Queueing-Based Performance Analysis of 5G Service Function Chains

Mario Di Mauro†, Raffaele Peluso*
†*University of Salerno, Italy*, mdimauro@unisa.it
**Cloud9 Reply, Italy*, ra.peluso@reply.it

***Abstract*—In this work we propose a queueing-based framework for evaluating the performance of 5G Service Function Chains (SFCs), focusing on the impact of delays at virtualized network nodes on end-to-end service delivery. Our approach employs an $M/G/k$ queueing model to characterize the delays in control and data plane nodes. Additionally, we introduce a greedy optimization algorithm, *OptInst*, to determine the minimum number of instances (e.g., containers or processes) to be deployed on 5G nodes to meet performance constraints. Using a realistic testbed based on Open5GS and UERANSIM platforms, we estimate the service times of the nodes and identify the optimal SFC deployment that minimizes resource consumption while fulfilling delay constraints. Our findings demonstrate the effectiveness of the proposed model in optimizing 5G network performance and offer insights into balancing delay requirements with resource efficiency.**

***Index Terms*—SFC performance analysis, queueing models, delay evaluation, Open5GS.**

I. Introduction

In the context of network virtualization, Service Function Chains (SFCs) play a pivotal role in orchestrating network services. In an SFC, network traffic passes through multiple virtualized network functions sequentially, each performing a specific operation.

One of the most critical performance aspects for SFCs is delay, a factor that significantly influences the end-to-end performance and quality of service (QoS) of 5G networks [3]. As traffic traverses multiple nodes in the chain, delays accumulate, potentially leading to unacceptable latencies. In softwarized infrastructures, the latency introduced at each node can vary depending on the resources allocated, the node's processing capabilities, and the nature of the traffic. These delays can be exacerbated by network congestion, high traffic volumes, or resource contention, all of which are more pronounced in virtualized environments. Ensuring that the delay across the entire SFC stays within acceptable limits while maintaining efficient resource utilization and sustainability is a challenging optimization problem that must be addressed to meet the stringent performance requirements of 5G networks. Accordingly, it is essential to model and assess the performance of SFCs from a delay-centric perspective. In particular, ensuring that each individual node in the chain meets specific latency constraints and that the accumulated delay does not exceed predefined thresholds is crucial to maintaining a high-quality user experience [4].

To address these concerns, this paper proposes a queueing-based framework to evaluate the performance of 5G SFCs. By modeling delays across interconnected nodes, we aim to assess the overall system's ability to meet performance requirements. This analysis is useful for understanding the interplay between individual node delays and end-to-end service performance, ensuring that 5G infrastructures achieve the necessary responsiveness.

The main contributions of this paper are as follows:

- We employ a realistic $M/G/k$ queueing-based model to characterize the performance of the nodes involved in a 5G-based SFC. The goal is to evaluate the overall SFC performance in terms of delay, distinguishing between control and data traffic.
- We develop a greedy optimization routine (*OptInst*) to estimate the minimum number of instances (e.g., containers or processes) required at each node to achieve energy savings while meeting specific performance requirements.
- Using a realistic testbed based on the Open5GS [1] and UERANSIM [2] platforms, we: i) estimate the mean service times of the involved nodes, and ii) determine the optimal SFC deployment that fulfills delay constraints with the minimum number of elements.

II. Related Work

To analyze performance challenges in SFCs and other virtualized infrastructures, researchers and practitioners exploit a range of formal approaches. Among the most prominent and effective methods are optimization frameworks and queueing theory. Optimization frameworks are frequently utilized for performance assessments, particularly in the context of traffic engineering and resource tuning and/or allocation in virtualized environments. For instance, in [5], the authors define optimization problems to model the performance-resource relationship, capturing the interplay between Virtual Network Functions (VNFs) throughput and latency. In [6], a mixed integer linear programming (MILP) approach is proposed to map VNFs onto physical resources, focusing on meeting SFC latency requirements. Authors in [7] propose a mixed integer second-order cone programming (MISOCP) formulation to ensure compliance with end-to-end SFC latency constraints, and a multi-objective linear programming (MOLP) approach is presented in [8], aiming to reduce application service latencies and minimize VNF migration costs within an SFC. An optimization algorithm based on the breadth-first search has been devised in [9], to find the shortest path between the source node and the destination node of an SFC aimed at

978-3-903176-75-1 ©2025 IFIP

estimating the end-to-end delay. Respecting delay and resource constraints, work in [10] proposes an optimization problem that simultaneously seeks to maximize the SFC deployment's benefit-to-cost ratio and to minimize its energy consumption.

One drawback of optimization models is that they typically demand extensive information about network conditions and can be computationally intensive.

Conversely, queueing theory is recognized as a highly effective method for addressing performance challenges in virtualized networks, particularly when characterizing latency in SFCs. Compared to optimization-based approaches, queueing models are often more computationally efficient and analytically tractable, frequently yielding closed-form solutions or practical approximations that simplify performance predictions for SFCs. We find many examples in the literature where performance problems are tackled with the queueing formalism.

The study in [11] addresses a resource allocation problem in 5G networks, where each Virtual Network Function (VNF) follows a standard $M/M/1$ queueing model. We recall that that $M/M/1$ means: exponentially distributed inter-arrival times (the first M), exponentially distributed service times (the second M), and the presence of 1 server. Authors in [12] propose a method to evaluate the resilience of 5G network services, focusing on the detection of traffic variations, with the classic assumption of exponentially distributed arrival and service times. In [13], the $M/M/1$ queueing model is utilized to detect bottlenecks in SFCs by analyzing network queue occupancy. $M/M/1$ and $M/M/m$ queueing models have been employed to characterize SFC scheduling [14], to improve the reliability of SFCs in 5G services [15], and to evaluate the performance of a containerized IP Multimedia Subsystem (IMS) architecture [16].

However, all the studies previously discussed use queueing models that provide precise solutions through closed-form expressions. A key drawback of these models is that they often depend on assumptions, such as exponentially distributed service times, which may not fully reflect the dynamics of real-world systems. To address this limitation, we use a more flexible $M/G/k$ queueing model to represent the latency introduced by a node with multiple instances, and apply Cosmetatos approximation formula to resolve the issue of the non-closed-form solution. Moreover, this work extends [17], which focused solely on modelling control-plane traffic and did not include the detailed data-traffic analysis presented here.

III. The Open5GS-based testbed

The 5G testbed developed in this study consists of two primary components. The first one is UERANSIM [2], an open-source simulator for the 5G radio access network including the gNodeB (gNB) and the User Equipment (UE). The second module leverages Open5GS [1] to implement the core network functions including:

- **Access and Mobility Management Function (AMF)**: it oversees UE registration, mobility management, and ensures device reachability to maintain seamless availability.

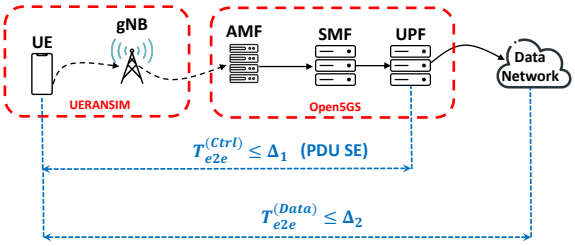

Fig. 1: The deployed testbed with UERANSIM and Open5GS, along with the constraints on the control plane (PDU SE) and on the data plane.

- **Session Management Function (SMF)**: it handles the setup, modification, and deletion of Protocol Data Unit (PDU) sessions, enabling end-to-end connectivity.
- **User Plane Function (UPF)**: it acts as a data router, optimizing packet transmission paths based on network status, QoS demands, and service priorities.

The nodes are logically traversed in a sequential manner (see figure 1), forming an SFC. In particular, we identify two *logical* SFCs responsible for two separate operational stages. The first is the *control plane* SFC, primarily involving gNB and AMF, which oversees the PDU Session Establishment (PDU SE) stage, allowing to create a data path for a UE (managing IP assignment, data tunnels, etc). The time to complete a PDU SE determines how fast a UE can join and become active in the 5G network, directly impacting user experience and service availability. We denote this time by $T_{e2e}^{(Ctrl)}$ which, according to European Telecommunications Standards Institute (ETSI), it must remain below a given threshold (the so-called T_{3580} timer that should be around 5 seconds [18]) denoted by Δ_1.

The second SFC is the *data plane* SFC, primarily involving gNB and UPF to handle data sessions. In this case, the 5G standard specifies end-to-end latency requirements tailored to different use cases to ensure performance objectives are met. Accordingly, we enforce the constraint $T_{e2e}^{(Data)} \leq \Delta_2$, where Δ_2 represents the Round-Trip Time (RTT). This value can vary significantly, depending on factors such as traffic type, external network conditions, and the codec used for multimedia sessions [20]. Specifically, in line with the latency budgets in Table 5.7.4-1 of 3GPP Technical Specification no. 23 501 [19], we set $\Delta_1 = 500$ ms and $\Delta_2 = 1$ ms.

IV. The Queueing-based approach and the OptInst Algorithm

In telecommunications systems like Open5GS, incoming requests are handled sequentially as they pass through the network nodes, each one with its distinct characteristics and service time. From a modeling perspective, we need to characterize each node involved in the SFC as a single queueing system. Specifically, we model each node as an *M/G/k* system. In practice, we assume that each node can run k homogeneous container or process instances —referred to as servers in classic queueing terminology— that concurrently

handle incoming 5G requests. In this approach, requests arrive as a Poisson process (*M*), service times follow a general distribution (*G*), and a limited number of servers (*k*) manage the workload. Notably, the adopted formalism scales gracefully to longer VNF chains: extra stages can be modeled simply by appending additional *M/G/k* systems in series. Consequently, the number of equations—and thus the computational burden—grows only linearly with the chain length, so the analysis remains tractable. It is useful to recall that $M/G/k$ queueing models do not admit closed-form solutions; thus, we need to employ some empirical approximations. Our approach involves three main steps: i) Collect and analyze node logs to estimate the mean service times; ii) Evaluate the expected value $\mathbb{E}(\cdot)$ and the standard deviation $\sigma(\cdot)$ of service times, to derive the coefficient of variation $CV = \sigma(\cdot)/\mathbb{E}(\cdot)$; iii) Evaluate performance metrics by applying the Cosmetatos approximation formula.

Aimed at designing a proper $M/G/k$ system, we begin by analyzing the "equivalent" $M/M/k$ queueing model employing classic queueing formulas. We then reconnect to the desired $M/G/k$ model through the application of Cosmetatos approximation [21] which, in case of medium/heavy-traffic condition and for a limited number of servers (typically, $k \leq 10$), it provides an excellent approximation [22], [23].

With reference to the equivalent $M/M/k_n$, let $1/\mu_n$ be the mean service time of node n, k_n the number of servers at the node n, and $\rho_n = \lambda_n/(k_n\mu_n)$ be the utilization factor at node n, where the condition $\rho_n < 1$ needs to be fulfilled to guarantee the model stability. Applying well-known formulas [24], the average queueing time at the generic node n is:

$$\mathbb{E}[Q_n]_{M/M/k_n} = \frac{\rho_n}{\lambda_n(1-\rho_n)} \cdot P_n, \tag{1}$$

where P_n represents the steady-state probability in the $M/M/k_n$ queueing model. Applying the Cosmetatos approximation to (1), it is straightforward to obtain an average queueing time for the $M/G/k_n$ model:

$$\mathbb{E}[Q_n]_{M/G/k_n} \approx \mathbb{E}[Q_n]_{M/M/k_n} \left(CV_n^2 + \frac{(1-CV_n^2)}{2\alpha_n} \right), \tag{2}$$

where:

$$\alpha_n = \frac{1}{1 + (1-\rho_n)(k_n-1)\frac{\sqrt{4+5k_n}-2}{16\rho_n k_n}}, \tag{3}$$

is the Cosmetatos approximation factor. Accordingly, the average response time to process a 5G request at node n is:

$$\mathbb{E}[T_n] = \frac{1}{\mu_n} + \mathbb{E}[Q_n], \tag{4}$$

where $\mathbb{E}[Q_n]$ corresponds to (2), and for simplicity, we omitted the subscript $M/G/k_n$. Finally, the end-to-end average time spent across the whole 5G SFC (T_{e2e}) can be simply interpreted as the sum of average times spent by request at each node, namely

$$T_{e2e} \approx \sum_{n=1}^{N} \mathbb{E}[T_n]. \tag{5}$$

Thus, we can use (5) to characterize performance of both control and data SFCs. In the former case, we need to estimate the mean service times of gNB and AMF during the control stage. In the latter case, we need to estimate the mean service times of gNB and UPF during the data transmission stage. It is important to notice that as the number of servers at node n increases, $\mathbb{E}[Q_n]$ decreases, leading to a reduction in $\mathbb{E}[T_n]$ and, consequently, in T_{e2e}. Therefore, our goal is to minimize the total number of instances (servers) deployed across 5G nodes while ensuring that both time constraints on control and data SFCs are fulfilled. This can be formulated as the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \sum_{n=1}^{N} k_n \\ &\text{subject to} \quad \begin{cases} k_n \geq k_{n0}, \quad k_n \in \mathbb{N}, \\ \\ T_{e2e} \leq \Delta_i, \quad i \in 1, 2. \end{cases} \end{aligned} \tag{6}$$

The first constraint in (6) is included to ensure queue stability ($\rho_n < 1$), where $k_{n0} = \lfloor \lambda_n/\mu_n \rfloor$ +1, with $\lfloor \cdot \rfloor$ denoting the integer floor operation. The second constraint ensures that T_{e2e} satisfies control and data conditions.

Algorithm 1: OptInst

Input: λ_n, μ_n, CV_n, Δ_i (delay thresholds)
Output: Optimal k_n, T_{e2e}

```
// Initialize server counts for all nodes
k_n ← 1
while True do
    for n ∈ {1, 2, ..., N} do
        ρ_n ← λ_n/(μ_n · k_n) // Utilization
        if ρ_n ≥ 1 then
            return ∞ // System is unstable
        end
        Calculate E[T_n] from (4) and T_e2e from (5)
    end
    if T_e2e ≤ Δ_i then
        return (k_n, T_e2e)
    end
    // Find node with highest delay and
       increment its servers
    n_max ← arg max_n E[T_n]
    k_{n_max} ← k_{n_max} + 1
end
```

We tackle the optimization problem in (6) with *OptInst*, a greedy heuristic inspired by Work-in-Process control in manufacturing. Because the server-placement problem generalizes the classic *Knapsack* problem and is therefore NP-hard [25], an exhaustive search quickly becomes impractical; *OptInst* delivers near-optimal solutions within polynomial time by incrementally eliminating the dominant bottleneck. The procedure starts with a single instance (server) per node—gNB/AMF for the control-plane chain and gNB/UPF for the data-plane chain—and uses the queueing model to evaluate the end-to-end delay. If the latency bound is violated, *OptInst* adds

one instance to the node contributing the largest share of delay and recomputes the performance; the loop repeats until the constraint is satisfied. Because control- and data-plane parameters differ, the algorithm runs independently on the two logical chains. By provisioning only what is strictly necessary, *OptInst* avoids over-allocation while keeping the computational burden low.

Finally, we note that the provided algorithm is computationally efficient. In the worst case, the number of iterations of the *while loop* (lines 2 – 15) is proportional to the required increase in the server count to achieve the target delay. This results in $O(k_{n_{max}})$ iterations, where $k_{n_{max}}$ is the maximum number of server increments needed across any node to meet the delay threshold. Conversely, the inner *for loop* (lines 3 – 9) iterates over all N nodes, performing constant-time operations. Therefore, the overall complexity is $O(k_{n_{max}} \cdot N)$.

V. Experimental Results

This section is divided into two parts. The first part describes the experiments conducted to estimate the service times of the 5G nodes, based on an analysis of internal node logs and Wireshark traces. The second part presents a numerical evaluation of our framework, using the parameters estimated in the previous stage. Specifically, we derive the optimal configurations for the 5G nodes to meet the desired performance requirements. Before describing the experiments, we provide some details about the implemented testbed. Specifically, we used two identical virtual machines (VMs) hosting UERANSIM and Open5GS, with the following characteristics: AMD Ryzen 7 4700U CPU, 2 GB RAM, and Ubuntu 20.04.6 LTS OS. To test network stability, particularly delay and jitter, we initially conducted repeated tests using ICMP traffic (ping command), which serves as an excellent diagnostic tool. The ping test was executed by the user equipment (UE) to a server hosted on another PC, which was connected to the testbed via a standard switch. Figure 2 shows the cumulative distribution functions (CDFs) of delay (in blue) and jitter (in red) associated with the ICMP test. Analyzing the results, we observe that 90% of the packets exhibit a delay value below 3.5 ms and a jitter value below 0.98 ms. These results suggest that both delay and jitter are low, indicating excellent 5G network performance with minimal variability.

A. Service Times estimation

To estimate service times of nodes involved in the control plane SFC (PDU SE management), we analyze and parse the log files of gNB and AMF nodes. Figure 3 (top) shows an example of gNB log file, where only crucial messages are reported. Although the gNB may not be busy for the entire duration of this phase, its service time can be reasonably estimated as the temporal difference between line **#1** (gNB initiates the RRC Setup procedure for UE with the configuration of radio bearers) and line **#4** (gNB confirms that the PDU session resources have been successfully set up for UE).

Figure 3 (bottom) shows an example of AMF log file, where only crucial messages are reported. Although the AMF may not be busy for the entire duration of this phase, its service time can be estimated as the temporal difference between line **#1** (AMF processes a Registration Request received from the UE) and line **#4** (UE Registration procedure is successfully completed for the UE identified by IMSI-9997001).

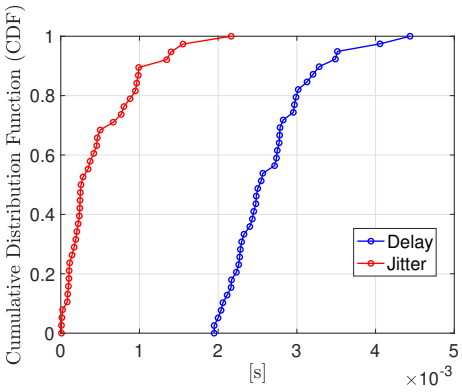

Fig. 2: Cumulative Distribution Functions (CDFs) of delay and jitter associated to the ICMP traffic.

gNB Log

#1 11:25:27.086 [rrc] [info] **RRC Setup for UE[1]**
#2 11:25:27.086 [ngap] [debug] Initial NAS message received from UE[1]
#3 11:25:27.112 [ngap] [debug] Initial Context Setup Request received
#4 11:25:27.332 [ngap] [info] **PDU session resource(s) setup for UE[1]**

AMF Log

#1 11:25:27.089: [gmm] INFO: **Registration request**
#2 11:25:27.089: [gmm] INFO: [suci-0-999-70-01] SUCI
#3 11:25:27.094: [amf] INFO: [imsi-9997001:1] Release SM context [204]
#4 11:25:27.319: [gmm] INFO: [imsi-9997001:1] **Registration complete**

Fig. 3: Example of gNB (top) and AMF log file (bottom).

Service times, estimated by parsing the aforementioned logs, have been represented in terms of boxplot distributions in figure 4. By averaging values over 100 trials, we obtained the following estimates: $1/\mu_{gNB} = 249.2$ msec and $1/\mu_{AMF} = 233$ msec, $CV_{gNB} = 0.0112$ and $CV_{AMF} = 0.0131$.

In contrast, to estimate service times of nodes involved in the data plane SFC (thus, extending the contribution offered in [17] which does not consider data plane), we analyze Wireshark *pcap* traces resulting from TCP and UDP traffic involving gNB and UPF nodes. To emulate TCP and UDP traffic we employ the *iperf* tool (imposing a data rate of 1Mbps[1]) thus producing about 32800 TCP data packets and about 23300 UDP data packets. Differently from SFC control logs, with TCP/UDP traffic we have calculated the time difference between a packet sent to a node connected to the testbed, and the pertinent response message.

[1]Although modest in throughput, this rate represents the worst-case *control-plane-dominated* load observed in IoT/URLLC bursts, where per-packet processing—not link capacity—dominates latency.

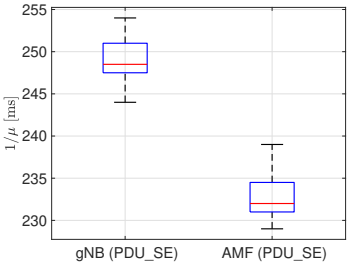

Fig. 4: Service times of gNB and AMF (Control plane).

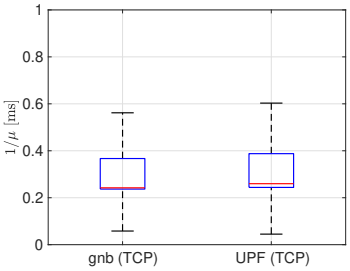

Fig. 5: Service times of gNB and UPF (Data plane - TCP).

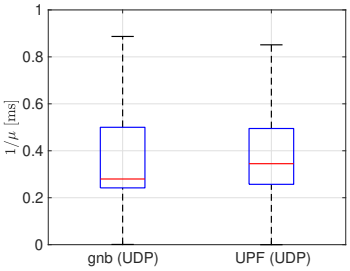

Fig. 6: Service times of gNB and UPF (Data plane - UDP).

We note that, in our local testbed, the one-propagation time between any two VNFs is below 50 μs. Under these circumstances the RTT of an ICMP echo request/reply largely reflects the time the target VNF spends receiving, scheduling and forwarding the packet rather than the physical propagation delay. Consequently, we adopt the RTT as a conservative upper-bound estimate of the VNF service time; if propagation were removed, the service time would only decrease, making our dimensioning decisions safe.

Figure 5 shows the boxplots of service times for gNB and UPF involved in TCP traffic, where the following averaged times are estimated: $1/\mu_{gNB} = 0.49$ msec and $1/\mu_{UPF} = 0.49$ msec, $CV_{gNB} = 1.91$ and $CV_{UPF} = 1.96$. Figure 6 shows the boxplots of service times for gNB and UPF involved in UDP traffic, where the following averaged times are estimated: $1/\mu_{gNB} = 0.44$ msec and $1/\mu_{UPF} = 0.44$ msec, $CV_{gNB} = 1.51$ and $CV_{UPF} = 1.60$. We note that the relatively high coefficients of variation are mainly attributable to experimental trials with *iperf* involving the data plane.

B. Performance Results

For the control plane SFC, we choose as external arrival rate $\lambda = 10$ req/sec that could represent a realistic scenario in a medium-sized 5G cell supporting thousands of devices, where session establishment requests occur sporadically. For the data plane SFC, we suppose as external arrival rate $\lambda = 100$ req/sec. In the same 5G cell, in fact, dozens or hundreds of devices might concurrently transmit TCP or UDP data for streaming, browsing, or gaming. We begin by analyzing the performance results for the control plane SFC, as presented in Table I. The first column lists the vector of instances associated with the gNB and AMF nodes. For example, $k = [2, 3]$ indicates that the gNB can be modeled as an $M/G/2$ system and the AMF as an $M/G/3$ system. The second column pertains to queue stability: satisfying $k_n \geq k_{n0}$ (the second constraint in (6)) ensures system stability ($\rho_n < 1$). The third column states whether the performance constraint is satisfied. All configurations with fewer than three instances for the gNB and AMF fail to meet the stability requirement (second column) and are therefore discarded. The configuration $k = [3, 3]$ satisfies the stability requirement but not the performance requirement. Consequently, the *OptInst* algorithm incrementally adds one instance to the node contributing the highest delay. The last row (highlighted in red) displays the optimal configuration, $k = [5, 5]$, which satisfies the condition $T_{e2e}^{(Ctrl)} \leq \Delta_1 = 500$ ms. Next, we examine the performance results for the data plane SFC. Table II summarizes the performance results for TCP traffic involving the gNB and UPF nodes. In this case, even the smallest configuration ($k = [1, 1]$) satisfies the stability requirement. However, achieving the performance constraint $T_{e2e}^{(Data)} \leq \Delta_2 = 1$ ms requires deploying two instances per node, yielding $k = [2, 2]$ (last row in red). Finally, Table III presents the performance results for UDP traffic. Although $k = [1, 1]$ meets the stability and performance requirements, we must choose $k = [2, 2]$ to satisfy both the TCP and UDP constraints simultaneously.

To conclude, we conduct a sensitivity analysis to assess the performance of control and data SFCs when arrival rates λ deviate from their nominal values. Figure 7 illustrates the behavior of $T_{e2e}^{(Ctrl)}$ as λ deviates from the benchmark value set at 10 req/s (indicated by the vertical dashed line). Unsurprisingly, performance improves for values below the benchmark, as the system experiences reduced load. However, when λ slightly exceeds the benchmark, the Δ_1 constraint is promptly violated, leaving very limited room for maneuver to maintain compliance with performance requirements.

Conversely, figure 8 presents the behavior of $T_{e2e}^{(Data)}$ as λ deviates from the benchmark value set at 100 req/s. In this case, the system demonstrates a higher degree of robustness.

The delay constraint is respected for TCP traffic until the arrival rate reaches roughly 450 requests per second, whereas

TABLE I: Performance results: Optimal allocation (in red) of instances for the **control plane (PDU SE) SFC**.

Vector of instances $k = [k_{\text{gNB}}, k_{\text{AMF}}]$	**Stability constraint** $k_n \geq k_{n0}$	**E2E Performance constraint** $T_{\text{e2e}}^{(\text{Ctrl})} \leq \Delta_1$
[1, 1]	NO	NO
...	...	...
[5, 5]	OK	(495 ms) OK

TABLE II: Performance results: Optimal allocation (in red) of instances for the **data plane (TCP) SFC**.

Vector of instances $k = [k_{\text{gNB}}, k_{\text{UPF}}]$	**Stability constraint** $k_n \geq k_{n0}$	**E2E Performance constraint** $T_{\text{e2e}}^{(\text{TCP})} \leq \Delta_2$
[1, 1]	OK	(1.099 ms) NO
...	...	...
[2, 2]	OK	(0.979 ms) OK

TABLE III: Performance results: Optimal allocation (in red) of instances for the **data plane (UDP) SFC**.

Vector of instances $k = [k_{\text{gNB}}, k_{\text{UPF}}]$	Stability constraint $k_n \geq k_{n0}$	E2E Performance constraint $T_{\text{e2e}}^{(\text{UDP})} \leq \Delta_2$
[1, 1]	OK	(0.949 ms) OK
[2, 2]	OK	(0.88 ms) OK

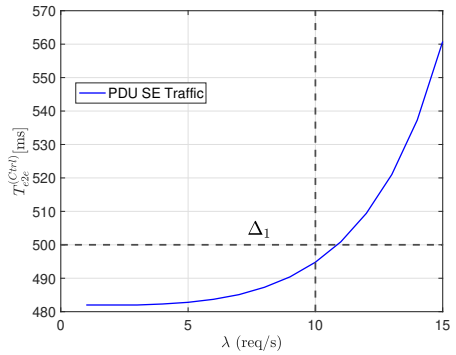

Fig. 7: Sensitivity analysis on the arrival request rates for the control plane SFC.

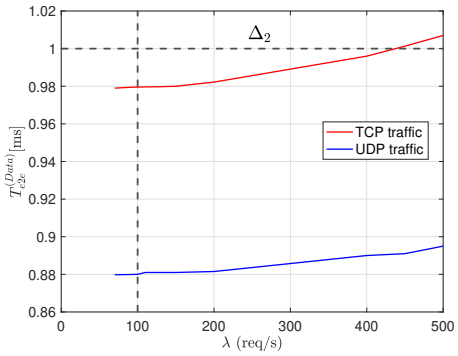

Fig. 8: Sensitivity analysis on the arrival request rates for the data plane SFC.

UDP flows stay within the same limit up to about 1270 requests per second—nearly three times higher. This contrast illustrates that control-plane chains require more conservative provisioning to absorb load spikes, while data-plane chains are inherently better suited to sustain bandwidth-intensive traffic volumes. All scripts and logs/traces used to obtain these results can be accessed in the project's GitHub repository [26].

VI. Concluding Remarks

We presented an $M/G/k$ analytical framework plus the $OptInst$ heuristic that dimension 5G service-function chains with the minimum number of VNF instances while still meeting latency targets. The method links each node's delay to the chain's end-to-end performance and works for both control- and data-plane paths.

VII. Acknowledgement

The work of Mario Di Mauro was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

References

[1] Open5GS, [Online]. open5gs.org.
[2] UERANSIM, [Online]. https://github.com/aligungr/UERANSIM.
[3] ETSI, "TS 129-514," [Online]. https://www.etsi.org/deliver/etsi_ts/129500_129599/129514/18.06.00_60/ts_129514v180600p.pdf.
[4] ETSI, "TS 122-261," [Online]. https://www.etsi.org/deliver/etsi_ts/122200_122299/122261/18.16.00_60/ts_122261v181600p.pdf.
[5] Y. Han, W. Meng, and W. Fan, "SFC Placement and Dynamic Resource Allocation Based on VNF Performance-Resource Function and Service Requirement in Cloud-Edge Environment," *Journal of Systems Engineering and Electronics*, vol. 35, no. 4, pp. 906–921, 2024.
[6] X. Li, H. Zhou, L. Ma, J. Xin, and S. Huang, "Cost and Latency Customized SFC Deployment in Hybrid VNF and PNF Environment," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 4, pp. 4312–4331, 2024.
[7] C. Zhang, T. Sato, and E. Oki, "Robust Deployment Model for Parallelized Service Function Chains Against Uncertain Traffic Arrival Rates," *IEEE Trans. Netw. Service Manag.*, doi: 10.1109/TNSM.2024.3515078, 2025.
[8] A. Karim, J. Ema, T. Yasmin, P. Roy, and M. A. Razzaque, "Latency and Cost-Aware Deployment of Dynamic Service Function Chains in 5G Networks," in *Proc. IEEE ISNCC*, 2023, pp. 1–6.
[9] G. Sun, Z. Xu, H. Yu, X. Chen, V. Chang, and A. V. Vasilakos, "Low-Latency and Resource-Efficient Service Function Chaining Orchestration in Network Function Virtualization," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5760-.5772, 2020.
[10] L. Tang, D. Zeng, J. Yang, Q. Hou, and Q. Chen, "The SFC Deployment Algorithm Based on Incremental Learning and Resource Awareness," *IEEE Internet Things J.*, DOI: 10.1109/JIOT.2025.3595654, 2025.
[11] S. Agarwal, F. Malandrino, C. F. Chiasserini, and S. De, "VNF placement and resource allocation for the support of vertical services in 5G networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 433–446, 2019.
[12] R. Li, B. Decocq, A. Barros, Y.-P. Fang, and Z. Zeng, "Estimating 5G network service resilience against short timescale traffic variation," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 2230–2243, 2023.
[13] A. Heideker and C. Kamienski, "Network queuing assessment: a method to detect bottlenecks in Service Function Chaining," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 4650–4661, 2022.
[14] J. Zu, G. Hu, D. Peng, S. Xie, and W. Gao, "Fair scheduling and rate control for Service Function Chain in NFV enabled data center," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 2975–2986, 2021.
[15] P. Kaliyammal Thiruvasagam, V. J. Kotagi, and C. S. R. Murthy, "The More the merrier: enhancing reliability of 5G communication services with guaranteed delay," *IEEE Netw. Lett.*, vol. 1, no. 2, pp. 52–55, 2019.
[16] M. Di Mauro and A. Liotta, "Statistical Assessment of IP Multimedia Subsystem in a Softwarized Environment: A Queueing Networks Approach," *IEEE Trans. Netw. Service Manag.*, vol. 4, no. 16, pp. 1493–1506, 2019.
[17] M. Di Mauro, "Performance Assessment of Multi-Class 5G Chains: A Non-Product-Form Queueing Networks Approach," *IEEE Trans. Netw. Service Manag.*, doi: 10.1109/TNSM.2025.3588304 2025.
[18] ETSI, "TS 124-501," [Online]. https://www.etsi.org/deliver/etsi_ts/124500_124599/124501/15.00.00_60/ts_124501v150000p.pdf.
[19] ETSI, "TS 123-501," [Online]. https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf.
[20] M. Di Mauro, A. Liotta, "An experimental evaluation and characterization of VoIP over an LTE-A network," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1626–1639, 2020.
[21] G. Cosmetatos, "Some Approximate Equilibrium Results for the Multi-Server Queue (M/G/r)," *Operation Research Quarterly*, vol. 27, no. 3, pp. 615–620, 1976.
[22] T. Kimura, "Approximations for multi-server queues: system interpolations," *Queueing Systems*, vol. 17, pp. 347–382, 1994.
[23] G. Bolch, S. Greiner, S. De Meer and K.S. Trivedi *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. New York, Wiley-Interscience, 1998.
[24] D.P. Bertsekas, R.G. Gallager, *Data Networks*. New York, Prentice-Hall International Editions, 1992.
[25] H. Frenk, M. Labbé, M. Van Vliet, S. Zhang "Improved Algorithms for Machine Allocation in Manufacturing Systems," *Operations Research*, vol. 42, no. 3, pp. 523–530, 1994.
[26] GitHub code. Available online: https://github.com/mariodim/5G_Performance.