case of HD gaming, we realize a latency gain of 83% at 15% congestion and 80% at 10% congestion. We see significant gains for packet loss rate and bandwidth compared against the default slice selection model at the different congestion levels. We were able to satisfy the requirements in Table 1. It can be surmised that our controller logic effectively assigns network slices based on the requirements of the service application, thus achieving the targeted latency, bandwidth and maintaining a consistent QoS. We utilized NETCONF to enhance network performance by adding switch layer resources, configuring OpenFlow tables, increasing buffer size, and setting QoS. We optimized code execution with the Dijkstra algorithm, multithreading, and multiprocessing, and evaluated algorithm performance using packet loss and throughput results. Our algorithm achieves desired QoS by combining optimal routing and resource usage while preventing switch overload. By leveraging a streaming API through the OpenDaylight SDN controller, our algorithm can push updates to switches and dynamically respond to network changes to ensure efficient traffic handling. Regular network performance monitoring allows for adjustments to resources and algorithm parameters to adapt to changing conditions and traffic patterns, and appropriate slice selection maximizes network resource utilization for specific services or applications. This contributes directly to the business case and viability aspect of 5G network slicing as commercial viability is critical. Poor management of resources directly impacts commercial models and revenue. We confirmed our model can help prevent network congestion by selecting the appropriate slice for a particular service or application.

5. CONCLUSION

Our research objectives were met by successfully orchestrating, reconfiguring and managing network slices in a dynamic 5G network environment. These slices satisfy the required service QoS and are managed within the QoS architectural framework we define in the MANO architecture. We optimized resource management and path selection performance using multithreading and multiprocessing with synchronization. As a result, we were able to dynamically orchestrate, reconfigure, and manage slices with QoS support for the reviewed vertical use cases. Our dynamic QoS-enabled network slice selection model generated forwarding rules, which enabled optimal service composition, resource management and slice lifecycle management within the MANO framework. We also developed an optimized streaming API for real-time statistics, which assists with QoS guarantees and enhances controller decision-making. Additionally, we implemented optimized NETCONF scripts for resource management to ensure QoS. Our proposed model's performance was evaluated and we confirmed that it satisfies the selected service QoS requirements. Through simulated results for selected eMBB use cases (gaming and video), we were able to demonstrate the model's ability to construct network slices that guarantee QoS as per the 3GPP and 5G Slicing Association, while adapting to dynamic network conditions. Compared to the default static model, our proposed dynamic slice selection outperformed it and most importantly, it was able to satisfy the QoS requirements for the evaluated eMBB use cases. In summary, our proposed dynamic slice selection improves resource allocation to satisfy QoS requirements, providing high-quality and reliable services to reviewed applications while optimizing resource utilization and network performance. Future work could involve optimizing the RAN, TN, and CN, as well as the OSS, to improve end-to-end performance and drive service demand. The incorporation of machine learning optimization techniques would further enhance the functionality of dynamic network slicing. Additionally, future work involves reviewing use cases and associated challenges for mMTC IoT, such as industrial automation, smart farming, smart wearables, and e-health, as well as URLLC use cases for VR gaming and autonomous vehicle applications.

REFERENCES

[1] P. Govender et al., “Quality of Service Enabled Network Slicing Model in 5G and the Next Generation Virtualized Networks,” in Proc. 2021 IEEE AFRICON Conf., 2021, doi: 10.1109/AFRICON51333.2021.9571005.

[2] IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond, Recommendation ITU-R M.2083-0, (09/2015).

[3] S. Abdelwahab et al., “Network Function Virtualization in 5G,” IEEE Commun. Mag., vol. 54, pp. 84-91, 2016.

[4] ETSI Report on Network Slicing Support with ETSI NFV Architecture Framework, ETSI Standard GR NFV-EVE 012, 2017.

[5] A. A. Barakabitze et al., “5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges,” J. ComNet., vol. 177, pp. 1-40, Feb. 2020, doi: 10.1016/j.comnet.2019.106984.

[6] C. Y. Chang, N. Nikaein, “Closing in on 5G control apps: enabling multiservice programmability in a disaggregated radio access network,” IEEE Veh. Technol. Mag., vol. 13, pp. 80-93, Sep. 2018.

[7] ECRI Final release of SONATA platform, ECRI SONATA D5.4 Standard, 2017.

[8] C. Parada et al., “5GTANGO: A Beyond-MANO Service platform,” in Proc. 2018 IEEE European Conf. on Networks and Communications, 2018, doi: 10.1109/EuCNC.2018.8443232.

[9] P. Twamley et al., “5GTANGO: An approach for testing NFV deployments,” in Proc. 2018 IEEE European Conf. on Networks and Communications, 2018, doi: 10.1109/EuCNC.2018.8442844.

[10] B. Sayadi et al., “SDN for 5G Mobile Networks: NORMA perspective,” in Proc. International Conf. on Cognitive Radio Oriented Wireless Networks, 2016, pp 751-753.

[11] 5G MoNArch Project Summary Report, 5G-MoNArch Standard 761445, 2019.

[12] G. Zhao et al., “Network Slice Selection in Softwarization based Mobile Networks,” in Proc. 2018 IEEE Global Communications Conf., 2019, doi: 10.1109/GLOCOM.2018.8647166.

[13] W. Guan et al., “A Service-oriented Deployment Policy of End-to-End Network Slicing Based on Complex Network Theory,” IEEE Access J., vol. 6, pp. 19691-19701, 2018.

[14] Z. Kotulski et al., “On end to end approach for slice isolation in 5G Networks. Fundamental challenges,” in Proc. 2017 Federated Conf. on Computer Science and Information Systems, 2017, doi: 10.15439/2017F228.

[15] A. Kammoun et al., “Admission Control Algorithm for Network Slicing Management in SDN-NFV Environment,” in Proc. 2018 IEEE 6th International Conf. on Multimedia Computing and Systems, 2018, doi: 10.1109/ICMCS.2018.8525945.

[16] M. Gragmalia et al., “Flexible Connectivity and QoE/QoS Management for 5G Networks: the 5G Norma View,” in Proc. 2016 IEEE International Conf. on Communications Workshops, 2016, doi: 10.1109/ICCW.2016.7503816.

[17] G. A. Carella et al., “A Network Function Virtualisation framework for Network Slicing of 5G Networks,” in Proc. 2017 22nd Information Technology Society in the VDE Conf., 2017, pp. 81-86.

[18] J. Adeel, “Understanding Dijkstra Algorithm,” SSRN Electronic Journal, 2013.

[19] J. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture," RFC7665, IETF, 2015, doi: 10.17487/RFC7665.

[20] ETSI Report on Network Functions Virtualisation Management and Orchestration Framework, ETSI Standard GR NFV-MAN 001, 2012.

[21] OpenDaylight Documentation Silicon documentation, OpenDayLight Documentation Guide, 2021. [Online]. Available: https://docs.opendaylight.org/en/stable-silicon/index.html.

[22] Mininet: An Instant Virtual Network on Your Laptop (or Other PC) – Mininet, Mininet Git Hub Documentation, 2021. [Online]. Available: https://github.com/mininet/mininet/wiki/Documentation.

[23] A. Botta et al., "D-ITG 2.8.1 Manual," COMICS, Oct. 2013.

[24] 3GPP System architecture for the 5G System (5GS), 3GPP Standard TS 23.501, 2021.

[25] 5G Network Slicing Self-Management White Paper, 5G Slicing Association, 2020.

AUTHORS

Poovendren Govender is an MEng student in the Electrical and Electronics Engineering Science Department at the University of Johannesburg, South Africa. He is a seasoned practicing telecommunications, IT and digital professional, member of the IEEE, senior member of the SAIEE and has a PrEng from ECSA with specialization in telecommunications. His research interests are in the interdisciplinary areas of mobile communications, cloud computing, AI, ML, data science and future network technologies.

Kingsley A. Ogudo is an associate professor/researcher at the University of Johannesburg's Department of Electrical and Electronic Engineering. He is a professional engineering technologist certified by ECSA and is a senior member of the IEEE Society. He is a senior member at SAIEE since 2011 and a fellow in 2021, and Secretary General for SAIEE Entrepreneur & Innovation chapter. His research interests are in the areas of electrical and electronics engineering communication, opto-electronics, photonics engineering and AI.

Chabalala Chabalala is a senior lecturer and researcher currently working at the School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg. His research interests are in the areas of wireless communications, cognitive radio networks and radio resource management.

