Elastic UPF Selection for NextG Cellular Networks

Marco Colocrese[*†], Nitinder Mohan[†], Georgios Iosifidis[†], and Fernando A. Kuipers[†]
[*]*Technology Department*, *VodafoneZiggo*, Utrecht, the Netherlands
[†]*Networked Systems group*, *Delft University of Technology*, Delft, the Netherlands
marco.colocrese@vodafoneziggo.com, {N.Mohan, G.Iosifidis, F.A.Kuipers}@tudelft.nl

***Abstract*—Cellular operators are actively exploring the migration of core network functionality to far-edge, edge, and cloud computing infrastructures. This shift is expected to position the User Plane Function (UPF) as one of the largest contributors to workload on emerging edge platforms. Yet, operational deployments use static, manually curated gNodeB-to-UPF associations, ignoring both the elasticity that edge-to-cloud-native cores enable and the energy footprint that distributed UPFs incur. We present a measurement-driven study of elastic UPF selection that combines live Packet Data Network Gateway (PGW) measurements from a production 5G Non-Standalone network - where PGWs serve as prospective UPF locations - complemented by controlled UPF experiments across far-edge, edge, and cloud testbeds. Our measurements show that how core functions are placed can dominate end-to-end latency, incurring penalties of 37–51% relative to a co-located baseline, while average PGW utilization across th observed sites stays below 12%. Building on these insights, we formulate per-session UPF selection as a Capacitated Facility Location Problem that maps directly onto the Session Management Function (SMF), enabling standards-compliant deployment. Our framework scales to thousands of sessions on commodity solvers, jointly reducing user latency and core-network energy on infrastructure operators already own.**

I. Introduction

The 5G core is rapidly migrating from monolithic appliances onto a cloud-native computing fabric that spans far-edge sites at the radio, metro edge data centers, and hyperscale clouds [1], [2]. At the heart of this transformation sits the User Plane Function (UPF), the network function responsible for all data-plane management (e.g., routing, packet processing, and traffic aggregation) for every cellular session. Among 5G core functions, the UPF is both the most performance-sensitive, since it sets the floor on end-to-end latency for every Protocol Data Unit (PDU) session, and the most resource-hungry, accounting for more than 90% of mobile core energy consumption [3]. This combination makes UPF traffic the dominant workload and the natural focal point for elastic, edge-aware resource management in NextG networks. *Yet, operators are not exploiting this flexibility.* Despite maintaining redundant connectivity from every gNodeB (the 5G base station) to multiple UPF sites, production networks statically assign each session through manually curated associations, updated only once every few years from demographic and capacity forecasts.

Two forces make this approach increasingly costly. *First*, mobile traffic is no longer human-dominated. Machine-to-machine (M2M) now drive cellular traffic growth faster than traditional human communication [4]. Unlike human traffic, which follows predictable diurnal patterns tied to population density, machine flows are application-triggered, bursty, and geographically uneven, so static assignments calibrated to historical human demand cannot track these shifts. *Second*, modern applications impose sharply heterogeneous requirements. Extended reality, connected robotics, and autonomous systems differ by orders of magnitude in their tolerance for latency [5], [6]. A one-size-fits-all user-plane policy cannot serve them efficiently. Meanwhile, energy consumption and the carbon footprint of edge infrastructure have become first-order concerns for operators. Distributing UPFs without elastic placement risks inflating both operating cost and embodied carbon. We argue in this paper that intelligent, per-session UPF selection across the edge-cloud continuum can deliver latency benefits while constraining the energy and carbon footprint of the cellular user plane.

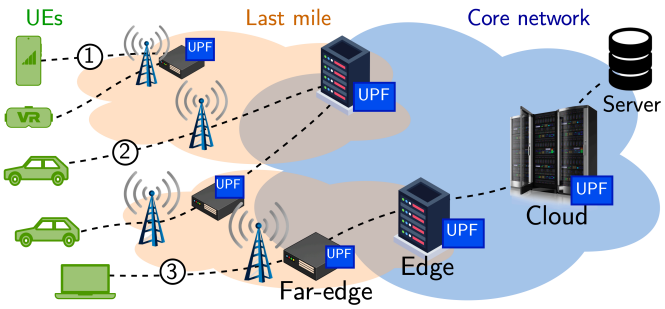

Fig. 1: Distributed UPF deployment across far-edge, edge, and cloud tiers for different traffic flows within cellular networks.

Figure 1 illustrates this opportunity across three representative session types. Local interactions ①, such as on-premise M2M traffic, are best served by a *far-edge* UPF co-located with the gNodeB. Cross-region flows ②, such as automotive telemetry traversing metropolitan distances, are best served by a *metro-edge* UPF that aggregates many radio sites. Internet-bound sessions ③ are best served from *cloud* regions where peering is dense. A single static gNodeB-to-UPF binding cannot serve all three patterns well. Dynamically matching placement to session demand can simultaneously cut user latency and consolidate energy onto fewer active UPFs.

Prior work spans three threads directly relevant to our scope. First, capacity-planning studies for UPF placement treat the decision as static and offline, determining where to instantiate UPFs once per deployment cycle [7]–[10]. We instead address per-session assignment: given the installed base, which UPF should each new PDU session use? Second, recent measurement work characterizes UPF performance in cloud-based 5G

core deployments [11], and a parallel line accelerates the UPF data path with programmable hardware [12]. Both are complementary to our focus on *per-session assignment* across an already-deployed, multi-tier network. Third, the few works addressing dynamic UPF selection rely on simplified single-tier models, omit energy as an objective, and lack grounding in real operator measurements [13], [14]. The broader edge computing community has independently identified network function placement and edge sustainability as central concerns [15]. Our work addresses both by combining live operator measurements, cross-tier power characterization, and a joint latency-energy optimizer that deploys inside the existing SMF. Concretely, this paper makes the following contributions.

1) **Live network measurement study** of PGW placement (the equivalent of UPF in 5G Non-Standalone networks for our scope) in a major European 5G operator, surfacing three insights about how today's static gNodeB-to-UPF binding under-uses the operator's edge capacity (§III). Routing through a non-closest core site over 100–200 km induces latency penalties of 37% to 51% relative to a co-located baseline, showing that core placement can be comparable to the radio link as a contributor to end-to-end delay. Average PGW utilization stays below 12% across the observed sites, and persistent load imbalance manifests despite manual rebalancing. By contrast, PGW *position* along a fixed source-to-destination path has minimal latency impact. This opens a degree of freedom for optimizing secondary objectives without sacrificing performance.
2) **Cross-tier power-performance characterization** of UPF deployments across far-edge, edge, and cloud archetypes (§IV), yielding the quantitative basis for any energy-aware selection decision. The three tiers operate in distinct power regimes and exhibit qualitatively different load coupling. The far-edge tier saturates at a hard 70% CPU threshold beyond which forwarding latency becomes unpredictable. The edge tier follows a two-regime piecewise-linear scaling driven by core activation. At the cloud tier, baseline draw dominates so heavily that consolidation, not load balancing, becomes the primary energy lever. We distill these regimes into per-tier power–throughput models that are computationally cheap enough to embed inside a dynamic selection loop directly observable at the SMF.
3) **Standards-compatible Capacitated Facility Location Problem** (CFLP) optimization framework that formulates elastic, per-session UPF selection as a CFLP jointly minimizing end-to-end latency and energy consumption at PDU-session granularity (§V). The decision logic resides natively in the Session Management Function (SMF) defined in 3GPP TS 23.501 [16] and requires no changes to the radio access network (RAN), UE, or data plane, making it deployable as a drop-in extension of existing 5G cores rather than a green-field redesign. Parameterized by the per-tier models from the previous contribution, the formulation scales to thousands of PDU sessions on commodity solvers and surfaces concrete latency–energy trade-offs against static-assignment baselines (§VI).

Crucially, our approach exploits the edge infrastructure operators have already deployed. With measurement-informed selection, operators can simultaneously improve user experience and reduce the energy and carbon footprint of the cellular user plane on infrastructure they already operate.

We release our performance measurements and optimization framework to facilitate future research [17].

II. Background and Related Work

UPF Placement. Prior work has extensively studied UPF placement to reduce latency or deployment cost in 5G networks. Many proposals focus on vertical-specific scenarios, including vehicular and IoT systems [7], [18], [19], and consistently show that placing UPFs closer to users improves performance. However, these studies typically rely on simplified network abstractions, target a single service type, and assume static deployments, limiting their applicability to large-scale public networks with heterogeneous traffic. Furthermore, UPFs are usually positioned near just one end of the connection, overlooking the other; considering both ends is essential for systematic performance gains. Other efforts jointly place UPFs and Multi-access Edge Computing (MEC) applications [8]–[10], often formulating offline optimization problems that minimize cost or resource usage under Quality of Service (QoS) constraints. While effective for capacity planning, these approaches assume relatively stable workloads and do not address runtime UPF selection across geographically distributed edge and cloud sites. Designs that push UPFs deep into the access network, such as local breakout or gNB-integrated UPFs [2], reduce latency but require extensive over-provisioning and scale poorly.

From a distributed systems perspective, these limitations mirror well-known challenges of static service placement. Prior work has shown that online, telemetry-driven decisions significantly outperform fixed configurations under dynamic workloads [20]. This is further evidenced at the MEC orchestration level, where static resource schedulers have been shown to fail under resource-constrained, multi-tenant edge deployments; even sophisticated cloud-native tools such as Kubernetes and AutoPilot prove inadequate for edge environments [21]. Elastic and adaptive mobile core designs that leverage distributed edge-cloud infrastructures are essential for maintaining performance and resilience in distributed environments [22]. Similarly, prior work has shown that maximizing server utilization is essential to avoid inefficient deployments [23], raising the question of how UPF utilization can be increased while preserving latency performance.

UPF Selection and Adaptive Control. Compared to placement, UPF selection at runtime has received limited attention. One of the few examples [13] applies deep reinforcement learning to select UPFs under latency and power constraints, but latency is modeled as a binary constraint rather than an optimization objective, and the approach does not incorporate real network measurements, routing dynamics, or link asymmetry. Measurement-based analyses of 5G core components

show that power consumption is tightly coupled with traffic load and platform characteristics [3], suggesting that effective selection policies must be informed by empirical power-performance models rather than simplified assumptions.

Traffic Granularity. Most existing models operate at the UE level, implicitly assuming homogeneous traffic per device. However, in practice, a single UE may establish multiple PDU sessions with distinct QoS requirements and destinations. Only a small number of works consider PDU-session granularity, such as [13], which lack mechanisms to exploit application- or destination-specific performance differences. Distributed systems research has established that finer-grained, flow-level decisions enable better latency control and resource efficiency in wide-area systems [24], and multi-server task scheduling [25] further demonstrate the benefits of runtime, per-task resource allocation for performance and load balancing.

Standards and Deployment. The 5G core architecture explicitly supports multi-UPF deployments and flexible anchoring. 3GPP TS 23.501 [16] specifies that the SMF selects a UPF during PDU session establishment; clause §6.3.3.3 enumerates the admissible inputs, including parameters exploited by our framework, such as the UE's location, the target Data Network Name (DNN), and the session's user-plane latency requirements. Crucially, the standard fixes this mechanism and context but leaves the decision rule open: as it states explicitly, the exact set of selection parameters "is deployment specific and controlled by the operator configuration" (§6.3.3.1). In operational networks this flexibility is typically realized through static, location-based rules. Although runtime re-selection is supported via SSC mode 3 (TS 23.501 §5.3.2.2) for mobility reasons, the standards do not define when re-selection should occur to optimize QoS, network KPIs or how the target UPF should be chosen, and public deployments remain rare (to the best of our knowledge). At the same time, vendors have introduced Deep Packet Inspection (DPI)-capable UPFs [26]–[28], and 3GPP discussions [29] have proposed packet inspection functionality to better inform UPF selection. These developments expose the necessary telemetry but fall short of defining concrete, adaptive policies that can be leveraged by operators.

Our positioning. In contrast to prior UPF placement work, which focuses on static, offline decisions, we address *UPF selection*: treating UPFs as distributed data-plane services whose suitability depends on live network conditions and hardware efficiency. Unlike prior work that optimizes latency or energy in isolation, we jointly minimize both metrics grounding on real-world measurements from a live public cellular network. We operate at PDU-session granularity rather than per-UE, enabling service-aware decisions that exploit heterogeneity in network paths and edge-cloud resources. By formulating the problem as a CFLP, we obtain principled, standards-compliant solutions that scale using established optimization techniques. Concretely, our optimizer is an instantiation of the operator-defined policy the specification leaves open: it consumes only what a standards-compliant SMF already collects and introduces no new interface, message, or telemetry plane.

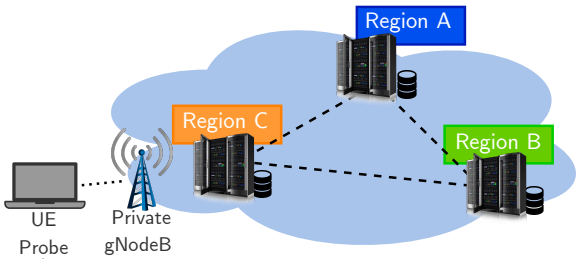

Fig. 2: [1]Illustration of the measurement setup utilizing three sites (A, B, C) within the ISP network. Each site hosts a PGW[2] along with core network components. All sites include Speedtest servers, while Region C hosts the client UE.

III. Real-World Network Performance

To quantify the impact of UPF placement, we perform measurements in a live public cellular network, evaluating how the core network and backbone topology affect latency. We investigate whether strategic placement can yield improvements for both operators and end users.

Measurement Methodology. We conduct our experiments in a major European mobile network operator's production 5G network, which ranks among the top three providers in the country, serving over 5 million subscribers and covering more than 98% of the population. Figure 2 illustrates the network architecture involved in our tests. The operator maintains different geographically distributed core locations, each hosting a PGW[2] and essential control-plane components. Among the three analyzed, two locations (A and B) also host edge application servers, while the third (C) includes both an application server and a user equipment (UE) device probe positioned in proximity to its PGW. High-capacity backbone links interconnect the three sites, spanning distances of 50 to 150 km. This architecture reflects current operational practice in public cellular networks, where multiple core sites provide redundancy and load distribution.

We measure network performance using the Ookla Speedtest [30], a widely adopted tool that captures latency, throughput, and jitter. We fix the UE at location C, connected to a private gNodeB within the same data center as the core network. The live network simultaneously handles public traffic from millions of users during our measurements. To isolate the effects of PGW placement and eliminate wireless channel variability, the operator provided a private gNodeB in an interference-free location, connected to the live core. Each test measures end-to-end throughput, latency, and jitter from the UE to application servers at locations A, B and C, while varying the PGW location among the three sites.

[1]In Figures 2–6 and Table I, the Core Network regions are color-consistent: blue is Region A, green is Region B, and orange is Region C.

[2]The analyzed network operates in 5G Non-Standalone (NSA) mode, so PGWs are used instead of UPFs. The PGW sites represent prospective locations for UPFs, and this distinction does not affect the validity of the experiments. We use UPF and PGW interchangeably throughout our measurements.

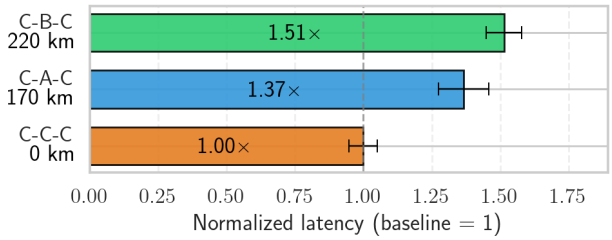

Fig. 3: Normalized latency for different PGW and server location combinations, shown as ⟨source, PGW, destination⟩. The ⟨C, C, C⟩ configuration represents the co-located baseline where UE, PGW, and server reside at the same site.

Measurements span more than one month with hourly repetitions across all configurations, capturing diurnal patterns and network dynamics.

A. Impact on Network Quality of Service

We focus on latency as the primary QoS metric. Throughput and jitter measurements do not exhibit a similarly strong dependence on PGW placement and are omitted for brevity.

We normalize all latency measurements to the fully co-located configuration ⟨C, C, C⟩. In this configuration, the UE, UPF, and application server all reside at location C and connect through a private gNodeB within the same data center. Rather than an artificial lower bound, this baseline reflects real operational conditions with minimal confounds. The private gNodeB eliminates radio interference and channel variability, the co-located core removes backbone propagation delay, and the shared data-center fabric minimizes switching latency. This configuration therefore captures the lowest achievable latency along the measured path and serves as our reference point for evaluating the cost of non-optimal PGW placement. Since confidentiality constraints with our operator partner prevent us from disclosing absolute latency values, we present all results normalized to the ⟨C, C, C⟩ baseline.

Figure 3 presents normalized latency measurements across different placement configurations. When the UE and application server both reside at location C but traffic routes through a remote PGW at A or B, latency rises to 1.37× and 1.51× the fully co-located baseline, respectively. These represent penalties of 37% and 51% incurred solely from suboptimal PGW placement over distances[3] of only 100 to 200 km. The baseline configuration already incorporates wireless channel latency and PGW processing delay under minimal load. The additional latency in remote-PGW configurations therefore originates almost entirely from network propagation and core routing overhead. Even under favorable wireless conditions, suboptimal core placement can degrade end-to-end latency by 50% or more. This qualifies the framing of prior work [31], [32], which treats the radio link as the dominant contributor to end-to-end delay.

[3]We report geographic (straight-line) distance. The operator's fiber topology correlates strongly with actual path length, estimated at approximately 1.4× the geographic distance based on regional infrastructure data.

Source location	PGW location	Destination location	Normalized latency
C	A	A	1.15
C	C	A	1.15
C	B	B	1.22
C	C	B	1.20
C	C	C	1

TABLE I: Normalized latency comparison for fixed source-destination paths with varying PGW placement. Same-color values denote the latency impact of changing core locations (UPF/PGW sites) between the same client-server pair.

Does UPF Position Along a Fixed Path Matter? The above results establish that PGW location relative to source and destination significantly affects latency. The natural follow-up is whether the PGW position along a fixed source-destination path also affects performance. We hypothesize that once the traffic path is determined, the exact UPF/PGW position becomes less critical, provided no server is overloaded. §III-A validates this hypothesis. For both source-destination pairs (C→A and C→B), we compare configurations where the PGW resides at the source endpoint versus the destination endpoint. The resulting latency distributions are nearly identical within the observed measurement variability, indicating that placement at either endpoint yields effectively equivalent performance. This invariance holds because, under low-to-moderate load, the dominant latency components (propagation delay through the backbone and baseline PGW processing time) remain constant regardless of where along the path the PGW resides.

This result carries significant architectural implications. Rather than rigidly anchoring UPFs near users or applications, operators can treat UPF placement as a degree of freedom for optimizing load distribution, energy consumption, or operational cost. The key constraint is avoiding UPF overload. Within that bound, placement flexibility incurs negligible latency penalty. More broadly, these findings indicate that effective UPF placement strategies must account for network topology, not merely computational capacity. A UPF with ample headroom can still degrade service if its position induces unnecessary backbone traversal. Conversely, geographic optimization alone is insufficient if it concentrates load on a single site.

Takeaway #1 — UPF placement relative to the UE and application server significantly affects latency, with variations up to 50% observed in a production cellular network. However, given a fixed source-to-destination path, the specific UPF position along that path has minimal performance impact, provided UPF load remains within acceptable limits.

B. Impact on Server Performance

We now examine how operators manage UPF assignments and identify inefficiencies arising from their design choices. Current operational practice relies on static traffic engineering. A network operator typically partitions gNodeB deployments in a geographical region among core locations through manual

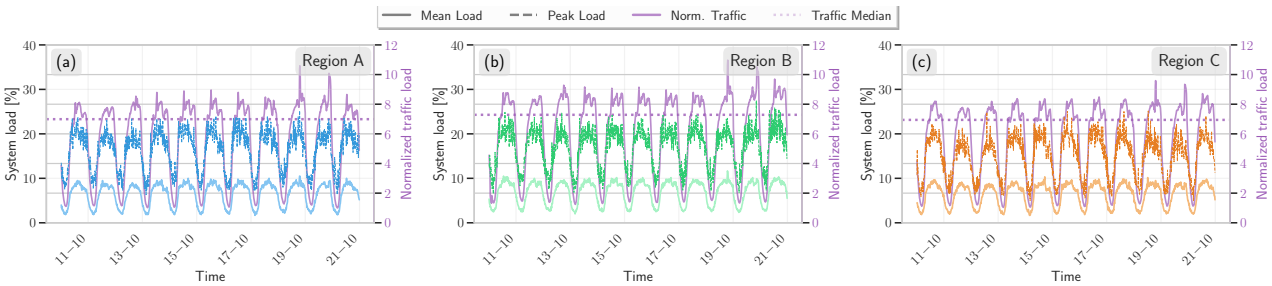

Fig. 4: Average CPU load, peak CPU load and normalized traffic load in the three core regions, measured at 15-minute granularity over a 10-day period (11-10-2025 to 21-10-2025).

configuration, updating these assignments infrequently, typically once every few years, based on demographic projections and capacity forecasts [33]. Our analyzed ISP enforces every gNodeB to maintain physical connectivity to multiple core sites for redundancy. However, only one path carries traffic under normal operation; the backup activates solely during failures. Figure 4 shows the traffic and server load across the three core locations over 10-days, examining both mean and peak utilization at 15-minute granularity. Our analysis reveals three significant consequences of this approach.

Hardware Overprovisioning. We find that the core infrastructure is substantially overprovisioned relative to actual demand. For instance, the average CPU load remains remarkably low: below 12% across all sites, falling to approximately 3% during nighttime hours. Peak utilization within 15-minute windows, which captures transient bursts lasting only seconds, rarely exceeds 30%. These figures indicate that the deployed infrastructure operates at a small fraction of its capacity for most of the day. This overprovisioning carries tangible costs and we quantify associated potential savings in §VI.

Persistent Load Imbalance. Static assignments also produce uneven load distribution that persists over time. Specifically, we find ≈ 5-10% relative difference in median load across sites in fig. 4. Further investigation reveals that the trend does not follow the geographical density of gNodeBs or population. For instance, region C, which covers the country's capital region with highest population density, consistently experiences lower load than region B. This imbalance reflects two factors: (i) demographic shifts since the original capacity planning (region B has experienced more metropolitan growth in the last decade), and (ii) the growing prevalence of machine-type traffic that does not conform to predictable diurnal patterns of human-generated traffic. In the analyzed country, M2M subscribers grew from roughly 2 million in Q4 2014 to 7 million by Q4 2019 and to over 18 million by Q4 2024, an increase of approximately 771% since the original 4G deployment. Static assignments calibrated to historical human traffic cannot adapt to these shifts.

The imbalance also varies throughout the day. Figure 5 disaggregates traffic by time slot, revealing that the relative ordering of locations changes. Location A consistently carries the lowest load, but the relationship between B and C fluctuates: during morning hours (8-10am), location C exceeds B by nearly 5%, while at other times the gap ranges from 3% to

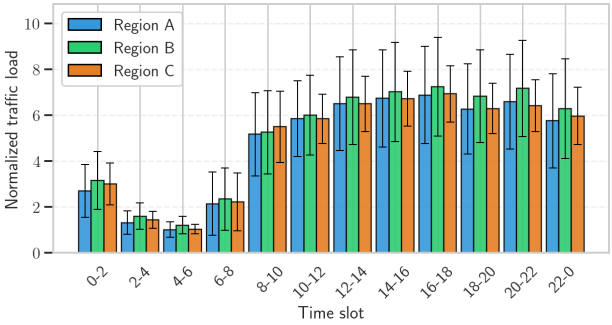

Fig. 5: Normalized traffic load across the three region during different time slots, showing time-varying imbalance patterns.

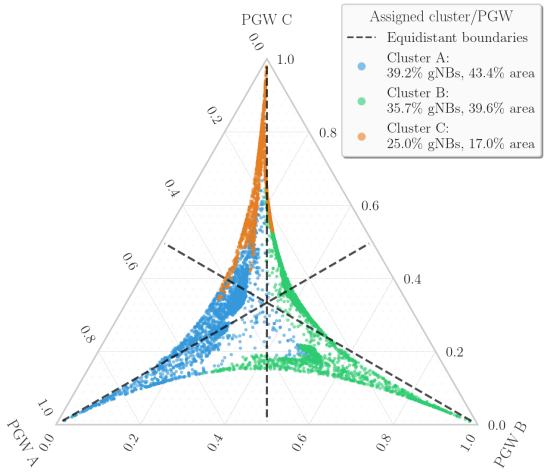

Fig. 6: Ternary plot of gNodeB assignments. Each point represents a gNodeB positioned by its normalized distance to the three PGWs. Dashed lines indicate Voronoi boundaries (equidistant points). Color indicates the assigned PGW.

11% in either direction. A static assignment optimized for one time period necessarily performs suboptimally during others.

Proximity-Latency Tradeoffs. The static assignment policy also sacrifices latency to achieve balanced load. Figure 6 visualizes this tradeoff using a ternary plot marking the normalized distance of several thousand gNodeBs operated by the ISP to each of the three core locations (denoted as PGWs in the figure). Dashed lines mark Voronoi boundaries where distances to two PGWs are equal and the color indicates the actual assigned PGW. Under a latency-minimizing policy, each gNodeB would connect to its nearest PGW, placing all

points within their respective Voronoi regions. However, the actual assignments deviate substantially from this ideal. Many gNodeBs, particularly those near the boundaries where three Voronoi cells meet, connect to PGWs that are not closest. Note that this assignment is deliberate with the aim of balancing load across the three core locations. However, we observe that such static assignments do not stand the test of time, as load patterns evolve and the initial balance degrades. For instance, core A serves 39.2% of gNodeBs despite covering 43% of the geographic area, while core C handles 25% of nodes from just 17% of the territory. The result is a persistent latency penalty for users whose gNodeBs connect to distant PGWs. Under the current low-utilization regime, with average load between 3% and 12%, this tradeoff is unnecessary. The network has ample capacity at every location, yet users still pay a latency penalty for assignments calibrated to outdated traffic patterns.

These observations motivate a fundamental shift in UPF selection strategy. Rather than static, long-term assignments that optimize for a single objective (typically load balance), we propose per-session selection that adapts to instantaneous conditions. When utilization is low, the system can prioritize proximity, assigning each session to its nearest available UPF to minimize latency. When localized congestion emerges, it can redistribute load to maintain acceptable performance across all sites. This flexibility requires no additional hardware as each gNodeB connects to multiple core locations, and the network maintains substantial spare capacity at all sites.

Takeaway #2 — Static gNodeB-to-UPF assignments, the prevailing operational practice, produce severe overprovisioning, persistent load imbalance, and suboptimal latency due to assignments that prioritize historical balance over current proximity. Elastic per-session selection can exploit existing spare capacity to improve latency without compromising load management.

IV. UPF Power Consumption and Capacity

The previous section revealed that static UPF assignments produce substantial inefficiencies: severe overprovisioning, persistent load imbalance, and elevated latency from suboptimal geographic placement. The analysis also established a key enabling result: once the source-destination path is determined, the specific UPF position along that path has minimal latency impact, provided server load remains acceptable. This flexibility opens the door to dynamic, per-session UPF selection that optimizes for secondary objectives such as energy efficiency and load balancing. *Realizing this opportunity, however, requires understanding how different UPF deployment platforms trade off power consumption, latency, and capacity.* A session routed to a low-power edge device may save energy but risk latency degradation under load. A cloud server guarantees stable, if higher, latency but consumes orders of magnitude more power. Without quantifying these tradeoffs, an optimization framework cannot make principled placement decisions. We therefore conduct controlled experiments across three representative deployment tiers: *Far-edge* (resource-constrained devices, e.g. basestation-colocated), *Edge* (moderate compute, e.g. aggregation nodes-colocated), and *Cloud* (core network datacenter servers). The resulting measurements inform the power models used in our optimization framework and establish practical operating constraints for each tier.

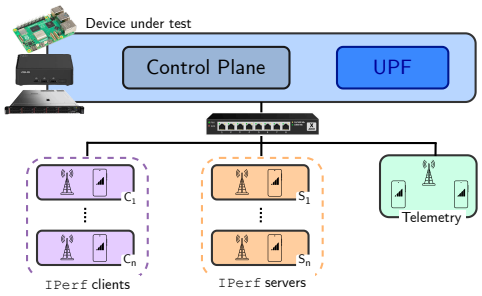

Fig. 7: Testbed for controlled UPF evaluation across far-edge, edge, and cloud tiers. The UPF under test (blue) runs on each tier-representative platform, with clients (pink) and servers (orange) injecting load and a dedicated telemetry node (green) measuring forwarding latency.

A. Experimental Setup

We evaluate each tier using identical methodology, varying only the UPF hosting hardware, its interconnect and the traffic intensity. All deployments use `Open5GS` [34], a widely used 5G core.

Our selection spans the continuum from a worst-case far-edge platform to a centralized cloud server, characterizing the constraints operators face when pushing UPF functionality toward the network edge. At the far-edge tier, compute is fundamentally limited. Base-station-co-located nodes share a constrained power envelope with radio hardware and typically operate within single-digit to low-tens of watts [35], [36]. We therefore adopt a Raspberry Pi 5 (`RPi`) as the **far-edge** representative, an Asus NUC 14 Pro (`NUC`) for the **edge** tier, and a rack server with an Intel Xeon Platinum 8444H for the **cloud** tier. Note that our hardware choice represents a conservative lower bound rather than a unique candidate for each tier: if a UPF selection strategy performs well with constrained hardware, it generalizes to any richer devices operators may employ. Furthermore, as we will show, the power-scaling behavior of each device mirrors the tier characteristics; with far-edge operating within 4-6 W, edge within 9-23 W, and cloud within 248-257 W under load.

Figure 7 illustrates the testbed. The UPF (*device under test*, in blue) runs on each tier representative, with three clients (in pink) and three servers (in orange) generating traffic across it. Each client/server device runs `UERANSIM` [37] to emulate the Radio Access Network and establish PDU sessions through the UPF. A gigabit Ethernet switch interconnects all components in the far-edge and edge experiments, matching the interface capacity of those platforms. The cloud-tier experiments instead run on our datacenter fabric, where nodes are interconnected

via 10 Gbps SFP+ links, accommodating the higher throughputs sustained by the server platform.

We employ two classes of emulated UEs to generate and monitor traffic. *Traffic UEs* run `iPerf3` clients (in pink) and servers (in orange) to inject unidirectional downstream user-plane traffic at specified throughput targets. A dedicated telemetry node (in green) continuously sends ICMP probes at 50 ms intervals to measure forwarding delay through the UPF under load. Packet traces captured at the UPF physical interfaces enable offline analysis of throughput and loss.

We capture power consumption via smart power sockets, following the methodology established in existing research [35], [38]. This records total device power, encompassing contributions from the CPU, memory, and network components. A whole-system measurement is essential for two reasons. First, the data-plane traffic that dominates UPF workloads stresses the NIC heavily, and this load is invisible to CPU utilization or load metrics alone. Second, total device power departs substantially from CPU-only power estimates reported by the operating system or produced by on-chip telemetry libraries. That gap reflects the contribution of the NIC, memory, and supporting subsystems that CPU-derived figures cannot account for. We confirm this empirically with a side-by-side measurement. Device power under a synthetic CPU stressor (`stress-ng`) differs substantially from device power under 5G data-plane traffic at the same CPU utilization. The difference is attributable to NIC and memory activity that 5G forwarding induces but CPU-only stressors do not. CPU utilization is logged via the Linux `top` utility.

B. Results

Figure 8 presents CPU utilization, power consumption, and forwarding latency as functions of throughput for each tier. The measurements reveal fundamental differences in how each platform trades off power, capacity, and latency. Note that the figure also includes dashed segments extending beyond the maximum tested throughput for each platform (imposed by hardware limitation), and are included to illustrate the expected behavior under higher loads.

Far-edge. `RPi` exposes the constraints of resource-limited deployments. CPU utilization climbs from near-idle to 90% at 300 Mbps, approaching saturation. Power consumption tracks this rise, increasing from 4.1 W at idle to 5.6 W under full load. Because idle power already accounts for roughly 75% of peak consumption, load reduction alone yields limited energy savings. Latency behavior reveals a critical operating boundary. Below 70% CPU utilization, forwarding latency remains stable and predictable. Beyond this threshold, latency variance increases sharply, with sporadic spikes reaching several milliseconds. We attribute these fluctuations to queuing delays as the device struggles to keep pace with incoming packets. The 70% boundary thus defines a hard constraint: sessions assigned to far-edge UPFs must not push utilization beyond this point if latency predictability is required.

Edge. `NUC` offers substantially greater capacity than the far-edge platform. At 975 Mbps, our maximum tested load, CPU utilization reaches only 43%, indicating that this mid-tier hardware can comfortably handle higher throughput. Power consumption ranges from 9 W at idle to 23 W under load, remaining moderate compared to datacenter equipment. Latency remains stable throughout the tested range, with one counterintuitive exception: at very low throughput, forwarding latency is slightly *higher* than at moderate load. We trace this effect to modern CPU power management. When traffic is light, the operating system assigns the UPF process to energy-efficient cores running at reduced clock speeds. As load increases, the scheduler migrates the process to high-performance cores, which process packets faster but consume more power. This behavior reveals a subtle tradeoff: operating UPF at minimum power in modern multi-core hardware does not necessarily minimize latency.

Cloud. The datacenter server operates in a fundamentally different power regime: baseline consumption dominates, and load has minimal impact. The server draws approximately 248 W at idle, rising only to 257 W at 1.7 Gbps throughput. This 9 W increase represents less than 4% of total power, meaning that traffic volume barely affects energy consumption once the server is running. This behavior has direct implications for energy optimization. Unlike edge platforms where distributing load reduces per-device power, Cloud UPFs benefit primarily from server consolidation. Powering down an idle server saves $\approx$248 W, whereas perfectly balancing load across active servers yields negligible savings. Latency remains consistently low and stable across the tested range, with none of the scheduler-induced variability observed on the edge device. Cloud deployments thus guarantee predictable latency performance, but at substantial baseline energy cost.

Cross-Platform Correlation. Figure 9 directly compares latency distributions across all three platforms under low, medium, and high throughput regimes. The comparison reveals how the power-latency tradeoff shifts with traffic intensity.

Low throughput (75-100 Mbps): All three platforms achieve tight, nearly indistinguishable latency distributions, with 90^{th} percentile values below 250 μs. Power consumption, however, differs by two orders of magnitude: `RPi` draws 4.6 W, `NUC` draws 17 W, and Cloud draws 248 W. For lightly loaded scenarios, `RPi` and `NUC` deliver equivalent latency performance at a small fraction of Cloud energy cost.

Medium throughput (150-250 Mbps): Platform differentiation emerges at moderate loads. `RPi` remains functional but exhibits growing latency variability as CPU utilization approaches the 70% threshold identified earlier. `NUC` maintains stable, predictable latency while consuming 18-20 W. Cloud continues to deliver consistent performance with minimal power increase. This regime exposes the core tradeoff: `RPi` sacrifices latency predictability for energy efficiency, while `NUC` and Cloud trade higher power for stability.

High throughput (300 Mbps and above): `RPi` approaches saturation at these loads, with latency spikes reaching several milliseconds that disqualify it for latency-sensitive applications. `NUC` continues to perform reliably at 22-23 W; our

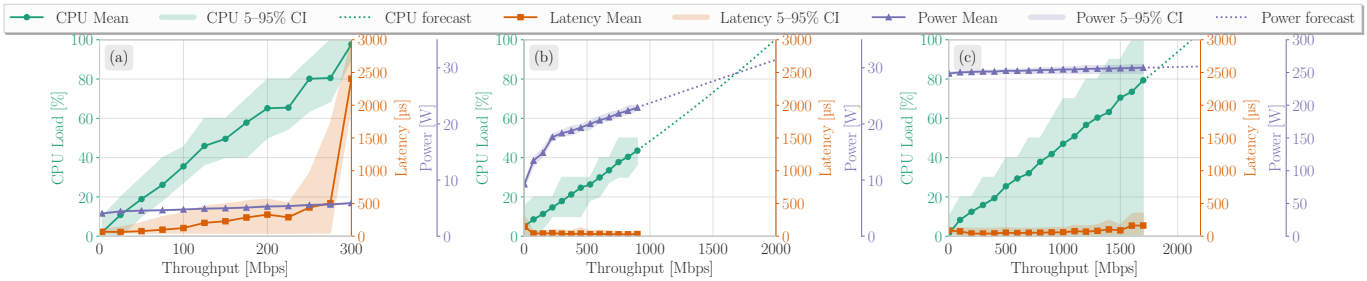

Fig. 8: CPU utilization, power consumption, and forwarding latency as functions of throughput for `RPi` (left), `NUC` (center), and Cloud server (right). Dashed segments extrapolate fitted models beyond the maximum tested throughput for each platform.

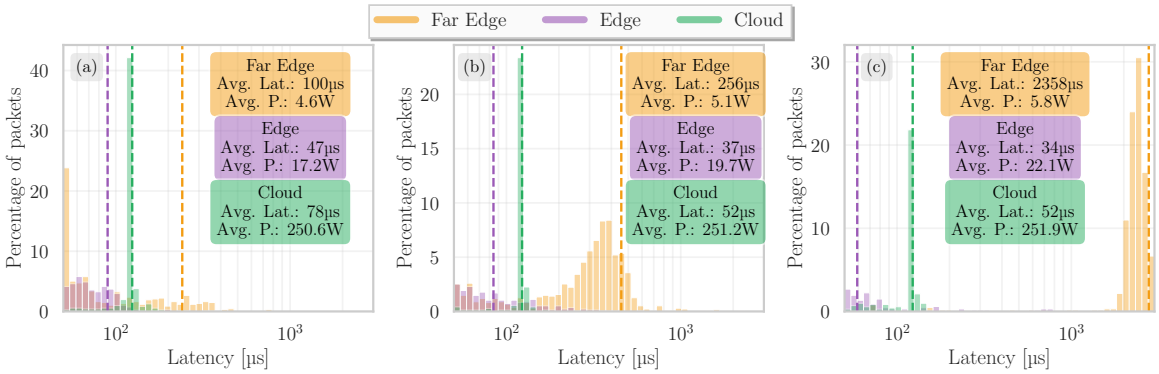

Fig. 9: Latency across tiers under low (75-100 Mbps), medium (150-250 Mbps), and high (300+ Mbps) throughput regimes.

tests reached 975 Mbps without observing degradation. Cloud handles the full tested range (up to 1.7 Gbps) with consistently stable latency as no queues build up.

Our measurements yield several concrete guidelines on device selection for UPF placement.

- **Far Edge (`RPi`):** Best suited for low-load, energy-constrained deployments where occasional latency variability is acceptable. The hard constraint is maintaining CPU utilization below 70% for latency predictability.
- **Edge (`NUC`):** Provides a balance between latency stability and power consumption. Appropriate for medium-to-high traffic requiring consistent, predictable performance.
- **Cloud:** Guarantees stable latency across all tested loads, but incurs high baseline power regardless of utilization. Best suited for latency-critical or high-throughput applications where energy cost is secondary.

One operational insight merits emphasis: for moderate aggregate loads (200-250 Mbps), deploying *multiple* `RPi` devices can outperform a single `NUC` on both energy and latency. Three `RPi` units consume approximately $3 \times 4.2 \approx 12.6$ W total, compared to 17-20 W for one `NUC`, while distributing the load keeps each device comfortably below its 70% utilization threshold. This observation motivates the multi-UPF selection strategies which we explore in our framework (§VI).

C. Power-Throughput Models

The per-tier observations of §IV-B are qualitative. For an online UPF selector to act on them, each tier needs a closed-form expression that maps the offered load to the device power it will draw. We therefore derive linear (and, for the edge tier, piecewise-linear) models relating throughput T (Mbps) to power consumption P (W). Note that the models are cheap enough to evaluate inside an optimization loop and parameterized by a quantity the operator already exposes at the Session Management Function (SMF).

We model power as a function of throughput rather than CPU utilization for two reasons. *First*, our `stress-ng` experiments demonstrate that CPU load is a poor proxy for UPF power. At the same measured CPU utilization, 5G data-plane traffic draws substantially more power than a synthetic CPU stressor, because NIC and memory activity contribute load that CPU counters cannot see. Prior measurement of virtualized 5G core power similarly finds device draw tightly coupled with traffic load [3]. We extend that observation by deriving per-tier models across heterogeneous deployment platforms suited for use inside an optimization framework. *Second*, throughput is the operator's natural control variable. Sessions are provisioned by bandwidth requirement rather than CPU budget, and per-session throughput is already observable at the SMF without additional per-UPF telemetry. This matters in practice, since fine-grained resource monitoring in MEC environments is itself non-trivial and cloud-native tools have been shown too coarse for accurate resource characterization [39].

Operators can obtain these models with a one-time offline profiling step. The UPF is driven under controlled synthetic load (e.g., `iPerf` or DPDK `pktgen`) at several throughput levels, and a linear or piecewise-linear fit to the measured power yields the per-tier coefficients reported below. This mirrors standard practice in network energy management before new device classes are deployed at scale. The fitted coefficients

(α_{FE}, β_{FE}, ...) are the only hardware-specific inputs the optimizer uses; the selection logic itself is device-agnostic. An operator running different silicon would repeat this profiling once per device class and substitute the fitted values, leaving the formulation of §V unchanged. The Raspberry Pi, NUC, and Xeon platforms should therefore be read as representative power classes rather than prescriptive hardware.

Far-edge. Power increases linearly with throughput:

$$P(T) = \alpha_{\mathrm{FE}} + \beta_{\mathrm{FE}}\, T, \tag{1}$$

with $\alpha_{\mathrm{FE}} = 4.20$ W (idle power) and $\beta_{\mathrm{FE}} = 5.44 \times 10^{-3}$ W/Mbps (marginal power per unit throughput). Each 100 Mbps of traffic adds 0.54 W, $\approx 13\%$ of idle power.

Edge. A piecewise-linear model captures the two-regime behavior induced by core activation, with breakpoint at $T_0 = 225$ Mbps:

$$P(T) = \begin{cases} \alpha_{\mathrm{E},1} + \beta_{\mathrm{E},1}\, T, & T \le T_0, \\ \alpha_{\mathrm{E},2} + \beta_{\mathrm{E},2}\, T, & T > T_0, \end{cases} \tag{2}$$

with $\alpha_{\mathrm{E},1} = 9.75$ W, $\beta_{\mathrm{E},1} = 3.6 \times 10^{-2}$ W/Mbps (low-load regime) and $\alpha_{\mathrm{E},2} = 16.05$ W, $\beta_{\mathrm{E},2} = 7.5 \times 10^{-3}$ W/Mbps (high-load regime). Below 225 Mbps, power rises steeply (3.6 W per 100 Mbps) as cores activate. Above this threshold, the system operates at full capacity and additional traffic has marginal impact (0.75 W per 100 Mbps).

Cloud. Baseline power dominates, with minimal load-dependent variation:

$$P(T) = \alpha_{\mathrm{C}} + \beta_{\mathrm{C}}\, T, \tag{3}$$

with $\alpha_{\mathrm{C}} = 248.46$ W and $\beta_{\mathrm{C}} = 5.1 \times 10^{-3}$ W/Mbps. Each 100 Mbps adds only 0.51 W, less than 0.2% of baseline. Energy optimization for Cloud UPFs must therefore target server consolidation rather than load distribution.

These models, combined with the latency thresholds identified above, provide the quantitative foundation for the optimization framework developed in the following section.

Takeaway #3 — UPF platforms exhibit distinct power-performance characteristics that must inform dynamic selection. Far-edge (4-6 W) offers energy efficiency but degrades beyond 70% CPU utilization; deploying multiple `RPi` units can outperform a single edge device for moderate loads. Edge (9-23 W) balances latency stability with moderate power consumption. Cloud servers (248-257 W) guarantee predictable latency but incur high baseline cost; energy savings require server consolidation rather than load balancing.

V. Optimization Framework

A. Problem Setting

UPF placement decisions trade end-to-end session latency against core-network device power. Reducing latency favors placing UPFs close to users, which activates many distributed edge devices that each draw idle power regardless of load. Reducing energy consumption favors consolidating traffic onto fewer, larger servers that sit further from users and add propagation delay. A similar latency-energy trade-off has been studied for edge inference workloads [40], but not at the cellular core, where UPF placement governs both propagation delay and infrastructure power draw. The two factors also map to two operator-level objectives. Service Level Agreement (SLA) commitments cap end-to-end latency for delay-sensitive sessions. On the energy side, over 70 mobile operators have committed to net-zero emissions by 2050 under the GSMA Mobile Net Zero initiative [41].

Building on the live-network measurements (§III) and the per-tier characterization (§IV), we cast elastic UPF selection as a joint optimization at PDU-session granularity. Each session, defined by its ⟨source, destination⟩ pair, is routed through a selected UPF along one of the feasible network paths at its required traffic demand. The objective minimizes the aggregate session latency and the cost of UPF deployment and operation. Each tier's operating cost enters the optimizer as a single linear term: a fixed idle power charged when the UPF is active, plus a marginal coefficient p_u per unit throughput. For the far-edge and cloud tiers this is the linear fit of §IV-C directly; for the edge tier we use the high-load segment ($\alpha_{E,2}$, $\beta_{E,2}$) of the piecewise model in Eq. (2) as a single-line approximation, since an active edge UPF aggregating many sessions operates well above the $T_0 = 225$ Mbps breakpoint. These coefficients are the sole point of contact between the physical hardware and the optimizer: substituting operator's own profiled devices changes their numerical values but not the structure of the objective or constraints, making the framework portable across heterogeneous UPF deployments. For routing latency, we assume the contribution of a given UPF is approximately constant within its safe operating range, where queuing, jitter, and congestion remain dominated by the linear processing term. Production 5G deployments engineer UPFs to operate inside this regime, so the linear abstraction is sufficient for the design-time analysis pursued here.

Cadence and re-assignment. The framework runs periodically, at intervals on the order of an hour or whenever the offered demand shifts. Each run produces a UPF-to-session mapping. The mapping is consumed at the existing UPF-selection step that the Session Management Function (SMF) performs at every PDU-session establishment, per 3GPP TS 23.501 [16]. A session arriving after a recomputation therefore binds to the UPF the framework currently prescribes, with no change to the RAN, the UE, or the data plane. For sessions already active when the mapping changes, the simplest policy leaves each session on its current UPF until it terminates, so re-binding affects only future sessions. Alternatively, the operator can migrate active sessions mid-flight via methods like the standards-defined SSC mode 3 procedure, which re-anchors a PDU session at a new UPF at the cost of a UE IP-address change. The evaluation in §VI adopts the leave-them-be policy and assumes new sessions inherit the latest mapping. The transient QoS cost of mid-session re-anchoring is the subject of ongoing standardization effort, which we

Symbol	Description
S	Set of PDU sessions
U	Set of UPFs
T_s	Session s traffic demand (Mbps)
T_u	UPF u throughput capacity
$x_{s,u}$	Binary decision variable for assigning session s to UPF u
y_u	Binary decision variable for UPF u activation
p_u^{idle}	Idle power of UPF u
p_u	Load-dependent power coefficient of UPF u
$d_{s,u}$	Latency of session s via UPF u
$d_s^{\min}$	Minimum achievable latency for session s
$d_{s,u}^{\text{fix}}$	Fixed propagation delay to UPF u
l_u	Latency introduced by UPF u routing
α	Weight for latency term
β_s	Session s latency sensitivity
γ	Weight for UPF power term

TABLE II: Notation Summary

summarize in §VII.

B. Optimization Model

We now distinguish between problem inputs and modeling assumptions with table II describing the used notation.

Problem Inputs. For each PDU session $s \in S$, we have: (i) a session traffic demand T_s (Mbps), and (ii) the fixed propagation delay d^{fix} to each UPF $u \in U$, determined by the physical network topology and fiber path length. Each UPF u has a finite processing capacity T_u and can be either active or inactive. In line with the observations in fig. 8, we assume that the processing delay introduced by a UPF can be modeled as a fixed value l_u, considering that it operates in a safe operating range. The power consumption of an active UPF consists of a fixed idle component p_u^{idle} and a load-dependent term proportional to the carried throughput, modeled as $p_u T_s$. The objective is to minimize a weighted combination of two components: (i) a latency-aware term that penalizes the gap between the latency experienced by a session and its minimum achievable latency $d_s^{\min}$, where $d_s^{\min}$ denotes the latency attainable in an unloaded network with all UPFs active; and (ii) the power consumption of active UPFs.

We focus on minimizing the deviation from $d_s^{\min}$ rather than the absolute latency, so as to explicitly account for the physical limits imposed by the network topology and geographical distances. The resulting optimization problem is:

$$\min_{\{x_{s,u},\, y_u\}} \quad \alpha \cdot \frac{1}{|S|} \sum_{s\in S} \sum_{u\in U} \beta_s \frac{x_{s,u}\big(d_{s,u}^{\text{fix}} + l_u - d_s^{\min}\big)}{\max_{s,u}(d_{s,u} - d_s^{\min})} + \gamma \cdot \frac{\sum_{u\in U} p_u^{\text{idle}} y_u + \sum_{s\in S}\sum_{u\in U} p_u\, y_u\, x_{s,u} T_s}{\sum_{u\in U} p_u^{\text{idle}}} \tag{4}$$

$$\text{s.t.} \sum_{u\in U} x_{s,u} = 1, \qquad \forall s \in S \tag{4.1}$$

$$\sum_{s\in S} T_s x_{s,u} \le T_u y_u, \qquad \forall u \in U \tag{4.2}$$

$$x_{s,u} \in \{0,1\},\ y_u \in \{0,1\}, \qquad \forall s \in S, u \in U \tag{4.3}$$

In the formulated problem, the binary variable $x_{s,u}$ indicates whether PDU session s is assigned to UPF u. Accordingly, each session must be associated with exactly one UPF, which is enforced by (4.1)[4]. The binary variable y_u determines whether a UPF is activated at location u. A session can only be routed through an active UPF, and the aggregate throughput of the sessions assigned to a UPF must not exceed its processing capacity. Both conditions are captured by (4.2).

Finally, the weighting parameters α and γ regulate the trade-off between latency performance and energy efficiency, respectively. The coefficient β_s captures the latency sensitivity of session s, allowing the model to differentiate among heterogeneous classes of sessions with varying latency requirements.

C. Complexity Discussion

Now that the problem has been formulated, it is important to understand its computational complexity to assess whether it can be solved efficiently at the intended time scale. To this end, we examine the structure of the problem and identify its connection to a well-known NP-hard problem, namely the Capacitated Facility Location Problem (CFLP) [42], [43].
Firstly, we aggregate constants to reach a simpler formulation:

$$\alpha' = \frac{\alpha}{|S| \cdot \max_{s,u}(d_{s,u} - d_s^{\min})}, \quad \gamma' = \frac{\gamma}{\sum_{u\in U} p_u^{\text{idle}}}.$$

Then, to see this correspondence, we regroup terms in the objective according to whether they depend on UPF activation variables y_u or session assignment variables $x_{s,u}$. The optimization problem can then be rewritten as:

$$\min_{x,y} \underbrace{\sum_{u\in U} \Big(\gamma' p_u^{\text{idle}} y_u + \gamma' p_u y_u \sum_{s\in S} T_s x_{s,u}\Big)}_{\text{Power consumption term}} + \underbrace{\sum_{s\in S}\sum_{u\in U} \alpha' \beta_s\, x_{s,u}\big(d_{s,u}^{\text{fix}} + l_u - d_s^{\min}\big)}_{\text{Latency term}} \tag{5}$$

Subject to constraints (4.1), (4.2) and (4.3).
Eq. (5) now matches the Mixed Integer Linear Programming (MILP) definition of CFLP:

$$\min_{x,y} \sum_{u\in U} F_u y_u + \sum_{s\in S}\sum_{u\in U} C_{s,u} x_{s,u}, \tag{6}$$

where

- $F_u = \gamma' p_u^{\text{idle}}$ *fixed cost* of opening (activating) UPF u;

[4]This formulation implicitly assumes that all sessions are admissible. The model can be extended to account for session rejection if required.

- $C_{s,u} = \alpha' \beta_s (d^{\text{fix}}_{s,u} + l_u - d^{\min}_s) + \gamma' p_u T_s$ acts as the *assignment cost* of serving session s via UPF u, combining both latency and traffic-dependent energy cost.

Unlike the classical CFLP, our formulation introduces a coupling term ($p_u T_s$) in the assignment cost, making it load-dependent. Nevertheless, this structure preserves the same binary nature and constraint set of the CFLP. The mapping enables direct reuse of the algorithmic toolbox developed for the CFLP [44], including exact and heuristic methods that can scale with known optimality guarantees.

VI. Optimization Evaluation

A. Setup and Methodology

We evaluate the optimization framework on an operator-derived topology, comparing optimized UPF assignments against three heuristic baselines across a range of traffic loads and topology sizes. The network topology reflects the infrastructure of the real mobile operator analyzed in this work (depicted in fig. 1). It follows a hybrid hierarchy with three layers: cloud datacenters, regional clusters, and local clusters. Each local cluster forms a mesh of ≈ 5 gNodeBs connected through a local aggregation node. Each regional cluster aggregates ≈ 10 such local clusters (approximately 50 gNodeBs), covering a total area of about $25000\ \text{km}^2$. This structure matches independently documented topologies from other mobile networks including AT&T's [45].

The number of PDU sessions ranges from 50 to 5,000. We test two topology sizes: 112 UPFs with 500 gNodeBs, and 222 UPFs with 1,000 gNodeBs. Topology scale has negligible impact on solver complexity compared to session count, consistent with the $\mathcal{O}(|\mathcal{S}| \cdot |\mathcal{U}|)$ growth of assignment variables in the CFLP formulation. Sessions are distributed evenly across four traffic patterns: (i) 25% directed to external destinations through the central datacenter (representing Internet Exchange Point coupling or cloud/CDN - Content Delivery Network - traffic), (ii) 25% to destinations attached to the same local far-edge router (edge applications), (iii) 25% to endpoints reachable through the edge UPF, and (iv) 25% to edge-cloud datacenters hosting edge UPFs (cloud/CDN workloads). These four classes span the major session archetypes in cloud-native 5G deployments, corresponding to the local, regional, and Internet-bound traffic types illustrated in fig. 1. The equal split is a conservative baseline. Operators with measurement-based traffic profiles can substitute observed distributions directly.

Each session is further parameterized by a minimum throughput requirement T_s and a latency-sensitivity weight β_s. Session traffic demands are drawn uniformly at random from $[1, 100]$ Mbps, reflecting the per-session bandwidth of typical PDU sessions, while β_s is drawn uniformly from the discrete set $\{0, 1, 5, 10\}$, spanning latency-insensitive background flows ($\beta_s = 0$) to delay-critical sessions ($\beta_s = 10$). These parameters are synthetically generated; the ranges are chosen to exercise the full latency-energy trade-off across heterogeneous session classes. As with the traffic split, operators with measurement-based profiles can substitute empirical T_s and β_s distributions directly, leaving the formulation unchanged.

Power consumption follows the per-tier models from §IV-C: local aggregation nodes map to far-edge UPFs, regional aggregation nodes to edge UPFs, and the central datacenter to the cloud UPF. Each UPF's usable capacity T_u is capped at 80% of its nominal throughput, reserving headroom for load fluctuation and preventing assignments from driving any device to saturation. We model UPF-introduced latency as constant, motivated by the small values observed in our live-network measurements (§III-A).

We solve the optimization using *MOSEK* [46] and *Gurobi* [47] as independent cross-validators. For all instances up to 5,000 sessions, both solvers converge to nearly identical objective values, confirming practical optimality of the solutions. We only show Gurobi for brevity as MOSEK's results are nearly identical. We compare the optimized solutions against three widely used heuristic baselines.

- **Source Centric** (inspired by [2], [7], [18], [19]): each session is assigned to the nearest feasible UPF to its originating node — UPFs are ranked by delay-weighted shortest-path proximity, and a session spills over to the next-nearest UPF whenever its closest choice has reached capacity. This minimizes latency but activates many UPFs.
- **Destination Centric** (inspired by [8]–[10]): identical spillover logic as the Source centric, but sessions rank UPFs by proximity to the destination endpoint rather than the source, partially balancing energy and delay.
- **Cloud UPF only** (inspired by live-network observations): sessions are routed to the most central core sites first, spilling to the next central location only when capacity is exhausted. At low session counts a single central site absorbs all traffic; additional central sites activate only as load grows. This mirrors the *static, centralized* character of current operator practice (§III-B), where static gNodeB-to-core assignments concentrate traffic and are rebalanced only through infrequent manual reconfiguration

Together these baselines span the practical policy space against which an elastic optimizer must be judged: latency-greedy with capacity fallback (Source/Destination Centric) and consolidation-greedy (Cloud UPF), the latter reflecting the static, infrequently rebalanced assignment operators deploy today (§III-B).

The optimization uses two weights: α for latency and γ for power, with $\alpha = 1$ fixed throughout. We sweep γ over $[0.01, 100]$ to explore the full latency-energy trade-off. When $\gamma < 1$, the latency term dominates and the optimizer activates many UPFs, preferring geographically closer devices to reduce end-to-end session delay. When $\gamma > 1$, the energy term dominates and the optimizer consolidates sessions onto fewer active UPFs to reduce total power consumption.

All experiments run on a single commodity server (12-core Intel Xeon E5-2620 v3 2.40GHz, 32GB RAM), under a user-defined solver time limit of 600s. For instances below 500 sessions both solvers close the optimality gap and certify the solution within this budget. For larger instances Gurobi occasionally returns at the time limit before formally certifying optimality; in every such case the terminal MIP gap stays

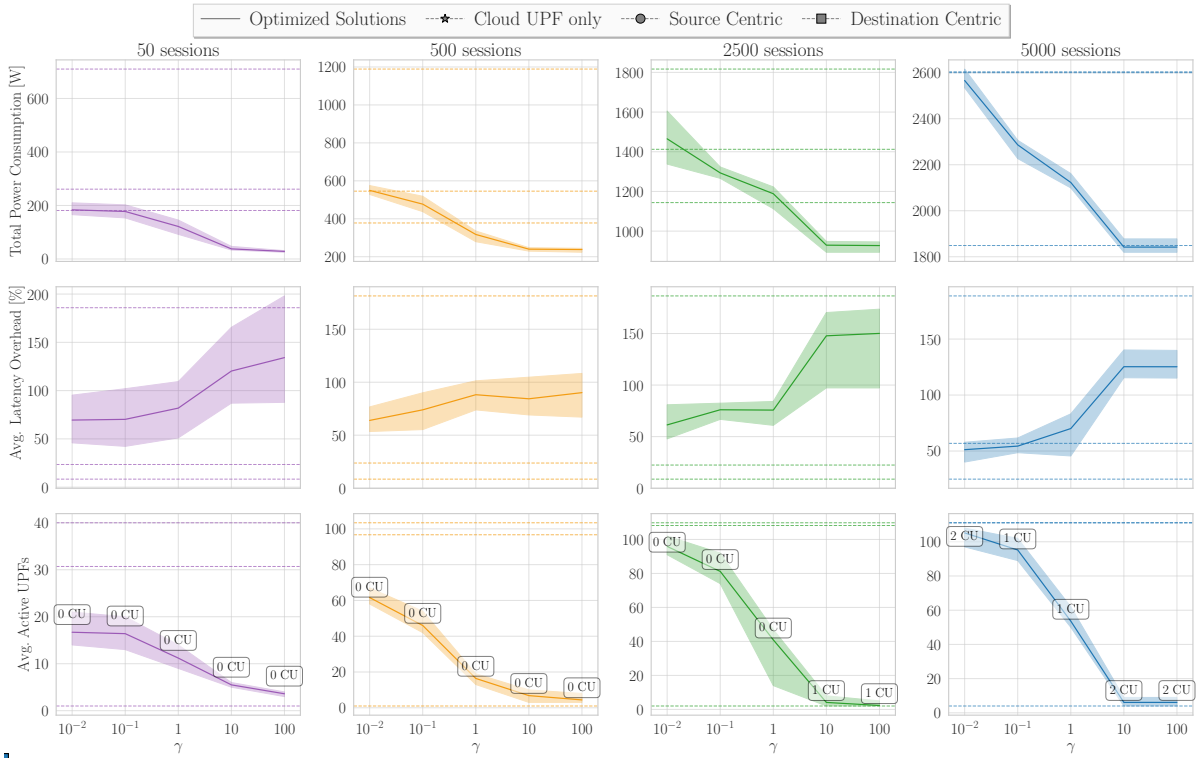

Fig. 10: Optimized UPF selection and three heuristic baselines evaluated across session volumes at varying energy weight γ. The first row reports aggregate power drawn by all active UPFs. The second row reports average latency overhead relative to an ideal zero-load reference, where all UPFs are available and unloaded. The third row shows the average number of active UPFs, with active cloud UPF (CU) counts annotated per panel. Horizontal dashed lines mark heuristic solutions, whose performance is invariant to γ. Note the different y-axis scales across plots.

below 0.01% and the returned assignment coincides with the independently computed MOSEK objective, so the solution is optimal for all practical purposes. The time limit is itself set orders of magnitude below the planning cadence at which the framework runs (§V-A), so early termination carries no operational cost.

B. Quantitative Results

Figure 10 compares the optimized solutions against all three baselines across increasing session volumes on both the power and latency objectives. The first row shows total power consumed by all active UPFs. The second row reports average *latency overhead*: the deviation from the ideal reference latency in which all UPFs are available, unloaded, and free of concurrency. Results are averaged over 10 runs with different random seeds, each producing distinct topology and traffic realizations with identical node and session counts. Shaded regions denote min-max bounds across runs. The third row tracks the average number of active UPFs, with the count of active cloud UPFs (CU) annotated within each panel.

The heuristic baselines rely on fixed anchoring rules, activating many lightly loaded UPFs and yielding inferior power-latency trade-offs. The optimized solutions, by contrast, exhibit clear elasticity. By tuning α/γ, the optimizer smoothly shifts traffic between edge and cloud UPFs, consolidating flows when beneficial. For low traffic volumes (e.g., 50 sessions), the *Source Centric* heuristic can consume less power than the *Central UPF* baseline by exploiting nearby edge devices. As traffic increases, however, source- and destination-centric heuristics activate too many edge UPFs without sufficient utilization to justify the activation cost. The optimizer instead selectively enables fewer edge devices while still exploiting their geographic proximity, achieving lower power consumption at the same session count. At high loads (e.g., 5,000 sessions), the optimized solution converges toward the *Central UPF only* baseline, since edge UPFs alone lack sufficient capacity and all strategies must ultimately rely on the cloud. The optimizer reaches this regime without the unnecessary edge activation that the heuristics incur at intermediate loads. In terms of latency, the optimized solutions span the full interval of achievable behaviors defined by the heuristic baselines. For small γ, latency closely resembles that of the latency-oriented heuristics. Increasing γ progressively shifts the solution toward higher latency overhead as power minimization becomes dominant, consistent with the formulation in §V. No fixed-rule baseline can access this continuous range. The optimizer is the only mechanism that places the operating point anywhere on the latency-energy Pareto frontier, adjusting to heterogeneous operator priorities without changing the underlying infrastructure.

The average number of active UPFs (third row of fig. 10)

reveals an additional behavioral pattern. At small γ, consistent with the latency-dominant regime, the optimizer avoids the cloud UPF and distributes sessions across many geographically proximate far-edge and edge devices. At large γ and high session counts, the energy-dominant regime instead consolidates traffic onto the cloud UPF. A single cloud server is more energy-efficient than tens or hundreds of simultaneously active far-edge and edge devices, each drawing baseline idle power. An instructive exception arises at 5,000 sessions. Even under strong latency prioritization (small γ), the optimizer activates two cloud UPFs. This counterintuitive cloud preference at small γ follows from two earlier findings. First, far-edge devices degrade unpredictably beyond 70% CPU utilization, as shown in §IV-B. At 5,000 sessions, the far-edge and edge devices along the optimal paths approach this saturation threshold, making additional load inadvisable. Second, the specific UPF position along a fixed source-destination path has minimal latency impact, provided the device is not overloaded (§III-A). The cloud UPF, lying along the same paths, therefore becomes the preferred assignment. The optimizer selects it not for lower propagation delay, but because the otherwise-optimal edge devices are already saturated. This effect disappears at intermediate γ (e.g., $\gamma \in [0.1, 1]$). At those settings, the optimizer turns off one cloud UPF, preferring mild edge load over the cloud's high baseline power cost. At large γ (e.g., $\gamma \geq 10$), the optimizer reactivates cloud UPFs to consolidate energy. The active device count collapses from nearly 100 to a few units, erasing the idle-power penalty accumulated across the far-edge and edge tier.

C. Scalability and Practical Deployment

While we evaluate up to 5,000 sessions in this work, the CFLP formulation maps to a richly studied combinatorial optimization class with a mature algorithmic ecosystem. Our MILP has $\mathcal{O}(|\mathcal{S}| \cdot |\mathcal{U}|)$ binary variables and $\mathcal{O}(|\mathcal{S}| + |\mathcal{U}|)$ constraints, well within the benchmark regimes where specialized heuristics, decomposition methods, and Lagrangean relaxation achieve near-optimal solutions at a fraction of MILP cost [43], [48]. Recent approaches, including Sinkhorn-OT solvers [49], Kernel Search heuristics [43], [50], RAMP algorithms [51], and large-scale network heuristics [52], routinely handle millions of assignments at near-optimal quality. The formulation supports geographic decomposition, where subregions are solved independently and combined without loss of feasibility. Operator-grade deployments can therefore combine exact MILP for moderate instance sizes with modern CFLP heuristics or decomposition strategies for very large instances.

Crucially, the optimization does not need to run in real time. Session-to-UPF assignments remain stable within planning intervals, so operators can execute the optimizer periodically (e.g., hourly or upon significant traffic-pattern changes).

VII. Discussion and Conclusion

Cloud-native 5G deployments spread UPFs across far-edge, metro-edge, and cloud tiers, yet production networks still rely on static gNodeB-to-UPF bindings updated only once every few years. Measurements in a live European operator network show that suboptimal core placement inflates end-to-end latency by 37% and 51% over paths of 100–200 km. Average PGW (our proxy for UPF) utilization stays below 12%, and load imbalance persists despite periodic manual rebalancing. We formulated per-session UPF selection as a Capacitated Facility Location Problem that jointly minimizes latency and energy. The optimizer reaches operating points no fixed-rule baseline can access, with tractable solutions on commodity solvers and a proven scale-out path via established CFLP algorithms (§VI-C). The framework requires no new hardware and deploys as a standards-compatible SMF extension with no changes to the radio access network, the UE, or the data plane.

We note the following limitations and directions for future work. The power-performance characterization in §IV uses three platforms spanning the far-edge, edge, and cloud power classes (4–6 W, 9–23 W, 248–257 W baseline), which may not match every operator's deployed hardware. However, the profiling methodology in §IV-C is fully repeatable. Any operator can characterize their own UPF hardware under controlled synthetic load, fit a linear or piecewise-linear power model, and substitute the resulting coefficients into the framework unchanged. The evaluation also targets steady-state placement, applying updated assignments only to newly arriving sessions and leaving active sessions undisturbed (§V). Operators requiring mid-session re-anchoring can invoke the standards-defined procedures, such as the one used in SSC Mode 3, at the cost of a UE IP-address change. Modelling the transient QoS cost of session migration under SSC Mode 3 or similar approaches is an important direction for future work. Finally, while we evaluate up to 5,000 sessions, the CFLP formulation maps to a well-studied combinatorial optimization class with a mature algorithmic ecosystem (§VI-C). For operator-grade deployments involving very large numbers of sessions or UPF sites, exact MILP solvers can be complemented by specialized heuristics, decomposition strategies, or metaheuristic refinements [43], [44], [48]–[52]. We consider the systematic evaluation of these techniques under realistic operator-scale topologies an important direction for future work.

References

[1] M. Hogan, G. Wan, Y. Qiu, S. Agarwal, R. Beckett, R. Singh, and P. Bahl, "Efficient Multi-WAN transport for 5g with OTTER," in *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*, 2025.

[2] R. Paudyal, R. Upadhyay, A. N. B. Emran, and D. Wijesekera, "gNB-based Local Breakout for URLLC in industrial 5G," in *2025 IEEE 30th CAMAD*, 2025.

[3] A. Bellin, F. Granelli, and D. Munaretto, "A Measurement-Based Approach to Analyze the Power Consumption of the Softwarized 5G Core," *Computer Networks*, 2024.

[4] The Organisation for Economic Co-operation and Development (OECD), "Broadband statistics," https://www.oecd.org/en/topics/broadband-statistics, 2024.

[5] N. Mohan, L. Corneo, A. Zavodovski, S. Bayhan, W. Wong, and J. Kangasharju, "Pruning edge research with latency shears," in *19th ACM Hot Topics in Networks*, 2020.

[6] A. Mallik, J. Xie, and Z. Han, "A performance analysis modeling framework for extended reality applications in edge-assisted wireless networks," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, 2024.

[7] P. Fondo-Ferreiro, D. Candal-Ventureira, F. J. González-Castaño, and F. Gil-Castiñeira, "Latency Reduction in Vehicular Sensing Applications by Dynamic 5G User Plane Function Allocation with Session Continuity," *Sensors*, 2021.

[8] E. Goshi, H. Harkous, S. Ahvar, R. Pries, F. Mehmeti, and W. Kellerer, "Joint UPF and Edge Applications Placement and Routing in 5G & Beyond," in *2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*, 2024.

[9] I. Leyva-Pupo, A. Santoyo-González, and C. Cervelló-Pastor, "A Framework for the Joint Placement of Edge Service Infrastructure and User Plane Functions for 5G," *Sensors*, 2019.

[10] Y. Li, X. Ma, M. Xu, A. Zhou, Q. Sun, N. Zhang, and S. Wang, "Joint Placement of UPF and Edge Server for 6G Network," *IEEE Internet of Things Journal*, 2021.

[11] T. Atalay, D. Stojadinovic, A. Famili, A. Stavrou, and H. Wang, "5G-MAP: Demystifying the Performance Implications of Cloud-Based 5G Core Deployments," in *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2025.

[12] A. Bose, S. Kirtikar, S. Chirumamilla, R. Shah, and M. Vutukuru, "AccelUPF: Accelerating the 5G User Plane using Programmable Hardware," in *Proceedings of the Symposium on SDN Research (SOSR)*. ACM, 2022.

[13] A. Bellin, N. Di Cicco, D. Munaretto, and F. Granelli, "Power consumption-aware 5g edge upf selection using deep reinforcement learning," in *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2024.

[14] Y. Liu, Y. Mao, X. Shang, Z. Liu, and Y. Yang, "Energy-Aware Online Task Offloading and Resource Allocation for Mobile Edge Computing," in *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*, 2023.

[15] J. Sun, Y. Zhang, F. Liu, H. Wang, X. Xu, and Y. Li, "A survey on the placement of virtual network functions," *Journal of Network and Computer Applications*, 2022.

[16] 3GPP, "3GPP TS 23.501 - System Architecture for the 5G System (5GS)," 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 23.501, 2024, version 18.6.0.

[17] Elastic UPF Selection for NextG Cellular Networks - Source code. [Online]. Available: https://github.com/SPEAR-Research-Lab/Elastic_UPF_Selection_for_NextG_Cellular_Networks

[18] P. Fondo-Ferreiro, F. J. Gil-Castiñeira, F. J. González-Castaño, D. Candal-Ventureira, J. Rodriguez, A. J. Morgado, and S. Mumtaz, "Efficient Anchor Point Deployment for Low Latency Connectivity in MEC-Assisted C-V2X Scenarios," *IEEE Transactions on Vehicular Technology*, 2023.

[19] N. Makondo, H. I. Kobo, T. E. Mathonsi, and D. P. D. Plessis, "Implementing an Efficient Architecture for Latency Optimisation in Smart Farming," *IEEE Access*, 2024.

[20] G. Miotto, M. Caggiani Luizelli, W. Cordeiro, and L. Gaspary, "Adaptive placement & chaining of virtual network functions with nfv-pear," *JISA*, 2019.

[21] A. Haroon, L. Jin, R. Stoleru, M. Maurice, and R. Blalock, "Edgecore: Resource dependency-aware multi-tenant orchestration for mobile edge clouds," in *2024 IEEE/ACM Symposium on Edge Computing (SEC)*, 2024.

[22] J. Larrea, A. E. Ferguson, and M. K. Marina, "Corekube: An efficient, autoscaling and resilient mobile core system," in *ACM MobiCom*, 2023.

[23] A. Santoyo-González and C. Cervelló-Pastor, "Network-aware placement optimization for edge computing infrastructure under 5g," *IEEE Access*, 2020.

[24] G. L. Santos, D. de Freitas Bezerra, Élisson da Silva Rocha, L. Ferreira, A. L. C. Moreira, G. E. Gonçalves, M. V. Marquezini, Ákos Recse, A. Mehta, J. Kelner, D. Sadok, and P. T. Endo, "Service Function Chain Placement in Distributed Scenarios: A Systematic Review," *JNSM*, 2022.

[25] C. Li, R. Fan, H. Wang, M. Han, S. Wu, F. Li, and P. Hu, "TSAJS: Efficient Multi-Server Joint Task Scheduling Scheme for Mobile Edge Computing ," in *IEEE 45th ICDCS*, 2025.

[26] Enea, "Deep Packet Inspection and Traffic Intelligence for 5G User Plane Function (UPF)," https://www.enea.com/solutions/deep-packet-inspection-traffic-intelligence/networking/5g-user-plane-function/, 2024.

[27] Z. Corporation, "ZTE's High-Performance 5G UPF DPI Implementation Based on Intel TADK," https://builders.intel.com/docs/networkbuilders/zte-s-high-performance-5g-upf-dpi-implementation-based-on-intel-tadk-1660802600.pdf, 2023.

[28] C. Systems, "Cisco Ultra Cloud Core UPF Configuration and Administration Guide," https://www.cisco.com/c/en/us/td/docs/wireless/ucc/upf/2024-01/config-admin/ucc-5g-upf-config-and-admin-guide_2024-01/m_inherited-dpi-charging-rules-action.html, 2024.

[29] 3GPP, "Tsg-sa working group 2 (sa2), meeting #162, s2-2407555," Maastricht, Netherlands, 2024.

[30] Ookla. Ookla, LLC. [Online]. Available: https://www.ookla.com

[31] T. K. Dang, N. Mohan, L. Corneo, A. Zavodovski, J. Ott, and J. Kangasharju, "Cloudy with a chance of short rtts: analyzing cloud connectivity in the internet," in *Proceedings of the 21st ACM IMC*, 2021.

[32] I. Hadžić, Y. Abe, and H. C. Woithe, "Edge computing in the epc: a reality check," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, 2017.

[33] C. de Andrade, A. Mahimkar, R. Sinha, W. Zhang, A. Cire, G. Rana, Z. Ge, S. Puthenpura, J. Yates, and R. Riding, "Minimizing effort and risk with network change deployment planning," in *2021 IFIP Networking Conference (IFIP Networking)*, 2021, pp. 1–9.

[34] Sukchan Lee, "Open5GS," https://github.com/open5gs.

[35] J. A. Ayala-Romero, I. Khalid, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Experimental Evaluation of Power Consumption in Virtualized Base Stations," in *IEEE ICC*, 2021.

[36] M. Bertuletti, Y. Zhang, A. Vanelli-Coralli, and L. Benini, "A 66-gb/s/5.5-w risc-v many-core cluster for 5g+ software-defined radio uplinks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2025.

[37] aligungr, "UERANSIM: Open source 5G UE and RAN (gNodeB) implementation," https://github.com/aligungr/UERANSIM.

[38] L. Ismail and H. Materwala, "Computing server power modeling in a data center: Survey, taxonomy, and performance evaluation," *ACM Comput. Surv.*, vol. 53, no. 3, Jun. 2020.

[39] K.-J. Hsu, K. Bhardwaj, and A. Gavrilovska, "Colibri: Efficient collection of fine-grained resource metrics necessary for mobile edge computing," in *2024 IEEE/ACM Symposium on Edge Computing (SEC)*, 2024.

[40] S. P. Rachuri, N. Shaik, M. Choksi, and A. Gandhi, "Ecoedgeinfer: Dynamically optimizing latency and sustainability for inference on edge devices," in *2024 IEEE/ACM Symposium on Edge Computing (SEC)*, 2024.

[41] GSMA, "Mobile net zero: State of the industry on climate action 2024," GSMA, Tech. Rep., 2024. [Online]. Available: https://www.gsma.com/solutions-and-impact/connectivity-for-good/external-affairs/climate-action/mobile-net-zero-2024/

[42] G. Cornuejols, R. Sridharan, and J. Thizy, "A comparison of heuristics and relaxations for the capacitated plant location problem," *European Journal of Operational Research*, 1991.

[43] G. Guastaroba and M. G. Speranza, "Kernel Search for the Capacitated Facility Location Problem," *Journal of Heuristics*, 2012.

[44] I. Gjergji, L. Kletzander, N. Musliu, and A. Schaerf, "Large neighborhood search for capacitated facility location with customer incompatibilities," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2025.

[45] Z. Zhang, A. Marder, R. Mok, B. Huffaker, M. Luckie, K. C. Claffy, and A. Schulman, "Inferring regional access network topologies: methods and applications," in *Proceedings of the 21st ACM IMC*, 2021.

[46] MOSEK ApS, *MOSEK Optimization Software and Documentation*, 2025. [Online]. Available: https://www.mosek.com

[47] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2024. [Online]. Available: https://www.gurobi.com

[48] P. Avella, M. Boccia, A. Sforza, and I. Vasil'ev, "An effective heuristic for large-scale capacitated facility location problems," *Journal of Heuristics*, 2009.

[49] E. Agarwal, K. S. Gurumoorthy, A. A. Jain, and S. Manchenahally, "A Scalable Solution for the Extended Multi-Channel Facility Location Problem," in *LNCS*, 2023.

[50] R. Mansini and R. Zanotti, "Two-phase Kernel Search: An Application to Facility Location Problems with Incompatibilities," in *11th Proceedings of ICORES*, 2022.

[51] T. Matos, Ó. Oliveira, and D. Gamboa, "RAMP algorithms for the capacitated facility location problem," *Annals of Mathematics and Artificial Intelligence*, 2021.

[52] K. Unnu and J. A. Pazour, "A Large-Scale Heuristic Approach to Integrate On-Demand Warehousing into Dynamic Distribution Network Designs," *Computers & Industrial Engineering*, 2023.