UPF Autoscaling in Private 5G Networks for Video Transmission: A Comparative Analysis of Reactive and Predictive Approaches

Pedro Antônio de Andrade da Silva[1], **Paulo Ditarso Maciel Jr.**[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB)
João Pessoa – PB – Brazil

andrade.antonio@academico.ifpb.edu.br, paulo.maciel@ifpb.edu.br

Abstract. *Industry 4.0 applications demand high reliability and low latency, driving the adoption of Service-Based Architectures in Private 5G networks. However, resource orchestration remains challenging; traditional reactive mechanisms, such as the Kubernetes Horizontal Pod Autoscaler (HPA), fail to address the session affinity of the GPRS Tunneling Protocol. This leads to unbalanced loads where active units saturate while new instances sit idle. This paper proposes a predictive scaling mechanism for the User Plane Function (UPF) that anticipates traffic and instantiates resources prior to session establishment. Validated in a virtualized testbed, the predictive approach ensures proper load balancing and prevents connection drops typical of reactive scenarios.*

1. Introduction

The consolidation of Industry 4.0 has accelerated the emergence of applications with increasingly strict requirements, driven by the need for real-time interaction among humans, cyber-physical systems, and industrial processes. While conventional services typically operate under moderate network demands, applications such as augmented reality, computer vision, and autonomous logistics require high reliability, high throughput, and, critically, low latency [Aijaz 2020].

To support such requirements, fifth-generation mobile networks (5G) have evolved into a service-oriented platform capable of supporting multiple vertical domains, including healthcare and manufacturing [Strinati et al. 2020]. The Service-Based Architecture (SBA) enables network functions to be implemented as independent microservices, improving flexibility and scalability [Choudhari et al. 2022]. In this context, Private 5G Networks (PNs) are particularly relevant because they provide controlled environments where Quality of Service (QoS) can be enforced for time-sensitive video traffic [Batalla 2020].

Despite the flexibility of SBA, efficient resource management in virtualized platforms such as Kubernetes remains challenging. The User Plane Function (UPF) acts as the data-plane "anchor" and is therefore a natural bottleneck. Although edge placement of UPFs can reduce latency [Mendes de Souza et al. 2025], dynamic traffic variations still require elastic scaling mechanisms to avoid either resource underutilization or Quality of Experience (QoE) degradation due to congestion.

Recent literature highlights Artificial Intelligence (AI) and Machine Learning (ML) as key enablers for autonomous orchestration, overcoming the limitations of purely

reactive methods [Yeh et al. 2024]. However, most autoscaling solutions, including the native Kubernetes Horizontal Pod Autoscaler (HPA), rely on reactive CPU and memory thresholds. In 5G systems, this behavior interacts with a critical architectural constraint: GTP (GPRS Tunneling Protocol) session affinity. Since PDU (Packet Data Unit) sessions remain anchored to the UPF selected at session establishment, newly created replicas do not absorb already active traffic flows. As a consequence, legacy UPFs can remain overloaded while newly instantiated UPFs stay underutilized.

This work proposes a predictive UPF autoscaling strategy that provisions resources before new PDU sessions are established. By aligning resource availability with expected demand, the approach enables more effective distribution of new sessions and reduces service instability under traffic growth.

The remainder of this paper is organized as follows. Section 2 reviews related work on 5G orchestration and UPF scalability. Section 3 presents the proposed architecture and experimental design. Section 4 reports and discusses the comparative results between native reactive HPA and the predictive strategy. Finally, Section 5 concludes the paper and outlines future directions.

2. Related Work

Recent studies have explored Artificial Intelligence (AI) and Machine Learning (ML) to overcome the limitations of reactive autoscaling, particularly in handling dynamic and time-dependent workloads across different network domains, from Radio Access Networks (RAN) to the 5G Core (5GC).

In the RAN domain, [Yeh et al. 2024] propose an intelligent and automated network slicing framework for Open RAN (O-RAN) environments. Their approach applies deep learning to forecast traffic demand and allocate radio resources while preserving service-level guarantees for multiple tenants. Although these results demonstrate the benefits of predictive orchestration, the scope remains centered on physical resource block allocation and RAN Intelligent Controller integration, rather than user-plane scalability in the 5G core.

In the 5GC context, the User Plane Function (UPF) is a critical bottleneck because it processes and forwards user traffic. The authors in [Veeck et al. 2025] investigate UPF scalability in Kubernetes-based virtualized environments by comparing native reactive HPA with a proactive GRU-based strategy. Their findings show that proactive control better preserves latency and service stability during load peaks by instantiating pods before resource exhaustion occurs.

However, an important limitation is often underexplored in cloud-native telecom autoscaling: the interaction between Kubernetes load-balancing behavior and the persistent nature of GTP tunnels [Botez et al. 2021]. Even when predictive approaches improve CPU- and latency-oriented indicators, the practical implications of session affinity on flow redistribution are frequently under-detailed.

This paper addresses that gap by explicitly analyzing the interaction between autoscaling mechanisms and GTP session affinity in UPF scaling. Unlike stateless workloads, where new replicas immediately receive traffic, established 5G PDU sessions remain anchored to previously selected UPFs, limiting the effectiveness of both reactive

and predictive approaches. Therefore, the contribution of this work is not only to apply predictive scaling, but to demonstrate how scaling timing influences session distribution at establishment time, enabling more effective load balancing and reducing service disruption risk.

3. Proposed Approach

Resource management in Kubernetes is typically implemented through two mechanisms: the *Horizontal Pod Autoscaler* (HPA) and the *Vertical Pod Autoscaler* (VPA) [Nguyen et al. 2020]. HPA scales applications horizontally by changing the number of pod replicas according to demand, whereas VPA adjusts the CPU and memory allocated to existing replicas.

In 5G environments, VPA presents an important limitation: applying new resource limits commonly requires pod restarts. For stateful user-plane services, this behavior can interrupt active sessions and degrade both service continuity and Quality of Experience (QoE), which is undesirable for mission-critical applications [3GPP 2022]. For this reason, our work focuses on HPA-based scaling.

Nevertheless, native HPA is reactive: new replicas are created only after predefined thresholds (e.g., CPU usage) are exceeded. In 5G core architectures, this characteristic interacts with PDU session affinity. When a UPF becomes overloaded and HPA creates a new replica, already established sessions remain attached to the original UPF due to GTP tunnel persistence. As a result, the new UPF receives only future session requests, often producing a prolonged imbalance in which one UPF remains congested while others are underused [3GPP 2023, Veeck et al. 2025].

To mitigate this behavior, we propose predictive HPA control. By forecasting traffic evolution, the orchestrator instantiates additional UPFs *before* new session establishment, so incoming sessions can be distributed over ready instances from the start. In order to evaluate this strategy, we developed a virtualized 5G testbed that reproduces controlled user-plane operation under progressive load increments, illustrated in Figure 1.

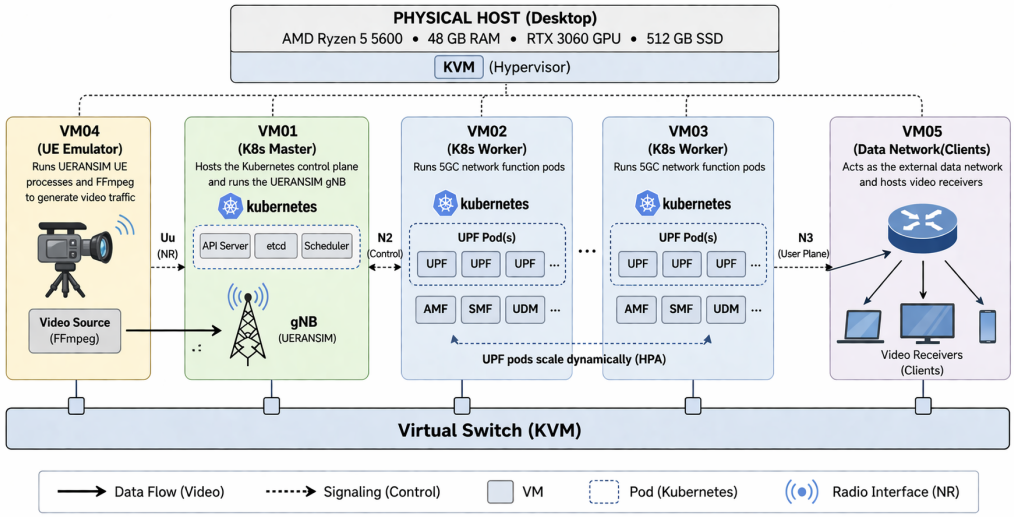


Figure 1. Experimental virtualized testbed.

The physical host is a desktop machine with an AMD Ryzen 5 5600 CPU, 48 GB RAM, an RTX 3060 GPU, and a 512 GB SSD. Virtualization is managed by KVM[1], where five virtual machines (VMs) are deployed as follows:

- **VM01 (K8s Master):** Hosts Kubernetes control-plane and runs the UERANSIM gNodeB (gNB) component[2], interconnecting radio access and core functions;
- **VM02 and VM03 (K8s Workers):** Execute 5GC network-function pods, where UPF instances are dynamically created by autoscaling mechanisms;
- **VM04 (UE Emulator):** Runs UERANSIM UE processes and FFmpeg[3] to generate video traffic;
- **VM05 (Data Network/Clients):** Acts as the external data network and hosts video receivers.

The 5G core is implemented with Open5GS[4] and deployed on Kubernetes via Helm, which automates the provisioning of the required deployments and services. The logical data path follows an end-to-end configuration: traffic generated by UEs in VM04 is first transmitted to the gNB in VM01, then encapsulated over GTP-U toward the UPF hosted in VM02 or VM03, and finally decapsulated and forwarded to application endpoints in VM05. To evaluate both QoE and resource utilization, two traffic profiles were considered:

1. **Synthetic traffic (IPerf):** used to characterize the operational limits and saturation behavior of the testbed;
2. **Real video traffic:** Reference video 1080p/60 FPS[5], transmitted from UEs to clients through FFmpeg.

The collected metrics were: (i) received frames per second (FPS), used as an application-level QoE indicator; and (ii) UPF pod CPU consumption (in millicores), used to quantify packet-processing cost. The evaluation compares native reactive HPA with the proposed predictive strategy under an identical dynamic workload scenario. The scenario begins with a single active UPF instance and one active gNB. Initially, one UE (UE1) transmits video to Client1. Subsequently, every 60 seconds, one additional UE–Client pair is activated until a total of five simultaneous pairs is reached. New sessions are then assigned according to the 5GC round-robin policy across available UPFs.

The objective is to determine whether the system can provision additional UPF instances early enough to prevent video-quality degradation and whether newly established sessions are effectively distributed to newly instantiated UPFs.

The training data for the predictive model was obtained from a baseline execution of the video transmission experiment, conducted without any active autoscaling mechanisms. This dataset captures the raw throughput variations of the 1080p/60 FPS video streams as user sessions were progressively established across the testbed. By training on this uncontrolled traffic profile, the model learns to identify the characteristic patterns and temporal dependencies of the video workload in its native state, providing a basis for

[1]https://linux-kvm.org/page/Main_Page
[2]https://github.com/aligungr/UERANSIM
[3]https://www.ffmpeg.org/
[4]https://open5gs.org/
[5]https://peach.blender.org/

predicting future demand independently of any resource scaling actions [Yeh et al. 2024].

Following recent advancements in autonomous orchestration for the 5G Core [Veeck et al. 2025, Yeh et al. 2024], a predictive autoscaling strategy based on a Long Short-Term Memory (LSTM) regression model was developed. While alternative recurrent architectures such as Gated Recurrent Units (GRU) have been investigated for UPF scalability, the LSTM network was selected for its capacity to capture long-term temporal dependencies in network traffic. The model architecture consists of an LSTM layer with 64 units and ReLU activation, followed by two dense layers of 64 and 32 neurons, respectively, and a linear output layer with 1 neuron for throughput prediction ($\hat{V}$). Training was performed using the Adam optimizer with a learning rate of 0.001, minimizing the Mean Squared Error (MSE) over 100 epochs with a batch size of 16.

The predictive analysis operates on a sliding window of 20 seconds of historical throughput data, collected in real-time via Prometheus. The model was configured with a 15-second look-ahead horizon, allowing the orchestrator to anticipate demand peaks. During execution, the predicted throughput ($\hat{V}$) is converted into the required number of replicas (R) through the relationship $R = \lceil \hat{V}/C \rceil$, where $C = 17$ Mbps represents the estimated nominal processing capacity of each UPF instance. To prevent excessive oscillations, the control logic incorporates a hysteresis mechanism with a 150-second cooldown period for scale-down operations.

4. Results

This section presents the comparative experimental results for native reactive HPA and the proposed predictive strategy. The discussion is organized into three parts: bandwidth characterization of the testbed, comparative analysis of FPS and resource consumption, and evaluation of UPF instantiation and UE association dynamics.

4.1. Bandwidth Analysis

Before the autoscaling experiments, IPerf was used to characterize testbed limits, revealing instability in Open5GS GTP tunnels under sustained high-throughput traffic without bandwidth control.

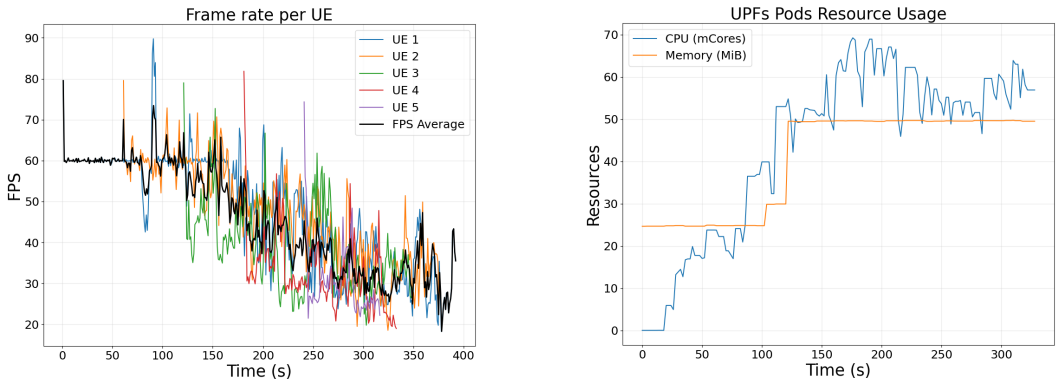
When IPerf was configured to use the full available capacity (approximately 300 Mbps in this environment), the GTP tunnel became unstable and complete service interruption occurred after approximately 2–5 minutes. This behavior is consistent with UPF resource-exhaustion conditions and required UE session re-establishment to restore connectivity.

Based on this finding, the subsequent video experiments were configured below the identified threshold, enabling a controlled evaluation of how each autoscaling strategy handles progressive load increases before reaching total service disruption.

4.2. Comparative Analysis of FPS and Resource Consumption

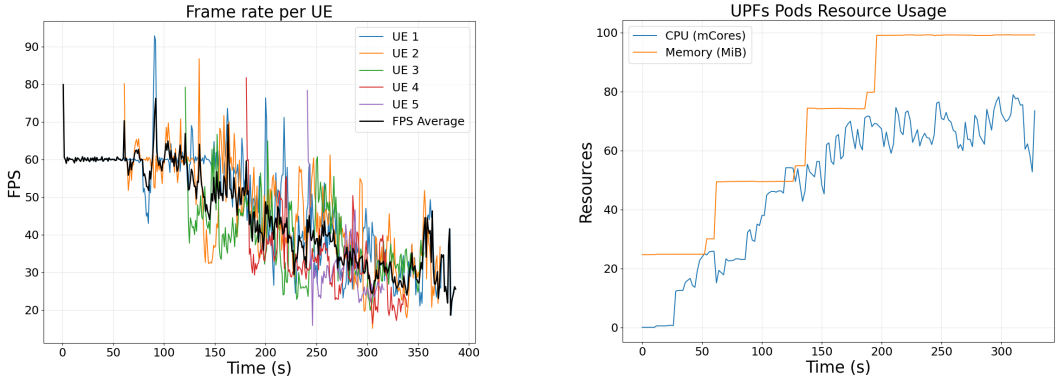
Figures 2 and 3 present the QoE and resource-consumption behavior observed in the reactive and predictive scenarios, respectively. In both cases, panel (a) shows FPS variation across UEs and panel (b) reports CPU and memory consumption.

Table 1 summarizes average FPS, memory usage, and CPU consumption across both strategies.



(a) FPS variation across UEs. (b) CPU and memory consumption.

Figure 2. Native reactive HPA baseline: (a) QoE behavior and (b) resource usage.



(a) FPS variation across UEs. (b) CPU and memory consumption.

Figure 3. Predictive HPA scenario: (a) QoE behavior and (b) resource usage.

The aggregate indicators show that both approaches operate in a similar performance range from the perspective of average FPS and CPU consumption. Average FPS remains close (39.29 in native HPA versus 38.83 in predictive HPA), and CPU usage is also comparable (44.32 versus 49.14 mCPU). The main quantitative difference appears in memory consumption, which is higher in the predictive strategy due to proactive maintenance of additional UPF replicas, reflecting a trade-off between resource efficiency and improved operational robustness.

It is noteworthy that the absence of significant variations in the received video FPS may be explained by the buffering mechanism on the receiver side, which mitigates delay and jitter fluctuations and preserves the playback rate. Therefore, under a purely average-metric perspective, both scenarios appear close in efficiency. However, average

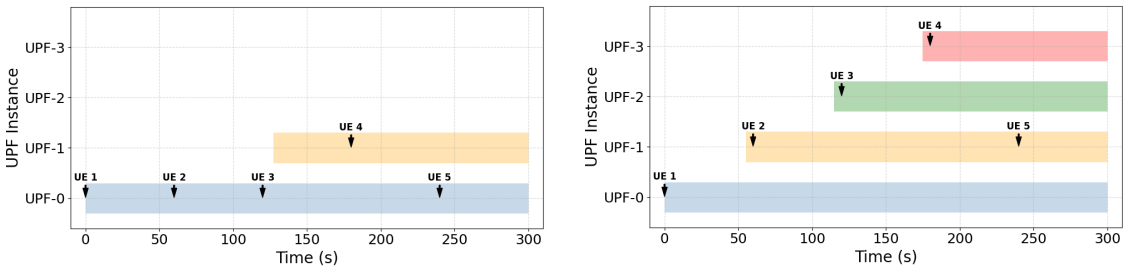
Table 1. Performance and resource comparison: native vs. predictive HPA

Method	Average FPS	Average Memory (MB)	Average CPU (mCPU)
Native HPA	39.29	40.69	44.32
Predictive HPA	38.83	69.77	49.14

values alone do not capture how each strategy behaves during session-arrival transitions, where autoscaling timing and session placement become decisive for service continuity.

4.3. UPF Instantiation and UE Association Dynamics

To examine control-plane timing and session-placement effects, Figure 4 compares the chronology of UPF instantiation and UE association for reactive and predictive scaling.



(a) Reactive HPA (native). (b) Predictive HPA (proposed).

Figure 4. UPF instantiation and UE association comparison: (a) reactive mode, delayed instantiation overloads UPF-0; (b) predictive mode, early instantiation enables balanced assignment.

In the reactive scenario (Figure 4a), new UPF instances are created only after load thresholds are exceeded. As a consequence, UEs requesting session establishment before replica readiness are anchored to already active UPFs, reinforcing load concentration. Because established GTP tunnels are persistent, these sessions are not migrated after new replicas become available, which amplifies transient overload and increases the probability of service degradation.

In contrast, the predictive strategy (Figure 4b) provisions UPF replicas before new session arrivals. This anticipation ensures that ready UPFs are available at session-establishment time, allowing the SMF round-robin policy to distribute UEs more consistently. The result is more balanced UE-to-UPF association, lower contention peaks, and improved robustness under progressive load.

Accordingly, even though average FPS and CPU metrics are similar across scenarios, the predictive approach yields superior operational performance for stateful 5G user-plane workloads, i.e., UPF instantiation timing and UE association quality.

5. Conclusions

This paper examined UPF autoscaling in private 5G environments supporting real-time video transmission, emphasizing the impact of GTP session affinity on cloud-native orchestration. While the native Kubernetes HPA can scale resources based on immediate CPU or memory pressure, it does not consider the persistence of established PDU sessions. As a result, existing UPF instances may remain overloaded while newly created replicas receive little or no traffic.

To mitigate this issue, the study evaluated a predictive autoscaling strategy that provisions new UPF instances before expected session arrivals. The experimental results, obtained in a virtualized Open5GS/Kubernetes testbed, show that the predictive approach

improves session distribution across UPFs and prevents the severe connectivity disruptions observed with the reactive HPA-based baseline. Although this strategy leads to higher average memory consumption, it offers better service continuity and reduces the operational impact of UPF saturation.

The findings indicate that predictive autoscaling mainly improves robustness for stateful 5G user-plane workloads rather than directly optimizing performance. Future work will consider broader traffic scenarios, enhanced forecasting models, and richer telemetry to jointly manage QoE, latency, and resource efficiency.

Acknowledgments

We acknowledge the Chamada Interconecta IFPB, by the Pró-Reitoria de Pesquisa, Inovação e Pós-Graduação (PRPIPG) of the Instituto Federal da Paraíba (IFPB), for the institutional support and funding that made the development of this research possible.

References

3GPP (2022). System Architecture for the 5G System (5GS); Stage 2 (Release 16). Technical Report TS 23.501 V16.10.0, 3rd Generation Partnership Project (3GPP).

3GPP (2023). Procedures for the 5G System (5GS) (Release 16). Technical Report TS 23.502 V16.16.0, 3rd Generation Partnership Project (3GPP).

Aijaz, A. (2020). Private 5G: The Future of Industrial Wireless. *IEEE Industrial Electronics Magazine*, 14(4):136–145.

Batalla, J. M. (2020). On Analyzing Video Transmission Over Wireless WiFi and 5G C-Band in Harsh IIoT Environments. *IEEE Access*, 8:118534–118541.

Botez, R., Costa-Requena, J., Ivanciu, I.-A., Strautiu, V., and Dobrota, V. (2021). SDN-Based Network Slicing Mechanism for a Scalable 4G/5G Core Network: A Kubernetes Approach. *Sensors*, 21(11):3773.

Choudhari, C. S., Patil, R., and Saraf, S. (2022). Deployment of 5G Core for 5G Private Networks. In *2022 International Conference on Industry 4.0 Technology (I4Tech)*.

Mendes de Souza, L., de Andrade da Silva, P. A., dos Santos Neto, A. A., Forcelli Silva, I., and Maciel Jr., P. D. (2025). Impacto do Posicionamento da UPF na Borda sobre a Qualidade de Serviço em Redes 5G Privadas. In *Anais do XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2025)*, Natal, RN, Brasil.

Nguyen, T.-T., Yeom, Y.-J., Kim, T., Park, D.-H., and Kim, S. (2020). Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration. *Sensors*, 20(16).

Strinati, E. C. et al. (2020). Beyond 5G Private Networks: the 5G CONNI Perspective. In *2020 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6.

Veeck, C., Barbosa, M., and Dias, K. (2025). Reagir ou Antecipar? Uma Comparação entre HPA e ML para Balanceamento de Carga. In *Anais do XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2025)*, Natal, RN, Brasil.

Yeh, S.-P., Bhattacharya, S., Sharma, R., and Moustafa, H. (2024). Deep Learning for Intelligent and Automated Network Slicing in 5G Open RAN (ORAN) Deployment. *IEEE Open Journal of the Communications Society*, 5:64–70.