Stress Testing Open5GS UPF Implementation

Measuring resource consumption and latency in virtual environment

Kevin Ji Shan[1]
Supervisor(s): Nitinder Mohan[1]**, Marco Colocrese**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 10, 2025

Name of the student: Kevin Ji Shan
Final project course: CSE3000 Research Project
Thesis committee: Nitinder Mohan, Marco Colocrese, Guohan Lan

Abstract

The global adoption of 5G technology is rapidly accelerating and 5G traffic is growing exponentially. This increase in demand compels network operators to evaluate whether their current and upcoming 5G infrastructure can effectively accommodate the growing data traffic. A key component within the 5G network is the User Plane Function (UPF), which connects the end-devices to the data networks. Therefore, it is vital for both equipment manufacturers and service providers to analyze the performance of existing UPF implementations. This paper presents an initial approach to assess the Open5GS UPF performance by conducting stress testing in virtualized environment. We focus on finding the optimal UPF configuration by measuring the resource consumption and latency of the UPF under varying intensity of generated traffic, and providing a simple queueing model for the UPF. Numerical results show that a CPU load of 70-80% balances between latency and throughput while ensuring that 99% of the packets are forwarded within 150 ms.

1 Introduction

5G or even 6G mobile networks represent a significant change in how network resources are being managed and used. As decided by the Third Generation Partnership Project (3GPP), services that the 5G should provide include enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (URLLC) [1]. These could provide an answer for many challenging demands, including immersive virtual reality, remote healthcare and traffic regulation for autonomous vehicles. Additionally, Ericsson stated in their technical report [2] that 5G mobile subscriptions will reach 6.3 billion in 2030, equaling 67% of total mobile subscriptions. The increasing 5G and the potential 6G traffic raise the need to assess how well existing technology can meet the performance demands for real-world applications [3].

Within a 5G network, the core network is the central component responsible for managing data, sessions and service delivery. The core network consists of multiple network functions, including the User Plane Function which forwards all user data from the RAN to the Data Network [4]. A simplified overview of a 5G network can be found in Figure 1. Maximizing data throughput introduces long waiting queues at the UPF that significantly impact latency [5], while minimizing latency typically leads to underutilized resources and suboptimal throughput, thus prolonging overall transmission time [6]. With the global daily 5G traffic reaching exabytes [7], it is important to assess how well the existing 5G network implementation meets the desired low-latency performance while maintaining high throughput. Such evaluation is key to optimizing resource utilization and maintaining a high quality of experience for end users.

While existing studies provide valuable insights into UPF performance and demonstrate methods to measure resource consumption, no research has measured the latency across different traffic scenarios while proposing a suitable mathematical model describing the relation between latency and load. This research begins by measuring UPF's CPU, memory usage, throughput, and latency as a function of load. The measured results will be compared with known queueing models such as M/M/1 to characterize the relationship between latency and load. Ultimately, a recommendation is provided for optimal UPF load level. This research is expected to provide insight into how the load on a UPF should be balanced between resource consumption, throughput and responsiveness based on the measured results. In addition, the paper contributes by proposing a mathematical model that reflects the behavior of latency under varying traffic loads, enabling better prediction and management of UPF performance.

The paper is organized as follows: Section 2 provides the background of the research and discusses related works in depth. Section 3 features a description of the architecture of the experiment and the setup used. Measured results are presented in Section 4 along with the analysis. Section 5 features the responsible research and Section 6 presents the conclusion and potential future research.

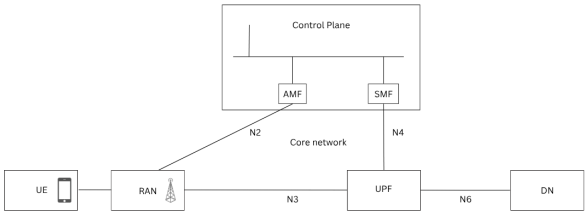


Figure 1: A simplified overview of the 5G infrastructure

2 Background and Related Works

2.1 5G infrastructure and Queueing model

The 5G infrastructure consists of User Equipment, Radio Access Networks and the Core Network (See Figure 1). User Equipment (UE) is connected to the gNB based RAN [4] and the RANs are connected to the 5G Core Network (CN). The CN consists of the control and the user plane [8] according to the Control and User Plane Separation (CUPS) [1]. The control plane includes network functions like Access and Mobility Management Function (AMF) and the Session Management Function (SMF). These network functions manage the authentication of users, establish the connection between UEs and Data Networks (DN) [9]. The RAN and UPF are part of the user plane, providing ways to forward data between UE and data networks. The connection between the RAN and a specific UPF instance is encapsulated using the GPRS Tunneling Protocol - User Plane (GTP-U) [10]. Open5GS is an open-source 5G Core implementation supporting UPF, SMF, and other control-plane functions [11].

Queueing theory provides a mathematical model for analyzing systems where tasks arrive, wait for service and eventually depart [12]. This is essential for generalizing and understanding a queueing system's theoretical performance as it allows various models that characterize different arrival patterns and configurations. The M/M/1 queue represents the most fundamental queueing model,

having a single first-in-first-out server with infinite buffer size [13]. This model has two parameters: λ, μ being the mean arrival and service rate, while the inter-arrival time and service time follow an exponential distribution with respective mean $\frac{1}{\lambda}, \frac{1}{\mu}$ [14]. The model assumes $\lambda < \mu$ such that the queue size will reach a steady state [15]. The sojourn time S can then be described using λ and μ [16]:

$$S = \frac{1}{\mu - \lambda} \tag{1}$$

In a M/M/1 queue, the sojourn time is exponentially distributed [14], meaning that:

$$P(S > aE(S)) = e^{-a} \tag{2}$$

This property is significant because it provides a guarantee that the majority of the packets will have a latency under a certain value [17]. In addition, the latency of a packet in the UPF can be modeled using the sojourn time S and some additional parameters to scale the results to ms.

$$L = \frac{fac}{\mu - \lambda} + c \tag{3}$$

We apply queueing theory principles to analyze UPF performance under varying traffic loads, using M/M/1 models as a baseline for understanding the relationship between traffic intensity and latency performance [18] as this model allows us to understand and verify the results quickly. This has been effective in the past research [19] and it is suitable as both the M/M/1 and the Open5GS UPF operate according to the FIFO principle [20]. This theoretical framework ultimately builds toward generalizing UPF performance for different setups.

2.2 Related Work

Recent studies have explored UPF performance from different perspectives, yet critical gaps remain in the exact relation between load and latency. Only limited research exists on benchmarking UPF implementations and finding the corresponding mathematical model. Mamushiane et al. [4] stress-tested the Open5GS UPF using UERANSIM, measuring the CPU and memory usage against 50 to 200 simultaneous UE connections. While their work provided insights into the resource consumption of the UPF during stress testing. Their reliance on ICMP ping traffic fundamentally limits the study's applicability, as it fails to replicate internet traffic patterns of mixed packets types and sizes. It also missed the opportunity to measure the resource consumption and throughput under realistic, extreme traffic load since the UERANSIM only allows up to 200 connections. Similarly, Scheich et al. [21] evaluated a Docker-based UPF accelerated with eXpress Data Path (XDP), showing significant throughput gains for GTP-U traffic but omitting analysis of resource consumption or latency. Christakis et al. [22] evaluated UPF implementations like SPGWU, VPP, and SmartNIC-P4, highlighting their difference in terms of cost, performance and resource consumption. However, the research did not research the impact of different traffic loads on the performance.

Arteaga et al. [23] provided a PEPA-based method to evaluate the throughput, average response time and scalability metric of network functions like the NRF and SMF. However, they did not examine the UPF performance. Aliyu et al. [19] proposed a M/M/C/∞/∞ model to describe the 5G core network and they have shown that the probability of arrival for packets at the server follows the expected Poisson behavior for different λs. Rotter et al. [9] proposed a queueing model for a threshold based scheduling algorithm that controls the numbers of UPF instances in 5G systems. Using both analytical and numerical results, they showed that the algorithm is able to automatically adjust based on traffic load and minimize resource consumption compared to the native implementation. However, they did not measure the latency that the packets experience through the queue of UPF. This is fundamental as it provides insight on how the end user would be affected.

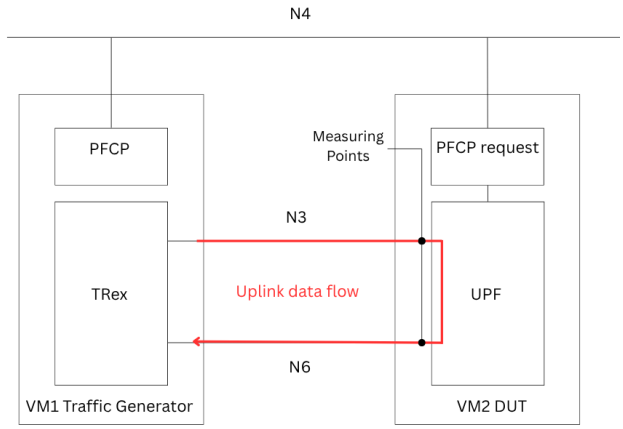


Figure 2: Experiment setup: Traffic Generator (VM1) and UPF (VM2) connected via PFCP, N3 (GTP-U) and N6 (IP) networks.

3 Methodology

3.1 Experiment Architecture

The overall setup is shown in Figure 2. The testbed configuration uses two virtual machines connected with the simulated N3, N4 and N6 interfaces respectively. In addition, the deployment details of the two virtual machines are summarized in Table 1. Open5GS and TRex are chosen due to their open source license and ease of deployment compared to their alternatives [11], [24]. However, the same configuration could utilize any other alternative core network implementation such as Open5G Core or free5GC.

As only the UPF performance is relevant to this research, the remaining core network is simulated: a PFCP connection to the abstracted control plane is established via the N4 interface after a request, such that the UPF receives predefined rules for packet forwarding. To generate traffic, the traffic generator TRex is used in the stateless mode. This allows generating and measuring traffic of predefined type and volume through any network interface. In the testbed scenario, TRex acts as the UE, RAN and DN. The traffic generator generates predefined GTP-U encapsulated data to the N3 interface of the UPF, simulating data from the RAN. According to the PFCP rules, the UPF decapsulates the GTP-U packet and sends the inner UDP payload to the N6 interface. TRex simulates the Data

Table 1: Deployment Specifications

Device	Traffic Generator	UPF
Software	TRex 3.04	Open5GS v2.7.5
Hosting Software	VirtualBox 7.1.10	VirtualBox 7.1.10
Operating System	Ubuntu 24.04	Ubuntu 20.04
CPU	i5-12600KF @3.7 GHz	i5-12600KF @3.7 GHz
Allocated Cores	3 vCPUs	4 vCPUs
Memory	8 GB	8 GB
Storage	20 GB SSD	20 GB SSD

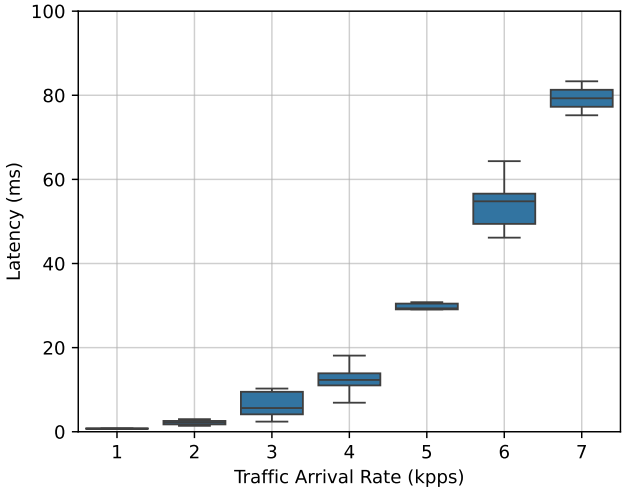
Network and measures the throughput of successfully forwarded packets, while packets that are not forwarded are dropped due to resource constraints such as CPU overload or buffer capacity limitations. This research aims to measure the resource usage and the latency of the UPF. The Linux tool pidstat is used to sample the CPU and memory usage of the UPF process every second during the traffic. The processing delay is measured using tcpdump at the N3 and N6 interfaces, which represents the time between the arrival and forwarding of a packet at the UPF. This approach provides more accurate measurements than end-to-end measurements on TRex, which may include additional network and processing overhead caused by the operating systems.

3.2 Experiment Procedure and Setup

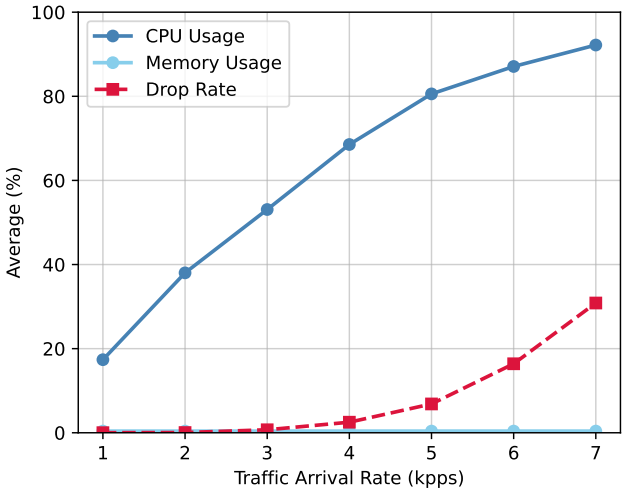
This section will describe the experimental methodology and the configuration procedures. The UPF configuration YAML file needs to be adjusted before the deployment of UPF, to define the address range of the N3, N4 and N6 interfaces. Additional information of configuring Open5GS UPF can be found here [11]. Before generating traffic, tcpdump and pidstat should be initialized to capture the UPF related metrics. A PFCP session establishment is initiated from VM1 to VM2 via the N4 interface to configure packet forwarding rules. The traffic generator needs to be configured with the correct address range and started in the stateless interactive mode without the Scapy server. A custom Python script defines the packet structure, which consists of GTP-U encapsulated UDP packets. The payload of the UDP is a 4 byte counter followed by 1396 bytes of mock data to mimic the packet payload size in a realistic scenario [25]. To differentiate sequential packets, the traffic generator STLVM is used to modify the packet through overwriting the payload [26]. Therefore, UDP checksum is disabled to prevent packets from being dropped at the UPF.

The experiment is repeated 10 times for each λ from 1000 to 7000. Each experiment includes sending around 5 million generated packets, to reach steady state and generalize the measured results [27]. Using the results, a grid search is performed over the parameter μ , fac and c to find the M/M/1 latency curve with the lowest MSE using Equation 3. The Xth percentile for different λs can be calculated using Equation 2 where

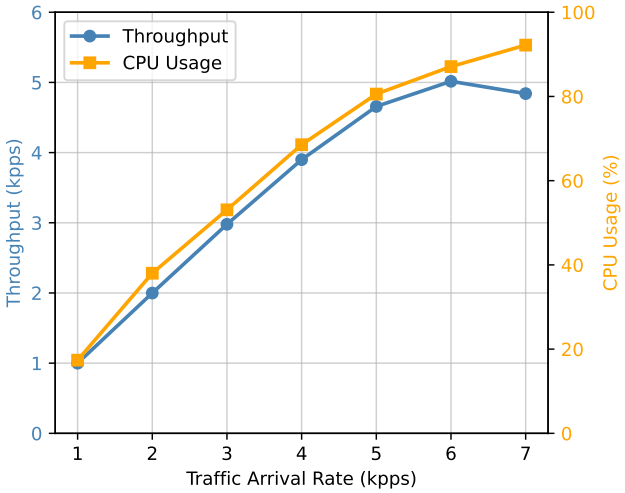
$$a = -e^{1-\frac{X}{100}} \tag{4}$$



(a) Average Latency



(b) CPU, Memory usage and Drop rate



(c) CPU usage and Throughput

Figure 3: The impact of traffic on UPF performance and resource consumption

4 Results

4.1 Performance metrics

The performance metrics including the average CPU, memory usage, drop rate and latency versus the traffic arrival rate are plotted in Figure 3. It can be observed that the latency increases linearly when the traffic intensity is low and rises sharply when the traffic reaches near saturation. This is in line with the exponential latency of the M/M/1 model. Additionally, the drop rate in Figure 4b is negligible for lower intensities but increases to 30% for higher traffic rates. Memory utilization remains constant at 0.4% for all traffic intensities while the CPU utilization grows linearly with the amount of traffic, as expected [13]. Another observation is that the throughput starts to drop at traffic intensities higher than 6000 per second. This is due to the fact that receiving and dropping packets beyond the service rate leads to unnecessary CPU overhead, which will contend with the actual forwarding. The optimal arrival rate is around $4000 < \lambda < 5000$. Compared to $\lambda > 5000$, this range utilizes less CPU resources, achieves a lower average latency while guaranteeing a throughput of 4000 - 4600 packets per second. This shows that the 70 - 80% CPU load is optimal [28].

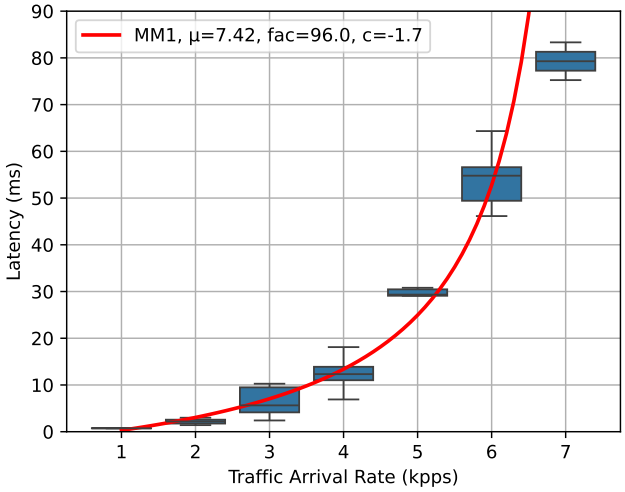
4.2 Comparison with M/M/1 Model

After performing a grid search over the parameters μ, fac and c, the M/M/1 model with the least MSE is shown in Figure 4. Although the actual behavior of the UPF deviates from the theoretical model, it is possible to accurately model the average latency. This leads to the analysis of the 95th and 99th percentiles. The 95th and 99th percentile latency is expected to increase exponentially as load increases [14]. However, this is not in line with the observed data, indicating that the M/M/1 model has limited meaning in predicting 99th percentile latency of TRex generated packets.

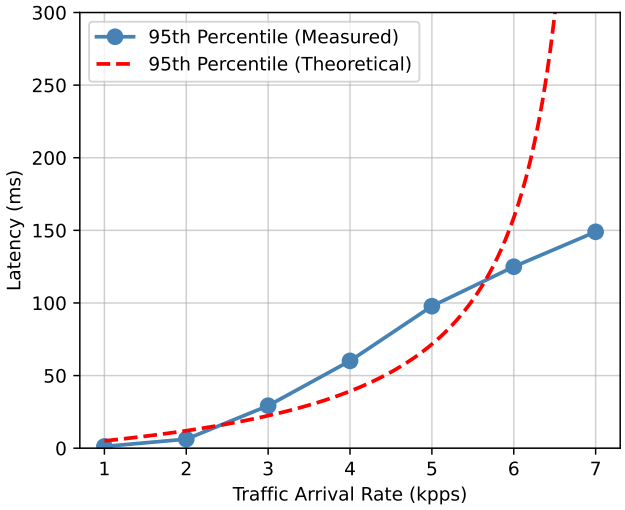
While the testbed provides a reasonably good starting point for measuring and modeling UPF performance under various traffic scenarios, it should be noted that the M/M/1 queue has an infinite buffer size and assumes an arrival rate not larger than the service rate [15]. In contrast, the UPF drops packets when the buffer space runs out or the CPU is overloaded, which explains why the latency is not exponentially increasing to infinity for $\lambda >= \mu$. Additionally, TRex is not configured to have exponential interarrival times assumed in the M/M/1 model, this could explain the lack of accuracy in the 95th and 99th percentile prediction as the measured latency does not satisfy Equation 2. Furthermore, since all processes were run on virtual machines and on the same physical machine, performance estimates might differ from a real-world scenario, where packets would require more inspection and thus more processing power before forwarding packets than in our setup.

5 Responsible Research

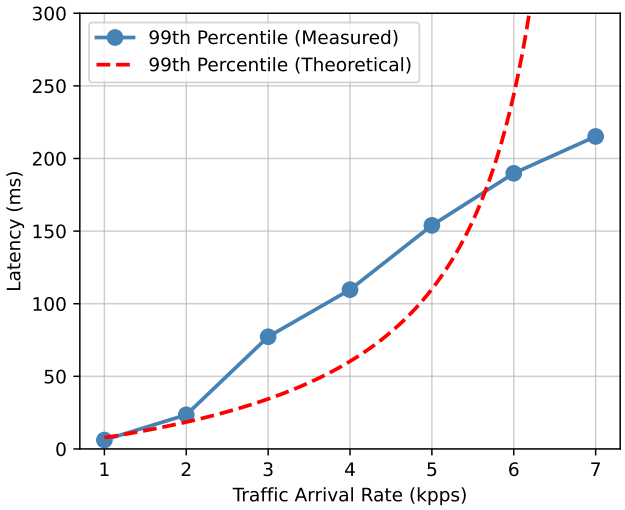
In this section, ethical aspects of the research will be discussed such as processing personal information and reproducibility. Our experiments were designed such that the packets generated only contain synthetic GTP-U traffic with no real information; no real user data or personally identifiable information was used or processed. The virtualized RAN and CN components operated entirely within the virtual machine, never traversing external or shared networks. This



(a) Average Latency



(b) 95th Percentile Latency



(c) 99th Percentile Latency

Figure 4: UPF latency metrics compared with M/M/1 Model

approach eliminates any risk of privacy violations or unintended data exposure while maintaining the validity of our measurements.

To ensure reproducibility, we have made our research methodology fully transparent by documenting all software versions, configurations and experimental parameters. We used open source implementations and the experiment can be reproduced by following the instructions in Section 3. All measurement tools (tcpdump, pidstat, Python) and analysis methods are standard and widely available utilities. Our experimental design includes multiple repetitions to ensure statistical validity. The queueing model parameter fitting is documented, allowing others to verify our mathematical modeling approach. We acknowledge the limitations of our experiments and clearly distinguish between our controlled experimental results and real-world deployment scenarios.

Another consideration is sustainability, we think that the experiment has little impact as the total active running time of the UPF is less than three hours. This is negligible compared to the potential benefits of optimizing the load and improving the energy efficiency of all future UPFs.

6 Conclusions and Future Work

In this work, we presented an evaluation of the Open5GS UPF's performance and resource consumption under controlled traffic loads in a virtualized environment. By generating GTP-U traffic with a traffic generator at rates from 1 kpps to 7 kpps and measuring latency, CPU utilization, and throughput, we observed that the UPF is most efficient between 70 and 80% CPU load while running on a single core, balancing throughput, latency, and energy efficiency. Beyond that point, queueing delays and packet drops rise sharply. We proposed an M/M/1 queueing model to predict the average latency, but the distribution of the measured latency is not exponential.

Future work could explore additional queueing models to approximate the 99th percentile latency more accurately, in order to provide a guarantee for the worst case scenario. Performance of the Open5GS UPF implementation while multithreading could also be measured and compared to alternative queueing models. In addition, improving the Open5GS UPF implementation to be more resource efficient could be considered.

References

[1] 3rd Generation Partnership Project (3GPP). [n. d.]. 5G System Overview. https://www.3gpp.org/technologies/5g-system-overview. Accessed: 2025-06-20.

[2] Ericsson. 2024. *Ericsson Mobility Report: November 2024.* Technical Report. Telefonaktiebolaget LM Ericsson, Stockholm, Sweden. https://www.ericsson.com/4adb7e/assets/local/reports-papers/mobility-report/documents/2024/ericsson-mobility-report-november-2024.pdf Accessed: Jun. 19, 2025.

[3] W.-E. Chen and C. H. Liu. 2020. High-performance user plane function (UPF) for the next generation core networks. *IET Networks* 9, 6 (2020), 284–289.

[4] Lusani Mamushiane, Albert A. Lysko, Thoriso Makhosa, Joyce Mwangama, Hlabishi Kobo, Alec Mbanga, and Rofhiwa Tshimange. 2023. Towards Stress Testing Open5GS Core (UPF Node) on a 5G Standalone Testbed. In *Proceedings of the 2023 IEEE AFRICON*. IEEE, Pretoria, South Africa. doi:10.1109/AFRICON55910.2023.10293284

[5] Donald Gross and Carl M. Harris. 1985. *Fundamentals of Queueing Theory.* Wiley, Chichester.

[6] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. 2006. Optimal throughput-delay scaling in wireless networks: Part I: The fluid model. *IEEE/ACM Transactions on Networking* 14, 6 (Dec 2006), 2568–2592. doi:10.1109/TNET.2006.875645

[7] Ericsson. 2024. Mobile traffic forecast – Mobility Report. https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast. Accessed: 2025-06-02.

[8] S. Redana and O. Bulakci. 2018. *View on 5G Architecture.* White Paper 22.891, Version 14.2.0. 5G-PPP Architecture Working Group.

[9] C. Rotter and T. V. Do. 2021. A Queueing Model for Threshold-Based Scaling of UPF Instances in 5G Core. *IEEE Access* 9 (2021), 81443–81453. doi:10.1109/ACCESS.2021.3085955

[10] 3GPP. 2020. *5G; Study on Scenarios and Requirements for Next Generation Access Technologies.* Technical Specification TS 38.913, Version 16.0.0, Release 16. 3GPP.

[11] Open5GS Project. 2025. *Open5GS Quickstart Guide.* Open5GS. https://open5gs.org/open5gs/docs/guide/01-quickstart/

[12] Investopedia [n. d.]. *Queuing Theory.* Investopedia. https://www.investopedia.com/terms/q/queuing-theory.asp Accessed: 2025-06-22.

[13] J. R. Artalejo and M. J. Lopez-Herrero. 2001. Analysis of the Busy Period for the M/M/c Queue: An Algorithmic Approach. *Journal of Applied Probability* 38, 1 (March 2001), 209–222.

[14] TU Eindhoven. [n. d.]. Queueing Theory Chapter 4. Course Material. https://iadan.win.tue.nl/que/h4.pdf Accessed: 2025-06-22.

[15] J. Sztrik. 2021. *Basic Queueing Theory.* GlobeEdit. https://irh.inf.unideb.hu/~jsztrik/education/16/SOR_Main_Angol.pdf Online book.

[16] Hao Jiang and Constantinos Dovrolis. 2005. Why is the Internet Traffic Bursty in Short Time Scales? *ACM SIGMETRICS Performance Evaluation Review* 33, 1 (June 2005), 241–252. doi:10.1145/1071690.1064240

[17] Micheal Kopp. [n. d.]. Why averages suck and percentiles are great. Dynatrace Blog. https://www.dynatrace.com/news/blog/why-averages-suck-and-percentiles-are-great/ Accessed: 2025-06-22.

[18] M. Rahouti, K. Xiong, Y. Xin, and N. Ghani. 2021. A priority-based queueing mechanism in software-defined networking environments. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE Press, 1–2. doi:10.1109/CCNC49032.2021.9369614

[19] A. L. Aliyu and J. Diockou. 2023. An Analytical Queuing Model Based on SDN for IoT Traffic in 5G. In *Advanced Information Networking and Applications (Lecture Notes in Networks and Systems, Vol. 655)*, L. Barolli (Ed.). Springer, Cham. doi:10.1007/978-3-031-28694-0_42 AINA 2023.

[20] Cisco Systems. 2024. *Cisco Ultra Cloud Core 5G User Plane Function Configuration and Administration Guide.* Cisco Systems, Inc. https://www.cisco.com/c/en/us/td/docs/wireless/ucc/upf/2024-04/config-admin/ucc-5g-upf-config-and-admin-guide_2024-04.pdf Accessed: 2025-06-22.

[21] Christian Scheich, Marius Corici, Hauke Buhr, and Thomas Magedanz. 2023. Performance Analysis of a 5G User Plane Function Accelerated with eXpress Data Path in Docker Containers. In *Proceedings of the 2023 IEEE Future Networks World Forum (FNWF)*. IEEE. doi:10.1109/FNWF58287.2023.10520617

[22] Sokratis Christakis, Theodoros Tsourdinis, Nikos Makris, Thanasis Korakis, and Serge Fdida. 2024. Evaluation of User Plane Function Implementations in Real-World 5G Networks. In *Proceedings of the IEEE INFOCOM Workshops (CNERT)*. IEEE. doi:10.1109/INFOCOMWKSHPS61880.2024.10620666

[23] C. H. T. Arteaga, A. Ordonez, and O. M. C. Rendon. 2020. Scalability and Performance Analysis in 5G Core Network Slicing. *IEEE Access* 8 (January 2020), 142086–142100. doi:10.1109/ACCESS.2020.3013597

[24] Cisco Systems. 2025. TRex Stateless Mode Documentation. https://trex-tgn.cisco.com/trex/doc/trex_stateless.html. Accessed: 2025-05-23.

[25] Geoff Huston. 2024. The Size of Packets. APNIC Labs. https://labs.apnic.net/index.php/2024/10/04/the-size-of-packets/ Accessed: 2025-06-21.

[26] Cisco Systems. [n. d.]. TRex Stateless Field Engine. TRex Traffic Generator Documentation. https://trex-tgn.cisco.com/trex/doc/cp_stl_docs/api/field_engine.html Accessed: 2025-06-21.

[27] [n. d.]. Network Performance Analysis. In *NCBI Bookshelf*. Michigan State University College of OM.

[28] Irian Leyva-Pupo. 2023. *Strategies for UPF Placement in 5G and Beyond Networks.* Ph.D. thesis. Universitat Politècnica de Catalunya. Advisor: Cristina Cervelló-Pastor.