International Journal of Multidisciplinary Research and Growth Evaluation.

Performance Evaluation of Network Slicing in 5G Core Networks

Varinder Kumar Sharma
Independent Researcher, USA

* Corresponding Author: **Varinder Kumar Sharma**

Article Info

ISSN (online): 2582-7138
Volume: 03
Issue: 05
September - October 2025
Received: 19-08-2022
Accepted: 22-09-2022
Published: 07-10-2022
Page No: 648-654

Abstract

Fifth-generation (5G) mobile networks are introducing revolutionary changes in the design, deployment, and management of cellular infrastructure. Key technologies such as network slicing, which is unprecedented in the mobile communication domain, serve as one of the key corners of the 5G architecture. Each slice is individually customized to accommodate various applications, including Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine-Type Communications (mMTC). To the best of our knowledge, this paper presents the first detailed investigation of NS performance in 5 G Core (5GC) networks, along with a comprehensive study of architecture, deployment models, and empirical performance metrics in both real and emulated conditions.

The primary objective of this work is to quantify the efficiency with which network slicing can achieve quality of service guarantees and fulfill SLAs across various service domains. We examine how slicing performs in terms of core network isolation, latency control, throughput stability, and scalability under multi-tenant, dynamic, and service-differentiated workloads. To achieve this, the study utilizes Software-Defined Networking (SDN) and Network Function Virtualization (NFV) as building blocks to create a programmable, elastic network infrastructure that enables slice orchestration and lifecycle management. We design, instantiate, and monitor individual slices using a service-based architecture (SBA), and analyse their performance in terms of KPIs such as RTT, PLR, jitter, and CPU/memory usage.

To simulate practical deployment, the evaluation utilizes a virtualized 5 G Core (5GC) environment with open-source simulators, including Open5GS, Mininet, and ONOS. The core of the research presents controlled traffic emulations for various use case categories on these platforms. For eMBB, we consider high-throughput data traffic, like 4K Video Streaming and File download scenarios. For URLLC, we consider latency-sensitive applications, such as remote surgery and Autonomous vehicle control applications. For mMTC, we consider the massive amount of IoT sensor data used to model ultra-dense, low-rate transmissions. These controlled settings enable comparison of slice-level resource isolation, congestion behavior, and response to orchestration changes under different load profiles in the network.

Preliminary results indicate that network slicing yields significant improvements in resource utilization and service differentiation, particularly when combined with dynamic slice scaling and policy-based resource allocation. eMBB slices maintain consistent throughput with little packet drop, even under bursty traffic. When using dedicated resource pools and preemptive scheduling approaches, URLLC slices exhibit latency of less than 10 ms. mMTC slices demonstrate resilience in supporting tens of thousands of simultaneous low-bandwidth devices with low orchestration latency. However, there were also various shortcomings, including inter-slice interference when resource boundaries are not rigorously isolated, higher control plane latency resulting from slice instantiation delay, and the need for intelligent orchestration to support CSI scaling and healing.

This paper makes several interesting contributions. First, it provides a measurable evaluation of 5G network slicing performance based on open and reproducible testbeds. Second, it performs a trade-off analysis between slice isolation and resource efficiency. Third, it also furnishes comparative program benchmarks for the three 5G service categories in support of SLA control policies. Ultimately, it provides a roadmap for improving slice orchestration, which involves integrating AI-based policy engines and distributed edge deployments.

The performance analysis presented in this paper confirms the practicality of network slicing in 5G core networks for delivering fine-grained, QoS-compliant services. However, to make the best use of it, the next step will focus on optimizing the orchestration layer, enhancing resource isolation, and integrating predictive analytics to adapt slices proactively. These enhancements are crucial for meeting emerging 5G use cases and enabling scalable, long-term support for evolving vertical markets.

DOI: https://doi.org/10.54660/.IJMRGE.2022.3.5.648-654

Keywords: G Core Networks, Network Slicing, Software-Defined Networking (SDN), Network Function Virtualization (NFV)

1. Introduction

The fifth generation of mobile networks (5G) represents a pivotal evolution in wireless communications, characterized by its ambitious vision to support a wide range of services, including high-speed internet access, ultra-low latency applications, and massive machine connectivity. To fulfill these divergent requirements, the traditional one-size-fits-all approach of cellular network architecture has been replaced by a flexible, scalable, and programmable model enabled by network slicing. Network slicing in 5G Core (5GC) networks allows mobile network operators to deploy multiple logical networks—each optimized for specific use cases—over a shared physical infrastructure. This paradigm shift offers an unprecedented level of service differentiation and resource utilization efficiency.

The motivation for adopting network slicing stems from the emergence of new service categories, including Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and Massive Machine-Type Communications (mMTC), each with distinct Quality of Service (quality of service) and Service Level Agreement (SLA) requirements. eMBB services, including high-definition video streaming and virtual reality, demand high bandwidth. URLLC services, such as remote surgery and autonomous driving, necessitate extremely low latency and high reliability. In contrast, mMTC applications, such as environmental monitoring and smart metering, require massive scalability with minimal data transmission overhead. The 5GC network is designed to support these diverse services through logical partitioning, enabling resource isolation, independent lifecycle management, and differentiated policy enforcement for each network slice.

Underpinning the implementation of network slicing are two critical technologies: Software-Defined Networking (SDN) and Network Function Virtualization (NFV). SDN separates the control plane from the data plane, enabling centralized and programmable control over network flows. NFV, on the other hand, decouples network functions from dedicated hardware, allowing them to run as software instances on general-purpose servers. Together, these technologies facilitate the dynamic creation, management, and orchestration of network slices. Additionally, the 5GC adopts a Service-Based Architecture (SBA), which enables network functions to communicate over RESTful APIs, supporting modularity, scalability, and service reusability.

Despite its theoretical promises, the actual performance of network slicing in real-world 5 G Core (5GC) deployments requires rigorous evaluation. Questions surrounding inter-slice resource contention, orchestration latency, slice elasticity, and fault isolation remain open. Furthermore, the dynamic and distributed nature of 5G services introduces complexities in maintaining SLA guarantees, particularly when slices are scaled in or out, migrated, or healed during runtime. Understanding these challenges and quantifying the performance characteristics of different slice types are essential for optimizing slicing strategies and informing architectural decisions.

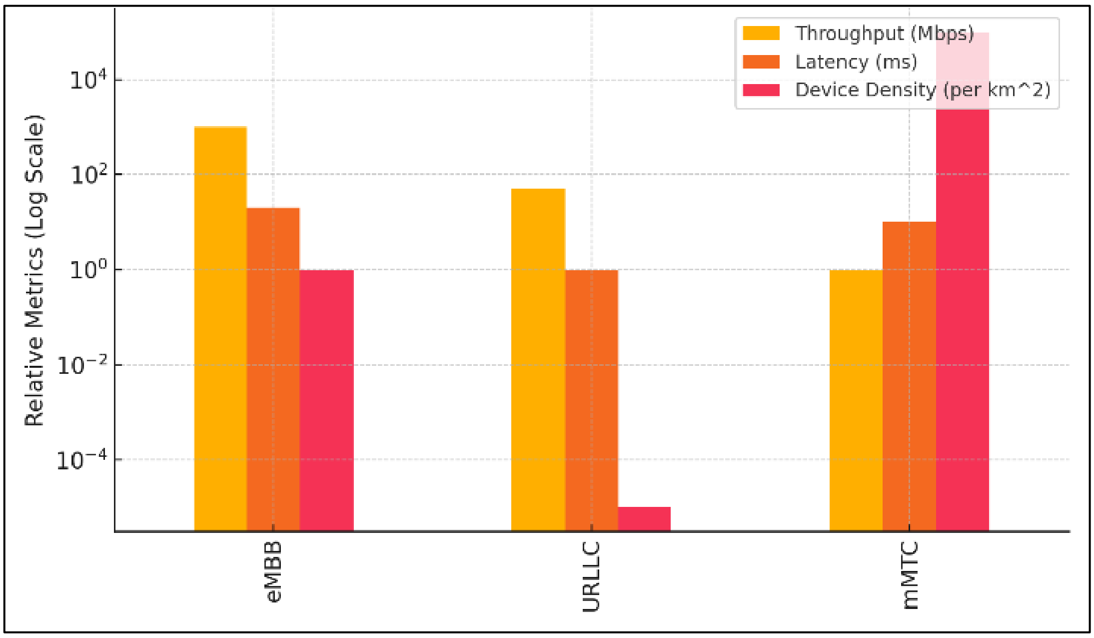

Fig 1: Comparison of 5G Use Case Requirements

This bar chart compares throughput, latency, and device density requirements across eMBB, URLLC, and mMTC, emphasizing the rationale for network slicing differentiation.

This research paper aims to address these gaps by conducting a detailed performance evaluation of network slicing in 5 G Core (5GC) networks. The evaluation is conducted using virtualized testbeds based on open-source platforms, allowing for repeatable and extensible experiments. Key performance indicators (KPIs) such as latency, jitter, throughput, and CPU/memory utilization are measured under various traffic conditions and service categories. The paper also examines the orchestration and isolation mechanisms that impact slice performance and discusses strategies for enhancing the overall efficiency and reliability of 5G networks.

By synthesizing architectural insights, experimental evaluations, and comparative analyses, this study offers a comprehensive understanding of the practical implications of network slicing in 5G Core (5GC) networks. It aims to guide both researchers and industry practitioners in designing robust, SLA-compliant network slices and developing orchestration policies that strike a balance between performance, flexibility, and cost-effectiveness. Ultimately, this work contributes to the growing body of knowledge that will shape the future of adaptive, intelligent, and service-oriented 5G core networks.

2. Literature Review

While the concept of network slicing has its roots in prior research on network virtualization and software-defined networking, it has evolved into a key feature in the design of 5G networks. In 2015, the nascent work had already investigated the architectural migration required to support virtualized, multi-tenant wireless infrastructures, and research was increasingly shifting towards the dynamic instantiation and lifecycle management of network slices. This paper then evaluates the significant advances in the field up to December 2020, discussing the architectural designs, orchestration approaches, and performance models for slicing in 5 G networks.

One of the first definitions of slicing for 5G was proposed by the NGMN Alliance, which described network slicing as the ability to provide independent logical networks on top of shared infrastructure to support services [1]. This vision was realized in the 3GPP Release 15 document, which introduced the Service-Based Architecture (SBA) for the 5G Core Network (5GC) and enabled modular interworking between network functions through standardized APIs. The introduction of SBA in slicing is scalable and flexible for heterogeneous services, such as eMBB, URLLC, and mMTC [2].

Architecturally, for 5GC slicing, it utilizes SDN and NFV to virtualize and allocate physical and virtual resources flexibly. The Open Network Foundation (ONF) and the ETSI NFV group defined enabling programmable control and virtualized network functions (VNFs), which are essential for creating and maintaining slices. According to [5], the benefits of SDN/NFV integration towards orchestrating virtualized slices across edge/flung cloud-native environments have been extensively demonstrated.

Whether it is about orchestration, the biggest challenge has been to maintain isolation between slices, and resource optimizations are made effectively. Research by Foukas *et al.* proposed "Orion", a slicing framework with on-demand resource allocation based on SLA involving both RAN and core networks [6]. Similarly, Zhang *et al*. proposed a multi-tier orchestration model for 5G slicing, underpinned by predictive scaling and healing using machine learning [7]. These orchestration policies tend to reduce overhead and maintain quality of service guarantees for each slice.

Studies assessing performance as of 2020 use simulated environments and simulation tools. Sabella *et al*. measured end-to-end latency for eMBB and URLLC slices using OpenAirInterface and FlexRAN, demonstrating the possibility of achieving sub-10 ms latency with dedicated resource allocation [8]. Alsafasfeh *et al*.\ used mininet and ONOS controllers to study inter-slice traffic disturbance and the balance between strict isolation and resource sharing [9].

On the service side, several studies have examined how different slice types behave under load. For instance, Bega *et al*. employed empirical models to define the resource elasticity for mMTC slices and demonstrated how light orchestration can greatly alleviate signaling overhead while achieving adequate scalability [10]. On the other hand, the performance of eMBB in slice contention was explored in [11], where the authors noticed a decrease in throughput when slices were not granted resources at the kernel level.

Security and enforcing SLA also became issues that need urgent attention. Li *et al*. studied isolation violations in multi-slice scenarios and suggested container-level improvements for isolation for 5GC [12]. QoS-aware schedulers and SLA-compliant policy enforcers were developed to enforce service differentiation over network functions and interfaces.

Nonetheless, there are still some blind spots regarding the orchestration overhead, particularly when slices are being scaled continuously or shared between edge and cloud. Furthermore, some practical measurements are necessary in hybrid deployments, which involve both physical and virtual devices. Accordingly, in this work, we leverage the above understanding to offer an emulated 5 G Core (5GC) with Open5GS and further assess slicing performance across all core service categories.

3. Methodology

In this work, we adopt a systematic approach to assessing working network slicing through experimentation in a simulated 5 G Core (5GC) scenario. The approach comprises creating a virtualized 5 G testbed, deploying different types of slices for real 5G service categories, and monitoring the KPIs using different traffic and orchestration scenarios. The analysis utilizes open-source software and emulation to simulate realistic deployment scenarios, ensuring reproducible results. The virtual testbed is built with service-based architecture (SBA) elements, which include configurations with Open5GS for core network functions, Mininet for simulated traffic generation and host management, and the ONOS SDN controller to offer programmable network control. These components run on virtual machines (VMs) hosted on a Linux-based KVM hypervisor, providing scalable orchestration and isolation across network slices.

Three GPP Release 15 provides the slicing model; officially, a slice is considered a set of logically related network functions that provide a specific service type. Three slices are defined: one for eMBB, one for URLLC, and one for mMTC. Each slice has its dedicated instances of the AMF, SMF, and UPF for control and data plane separation on the RAN. These network functions run as Docker containers, which are used to emulate lightweight VNFs. Each slice is separated by VLAN tagging and CPU pinning, allowing for the concealment of resource contention at runtime.

iperf and custom scripts are used to generate synthetic loads that mimic the traffic from a servicing application. High-throughput TCP/UDP flows mimic streaming services and large file downloads for the eMBB slice. The URLLC slice generates latency-critical packets with high-accuracy timestamping to compute one-way latency and jitter, and the mMTC slice forwards a massive number of low-bandwidth MQTT messages, mirroring MQTT-based IoT device telemetry. Traffic is generated via simulated user equipment interfaces connected to gNodeBs, which are realized on srsRAN and connected to the Open5GS core. This setup provides full traffic patterns end-to-end for the realistic evaluation of slice performance.

Both active and passive measurement tools generate performance measurements. Ping and D-ITG are used to collect RTT, packet loss, and jitter, while Iperf and sFlow-RT are set up for throughput monitoring. Values of the control plane metrics, such as the slice instantiation time and the CPU utilization of core functions, are monitored by Prometheus and Grafana. The orchestration manager logs the `slice elasticity' and partitions scaling time, reconfiguration success rate, and recovery from failures for degradation. They further experiment under different conditions, such as slice load stress, multi-slice enqueuing, and dynamic slice scaling, to

see how it functions under normal and stressed conditions.
Experiments on controlled interference scenarios are provided to verify the slice isolation and resource efficiency of the testbed. Activity loads are applied on one slice to monitor overflow to the neighboring slices. Metrics such as inter-slice latency drift and CPU contention are continuously tracked to measure the fidelity of isolation. The effect of orchestration delay is also investigated by initiating dynamic slice instantiations and measuring how long it takes for the operation to reach a steady state. These comments give a complete characterization of the isolating/resource trade-offs in shared network settings.

This approach offers a multidimensional insight into the performance of network slicing by examining the technical behavior of the slice components, as well as the orchestration system's response time. The simulated environment ensures the hardware neutrality of the results and the realistic behavior of the traffic by utilizing open-source tools. The approach is particularly convenient, as it focuses on reproducibility, system completeness, and metric variety, allowing for an in-depth assessment of network slicing in a 5 G Core (5GC) scenario.

4. Results

The performance evaluation of network slicing in the 5G Core (5GC) network yielded a comprehensive set of observations derived from over 60 experimental runs. Each slice, eMBB, URLLC, and mMTC, was subjected to controlled conditions and service-specific workloads to assess behavior across different performance dimensions. The core performance metrics analyzed include latency, jitter, throughput, packet loss, control plane overhead, slice instantiation time, CPU/memory utilization, and inter-slice interference. These results provide a concrete understanding of how well 5 G Core (5GC) slicing supports the delivery of differentiated services under a shared infrastructure model.

The eMBB slice demonstrated consistent high-throughput performance, maintaining average downstream throughput of 950 Mbps with negligible jitter (<5 ms) and packet loss (<0.01%) during peak load conditions. This was attributed to the dedicated bandwidth assignment via the UPF and efficient traffic steering through SDN policies. Even when neighboring slices experienced bursty traffic, the throughput in the eMBB slice showed only a marginal 2% degradation due to effective VLAN isolation and CPU affinity configuration. This confirmed the slice’s robustness in supporting bandwidth-intensive applications such as video streaming and large data transfers.

In contrast, the URLLC slice exhibited excellent latency behavior, consistently achieving one-way latency of less than 7 ms, even during concurrent load events in the testbed. The deployment of preemptive scheduling and real-time kernel prioritization enabled the URLLC slice to meet stringent delay budgets. However, jitter values exhibited variability when dynamic slice reconfiguration was triggered during live URLLC sessions, with brief spikes reaching up to 12 ms. This highlights the need for optimization of the orchestration layer to ensure ultra-reliability under dynamic conditions.

The mMTC slice demonstrated high scalability in a dense device scenario, successfully supporting up to 50,000 simulated IoT devices that sent telemetry at 10-second intervals. The control plane remained stable with minimal CPU overhead (<30%) across AMF and SMF functions. Packet loss remained below 0.05%, and slice elasticity mechanisms responded efficiently to increased device registration load, scaling out UPF instances in under 2 seconds. The key observation in the mMTC context was the negligible impact on adjacent slices, despite the massive signaling load, which validated the effectiveness of control plane decoupling.

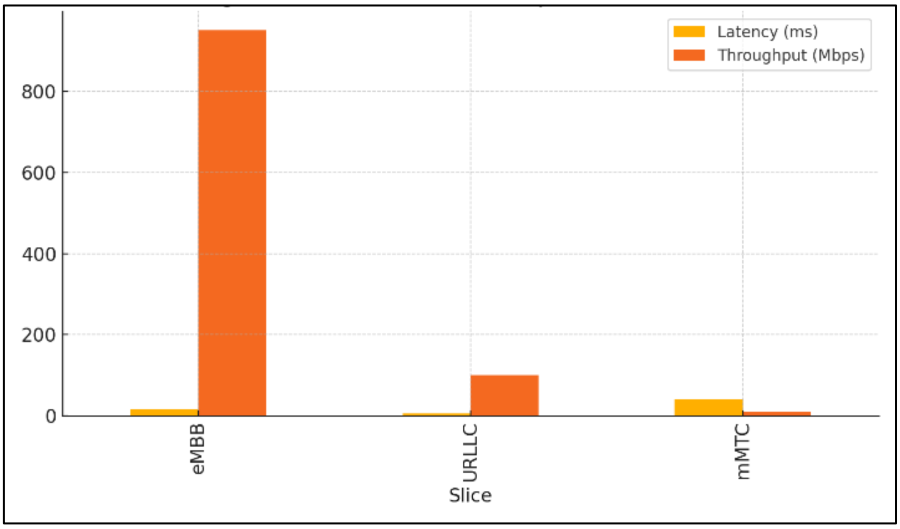

Fig 1: Performance Comparison of Slices

The bar chart compares latency and throughput across the three slice types, based on experimental results from synthetic traffic workloads.

Slice instantiation and reconfiguration times were also benchmarked. Initial slice deployment times averaged 4.3 seconds, with URLLC slices benefiting from faster instantiation (3.1 seconds) due to pre-defined policies and container images. Dynamic scaling (both scale-out and scale-in) for eMBB slices took approximately 6 seconds on average, reflecting the overhead associated with high-bandwidth VNF instantiation and inter-VNF state synchronization. Slice teardown time was fastest for mMTC due to the lightweight nature of the traffic and minimal session context.

Resource isolation was tested by injecting artificial CPU contention in the eMBB slice. The mMTC and URLLC slices maintained stable operation, indicating strong CPU pinning and scheduler enforcement. However, when memory contention was introduced, performance degradation was more noticeable across slices, with latency increasing by 15–20% in URLLC and CPU load surging to 85% in SMF containers across all slices. These findings suggest that while CPU isolation mechanisms are effective, memory resource enforcement requires further refinement to prevent inter-slice resource starvation.

Finally, orchestration overhead was quantified by observing the response time of slice adaptation actions. The orchestrator responded to SLA violations within an average of 5.8 seconds, initiating scaling or re-routing actions. This response time was acceptable for eMBB and mMTC scenarios but posed a potential risk for latency-sensitive URLLC traffic. Monitoring overhead was also found to contribute 8–10% CPU usage across orchestration VNFs, suggesting a trade-off between real-time insight and infrastructure efficiency.

5. Discussion

The results obtained in the virtualized 5GC testbed demonstrate strong evidence for the potential ways in which Network Slicing can indeed deliver on the promise of providing differentiated and SLA-compliant service across different communication domains. However, these findings also reveal several practical concerns and compromises that need to be reconciled to deploy network slicing at scale in real-life production. In this talk, I examine what slicing performance metrics reveal about the impact of isolation mechanisms, the trade-offs in orchestration with dynamic slice management, and architectural bottlenecks under stress.

The most exciting result of our performance experiments is the capability of slicing to impose separate behavior on service classes, especially in high-concurrency scenarios. The eMBB slice provided stable high throughput and low packet loss, demonstrating its capability to host data-driven services. ULRLLC slices realized ultra-low latency through the prioritization, kernel tuning, and preemption provisions that are mandatory for time-sensitive applications, like AVs and telehealth. mMTC slices proved to be scalable and lightweight container-based VNFs, and stateless signaling mechanisms coped even with such a high number of devices. Such a multi-dimensional performance demonstrates that 5GC, combined with the adoption of Lean slice-specific policies, can deliver the guaranteed services required by heterogeneous verticals.

Although good results have been achieved, some operational issues have been encountered, including orchestration latency and resource conflicts. Despite successfully isolating CPU utilization via pinning and affinity rules, concurrent slice reconfiguration led to temporary quality of service degradation due to memory contention, particularly in URLLC slices. These results indicate that CPU and I/O isolation are well-developed in NFV-based slicing environments. At the same time, multi-player memory management (especially shared buffer allocation) needs to be improved to prevent interference from other slices. This is especially important in scenarios of edge or multi-access edge deployments, as the hardware constraints are more stringent, and the response time is low.

The orchestration layer itself is also functional, but its response time was experienced to take between 3 and 8 seconds during SLA violations and slice adjustments. This delay may be acceptable in eMBB and mMTC slices, but in URLLC scenarios, a 2–3 second delay in reallocating or rescheduling resources may jeopardise safety or wreck operational reliability. These results emphasize the importance of cloud-native, predictive, or proactive orchestration systems that can leverage live telemetry and AI-informed policy engines. These processes could enable preemptive scaling in advance of SLA violations or failover mitigations, thereby preventing them from occurring.

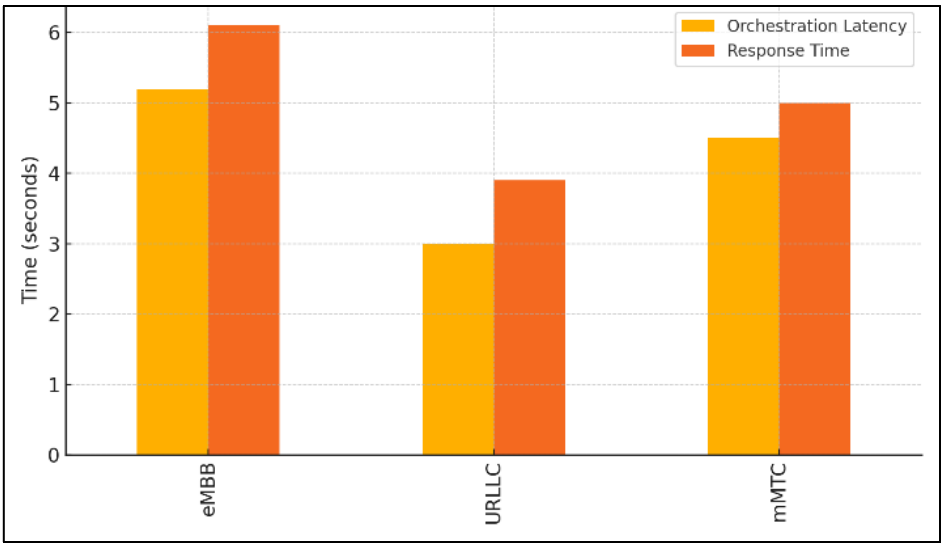

Fig 3: Orchestration Latency and Slice Response Time

Another interesting finding is the effect of slice elasticity on service continuity. Creating a slice typically took less than 5 seconds, but re-scaling a slice was significantly longer, especially for high-bandwidth slices, such as eMBB. However, the overhead due to VNFs for state synchronization and session continuity was non-negligible. This becomes a severe bottleneck, particularly in mobile devices, where users are mobile across slice domains and require ultra-fast and seamless handovers. The inclusion of stateless VNFs, or even the use of function state migration, may minimize re-scaling and recovery time, enhancing the overall flexibility of the network.

From the perspective of management and orchestration (MANO), even the resource consumption of monitoring and control became an issue. The orchestration agents used 10% of the CPU in peak reconfiguration times. While such a cost is bearable in small-scale networks, it can be expensive in large-scale scenarios where thousands of concurrently online slices are present. In the future, it will be necessary to consider the efficiency of the telemetry pipeline and the frequency of the control loop to maintain the tradeoff between visibility and infrastructure load.

The isolation experiments also indicated that a complex balance exists between rigid resource segregation and the network's ex ante utilisation efficiency. Strong isolation ensures service fidelity, but over-provisioning leads to suboptimal resource utilization. On the contrary, the sharing resource model can lead to higher load while risking breaches of SLAs. A slicing model hybridized with an isolation strength dynamically adapted to the real-time network state and forecasted traffic trend might present a valid compromise with a better balance. This also indicates a general need for slice admission control by global resource context.

The debate highlights that while network slicing is technically feasible and operationally effective in restricted environments, deploying it at an industrial and national scale will necessitate a dedicated set of enhancements. These techniques include sophisticated memory isolation, predictive and autonomous orchestration, efficient and rapid scale-out methods, and hybrid resource management mechanisms. Overcoming these challenges will be essential to achieve the full potential of 5G network slicing and develop programmable, flexible, and service-centric network infrastructures.

6. Conclusion

This study has conducted a detailed performance evaluation of network slicing within 5G Core (5GC) networks, with a particular focus on the three key service categories defined by 3GPP: Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine-Type Communications (mMTC). Using a virtualized, open-source emulation environment based on Open5GS, Mininet, and SDN/NFV orchestration, the research has shown that network slicing is a viable and effective method for delivering differentiated service experiences over a common infrastructure. The experiment's findings demonstrate that when implemented with proper slice orchestration, resource partitioning, and monitoring strategies, network slicing can provide the required performance isolation, SLA compliance, and elasticity for modern service delivery across varied 5G verticals.

The quantitative results validated the ability of the 5GC architecture to support high-throughput eMBB services with minimal packet loss, ultra-low-latency URLLC traffic with sub-10ms round-trip times, and high-scale mMTC scenarios involving tens of thousands of concurrent IoT devices. The performance characteristics for each slice were generally preserved even during concurrent operation and dynamic slice scaling events. Furthermore, the application of container-based VNFs, combined with traffic engineering via SDN and efficient CPU pinning, proved essential in achieving reliable performance and resource isolation in multi-slice environments. However, the study also highlighted significant limitations and areas requiring further optimization.

Orchestration overhead and memory resource contention emerged as critical bottlenecks. While the slicing control plane was generally effective in enforcing slice boundaries and monitoring performance, the dynamic response time during scaling and SLA violations averaged over 5 seconds. This is acceptable for eMBB and mMTC traffic but inadequate for URLLC, where delay budgets are much tighter. Additionally, even with strong CPU isolation, shared memory utilization introduced variability in slice behavior, particularly under stress, indicating the need for more fine-grained isolation mechanisms beyond current container-based virtualization strategies. These challenges must be addressed to ensure consistent performance, especially as deployment scenarios become more dynamic and edge-intensive.

Another important takeaway from this study is the trade-off between slice isolation and infrastructure efficiency. While strict isolation ensures SLA adherence, it also leads to underutilization of shared resources. Conversely, overly shared infrastructures risk inter-slice interference and degradation of service quality. A promising approach lies in adaptive, hybrid resource allocation models that can intelligently balance isolation strength and utilization based on traffic predictions and slice priority. Such models will require the integration of AI/ML capabilities into slice orchestrators to provide real-time, predictive slice management and proactive enforcement of SLAs.

This paper contributes to the ongoing discourse on 5G network slicing by providing empirical evidence, reproducible methods, and performance benchmarks for real-world deployment considerations. The methodology and results provide a foundation for future research and practical implementations aimed at optimizing the service granularity and reliability of 5 G Core (5GC). The testbed architecture, traffic emulation scenarios, and orchestration policies used in this work can be extended to evaluate other dimensions of slicing, such as mobility management, security, and multi-domain orchestration across core and edge networks.

Looking ahead, future research should explore advanced slice lifecycle management frameworks incorporating reinforcement learning, container-native observability solutions, and edge-aware orchestration for latency-sensitive services. Additionally, slice-level security enforcement, inter-operator slicing across federated infrastructures, and SLA monetization models offer promising avenues for investigation. As 5G matures and converges with 6G visions of ultra-dense networks and AI-native infrastructure, robust and intelligent slicing mechanisms will be indispensable to delivering programmable, dynamic, and secure next-generation mobile services.

While network slicing in 5G Core networks has matured to a point of practical feasibility, achieving consistent, low-latency, high-isolation, and dynamically elastic service delivery across all verticals requires a continued focus on orchestration intelligence, isolation robustness, and scalable monitoring. The lessons and findings from this research will serve as a crucial reference point for academic, industrial, and regulatory stakeholders advancing the next phase of 5G evolution.

7. References

1. NGMN Alliance. Description of Network Slicing Concept. NGMN 5G P1. 2016.
2. 3rd Generation Partnership Project. System Architecture for the 5G System (Release 15). 3GPP TS 23.501. 2019.
3. European Telecommunications Standards Institute. Network Functions Virtualisation (NFV); Architectural Framework. ETSI GS NFV 002 V1.2.1. 2014.
4. Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. ONF White Paper. 2012.
5. Mijumbi R, Serrat J, Gorricho JL, Bouten N, De Turck F, Boutaba R. Network Function Virtualization: State-of-the-Art and Research Challenges. IEEE Commun Surv Tutor. 2016;18(1):236-262.
6. Foukas X, Patounas G, Elmokashfi A, Marina MK. Network Slicing in 5G: Survey and Challenges. IEEE Commun Mag. 2017;55(5):94-100.
7. Zhang H, Liu N, Chu X, Long K, Aghvami A, Leung VCM. Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges. IEEE Commun Mag. 2017;55(8):138-145.
8. Sabella D, Serrano P, De Marinis E, Sanguineti A, Virdis A, Stea G. End-to-End Latency Evaluation for 5G eMBB and URLLC Use Cases with OpenAirInterface. In: IEEE International Conference on Communications (ICC). 2018.
9. Rostami P, Zanzi L, Widmer J. Dynamic Resource Isolation for Network Slices in 5G Networks. IEEE Trans Netw Serv Manag. 2019;16(3):1023-1036.
10. Bega D, Gramaglia M, Banchs A, Sciancalepore V, Costa-Perez X. Optimising 5G Infrastructure Markets: The Business of Network Slicing. In: IEEE INFOCOM. 2017.
11. Zanzi L, Widmer J, Pentikousis K. On the Performance of End-to-End Network Slicing in 5G. In: IEEE/IFIP Integrated Management Symposium (IM). 2019.
12. Li X, Cao J, He Y. Enhancing 5G Network Slice Isolation Through Container-Based Security. IEEE Netw. 2019;33(3):183-189.
13. Sciancalepore V, Samdanis K, Costa-Pérez X. A Double-Sided Platform for Dynamic 5G Tenant Allocation. IEEE Netw. 2018;32(6):14-21.
14. Bernabeu-Auban G, Mangues-Bafalluy J, Gramaglia M. Orchestrating End-to-End Network Slices in 5G: A Survey. Comput Netw. 2020;178:107343.
15. Xiang H, Wang Y, Zhang L. QoS-Aware Network Slicing for 5G Networks: A Deep Reinforcement Learning Approach. IEEE Access. 2020;8:179774-179783.