OutRAN: Co-optimizing for Flow Completion Time in Radio Access Network

Jaehong Kim Yunheon Lee Hwijoon Lim Youngmok Jung Song Min Kim Dongsu Han
KAIST

ABSTRACT

Traffic from interactive applications demanding low latency has become dominant in cellular networks. However, existing schedulers of cellular network base stations fall short in delivering low latency when prior information (i.e., dedicated Quality of Service (QoS)) is unavailable; they become service agnostic and perform towards maximizing the radio resource utilization or user fairness. We identify a new opportunity of providing a better latency for those latency-sensitive traffic flows by additionally taking the Flow Completion Time (FCT) into account in downlink scheduling at the base stations. However, the key challenges are 1) it can bring a severe cost in optimization metrics of the existing scheduler and 2) it should work without prior knowledge of the traffic.

To this end, we present OutRAN, a practical flow scheduler designed for Radio Access Network that co-optimizes the FCT and optimization objectives of the cellular scheduler. The resulting system does not require prior information. Through simulation and over-the-air evaluation, we demonstrate that OutRAN outperforms the legacy LTE/5G schedulers in FCT, which leads to the reduction in webpage load time of Android phones.

CCS CONCEPTS

• **Networks** → **Cross-layer protocols**; **Mobile networks**; **Wireless access points, base stations and infrastructure**.

KEYWORDS

radio access network, base station, resource scheduling

ACM Reference Format:
Jaehong Kim, Yunheon Lee, Hwijoon Lim, Youngmok Jung, Song Min Kim, Dongsu Han. 2022. OutRAN: Co-optimizing for Flow Completion Time in Radio Access Network. In *The 18th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '22), December 6–9, 2022, Roma, Italy*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3555050.3569122

1 INTRODUCTION

Interactive applications such as web browsing and social networking have become dominant in cellular networks [28, 41, 54]. These applications involve human (or machine) interaction with the remote server and generate short request and response traffic patterns. The 3GPP standard [8] defines such traffic as the Interactive class, one of the four generic traffic classes in cellular networks. The applications demand low latency for each of the short flows since even a small delay can degrade user experience [36].

To provide low latency for the interactive applications, one of the solutions considered in cellular networks is Quality of Service (QoS) provisioning [59, 63, 64, 83]. For this, 3GPP [7] defines fine-grained QoS classes that recommend different latency guarantee for each traffic type. In current operational cellular networks, however, most of the traffic, including the Interactive and Background traffic class (e.g., ftp), is typically serviced with the default QoS (Non Guaranteed Bit Rate (Non-GBR)) by the cellular network operators, also known as the best-effort service (refer to Table 1) [80]. The only exception is delay-critical traffic such as Conversational (e.g., VoIP) or Streaming traffic (e.g., real-time audio/video) and they are serviced with the dedicated QoS profile with Guaranteed Bit Rate (GBR) [83]. Consequently, the latency-sensitive Interactive traffic and heavy Background traffic become the same citizens in the network, hence experiencing the same best-effort service from the base station scheduler [24].

Unfortunately, when it comes to scheduling the best-effort traffic, current base station schedulers [24, 27, 42, 43, 50, 81] are service agnostic and do not consider the latency sensitivity of the traffic. Instead, they are designed to maximize the radio resource utilization and achieve fair resource allocation among users. For instance, when latency-sensitive short flows (e.g., web browsing) and long flows (e.g., bulky file transfer) compete for the bandwidth in the downlink, the base stations perform scheduling on a user-granularity – they prioritize the user experiencing the better channel quality at that moment, or the user with the lower past achieved service, but overlook the ones carrying latency-sensitive short flows. This results in poor latency for the short flows from latency-sensitive applications, especially in tail latency.

The goal of this paper is to provide better latency for the latency-sensitive interactive traffic even when it lacks the dedicated QoS profile and is considered best-effort traffic. This can be accomplished by finishing each of its short flows quickly, that is, minimizing the Flow Completion Time (FCT). FCT is known to be a simple, yet good proxy for user experience in latency of interactive applications, as users want the shortest possible completion time [14, 34].

Inspired by the recent works in datacenter [18, 26, 29], we find an opportunity to minimize the FCT without requiring prior knowledge of the flow size or service type. However, applying the approach to Radio Access Network (RAN) is not straightforward:

- Each user experiences a different link rate across time due to the varying wireless channel. Therefore, the spectral efficiency (bit/s/Hz) highly depends on how the switching fabric (i.e., base station) schedules the user flows with the available radio resource. Scheduling flows simply towards the transport-layer

CoNEXT '22, December 6–9, 2022, Roma, Italy

ACM ISBN 978-1-4503-9508-3/22/12…$15.00
https://doi.org/10.1145/3555050.3569122

metrics (i.e., FCT) may cause huge side-effects on the radio resource utilization.
- The fairness between users can be an issue since the flow scheduling preempts the users carrying longer flows. In contrast to the datacenter where fairness is not a primary concern [14, 40], it is a major requirement in RAN [24].

To address the problem, we present OutRAN, a practical flow scheduler tailored for RAN. OutRAN is a downlink scheduling scheme performed at the last mile base stations. The main idea of our work is to identify the user flows that best balance the satisfaction both for our optimization goal and the legacy scheduler's optimization objective. This way, we can effectively minimize the FCT for short flows at a minimal cost of the system's spectral efficiency and (user) fairness. OutRAN achieves this in two ways:
- OutRAN emulates Shortest Job First (SJF) on the flows sharing the same user (i.e., intra-user flows). It uses Multiple Level Feedback Queue (MLFQ) [16] which is an information agnostic scheduling that approximates SJF.
- OutRAN leverages the underlying link-layer information when scheduling the flows from different users. It gets the primary candidates of users that suit the link-layer scheduler's optimization objective (i.e., spectral efficiency, fairness) and opportunistically prioritizes the user carrying shorter flow.

Figure 1 illustrates our approach and difference from the existing scheduler. We implement OutRAN on top of the srsRAN [69], an open-source LTE/5G software radio suite that runs on a commercial Software Defined Radio (SDR) device. OutRAN only requires modification on the user-plane protocol stack (LTE/5G Layer 2) of the base station which is fully programmable in srsRAN.

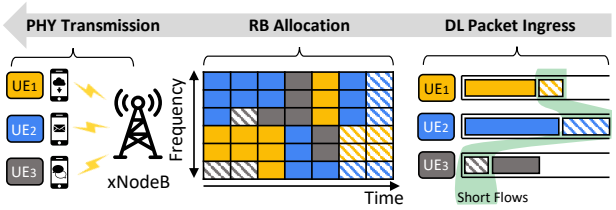
We conduct testbed experiments on *Colosseum* [31]—the large-scale wireless network testbed that emulates real-world wireless environments—and over-the-air using commercial Android phones. On Colosseum, under real-world RAN scenarios [21], OutRAN improves the average FCT by 32%. On the over-the-air testbed, OutRAN shows an average 14% (up to 34%) reduction in webpage loading time for the top 20 webpages from Alexa [17]. OutRAN achieves this with marginal overhead in CPU and memory usage, without compromising the processing throughput of LTE/5G xNodeB. The video samples showing PLT improvements are available at https://ina.kaist.ac.kr/projects/outran, and the corresponding snapshots of them are presented in Appendix §B.

Lastly, our simulation shows that OutRAN delivers comparable short flow FCT to the QoS-aware schedulers provided in NS-3 [20, 56] and outperforms them in spectral efficiency and user fairness.

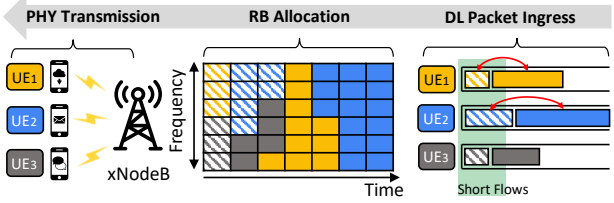
Contribution. To the best of our knowledge, OutRAN is the first work to establish the feasibility of co-optimizing the FCT in the cellular network domain by introducing transport-level flow scheduling to the base stations. We propose a practical design that is compatible with both LTE and 5G and is promising in providing better service for the best-effort Interactive traffic.

2 BACKGROUND

Cellular Network consists of three main components; the User Equipment (UE), the RAN, and the Core Network (CN). CN is the backbone of the cellular network that provides core functionalities such as QoS provisioning and the Internet connection. RAN



(a) Existing schedulers in operation, being flow-agnostic. Short flows (pattern-filled) suffer from queueing delay.



(b) OutRAN completes short flows first.

Figure 1: Existing base station schedulers vs. OutRAN.

consists of multiple radio base stations called xNodeBs (eNodeB in LTE, gNodeB in 5G) that offer a wireless connection between the UE and the CN. The xNodeBs are responsible for scheduling the downlink/uplink traffic from multiple users with the available radio resource in RAN. In this paper, we focus on scheduling the downlink user traffic at the xNodeBs. We will use the terms "base station" and "xNodeB" interchangeably.

5G New Radio (NR). 5G gNodeBs now operate with the 5G NR interface [58] which brings low latency "in air" [65, 66]. The new interface provides a scalable frame structure with multiple different sub-carrier spacings (SCs) [4, 10]. At higher SCs (numerologies), the OFDM symbol duration decreases, and hence the length of a slot, which is a scheduling resolution at the xNodeB (refer to Figure 5). This reduces the delay waiting for a transmit opportunity by the scheduler [65, 66]. OutRAN is compatible with the different 5G numerologies.

Flow scheduling. The optimal flow scheduling algorithm that minimizes the expected FCT over a single link is the Shortest Remaining Job First (SRJF) [14]. Yet, this requires perfect knowledge of the processing time which makes it impractical in cellular networks. Recent works in flow scheduling [18, 26, 29, 52] show that the classic information-agnostic (non-clairvoyant) scheduling policy such as Least-Attained Service (LAS) and Multiple Level Feedback Queue (MLFQ) can provide outstanding performance in FCT minimization.

3 MOTIVATION

Coarse-grained QoS provisioning in LTE/5G. The 3GPP standard specifies multiple QoS classes for 5G (LTE) through 27 5QIs (26 QCIs) [1, 7]. However, in most cases, they are not utilized except for the delay-critical traffic [83]. To confirm the idea, we conduct an experiment on a commercial-level 4G/5G testbed [35, 49] that supports 3GPP Release 15. Its CN shares the identical QoS parameter settings with the commercial CN run by the major cellular operator in South Korea. We use the XCAL-PU12 tool [11] which enables us to peek the 4G/5G signaling information exchanged between CN and UEs. We check the QoS negotiation process with the CN

Application	Traffic Class	Bearer	QCI	Service
VoIP (i.e., VoLTE)	Conversational	Dedicated GBR	1	Guaranteed Bitrate (GBR) =14 kbps
IMS signaling	Interactive	Default (ID=5)	5	High priority, Best-effort
Web browsing, Social networking	Interactive	Default (ID=6)	6	Low priority, Best-effort
TCP-based video, File transfer	Background	Default (ID=6)	6	Low priority, Best-effort

Table 1: QoS profiling of the mobile applications on a commercial-level 5G Non-Standalone (NSA) testbed. 5G SA showed the same 5G QI values as the LTE QCI.

using multiple representative mobile applications (e.g., Chrome, Instagram) on a commercial Samsung 5G phone, and Table 1 shows the summary.

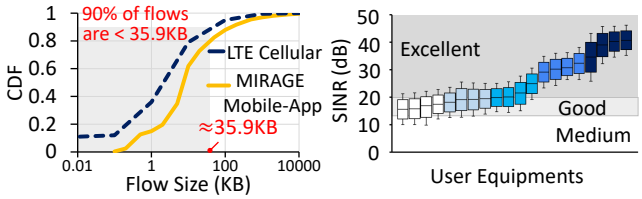
Notice that except for VoIP and IMS, all other internet-based applications (blue colored) share the same QoS profile (QCI/QI=6), which is the default QoS. We believe this is because the QoS provisioning in 5G is in its infancy and hard to implement in practice; 1) it requires a sophisticated Packet Detection Rules (PDRs) [9, 59], whose implementation is left to the operators and 2) the networks need to configure all the elements from wireless to the physical resources on CN interfaces. Even if the operator can detect latency-sensitive traffic and wants to enforce the dedicated QoS on it, it requires setting up a dedicated bearer[1] on demand. This takes an initialization delay since it would take an additional round trip from the base station to the CN [83].

Our goal is to provide better latency for latency-sensitive applications even when they are served with the same default QoS profile as the other traffic.

Heavy-tail traffic distribution. Traffic distribution in the cellular network is known to be skewed as wired networks, exhibiting strong heavy-tail distribution [12, 41, 82]; most flows are small (90% of flows are smaller than 35.9 KB), and large flows (i.e., heavy-hitter flows) occupy only a small fraction but they take up the majority of the traffic volume. In such distribution, short and long flows coexist in the network, and when they share the same service, it is problematic for latency-sensitive short flows [41, 47]. Throughout the paper, we use the downlink flow size distribution from [41] (Figure 2(a)) for LTE simulations as it represents the TCP traffic collected in real-world LTE eNodeBs. Since it is a regular TCP traffic (76% being HTTP), it falls into the default QoS class, which is our target traffic class. For 5G, we use the traffic more recently collected in [12].

Flow scheduling at xNodeBs. Using large buffers is essential in xNodeB to absorb the bursty traffic, channel fluctuation, and to store the unacknowledged packets for link-layer retransmissions [53]. However, this makes it even worse for latency-sensitive flows in terms of latency when they contend with the other flows. The large buffers cause queue-buildup and lead to long queueing delay, known as the bufferbloat issue [46, 47]. In the worst case, the packets of

[1]A logical channel between UE and Packet Gateway (P-GW). LTE QoS is enforced at the bearer level.



(a) Flow size distribution. **(b)** SINR distribution.

Figure 2: Downlink cellular traffic distribution [41] and channel quality distribution across UEs in our evaluations.

the short flows queued up behind by bursts of packets from large flows. The problem may become even more significant in 5G as the buffer size is expected to scale over 5× more than that of LTE [77].

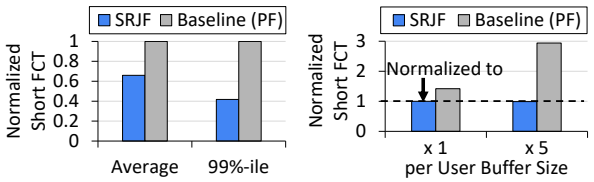
Fortunately, flow scheduling can significantly alleviate the problem and provide a better completion time for short flows. To quantify the potential benefit, we conduct a simulation using NS-3 comparing the FCT of short flows between the case with and without the flow scheduling at the xNodeB. For flow scheduling, we hypothesize that flow sizes are known a priori and use the optimal SRJF algorithm in datacenter [40] to quantify the maximum possible benefit. The baseline is the Proportional Fair (PF) scheduler [27, 43, 50, 81], the de facto standard xNodeB scheduler. It maintains the balance between spectral efficiency and fairness between users.

We set a scenario where UEs (using LTE with default QoS) request a service from a remote server that generates downlink traffic according to a Poisson process with a size distribution that follows the LTE traffic distribution [41]. The transport protocol is TCP-Cubic [39] and the buffer size per-user at xNodeB is set to the default value of srsRAN [69]. For realistic channel dynamics, we use the channel fading traces provided by 3GPP [2] that emulates a pedestrian scenario. The resulting channel quality distribution across UEs is presented in Figure 2(b).

Figure 3(a) shows that with SRJF flow scheduling applied to eNodeB, the average and tail FCT of short flows (< 10 KB) improve 35% and 59% over the PF, respectively. Additionally, Figure 3(b) shows that even when the per-user buffer increases (×5), SRJF keeps the short flow FCT low while it increases dramatically for the PF compared to the short flow FCT of SRJF.

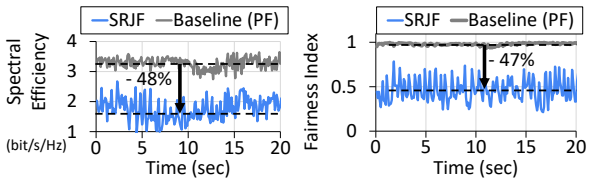
Challenge. Flow scheduling at the base station shows great benefits but it may cause side-effects to the xNodeB performance. Figure 4 shows the spectral efficiency and fairness index (eq.3 in §6) of resource allocation between users for the same experiment conducted in Figure 3(a). SRJF costs 48% of spectral efficiency and 47% of fairness respectively to minimize the short flow FCT. This is because the SRJF solely relies on the transport-layer flow size with being ignorant of the channel condition of each user, and it preempts the user carrying longer flows whenever there is a user with a shorter flow. The result proves that the flow scheduling should be carefully designed for xNodeBs to get benefits.

In this paper, we explore a practical flow scheduling design for the cellular network base stations by addressing the following issues. First, to be practical, our flow scheduling should work without prior knowledge of flow sizes. Second, it should minimize the FCT of short flows while not starving the long flows. Finally, such design goals should be compatible with the standard xNodeB radio resource scheduling while having a negligible impact on the system



(a) Improves short flow FCT. **(b)** Less sensitive to buffer size.

Figure 3: Benefit of flow scheduling at xNodeBs.



(a) Hurts spectral efficiency. **(b)** Hurts user fairness.

Figure 4: Side-effect of flow scheduling at xNodeBs.

performance. In the next section (§4), we describe how OutRAN addresses the relevant issues and challenges in making OutRAN more practical and effective.

Impact in 5G. Recently, edge computing and 5G New Radio (NR) have shown great potential in achieving low latency in backhaul network and RAN respectively [58, 77]. Given that the *base station* bridges the two, the aforementioned problem remains as bottleneck in 5G, and solving it becomes important for the next generation cellular network. In our main evaluation (§6.2), we show that OutRAN becomes even more attractive in 5G (Figure 17).

4 SYSTEM DESIGN

Design goals. We focus on designing a practical flow scheduler at xNodeBs that respects the optimization objectives of the legacy xNodeB schedulers and is compatible with underlying radio access technology.

- **Channel-aware cross-layer scheduling:** Our design should consider both the transport-layer and the link-layer information to minimize the FCT while imposing minimal side-effect on spectral efficiency and user fairness.
- **Information-agnostic FCT minimization:** In cellular networks, flow size information is transparent to the xNodeBs, and QoS provisioning by operators could be limited for the unidentifiable flows. OutRAN should achieve the FCT minimization without prior knowledge of the flows.
- **Compatibility:** To be practical, our design should be compatible with LTE and advanced 5G NR scheduling for the next generation RAN. Also, the design should not compromise the processing throughput of existing xNodeBs when it is applied in practice.

System overview. OutRAN consists of two key design components: *Intra-user flow scheduler* (§4.2) and *Inter-user flow scheduler* (§4.3). The *Intra-user flow scheduler* emulates SJF without any loss of the spectral efficiency and the (user) fairness that the underlying xNodeB scheduler provides. The *Inter-user flow scheduler* prioritizes a user carrying shorter flow with minimal impact on the spectral efficiency and the (user) fairness. Figure 5 shows the system design overview.

4.1 Downlink Scheduling at xNodeB

We design OutRAN to be agnostic to the underlying radio access technology, and thus it is compatible with both LTE and 5G downlink scheduling.

Radio resource in RAN is distributed across the time and frequency domain. The time domain is divided into multiple Transmission Time Interval (TTI), which is a scheduling interval (or scheduling resolution) of a xNodeB scheduler. In the frequency domain, the total bandwidth is divided into subchannels which consists of 12 consecutive OFDM subcarriers grouped together. A radio resource spanning over the single TTI and subchannel is called Resource Block (RB), and it is the smallest unit of resource that the xNodeB scheduler allocates to a user. The choice of TTI and subchannel size (i.e., bandwidth per RB) depends on the radio access technology, and the main difference between 5G and LTE in scheduling is the scale of the number of RBs (i.e., bandwidth) and scheduling resolution [27, 42]. For instance, LTE supports {1 ms, 180 kHz} and 5G NR numerology 3 supports {125 μs, 1440 kHz} for TTI and the subchannel size respectively [4, 65]. Finally, the total number of RBs available within a TTI depends on the system bandwidth and SCs. In LTE, a total of 100 RBs are available for 20 MHz and in 5G, a total of 273 RBs are available for 100 MHz (SC spacing=30 kHz) [3].

Downlink scheduling. For every TTI, MAC scheduler of xNodeB decides how to allocate RBs available in the bandwidth to different users. The resulting RB allocation highly impacts the spectral efficiency of the system because, 1) the channel condition of users varies for the same RB due to their different geographical locations and, 2) the channel condition of a user varies across different RBs due to the frequency-selective fading in the wireless channel. The scheduler, therefore, uses Channel Quality Indicator (CQI) values periodically reported by the users in order to assign each RB to ones experiencing the better channel quality at the current scheduling moment.

Optimization objectives and approach. There are various MAC schedulers with different optimization objectives. For example, the Maximum Throughput (MT) scheduler aims to maximize the spectral efficiency while the PF scheduler provides the balance between the spectral efficiency and the fairness between users.

Consider the xNodeB having users $\mathcal{U}$ are under its service and RBs $\mathcal{B}$ available in its downlink bandwidth. To find the best allocation of $\mathcal{B}$ to $\mathcal{U}$ for each scheduler's objective, the computational complexity simply becomes $O(|\mathcal{U}|^{|\mathcal{B}|})$. Considering that xNodeBs should schedule RB within short channel coherence time (<1 ms) and for a varying number of users, the computation is too complex and not scalable [24, 27, 42]. Thus, the xNodeB in practice, adopts a sub-optimal approach where scheduling is based on the comparison of the per-RB metrics, $m_{u,b}$, between users, which leads to much simpler computation. The per-RB metric for different schedulers is shown below:

$$m_{u,b}(t) = \begin{cases} r_{u,b}(t) & \text{MT scheduler} \\ r_{u,b}(t)/\tilde{R}_u(t-1) & \text{PF scheduler} \end{cases} \quad (1)$$

where $r_{u,b}(t)$ is the expected achievable rate of RB $b \in \mathcal{B}$ for user $u \in \mathcal{U}$ for TTI t, and $\tilde{R}_u(t-1)$ indicates the long-term average throughput of user u defined as the exponentially smoothed average of throughput of u up to $t-1$.

For each RB $b \in \mathcal{B}$, the scheduler iterates over users $\mathcal{U}$, and calculates the per-RB metric. Then, it allocates RB b to the user $\hat{u}$ having the best metric independently of other RBs. The computational complexity of the approach becomes $O(|\mathcal{U}||\mathcal{B}|)$ and we design OutRAN to work within the same computational complexity.

4.2 Intra-user Flow Scheduling

Switching between flows that share the same user can be achieved without harming the total spectral efficiency or user fairness. Motivated by this, our Intra-user Flow Scheduler emulates SJF on the intra-user flows based on the transport-layer information obtained from the PDCP (Packet Data Convergence Protocol) layer. It operates at the Radio Link Control (RLC) Layer (shown in Figure 5) of LTE/5G Layer 2.

Design rationale. Our goal is to minimize the FCT without prior knowledge. Recent works in Datacenter Network (DCN) [18, 26, 29] have confronted the similar challenge and proposed a simple MLFQ or LAS scheduling, where each switch fabric approximates SJF upon bytes-sent of a flow. The approach was simple enough to just do strict MLFQ scheduling on every egress flow since it can achieve minimal FCT while sustaining the network utilization. The reason for this is, all ports of a switch fabric maintain the same consistent link rate, which makes every single flow experience theoretically the identical link rate. However, the condition does not hold for the xNodeB; flows from different users experience diverse link rates due to their disparate wireless channel conditions, as we explained in §4.1. Since the strict MLFQ (or LAS) used in DCN schedules flows upon the flow size without considering the channel conditions of each flow, porting the same scheduler design directly on the xNodeB leads to an inevitable cost in spectral efficiency, which is similar to the SRJF scheduling in §4.

Still, the opportunity for flow scheduling with having no impact on the system performance exists, if we switch between the flows that have the "identical" channel condition (or achieved service so far from the xNodeB for fairness) at the scheduling moment. The flows destined to the same user can perfectly satisfy the requirement as they are always guaranteed to undergo the identical channel.

Identifying the flows with the same user is challenging in DCN switches, but it is feasible in xNodeB. This is because the cellular network readily supports per-user scheduling to apply the Policy and Charging Rule Function (PCRF), and xNodeB maintains a separate per-user buffer to handle different packet delivery rate, and its packet retransmission of a user. By leveraging this aspect, our scheduler maintains separate MLFQ scheduling per user, instead of using a single MLFQ per egress port as done in DCN [18]. This design preserves exactly the same spectral efficiency and user fairness that the original xNodeB scheduler provides.

MLFQ scheduling in xNodeB. Today's programmable base stations give us more freedom of implementing advanced scheduling algorithms [22]. Thus, OutRAN's base station (i.e., xNodeB) inspects packets and maintains per-flow states (i.e., sent-bytes per flow). The design is feasible because 1) cellular network operators have full control on the xNodeBs which are fully programmable, and 2) it does not require any modification on end hosts including UEs and MECs.

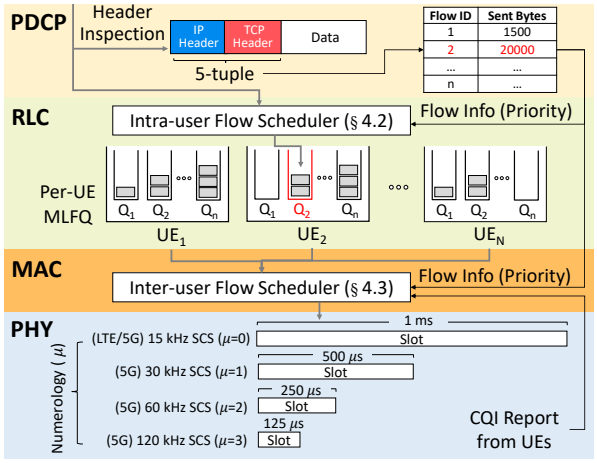


Figure 5: OutRAN overview with design components.

When a packet arrives at each user's buffer, our scheduler identifies the flow based on the five tuple information (src/dst IPs, src/dst ports, protocol) and updates the sent-bytes so far (or create a new entry if it is a new one). Next, using the sent-byte information, it enforces the MLFQ scheduling for each flow. The MLFQ scheduling consists of K priority queues, P_i $(1 \le i \le K)$ and $K-1$ thresholds, α_j $(1 \le j \le K-1)$. The priority decreases from P_1 to P_K, and strict priority queueing is applied based on the following rule [16, 18, 26, 29]:

- A new incoming flow starts from P_1.
- Packets of a flow having P_i priority enter the P_i queue of their destined user.
- A flow gets demoted from P_i to P_{i+1} when sent-bytes of a flow crosses the threshold α_i.

The policy mimics SJF as the short flows are likely to finish in high priority queues than the long flows. Another benefit of using MLFQ is it prevents starvation of the long flows by having a lower bound on the priorities. Beyond a certain size, all flows will get the same base priority and get fair service.

The users share the same thresholds but our design maintains a separate MLFQ structure for each user buffer. After each packet is assigned with the MLFQ priority, they are put into its corresponding priority queue.

Parameter choice. The performance of MLFQ depends on how we choose the number of queues K and the thresholds $\{\alpha\}$. We observe that for $K > 4$, the performance of OutRAN stays steady as similar results are shown in previous works [18, 29]. To find the best thresholds, we referred to the solution method presented in PIAS [18, 19], which solves the optimization problem of finding the MLFQ thresholds. We used the global optimization toolbox in SciPy [48] to solve the problem and used the solution values in our evaluation.

Limitation. Some applications (e.g., websites using QUIC [51] protocol, video streaming) maintain persistent TCP connections and reuse the same five tuples for sending multiple short flows. In these cases, sometimes it can mislead our scheduler to serve

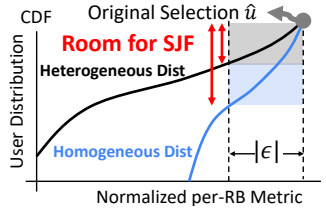
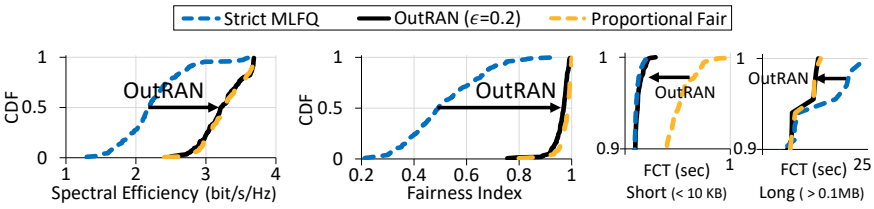


Figure 6: Illustration of how Inter-user Flow Scheduler works.



(a) Spectral Efficiency. **(b)** Fairness Index. **(c)** FCT Distribution.

Figure 7: Proof-of-concept: CDF of Spectral Efficiency, Fairness and FCT.

short flow at the low priority queue due to the large aggregated sent-bytes. To mitigate the problem, we can reset or promote the priorities in MLFQ for every period. We show this in our case study (§6.3). Our evaluation on PLT (§6.1) is conducted with QUIC [51] enabled in the Chrome browser. Even without the priority reset, OutRAN still improves the PLT by 14% compared to the vanilla xNodeB. The detail is in §6.1.

4.3 Inter-user Flow Scheduling

Our Inter-user Flow Scheduler opportunistically prioritizes the user carrying short flow while minimizing its impact on spectral efficiency and user fairness. The scheduler considers both the transport-layer information (MLFQ priority) passed down from the Intra-user Flow Scheduler (§4.2) and PHY-layer information. The scheduler operates at the MAC layer (shown in Figure 5) of LTE/5G Layer 2.

Design rationale. Since each user experiences different channel quality, switching between flows of different users inevitably results in different spectral efficiency (and user fairness) from what the original xNodeB scheduler provides. In the worst case, it can lead to a huge drop in both spectral efficiency and fairness as shown in §3.

Fortunately, we notice that the opportunity for flow scheduling still exists, if we exploit the case where users are experiencing a similar channel quality (or per-RB metric) at the scheduling moment. As we explained in §4.1, the original xNodeB scheduler assigns each RB to the user having the "best" per-RB metric value at each TTI. On the other hand, there can be other multiple candidates having suitable value for the optimization objective of the original scheduler with a marginal difference from the best one. In fact, such nodes are quite common for the current dense cellular networks. Thus, with a minimal cost on the per-RB metric, we can select another user set $\mathcal{U}'$ that has comparable per-RB metric values (if there is any) to that of $\hat{\mathcal{U}}$ (the originally selected user set) and has shorter flow size overall. This does not guarantee the exact spectral efficiency (and user fairness) that the original xNodeB provides, yet it still achieves a similar performance and makes enough room for our flow scheduling.

Design. Motivated by the idea, our Inter-user Scheduler expands the user space that the original scheduler explores by relaxing the per-RB metric it uses for every TTI. Specifically, we define a relaxation threshold ϵ that Inter-user Scheduler applies to the per-RB metric. Figure 6 illustrates the approach.

On the first iteration of RBs $\mathcal{B}$, the Inter-user Scheduler operates the same ways as the original xNodeB scheduler does for the RB allocation. It first selects the best user $\hat{u} = \arg\max_{u \in \mathcal{U}} m_{u,b}$ and updates the m_{max} for each RB, $b \in \mathcal{B}$ (Algorithm 1 in A, line 4-8). Then, it applies the relaxation threshold ($0 \le \epsilon \le 1$) to the maximum value $m_{\hat{u},b}$ selected from the original scheduler. Next, the scheduler does a secondary iteration to get the primary user candidates $\mathcal{U}'$ that at least have $(1-\epsilon)$ of the m_{max}. Among them, our scheduler re-selects the user $\hat{u}$ having shorter flow, comparing the MLFQ priority marked by the Intra-user Flow Scheduler (§4.2) (Algorithm 1 in Appendix §A line 14-23):

$$\hat{u} \leftarrow \arg\max_{u \in \mathcal{U}'} \left(\max_{f \in \mathcal{F}_u} Priority(f) \right)$$
$$where,\ \mathcal{U}' = \{u' \mid (1-\epsilon) \cdot m_{\hat{u},b} \le m_{u',b},\ u' \in \mathcal{U}\} \quad (2)$$

This way, our scheduler guarantees at least $|1-\epsilon|$ of the per-RB metric $m_{u,b}$ that the original scheduler uses while expanding the room $|\epsilon|$ for SJF flow scheduling.

Note that our approach is different from selecting the top K users, where it always guarantees a room space of K users, which can also lead to the side-effect similar to the SRJF. Our scheduler naturally condenses the room if the users are experiencing heterogeneous distribution in the per-RB metric as shown in Figure 6. Our scheduling logic requires one additional iteration among users. This still gives us $O(|\mathcal{U}||\mathcal{B}|)$ complexity which imposes negligible overhead to the xNodeB operation. We measure the overhead of the additional operation by increasing the number of RBs in §6.1.

Proof-of-concept. To validate our design, we conduct a simulation in a similar setting described in §3. We compare the performance between OutRAN ($\epsilon = 0.2$), which works on top of the PF scheduler (respecting its per-RB metric by 80%), and strict MLFQ, which expands the entire room for SJF. Figure 7 compares the CDF results of spectral efficiency, fairness, and the FCT between the two schemes and the original PF scheduler. Figure 7(a) and 7(b) are the CDF of the spectral efficiency and fairness values obtained from the xNodeB for every 50 TTIs.

OutRAN ($\epsilon = 0.2$) effectively minimizes the short flow FCT showing the comparable result to that of the strict MLFQ without starving the long flows compared to PF (Figure 7(c)). OutRAN achieves this showing almost the same spectral efficiency and fairness compared to PF (Figure 7(a), 7(b)). We also compare this with OutRAN ($\epsilon = 0$) where it operates only with the Intra-user Flow Scheduler. OutRAN with Inter-user Flow Scheduler ($\epsilon = 0.2$) further improves the tail FCT of short flows (10% in 95%-ile) compared to the ($\epsilon = 0$), and the benefit becomes greater when there are more users experiencing similar per-RB metrics.

Parameter choice. The threshold ϵ provides a trade-off between the optimization objective of the xNodeB scheduler and FCT minimization. Through our extensive simulation, we observe that for $\epsilon < 0.4$, OutRAN shows steady performance with FCT minimization while having minimal impact on the xNodeB objectives. We

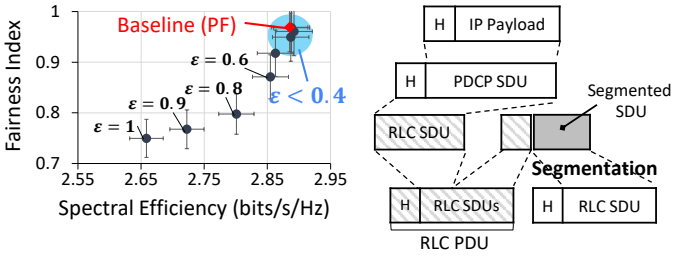


Figure 8: OutRAN sensitivity.

Figure 9: User-plane data flow.

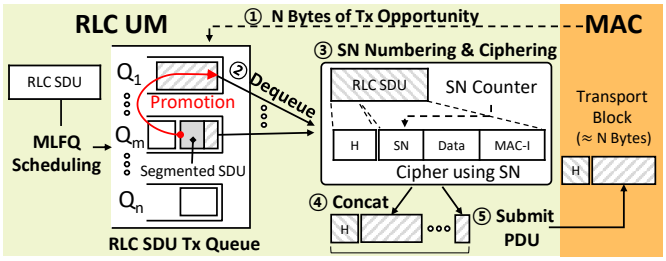


Figure 10: OutRAN workflow in practice.

chose $\epsilon = 0.2$ for OutRAN which provided the best balance between the two optimization objectives in the result. The sensitivity of OutRAN to the ϵ is shown in Figure 8. If desired, the ϵ can also be tuned toward FCT minimization depending on the operator's interest.

4.4 Integrating OutRAN into xNodeB

From a radio protocol stack (LTE/5G Layer 2) perspective, a number of aspects that are unique in xNodeB should be considered to make our scheduling design compatible with the standard xNodeB.

Segmentation & reassembly. The downlink transmission (tx) buffer for each user is maintained at the RLC layer, and the data unit of the tx queue is RLC Service Data Unit (SDU). When the RLC layer is notified of the tx opportunity by the MAC scheduler, the RLC dequeues SDU(s) from its tx buffer and submits it to the next MAC layer as an RLC Protocol Data Unit (PDU). The relation between the RLC SDUs and the PDUs is not univocal, and the payload of the RLC PDUs can contain multiple SDUs due to the different packet delivery rates of each user, as shown in Figure 9. The RLC layer, therefore, is responsible for segmentation & concatenation at the sender, and reassembly of the segmented SDUs at the receiver [6, 67].

If we stick to the strict MLFQ rule by maintaining the priority of the segmented SDUs, the transmission of the segmented SDU could be delayed by the SDUs in the higher priority queue, unlike the original FIFO scheduling. This becomes a problem at the RLC receiver entity since the reassembly at the RLC is performed on an RLC SDU basis; when the entity receives the segmented SDU, it waits for the remaining segmented SDU(s) in the next reception within the reassembly window. If the segmented SDU gets delayed and falls outside of the window, the RLC considers it as it cannot be reassembled, and discards it [6]. This eventually hurts the FCT as only the RLC SDUs that are fully available can be delivered to the upper PDCP layer. To avoid the case, we make an exception for the segmented SDU if there is any, and promote it to the head of the first priority queue.

Sequence numbering. In the cellular networks, the PDCP data PDU counter keeps track of the unique Sequence Number (SN) for each received PDU from the IP layer by incrementing the value. Then using the SN as a key, ciphering is performed on the payload at the PDCP layer before submitting it to the lower RLC layer for security [6, 67]. The SN is synchronized with the receiver-side (UE) so that only the destined user can decipher the received packet.

OutRAN, however, changes the order of transmitted PDU and therefore, the SN numbered on the transmitted PDU does not correspond to the one maintained at the receiver-side, which makes it receiver impossible to decipher the PDU. In order to prevent the case, OutRAN delays the PDCP's SN numbering & ciphering and performs the process at the RLC layer, right before submitting the RLC PDUs to the MAC layer. We implement the feature in our testbed and confirm that our design is still compatible with commercial android phones.

Retransmission. An RLC entity has two data transmission modes for user-plane: Unacknowledged Mode (UM) and Acknowledged Mode (AM). The UM provides unidirectional data transfer and only has a tx buffer. On the other hand, the AM mode provides a bidirectional data transfer service and supports link-layer retransmission. In the AM mode, there are three queues having different priority-level:

- Ctrl Q (1st priority) for Control PDUs (link-layer ACK/NACK).
- Retx Q (2nd) for RLC PDUs which are considered for retransmission. (e.g., NACKed PDUs)
- Tx Q (3rd) for the RLC SDUs that are waiting for the tx opportunity.

If the tx mode of RLC is configured as the AM, OutRAN complies with the priority levels of each queue specified in the 3GPP standard [5]. In detail, we only apply intra & inter-user scheduling on the TxQ and schedule the TxQ within the leftover tx opportunity bytes after scheduling the Ctrl and the RetxQ. The per-flow state is kept only for the TxQ.

In our main evaluations, we choose the UM as our RLC tx mode. In addition, we perform a case study in §6.3 showing that OutRAN also works with the AM mode and outperforms the PF MAC scheduler that operates with the RLC AM mode in FCT.

The overall workflow is shown in Figure 10. When a user is selected for the dl transmission and assigned N bytes of tx opportunity by our Inter-user Flow Scheduler ①, N bytes of RLC SDUs (pattern-filled) are dequeued according to our Intra-user Scheduling ②. Whenever it generates a segmented SDU, OutRAN promotes it to the topmost priority. Next, OutRAN performs delayed SN numbering & ciphering ③, concatenates the scheduled SDUs into the RLC PDU ④, and submits to the MAC layer ⑤.

5 IMPLEMENTATION

OutRAN is implemented on top of the srsRAN [69], an open-source LTE/5G software radio suite. We modified the user plane protocol stack (LTE/5G Layer 2) of srsENB in srsRAN, which is composed of the following three sub-layers; PDCP, RLC, and MAC. OutRAN consists of ~1.4K lines of new or modified code. More detail of our implementation is explained in Appendix §B. We believe that the deployment is feasible given that cellular networks, including the base stations are becoming open and programmable [21, 22, 31].

6 EVALUATION

Metrics. To show the improvements that OutRAN delivers, we use the FCT and Web Page Load Time (PLT). We referred to the PLT defined in [73] and use the Navigation Timing API [15] from W3C for measurement. Additionally, we compare the spectral efficiency (bit/s/Hz) and fairness index of the long-term average throughput among the users defined as:

$$Fairness(t) = \frac{(\sum_{u\in\mathcal{U}} \tilde{R}_u(t))^2}{|\mathcal{U}| \sum_{u\in\mathcal{U}} \tilde{R}_u(t)^2} \quad (3)$$

Baselines. We use the PF scheduler, the de facto standard MAC scheduler of xNodeB [27, 43, 50, 81] as our baseline. Specifically, we use the PF scheduler implemented in NS-3 [32, 61] and srsRAN [69]. In our simulations, we also include variants of PF schedulers with QoS support as the baselines. We explain the detail of them in §6.2. For a fair comparison for our information-agnostic scheduler, we set the default QoS class for every traffic flow except for the QoS-aware schedulers.

Platform overview. OutRAN is evaluated using a combination of testbed experiments and NS-3 simulations. Below is a brief overview of the platforms used:

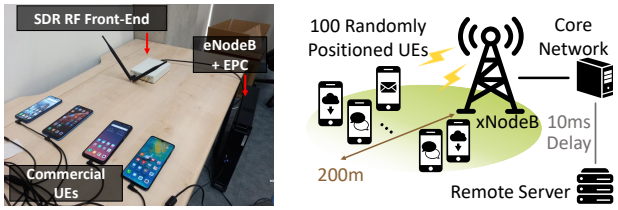
- **Over-the-air testbed (§6.1)** is a testbed setup using the srsRAN and android smart phones. We show the impact of OutRAN on interactive application (i.e., Chrome) by measuring the PLT of the top 20 visited web pages from Alexa [17].
- **Emulated environment (§6.1)**. To test OutRAN in more realistic RF scenarios, we deploy OutRAN implementation on Colosseum [31], which is a massive wireless systems testbed developed by DARPA. The platform provides remote access to SDRs and server nodes, which supports real-time emulation with real wireless signals and emulated channels.
- **NS-3 simulation (§6.2, §6.3)** is used to provide an evaluation with the large scale of users and different advanced settings (i.e., 5G numerology and mobility) that are not available in our testbed.

Summary. We summarize our main findings by answering the following questions:

- How does OutRAN improve the latency in an application level? On average, OutRAN improves the PLT 626 ms (14%) by improving the FCT of subflows of the webpage by 53 ms (20%) (§6.1).
- How well does OutRAN perform without prior knowledge? OutRAN shows comparable short flow FCT to that of SRJF and QoS-aware schedulers (§6.2). OutRAN works well under real-world RF scenarios (§6.1).
- Does it minimize the side-effects? OutRAN preserves 98% and 97% of spectral efficiency and fairness respectively that PF provides. It does not starve the long flows (§6.2).
- Is OutRAN practical? OutRAN works well in both LTE/5G (§6.2) and our performance benchmark suggests that OutRAN is applicable to an actual xNodeB in practice without compromising the processing throughput of xNodeB (§6.1).

6.1 Testbed Experiments

Settings. We build an end-to-end cellular network using srsRAN [69] and the setup is shown in Figure 11(a). The RAN consists of srsENB and srsEPC which are the base station and the Core Network (CN),



(a) Over-the-air Testbed. **(b)** Cell-scale Simulation.

Figure 11: Testbed and simulation topology.

respectively. They run on a single machine with Linux kernel 5.4.0 equipped with Intel Core i9 3.6 GHz CPU. For the srsENB RF frontend, we use Universal Software Radio Peripheral (USRP) B210 SDRs. For UEs, we use four commercial Android smartphones and apply the programmable SIM cards (sysmoISIM-SJA2) [71] to register the UEs to our HSS database. The CN enables the Internet access of each UE by masquerading its network interface.

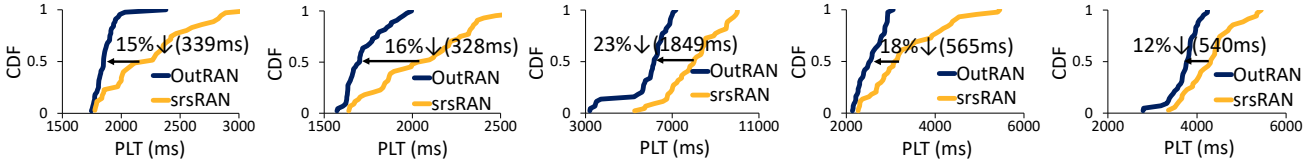
To model realistic channel dynamics, we program srsENB to use the CQI values read from the 3GPP channel trace [2] (pedestrian scenario) instead of an actual CQI report from each UE. In this way, we confirm that the srsENB schedules UEs based on the given CQI information, and each UE experiences different DL throughput across time. The srsENB operates in Band 7 (2680 MHz downlink) with 20 MHz bandwidth at 97 Mbps bitrate (256 QAM, SISO) which yields 4.85 bit/s/Hz spectral efficiency. The maximum buffer size of the RLC UM entity is set to the default value of srsENB (capacity of 128 RLC SDUs).

Workloads. We model a setup where Interactive traffic (i.e., Chrome web browsing) and heavy Background traffic competes for the downlink bandwidth. The Background traffic follows the web search service [13] with an average flow size of 1.92 MB. Each UE requests background flows (i.e., bulky file transfer) from our local server according to a Poisson process with a size distribution that follows the given background traffic distribution. The average cell load is set to 60% and we set the frequency of webpage requests to every 15 sec. The web pages are hosted on their original server. To synchronize each request of background flows to our Poisson process, we implement a workload generator that signals UEs when to start their assigned workload via Android Debug Bridge (adb).

Page load times for Alexa websites. For the experiment, we use the top 20 visited webpages from Alexa [17].[2] Since the contents of a webpage change dynamically over time, we averaged the results of the 50 experimental runs for each webpage. We enabled QUIC and HTTP 3.0 in the Chrome browser of every UE. Out of 20 webpages, 9 of them support QUIC and we show the PLT cumulative distribution of OutRAN and srsRAN for the top 5 popular webpages among the QUIC supported ones in Figure 12. The rest are presented in Appendix §B Figure 21.

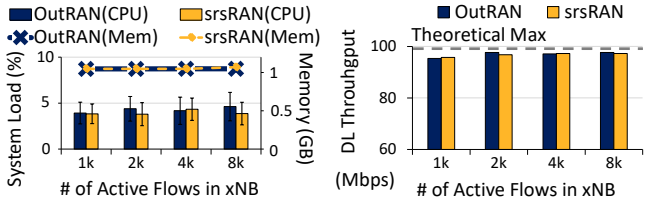
To sum up, for 20 webpages, OutRAN improves the FCT on average by 20% (53 ms) and up to 46%, which translates to 14% (626 ms) on average and up to 34% improvement in PLT compared to the srsRAN. This is because a webpage consists of multiple short sub-flows and OutRAN completes each of them quickly. Note that even a small improvement in a web application has a great impact

[2]The experiment is conducted between Sep. 7, 2021 and Sep. 11, 2021.



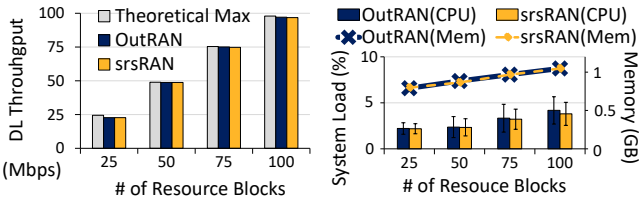
(a) Google.com **(b)** YouTube.com **(c)** Netflix.com **(d)** Facebook.com **(e)** Reddit.com

Figure 12: [Testbed] Sample PLT results of top 20 webpages from Alexa [17].



(a) Avg. CPU & memory usage. **(b)** Peak DL throughput.

Figure 13: [Testbed] Throughput & resource usage.



(a) DL Throughput. **(b)** Resource usage.

Figure 14: [Testbed] Scalability vs RBs.

on user experience since even a small delay can lead to a significant profit loss [36, 51].

For the QUIC supported webpages, OutRAN improves PLT by 14.2% (587 ms) on average except for the Zoom.us. This is because even when QUIC supports stream multiplexing in a single connection, the flow size of a single QUIC flow (avg. 147 KB, max 736 KB) is still short compared to our background flows (avg. 1.92 MB). For Zoom.us, OutRAN improves the sub-flow FCTs by 25% on average although the PLT has no improvement. We believe that this is because, for some web pages, other factors such as rendering time take up the dominant fraction in PLT [64]. The flow size breakdown on the QUIC flows of web pages is provided in Appendix §B Table 2. The video samples that show the PLT improvement by OutRAN are available at https://ina.kaist.ac.kr/projects/outran.

xNodeB performance. As we run the testbed experiment for Figure 12, we measured the actual spectral efficiency and the fairness achieved during runtime. The result shows that on average, OutRAN provides almost the same spectral efficiency compared to the baseline and costs only 4% of the fairness among users.

Throughput & resource usage. To measure the overhead of OutRAN, we evaluate CPU and memory usage by increasing the number of flows in an ingress downlink traffic of the base station with COTS UEs. We model an extreme traffic surge scenario where a large number of flows (from 1K to 8K) are ingressed and remain active during a short period (≈50 sec). Figure 13(a) shows that OutRAN requires almost the same memory usage and imposes marginal overhead (max +2.5% at peak) on CPU usage. The overhead has a negligible impact on the processing throughput of srsENB as shown in Figure 13(b), having only up to 2.73% performance gap from the theoretical max DL throughput. The additional scheduling delay of OutRAN is ≈150 ns per PDCP SDU, which is mainly due to the flow identification at the PDCP layer. Compared to the shortest scheduling TTI in NR (125 us), the latency is negligible.

To see if OutRAN can scale, we evaluate the same metrics with the same scenario by increasing the number of RBs (i.e., DL bandwidth). The overhead is negligible as shown in Figure 14. This is because the scheduler of OutRAN runs within the same computation complexity of the MAC scheduler as explained in §4.3. Our design guarantees the performance of the underlying MAC scheduler and its RF hardware.

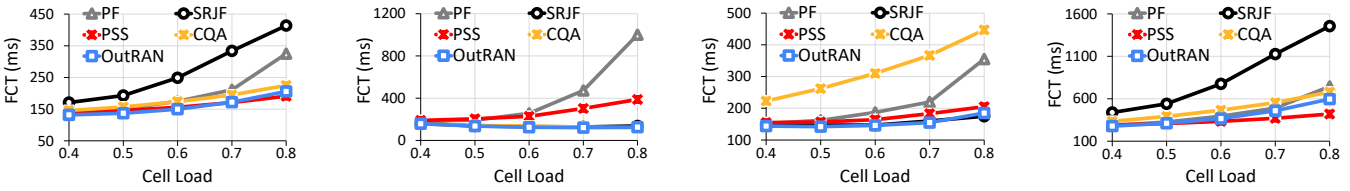
Experiments on Colosseum. To demonstrate that OutRAN works on real-world RF scenarios, we test OutRAN on Colosseum [31]. We build an LXC container of our srsENB codebase and deploy it on a Colosseum platform. We set up a four-cell topology that consists of 4 eNodeBs and 16 UEs, where each eNodeB maintains 4 UEs. For the RF scenario, we apply Rome, Boston, and POWDER provided in [21]. Each scenario has different characteristics of user mobility and geographical locations. Every UE and base station is equipped with USRP X310 as an RF frontend. We model traffic where each UE requests DL traffic from a server hosted behind the eNodeB, according to a Poisson process with a size distribution that follows LTE traffic distribution [41], for three different cell loads {20, 40, 60}%. OutRAN improves average FCT and short flow FCT by 32% and 56% respectively without having a negative impact on the long flows compared to the vanilla srsRAN. Due to the space limit, we present Figure 19 in Appendix§B.

6.2 Cell Scale Simulations

Topology. We use NS-3 to simulate a single-cell RAN topology that consists of a single xNodeB, CN, and a server with 10 ms wired link delay from the P-GW. For the TCP congestion control, we use the TCP-Cubic for both UEs and remote server. The UEs are positioned randomly within a 200 m radius from the xNodeB having random mobility with an average walking speed of 1.4 m/s. Figure 11(b) shows the topology.

NS-3 supports different modules for LTE [61] and 5G [32]. For the 5G module, we use the latest release and simulated with the maximum number of UEs supported on a gNodeB. Below is the main difference in the 5G/LTE setting:

5G setting. gNodeB operates in Band n257 (28 GHz) in Standalone (SA) mode with 100 MHz bandwidth. We use different types of numerologies (0-3) to see their impact. A single MIMO layer is used and the duplexing mode is TDD with 5DDDSU format as recommended in [38, 44]. A total of 40 UEs are attached to gNodeB. For



(a) Overall Average. **(b)** (0,10 KB] (S) 95%-ile. **(c)** (10 KB,0.1 MB] (M) Avg. **(d)** (0.1 MB,∞] (L) Avg.

Figure 15: [NS-3 LTE] [61] FCT across different cell load in LTE under LTE cellular workload [41].

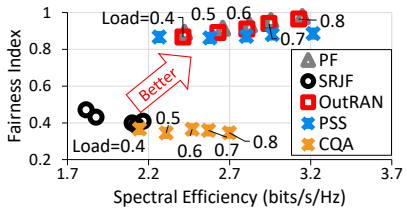


Figure 16: [NS-3 LTE] Overall spectral efficiency and fairness comparison.

Settings		Cell Load = 10%		Cell Load = 60%					
Server Location	Numerology / Slot (us)	①RTT (ms)		④ (S) 95%-ile FCT (ms)		③ (S) Q Delay		②Avg. Q Delay	
		PF	OutRAN	PF	OutRAN	PF	OutRAN	PF	OutRAN
Remote (Prop Delay = 20ms)	0 / 1000	59	61	191	188	2.1	2.2	7	6
	1 / 500	50	51	164	161	2.0	1.2	6	5
	2 / 250	45	46	157	146	3.1	0.6	7	6
	3 / 125	43	43	152	138	4.8	0.3	9	7
MEC (Prop Delay = 5ms)	0 / 1000	29	31	144	97	3.4	2.2	13	12
	1 / 500	20	21	138	71	4.3	1.1	15	14
	2 / 250	15	16	141	57	6.6	0.6	15	16
	3 / 125	13	13	138	49	8.8	0.3	16	18

Towards Low-latency

Queue Buildup

Figure 17: [NS-3 5G] [32] Impact of OutRAN in 5G RAN.

channel dynamics, we apply the urban channel scenario provided in the module [32].

LTE setting. eNodeB operates in Band3 (1805 MHz DL) with 20 MHz bandwidth in Transmission Mode 5 (MU-MIMO). A total of 100 UEs are attached to eNodeB. To model a realistic channel condition. we use 3GPP channel trace [2].

For both settings, the maximum buffer size of the RLC UM entity is set to the default value of srsENB [69].

Workloads. The setup is identical to the experiments on Colosseum but with cell loads of 40-80%. We create a total of 10 K flows on average for each simulation.

Baselines. In addition to PF, OutRAN is compared with the following baselines:

- **Shortest Remaining Job First (SRJF)** is an optimal flow scheduling scheme in DCN that has perfect knowledge of flow size. SRJF schedules flows based on the remaining flow size, being ignorant of the channel condition.
- **Priority Set Scheduler (PSS)** [56] and **Channel & QoS-aware (CQA) Scheduler** [20] are variants of PF scheduler that support QoS provisioning. We assume they are aware of the flow size of each flow, and apply QoS of low-latency service type (delay budget=50 ms) for short flows (< 10 KB). PSS and CQA are supported in the NS-3 LTE LENA module [61].

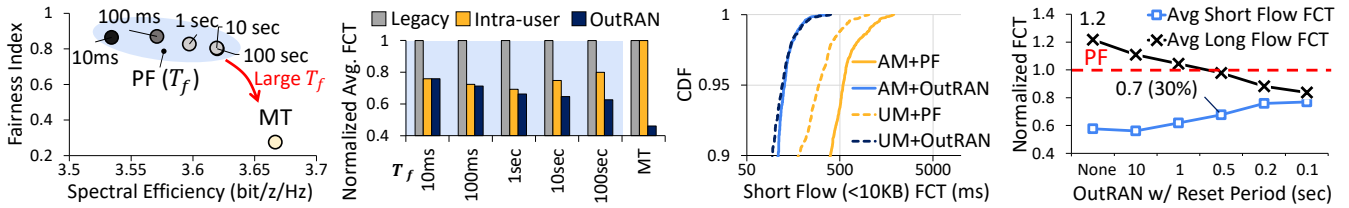
FCT minimization. Figure 15 shows the FCT results across different cell loads in LTE respectively. OutRAN effectively minimizes the average (figure omitted) and tail FCT (Figure 15(b)) of short flows, outperforming the PF, while showing comparable results to the SRJF. The result also shows that even when the load increases, OutRAN keeps the short flow FCT steady while in PF, it dramatically gets inflated. The 5G simulation result also shows the same trend and the results are presented in Appendix §B Figure 20.

Long flow FCT. Since OutRAN entails preemption of long flows, one may concern the starvation. That said, OutRAN does not starve long flows and provides slightly better FCT for them compared to PF in our evaluation. This is because as the cellular network exhibits heavy-tail distribution [41], prioritizing the short flows actually improves the majority of the flows as most flows in the network are small. Also, prioritizing short flows has little impact on long flows as short flows take up only a small amount of traffic volume in the network. In fact, it even helps the long flows by finishing the majority of short flows first and mitigating network contention [14, 40].

In contrast to OutRAN, Figure 15(d) shows that SRJF adversely affects the FCT of long flow in LTE, which leads to the worst performance in overall FCT (Figure 15(a)). This is because SRJF schedules solely upon the remaining flow size ignoring the actual channel quality of UEs. For example, in the worst case where SRJF starts to schedule a user flow that has the worst channel quality, the user will grab all the bandwidth (with poor spectral efficiency) to finish its flow. This may last a long time due to its poor throughput and will eventually starve the other user flows. OutRAN, on the other hand, prioritizes users with short flows among the ones who are going through a good channel condition at the moment.

Compatability. Figure 16 compares the overall spectral efficiency and fairness between schedulers. While other latency-optimized schedulers cost spectral efficiency or fairness of the PF by up to 33% and 65% respectively, OutRAN preserves at least 98% and 97% of spectral efficiency and the user fairness of that PF scheduler provides. The SRJF collapses in both metrics more severely because of the high variance in channel dynamics in the simulation trace.

vs. QoS-aware schedulers. The QoS-aware schedulers improve the FCT of PF, as it prioritizes short flows with a tight packet delay budget. However, this either provides suboptimal performance in short flow FCT as PSS performs in Figure 15(b), or entails starvation of other (user) flows as CQA performs in Figure 15(c). OutRAN outperforms QoS-aware schedulers in terms of spectral efficiency and fairness. Also, OutRAN shows similar short flow FCT compared to the CQA scheduler but ours does not require any prior knowledge of QoS or flow size which makes it more practical and robust.



(a) PF with different T_f. **(b)** Ablation study with T_f. **(c)** Case for RLC AM mode. **(d)** Resetting the priority.

Figure 18: [NS-3 LTE] Simulation results for case studies.

Impact in 5G. OutRAN becomes even more attractive in 5G considering its efforts towards low latency. According to the 5G measurements [58, 77], the latency reduction in today's 5G compared to LTE is mainly attributed to two factors; 1) the edgification of server and core networks (which significantly reduces the latency in the backhaul network), and 2) the 5G NR numerology used by the gNodeB in RAN (which brings low latency "in-air"). Our simulation presented in Figure 17 shows consistent results with the study. When a cell experiences modest traffic congestion (Load=10%), RTT significantly reduces as the server gets closer to gNodeB and more advanced NR numerology is used by the gNodeB (①).

However, we observe in our simulation that the downlink traffic arrives in more bursty patterns at the gNodeB when the end-to-end latency gets reduced and the traffic load increases. As a result, when cell load increases (Load=60%), such trend exacerbates the queue build-up in the gNodeB (②) and short flows suffer from the severe queuing delay (③). This leads to the inflation of short flow FCT (④) even with the most advanced RAN settings (MEC+numerology=3).

OutRAN alleviates this by reducing the queueing delay of short flows at gNodeB (③) which results in lower short tail FCT (④). Note that short flow FCT of OutRAN improves along with the more advanced RAN settings. OutRAN achieves this at a minimal cost in spectral efficiency and fairness. The result demonstrates that to fully exploit the low-latency aspect of 5G, traffic optimization at the *base station*—which is what OutRAN performs—is an essential part of the future 5G RAN.

6.3 Case Study

In this section, we provide different case studies of OutRAN with using a series of targeted simulations.

Ablation study. The PF scheduler provides a trade-off between the spectral efficiency and the user fairness and this trade-off can be adjusted by specifying a time window for imposing fairness among the users. Such time window is referred to as the *fairness window* (T_f) [37, 67] and it is used as a smoothing factor in calculating the long-term average throughput of the PF scheduler's per-RB metric (eq. 1). While PF with a small fairness window (T_f = 10 ms) performs similarly to the Round-Robin scheduling, PF with a large fairness window ($T_f \geq$ 100 sec) performs toward MT scheduling [24, 67] as shown in Figure 18(a).

We observe that when PF is more optimized towards fairness (i.e., tuned with a small T_f) and works with the Intra-user Flow Scheduler (i.e., OutRAN ($\epsilon = 0$)), the PF acts similarly to the Inter-user Flow Scheduler. This is because the PF scheduler guarantees service to every user within a certain time (i.e., fairness window), and this characteristic prevents starvation of the short-flows from inter-users, which ultimately results in a similar scheduling decision as the SJF.

To identify the contribution of each design component of OutRAN in detail, we conduct an ablation study with different T_f, and Figure 18(b) shows the result. The legacy scheduler is either PF with different T_f or MT, and Intra-user Scheduler is basically OutRAN without Inter-user Flow Scheduler. When PF uses a small T_f, we see that most of the improvement comes from the Intra-user Flow Scheduler[3]. On the other hand, the Inter-user Flow Scheduler becomes significant and provides a greater benefit when the PF scheduler uses a larger fairness window; it provides 11% more improvement over the Intra-user Scheduler when T_f = 10 sec.

According to the studies in PF schedulers [37, 57], the fairness does not need to be satisfied in a very short time (i.e., in a few milliseconds) and a few seconds for the fairness window should be sufficient. In practice, there is no standard for the cellular scheduler design and its parameter settings are often the manufacturer's "secret sauce". Even so, OutRAN can work with any scheduler variants; our result in Figure 18(b) demonstrates that both design components are meaningful and OutRAN with the two design components all together always outperforms the legacy scheduler in FCT.

RLC AM mode. The choice between the RLC UM and the AM mode is made during the bearer setup procedure according to the loss rate specified in the QoS [67]. We choose the RLC UM as the default mode since our work assumes the case when QoS is not available, and the UM is mainly utilized by delay-sensitive applications [67]. Still, OutRAN can also work with the RLC AM mode, and we show this by a simulation result. The setting is identical to the one in §6.2, and for the AM mode, the timers related to the retransmissions (e.g., t-PollRetransmit, t-statusProhibitTimer) [6] are set to the default values provided in NS-3 LTE LENA module [61]. Compared to PF, OutRAN shows consistent trends with the UM case shown in our main evaluations in terms of FCT (improving the average by 30%), spectral efficiency, and fairness (preserving 97% and 95% respectively). In particular, we observe the tail CDF of short flow FCT in Figure 18(c).

It is worth noting that when the PF operates with the AM mode, the FCT gets increased compared to the case when the PF operates with the UM mode. This complies with the general knowledge that the UM is better for latency-sensitive applications due to the simplified data transfer without the retransmission process. In fact, unless the timer values for the retransmissions are carefully set, the RLC AM entity could generate unnecessary retransmissions [55] which

[3]When it comes to the tail latency, Inter-user Flow Scheduler also provides improvement (10%) even for small T_f. The contribution may change depending on the channel dynamics, cell traffic load, and user distribution.

results in wasting the bandwidth. This leaves less transmission opportunities for the flows that are waiting in the transmission queue, and it especially poses an issue on the latency-sensitive short flows when they are queued behind the long flows.

OutRAN with the AM mode, however, still provides a good short flow FCT because it first prioritizes the short flows within the available bandwidth after scheduling the retransmission queue. It shows even better short flow FCT than the PF operating with the UM mode in our simulation results. Overall, OutRAN with the UM mode showed the best performance in FCT.

Priority reset. OutRAN does not starve the long flows in our evaluations. However, there could be some worst-case scenarios where too many short flows interrupt the long flows and thus, OutRAN hurts the performance of the flows that are latency-sensitive but long-lived (e.g., video streaming). To alleviate the issue, well-known solutions are already proposed [16, 18], and one of them is "Priority Boost", which is resetting the flow state of every flow and moving all flows to the topmost queue after some time period S.

Likewise, we suggest that the measure is also applicable to our system. In our simulation, we setup an extreme incast-like scenario where 8 KB of short flows arrive simultaneously to xNodeB taking up 10% of the traffic volume. The rest follows LTE traffic distribution [41] and the total load is set to 80%. We compare PF and OutRAN configured with the reset period S, and Figure 18(d) shows the average FCT of both long and short flows that are normalized to the ones of PF. Compared to PF, in OutRAN, the short flow FCT gets reduced by 40% while long flow FCT gets increased by 20%. As we apply the priority reset and shorten the period S, we can push down the long flow FCT while sacrificing the improvement for the short flows. In fact, when $S = 500$ ms, the long flow FCT remains almost the same as the PF, and OutRAN still provides significant improvement for short flow FCT (30% and 62% for average and 95%-ile respectively). The period S can be tuned according to the network operator's interest.

7 DISCUSSIONS & LIMITATIONS

When FCT is not the primary goal. The FCT may not be the best metric for streaming applications such as audio/video streaming. We clarify that these applications are not the main target of our system. If such applications require real-time service (i.e., video conferencing), they are classified as Conversational class and enforced with the dedicated QoS profile. Thus, we believe that their real-time traffic could be isolated from our target traffic class, which is the best-effort traffic with the default QoS profile.

One exception would be the non-realtime ones that fall into the Interactive or Background traffic class. When they are latency-sensitive but long-lived (i.e., TCP-based video streaming), our design could somehow sacrifice their performance. We introduce the safety measure that can be applied to alleviate the issue in §6.3.

Safeguard to prevent gaming. A user may try to game the system by intentionally splitting its flows into multiple short flows to get better service. However, such malicious behavior of a user will not be an issue for OutRAN if the underlying cellular scheduler of OutRAN is PF. Considering that today's most used cellular scheduler is the PF, OutRAN will maintain fairness among the users that PF provides as it respects its optimization objectives.

Overhead on a commercial xNodeB. OutRAN is implemented on top of srsRAN which performs baseband processing using CPU. In the commercial xNodeB, heavy PHY layer processing tasks (e.g., LDPC) [33] are offloaded to hardware such as FPGA or GPU [62]. If so, Layer 2 becomes a dominant processing part. The operation of OutRAN resides at the Layer 2, thus the CPU overhead of OutRAN could become a burden in those cases. To reduce the cost, flow identification, which is the main overhead of OutRAN, could also be offloaded to the User Plane Function (UPF) of the Core Network [45].

Handover. When the handover procedure is triggered, the source xNodeB forwards data freshly arriving (also the buffered data if it's a lossless handover) to the xNodeB of the target cell. This paper does not cover such case, but the flow state of a user can also be copied along with the data. The flow state requires 41 bytes per flow (37 bytes for the five tuple and 4 bytes for the sent-byte). If copying the flow state is considered a burden, we can reset the state at the new xNodeB and start fresh.

8 RELATED WORK

Cross-layer interaction. There has been a large number of efforts to improve the user experience in cellular networks based on cross-layer interaction. They include cross-layer optimization of transport protocols [60, 74, 79] and improving the application performance by leveraging the cross-layer information [23, 25, 75, 76]. We also propose an approach based on the cross-layer interaction but study a different topic and focus on improving the base station scheduling.

Optimizing the MAC scheduler. To meet the stringent time requirement in 5G NR scheduling, recent works [27, 42, 43] present GPU-based PF scheduler, which achieves the near-optimal performance of PF scheduler at low latency. OutRAN has a different optimization objective which is co-optimizing the FCT and the PF metric. We believe that OutRAN can also be combined with such a highly optimized scheduler by applying the same rationale of our scheduling design.

Software-based baseband processing. Existing works [33, 70, 72, 78] perform baseband processing at the base station in software. Agora [33], the state-of-the-art in the area, supports real-time massive MIMO on a single server. Our testbed performs baseband processing in software and we expect that OutRAN can also work with the existing software-based baseband processing.

9 CONCLUSION

In this work, we present OutRAN, a practical flow scheduler designed for Radio Access Network. Inspired by the transport layer scheduling design in the datacenter network, OutRAN demonstrates that it is possible to apply the concept of Flow Completion Time (FCT) to the cellular network by modifying the resource scheduler embedded in the base station. OutRAN deals with the unique challenge of applying the concept to the cellular network and solves it by co-optimizing the FCT with legacy cellular scheduler metrics. As a result, OutRAN provides better latency for interactive applications by completing their short flows quickly when scheduling downlink radio resource at the last-mile base station. The resulting system is practical since it does not require prior information on the traffic and is compatible with LTE/5G.

ACKNOWLEDGMENTS

We would like to thank our shepherd Qing Wang and anonymous reviewers for their constructive feedback. We appreciate Colosseum Team for providing the access to the Colosseum platform and Electronics and Telecommunications Research Institute (ETRI) for providing the 5G Open Test Lab. We also thank Jeong Hwan Kim from ETRI for helping us use the Test Lab. This work is supported by Samsung Electronics Co., Ltd. Modem S/W R&D Group and was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (Ministry of Science and ICT) [No.2018-0-00693].

REFERENCES

[1] 3GPP. 2011. 3GPP TS 23.203 version 9.7.0 Release 9, Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Policy and charging control architecture. https://www.etsi.org/deliver/etsi_ts/123200_123299/123203/09.07.00_60/ts_123203v090700p.pdf.

[2] 3GPP. 2017. 3GPP TS 36.141 version 14.3.0 Release 14. https://www.etsi.org/deliver/etsi_ts/136100_136199/136141/14.03.00_60/ts_136141v140300p.pdf.

[3] 3GPP. 2018. 3GPP TS 38.101-1 version 15.2.0 Release 15, 5G; NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone;. https://www.etsi.org/deliver/etsi_ts/138100_138199/13810101/15.02.00_60/ts_13810101v150200p.pdf.

[4] 3GPP. 2018. 3GPP TS 38.211 version 15.2.0 Release 15, Physical channels and modulation. https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/15.02.00_60/ts_138211v150200p.pdf.

[5] 3GPP. 2018. 5G; NR; Packet Data Convergence Protocol (PDCP) specification (3GPP TS 38.323 version 15.2.0 Release 15). https://www.etsi.org/deliver/etsi_ts/138300_138399/138323/15.02.00_60/ts_138323v150200p.pdf.

[6] 3GPP. 2018. 5G; NR; Radio Link Control (RLC) protocol specification (3GPP TS 38.322 version 15.3.0 Release 15). https://www.etsi.org/deliver/etsi_ts/138300_138399/138322/15.03.00_60/ts_138322v150300p.pdf.

[7] 3GPP. 2018. 5G; System Architecture for the 5G System (3GPP TS 23.501 version 15.3.0 Release 15). https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/15.03.00_60/ts_123501v150300p.pdf.

[8] 3GPP. 2018. Quality of Service (QoS) concept and architecture (3GPP TS 23.107 version 15.0.0 Release 15). https://www.etsi.org/deliver/etsi_ts/123100_123199/123107/15.00.00_60/ts_123107v150000p.pdf.

[9] 3GPP. 2019. Interface between the Control Plane and the User Plane nodes (3GPP TS 29.244 version 15.5.0 Release 15). https://www.etsi.org/deliver/etsi_ts/129200_129299/129244/15.05.00_60/ts_129244v150500p.pdf.

[10] 3GPP. 2021. 3GPP TS 38.300 version 16.4.0 Release 16, NR and NG-RAN Overall description. https://www.etsi.org/deliver/etsi_ts/138300_138399/138300/16.04.00_60/ts_138300v160400p.pdf.

[11] Accuver. 2022. XCAL-Pu12. https://www.accuver.com/sub/products/view.php?idx=5#\protect\protect\leavevmode@ifvmode\kern+.2222em\relax~\protect\protect\leavevmode@ifvmode\kern+.2222em\relaxtext=The%20Pu12%20can%20support%20up,is%20suitable%20for%20walk%20test.

[12] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, Valerio Persico, and Antonio Pescapé. 2019. MIRAGE: Mobile-app Traffic Capture and Ground-truth Creation. In *2019 4th International Conference on Computing, Communications and Security (ICCCS)*. IEEE, New York, NY, USA, 1–8. https://doi.org/10.1109/CCCS.2019.8888137

[13] Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data Center TCP (DCTCP). In *Proceedings of the ACM SIGCOMM 2010 Conference* (New Delhi, India) *(SIGCOMM '10)*. Association for Computing Machinery, New York, NY, USA, 63–74. https://doi.org/10.1145/1851182.1851192

[14] Mohammad Alizadeh, Shuang Yang, Milad Sharif, Sachin Katti, Nick McKeown, Balaji Prabhakar, and Scott Shenker. 2013. PFabric: Minimal near-Optimal Datacenter Transport. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (Hong Kong, China) *(SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 435–446. https://doi.org/10.1145/2486001.2486031

[15] W3C Navigation Timing API. 2021. W3C Navigation Timing API Official Document. https://www.w3.org/TR/navigation-timing/#dom-performancetiming-navigationstart.

[16] Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. 2018. *Operating Systems: Three Easy Pieces* (1.00 ed.). Arpaci-Dusseau Books, -.

[17] Amazon Web Service (AWS). 2021. Alexa Top Sites. https://www.alexa.com/topsites.

[18] Wei Bai, Li Chen, Kai Chen, Dongsu Han, Chen Tian, and Hao Wang. 2015. Information-Agnostic Flow Scheduling for Commodity Data Centers. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. USENIX Association, Oakland, CA, 455–468. https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/bai

[19] Wei Bai, Li Chen, Kai Chen, Dongsu Han, Chen Tian, and Hao Wang. 2017. PIAS: Practical Information-Agnostic Flow Scheduling for Commodity Data Centers. *IEEE/ACM Transactions on Networking* 25, 4 (2017), 1954–1967. https://doi.org/10.1109/TNET.2017.2669216

[20] Biljana Bojovic and Nicola Baldo. 2014. A new channel and QoS aware scheduler to enhance the capacity of voice over LTE systems. In *2014 IEEE 11th International Multi-Conference on Systems, Signals Devices (SSD14)*. IEEE, New York, NY, USA, 1–6. https://doi.org/10.1109/SSD.2014.6808890

[21] Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. 2021. SCOPE: An Open and Softwarized Prototyping Platform for NextG Systems. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (Virtual Event, Wisconsin) *(MobiSys '21)*. Association for Computing Machinery, New York, NY, USA, 415–426. https://doi.org/10.1145/3458864.3466863

[22] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. 2020. Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead. *Computer Networks* 182 (2020), 107516. https://doi.org/10.1016/j.comnet.2020.107516

[23] Kevin Boos, David Chu, and Eduardo Cuervo. 2016. FlashBack: Immersive Virtual Reality on Mobile Devices via Rendering Memoization. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services* (Singapore, Singapore) *(MobiSys '16)*. Association for Computing Machinery, New York, NY, USA, 291–304. https://doi.org/10.1145/2906388.2906418

[24] F. Capozzi, G. Piro, L.A. Grieco, G. Boggia, and P. Camarda. 2013. Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey. *IEEE Communications Surveys Tutorials* 15, 2 (2013), 678–700. https://doi.org/10.1109/SURV.2012.060912.00100

[25] Abhijnan Chakraborty, Vishnu Navda, Venkata N. Padmanabhan, and Ramachandran Ramjee. 2013. Coordinating Cellular Background Transfers Using Loadsense. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking* (Miami, Florida, USA) *(MobiCom '13)*. Association for Computing Machinery, New York, NY, USA, 63–74. https://doi.org/10.1145/2500423.2500447

[26] Li Chen, Justinas Lingys, Kai Chen, and Feng Liu. 2018. AuTO: Scaling Deep Reinforcement Learning for Datacenter-Scale Automatic Traffic Optimization. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (Budapest, Hungary) *(SIGCOMM '18)*. Association for Computing Machinery, New York, NY, USA, 191–205. https://doi.org/10.1145/3230543.3230551

[27] Yongce Chen, Yubo Wu, Y Thomas Hou, and Wenjing Lou. 2021. mCore: Achieving Sub-millisecond Scheduling for 5G MU-MIMO Systems. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, IEEE, New York, NY, USA, 1–10.

[28] Yongmin Choi, Cha-hyun Yoon, Young-sik Kim, Seo Weon Heo, and John A. Silvester. 2014. The impact of application signaling traffic on public land mobile networks. *IEEE Communications Magazine* 52, 1 (2014), 166–172. https://doi.org/10.1109/MCOM.2014.6710079

[29] Mosharaf Chowdhury and Ion Stoica. 2015. Efficient Coflow Scheduling Without Prior Knowledge. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (London, United Kingdom) *(SIGCOMM '15)*. Association for Computing Machinery, New York, NY, USA, 393–406. https://doi.org/10.1145/2785956.2787480

[30] Colosseum. 2020. Colosseum srsLTE 20-04 gihub repository. https://github.com/colosseum-wiot/colosseum-srslte-20-04.

[31] Colosseum. 2022. Colosseum Officcial Website. https://www.northeastern.edu/colosseum/.

[32] CTTC. 2022. NS-3 5G LENA module. https://5g-lena.cttc.es/.

[33] Jian Ding, Rahman Doost-Mohammady, Anuj Kalia, and Lin Zhong. 2020. Agora: Real-Time Massive MIMO Baseband Processing in Software. In *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies* (Barcelona, Spain) *(CoNEXT '20)*. Association for Computing Machinery, New York, NY, USA, 232–244. https://doi.org/10.1145/3386367.3431296

[34] Nandita Dukkipati and Nick McKeown. 2006. Why Flow-Completion Time is the Right Metric for Congestion Control. *SIGCOMM Comput. Commun. Rev.* 36, 1 (Jan. 2006), 59–62. https://doi.org/10.1145/1111322.1111336

[35] ETRI. 2022. 5G Open Testlab. https://www.5gdaejeon.or.kr/etri.

[36] Tobias Flach, Nandita Dukkipati, Andreas Terzis, Barath Raghavan, Neal Cardwell, Yuchung Cheng, Ankur Jain, Shuai Hao, Ethan Katz-Bassett, and Ramesh Govindan. 2013. Reducing Web Latency: The Virtue of Gentle Aggression. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (Hong Kong, China) *(SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 159–170. https://doi.org/10.1145/2486001.2486014

[37] Tolga Girici, Chenxi Zhu, Jonathan R. Agre, and Anthony Ephremides. 2010. Proportional fair scheduling algorithm in OFDMA-based wireless systems with QoS constraints. *Journal of Communications and Networks* 12, 1 (2010), 30–42. https://doi.org/10.1109/JCN.2010.6388432

[38] GSMA. 2020. 5G TDD Synchronisation Guidelines and Recommendations for the Coexistence of TDD Networks in the 3.5 GHz Range. https://www.gsma.com/spectrum/wp-content/uploads/2020/04/3.5-GHz-5G-TDD-Synchronisation.pdf.

[39] Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS operating systems review* 42, 5 (2008), 64–74.

[40] Chi-Yao Hong, Matthew Caesar, and P. Brighten Godfrey. 2012. Finishing Flows Quickly with Preemptive Scheduling. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication* (Helsinki, Finland) *(SIGCOMM '12)*. Association for Computing Machinery, New York, NY, USA, 127–138. https://doi.org/10.1145/2342356.2342389

[41] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. 2013. An In-Depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (Hong Kong, China) *(SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 363–374. https://doi.org/10.1145/2486001.2486006

[42] Yan Huang, Shaoran Li, Y. Thomas Hou, and Wenjing Lou. 2018. GPF: A GPU-Based Design to Achieve 100 us Scheduling for 5G NR. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 207–222. https://doi.org/10.1145/3241539.3241552

[43] Yan Huang, Shaoran Li, Y. Thomas Hou, and Wenjing Lou. 2022. GPF+: A Novel Ultrafast GPU-Based Proportional Fair Scheduler for 5G NR. *IEEE/ACM Transactions on Networking* 30, 2 (2022), 601–615. https://doi.org/10.1109/TNET.2021.3118005

[44] Huawei. 2020. 5G Spectrum Public Policy Position. https://www-file.huawei.com/-/media/corporate/pdf/public-policy/public_policy_position_5g_spectrum_2020_v2.pdf?la=en.

[45] Vivek Jain, Hao-Tse Chu, Shixiong Qi, Chia-An Lee, Hung-Cheng Chang, Cheng-Ying Hsieh, K. K. Ramakrishnan, and Jyh-Cheng Chen. 2022. L25GC: A Low Latency 5G Core Network Based on High-Performance NFV Platforms. In *Proceedings of the ACM SIGCOMM 2022 Conference* (Amsterdam, Netherlands) *(SIGCOMM '22)*. Association for Computing Machinery, New York, NY, USA, 143–157. https://doi.org/10.1145/3544216.3544267

[46] Haiqing Jiang, Zeyu Liu, Yaogong Wang, Kyunghan Lee, and Injong Rhee. 2012. Understanding Bufferbloat in Cellular Networks. In *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design* (Helsinki, Finland) *(CellNet '12)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/2342468.2342470

[47] Haiqing Jiang, Yaogong Wang, Kyunghan Lee, and Injong Rhee. 2012. Tackling Bufferbloat in 3G/4G Networks. In *Proceedings of the 2012 Internet Measurement Conference* (Boston, Massachusetts, USA) *(IMC '12)*. Association for Computing Machinery, New York, NY, USA, 329–342. https://doi.org/10.1145/2398776.2398810

[48] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2016. SciPy: Open source scientific tools for Python, 2001.

[49] KOREN. 2022. 5G Open Testlab brochure. https://bit.ly/5gopentestlabbrochure.

[50] H.J. Kushner and P.A. Whiting. 2004. Convergence of proportional-fair sharing algorithms under general conditions. *IEEE Transactions on Wireless Communications* 3, 4 (2004), 1250–1259. https://doi.org/10.1109/TWC.2004.830826

[51] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, Jeff Bailey, Jeremy Dorfman, Jim Roskind, Joanna Kulik, Patrik Westin, Raman Tenneti, Robbie Shade, Ryan Hamilton, Victor Vasiliev, Wan-Teh Chang, and Zhongyi Shi. 2017. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. Association for Computing Machinery, New York, NY, USA, 183–196. https://doi.org/10.1145/3098822.3098842

[52] Libin Liu, Chengxi Gao, Peng Wang, Hongming Huang, Jiamin Li, Hong Xu, and Wei Zhang. 2021. Bottleneck-Aware Non-Clairvoyant Coflow Scheduling with Fai. *IEEE Transactions on Cloud Computing* ., . (2021), 1–1. https://doi.org/10.1109/TCC.2021.3128360

[53] Xin Liu, Ashwin Sridharan, Sridhar Machiraju, Mukund Seshadri, and Hui Zang. 2008. Experiences in a 3G Network: Interplay between the Wireless Channel and Applications. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking* (San Francisco, California, USA) *(MobiCom '08)*. Association for Computing Machinery, New York, NY, USA, 211–222. https://doi.org/10.1145/1409944.1409969

[54] Cristina Marquez, Marco Gramaglia, Marco Fiore, Albert Banchs, Cezary Ziemlicki, and Zbigniew Smoreda. 2017. Not All Apps Are Created Equal: Analysis of Spatiotemporal Heterogeneity in Nationwide Mobile Service Usage. In *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies* (Incheon, Republic of Korea) *(CoNEXT '17)*. Association for Computing Machinery, New York, NY, USA, 180–186. https://doi.org/10.1145/3143361.3143369

[55] Jessica Mendoza, Isabel de-la Bandera, David Palacios, Ana Herrera-García, and Raquel Barco. 2020. On the capability of QoE improvement based on the adjustment of RLC parameters. *Sensors* 20, 9 (2020), 2474.

[56] G. Monghal, Klaus I. Pedersen, I. Z. Kovacs, and P. E. Mogensen. 2008. QoS Oriented Time and Frequency Domain Packet Schedulers for The UTRAN Long Term Evolution. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*. IEEE, New York, NY, USA, 2532–2536. https://doi.org/10.1109/VETECS.2008.557

[57] Sameh Musleh, Mahamod Ismail, and Rosdiadee Nordin. 2015. Effect of average-throughput window size on proportional fair scheduling for radio resources in LTE-A networks. *Journal of Theoretical and Applied Information Technology* 80, 1 (2015), 179.

[58] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, Feng Qian, and Zhi-Li Zhang. 2021. A Variegated Look at 5G in the Wild: Performance, Power, and QoE Implications. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (Virtual Event, USA) *(SIGCOMM '21)*. Association for Computing Machinery, New York, NY, USA, 610–625. https://doi.org/10.1145/3452296.3472923

[59] Netmanias. 2013. Netmanias Technical Document: LTE QoS: SDF and EPS Bearer QoS. https://www.netmanias.com/en/?m=view&id=techdocs&no=5908.

[60] Binh Nguyen, Arijit Banerjee, Vijay Gopalakrishnan, Sneha Kasera, Seungjoon Lee, Aman Shaikh, and Jacobus Van der Merwe. 2014. Towards Understanding TCP Performance on LTE/EPC Mobile Networks. In *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges* (Chicago, Illinois, USA) *(AllThingsCellular '14)*. Association for Computing Machinery, New York, NY, USA, 41–46. https://doi.org/10.1145/2627585.2627594

[61] NS-3. 2022. NS-3 LTE LENA module. https://www.nsnam.org/docs/models/html/lte-user.html.

[62] NVIDIA. 2021. NVIDIA Aerial SDK. https://developer.nvidia.com/aerial-sdk.

[63] Zafar Ayyub Qazi, Melvin Walls, Aurojit Panda, Vyas Sekar, Sylvia Ratnasamy, and Scott Shenker. 2017. A High Performance Packet Core for Next Generation Cellular Networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. Association for Computing Machinery, New York, NY, USA, 348–361. https://doi.org/10.1145/3098822.3098848

[64] Nimish Radio, Ying Zhang, Mallik Tatipamula, and Vijay K Madisetti. 2012. Next-generation applications on cellular networks: trends, challenges, and solutions. *Proc. IEEE* 100, 4 (2012), 841–854.

[65] Joachim Sachs, Gustav Wikstrom, Torsten Dudda, Robert Baldemair, and Kittipong Kittichokechai. 2018. 5G Radio Network Design for Ultra-Reliable Low-Latency Communication. *IEEE Network* 32, 2 (2018), 24–31. https://doi.org/10.1109/MNET.2018.1700232

[66] Samsung. 2017. 4G-5G Interworking. https://images.samsung.com/is/content/samsung/p5/global/business/networks/insights/white-paper/4g-5g-interworking/global-networks-insight-4g-5g-interworking-0.pdf.

[67] Stefania Sesia, Issam Toufik, and Matthew Baker. 2011. *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, .

[68] srsRAN. 2022. srsRAN 5G NSA Application Document. https://docs.srsran.com/en/latest/app_notes/source/5g_nsa_cots/source/index.html.

[69] srsRAN. 2022. srsRAN Official Website. https://www.srslte.com/.

[70] Gordon Stewart, Mahanth Gowda, Geoffrey Mainland, Bozidar Radunovic, Dimitrios Vytiniotis, and Cristina Luengo Agullo. 2015. Ziria: A DSL for Wireless Systems Programming. *SIGPLAN Not.* 50, 4 (March 2015), 415–428. https://doi.org/10.1145/2775054.2694368

[71] Sysmocom. 2022. Programmable SIM cards from Sysmocom. https://shop.sysmocom.de/sysmoISIM-SJA2-SIM-USIM-ISIM-Card-10-pack-with-ADM-keys/sysmoISIM-SJA2-10p-adm.

[72] Kun Tan, He Liu, Jiansong Zhang, Yongguang Zhang, Ji Fang, and Geoffrey M. Voelker. 2011. Sora: High-Performance Software Radio Using General-Purpose Multi-Core Processors. *Commun. ACM* 54, 1 (Jan. 2011), 99–107. https://doi.org/10.1145/1866739.1866760

[73] Xiao Sophia Wang, Arvind Krishnamurthy, and David Wetherall. 2016. Speeding up Web Page Loads with Shandian. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. USENIX Association, Santa Clara, CA, 109–122. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/wang

[74] Keith Winstein, Anirudh Sivaraman, and Hari Balakrishnan. 2013. Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. USENIX Association, Lombard, IL, 459–471. https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/winstein

[75] Xiufeng Xie, Xinyu Zhang, Swarun Kumar, and Li Erran Li. 2016. PiStream: Physical Layer Informed Adaptive Video Streaming Over LTE. *GetMobile: Mobile Comp. and Comm.* 20, 2 (Oct. 2016), 31–34. https://doi.org/10.1145/3009808.3009819

[76] Xiufeng Xie, Xinyu Zhang, and Shilin Zhu. 2017. Accelerating Mobile Web Loading Using Cellular Link Information. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (Niagara

Falls, New York, USA) *(MobiSys '17)*. Association for Computing Machinery, New York, NY, USA, 427–439. https://doi.org/10.1145/3081333.3081367

[77] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. 2020. Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication* (Virtual Event, USA) *(SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 479–494. https://doi.org/10.1145/3387514.3405882

[78] Qing Yang, Xiaoxiao Li, Hongyi Yao, Ji Fang, Kun Tan, Wenjun Hu, Jiansong Zhang, and Yongguang Zhang. 2013. BigStation: Enabling Scalable Real-Time Signal Processingin Large Mu-Mimo Systems. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (Hong Kong, China) *(SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 399–410. https://doi.org/10.1145/2486001.2486016

[79] Yasir Zaki, Thomas Pötsch, Jay Chen, Lakshminarayanan Subramanian, and Carmelita Görg. 2015. Adaptive Congestion Control for Unpredictable Cellular Networks. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (London, United Kingdom) *(SIGCOMM '15)*. Association for Computing Machinery, New York, NY, USA, 509–522. https://doi.org/10.1145/2785956.2787498

[80] Cisco Zeljko Savic. 2011. LTE Design and Deployment Strategies. https://www.cisco.com/c/dam/global/en_ae/assets/expo2011/saudiarabia/pdfs/lte-design-and-deployment-strategies-zeljko-savic.pdf.

[81] Honghai Zhang, Narayan Prasad, and Sampath Rangarajan. 2012. MIMO downlink scheduling in LTE systems. In *2012 Proceedings IEEE INFOCOM*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 2936–2940. https://doi.org/10.1109/INFCOM.2012.6195733

[82] Ying Zhang and Ake Årvidsson. 2012. Understanding the Characteristics of Cellular Data Traffic. *SIGCOMM Comput. Commun. Rev.* 42, 4 (sep 2012), 461–466. https://doi.org/10.1145/2377677.2377764

[83] Zhehui Zhang, Shu Shi, Varun Gupta, and Rittwik Jana. 2019. Analysis of Cellular Network Latency for Edge-Based Remote Rendering Streaming Applications. In *Proceedings of the ACM SIGCOMM 2019 Workshop on Networking for Emerging Applications and Technologies* (Beijing, China) *(NEAT'19)*. Association for Computing Machinery, New York, NY, USA, 8–14. https://doi.org/10.1145/3341558.3342199

A INTER-USER FLOW SCHEDULER ALGORITHM

Algorithm 1 Scheduling Inter-user Flows

procedure RB ALLOCATION ▷ Operates every TTI t
  for each $b \in \mathcal{B}$ **do**
    initialize m_{max} and the selected user $\hat{u}$
    for each $u \in \mathcal{U}$ **do** ▷ First iteration
      calculate per-RB metric $m_{u,b}(t)$
      if $m_{max} < m_{u,b}(t)$ **then**
        update $m_{max} = m_{u,b}(t)$
        update selected user $\hat{u} = u$
    /* Re-selection by our scheduler */
    initialize max MLFQ priority P_{max}
    for each $u \in \mathcal{U}$ **do** ▷ Second iteration
      if $(1-\epsilon) \cdot m_{max} \leq m_{u,b}(t)$ **then**
        $P_u = \max_{f \in \mathcal{F}_u} Priority(f)$ ▷ User priority
        if $P_u > P_{max}$ **then**
          reselect the user $\hat{u} = u$
          update $P_{max} = P_u$
    allocate RB b to user $\hat{u}$
end procedure

B SUPPLEMENTARY MATERIAL

Implementation detail. OutRAN is implemented on top of the srsRAN [69]. The IP/TCP header inspection and per-flow state reside at the PDCP layer. We implement the header inspection before the PDCP header compression and we maintain the per-flow state using a hash table that uses five-tuple as a key and sent-bytes so far as a value. We split the `tx_sdu_queue` of the RLC UM base into 4 MLFQ priority queues and modified `rlc_um::write_sdu` to check the assigned MLFQ priority of each ingressed RLC SDU and put it in its corresponding queue.

The Inter-user Flow Scheduler requires the status of the MLFQ (queued size for each priority queue) at the MAC layer scheduling. To pass over the status from the RLC layer to the MAC layer, we add the "priority" attribute to the Buffer Status Report (BSR) which is originally used to notify the RLC SDU buffer status of each UE to the MAC layer in downlink scheduling. Finally, we extend the `sched_dl_users` at the MAC layer to implement Inter-user Flow Scheduler.

QUIC supported webpages. Out of 20 webpages in our evaluation, 9 of them support QUIC, and Table 2 shows the statistics. The maximum size of a single QUIC flow in the dataset is 443 KB (from Instagram), which is still considered a short flow compared to the background flow (avg. 1.92 MB) from websearch workload [13].

Experiments on Colosseum. Figure 19 shows the FCT results of our experiments on the Colosseum under RF scenarios provided by SCOPE [21]. For srsRAN configurations, we use the configuration files provided by Colosseum github [30]. The main difference between Colosseum and our testbed configuration setup is the number of RBs available for srsENB. In Colosseum, the number of RBs is set to 15, and in our evaluation, the number of RBs is set to 100.

5G NSA testbed. To show that OutRAN works on 5G gNodeB, we implement OutRAN design on the latest release of srsRAN 21.10

Page	Page Size (KB)	Total QUIC Flow Size (KB) in Page	Total # of Flows in Page	Total # of QUIC Flows in Page
Facebook	381	206	33	21
Google	540	70	37	23
Google hk	541	70	38	23
YouTube	899	79	26	8
Instagram	1756	736	25	7
Netflix	1902	1	49	1
Reddit	1928	0.2	90	1
Zoom	2816	165	114	3
Sohu	3370	0.5	522	8

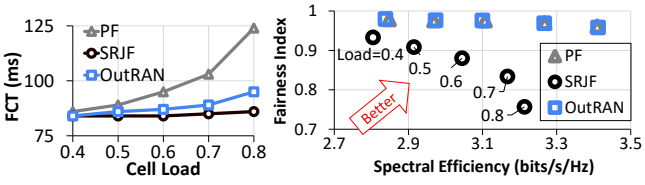
Table 2: Flow statistics for QUIC supported webpages.

that supports 5G Non-Standalone (NSA) gNodeB. However, at the time of this writing, we discovered that the latest release at the time has several limitations supporting fully functional 5G gNodeB, including the constraint of the number of RB (fixed to 50), Band (supports only n3/n78 with 15kHz SCs), and limited support of COTS UEs (only supports specific device) [68]. As an alternative, we test OutRAN on 5G NSA testbed with virtual RF hardware supported in srsRAN, which uses ZeroMQ networking library that transfers radio samples between applications. OutRAN achieves the same performance in DL throughput, CPU, and memory usage and improves the short flow FCT (59% in tail).

5G simulation result. The trend of the result is consistent with our LTE case except that the SRJF performs the best in terms of the FCT. This is because the channel dynamics of a trace provided in NS-3 5G module [32] are more stable and steady than the ones provided in the NS-3 LENA module [61]. In such cases, SRJF performs ideally just like in the datacenter networks.

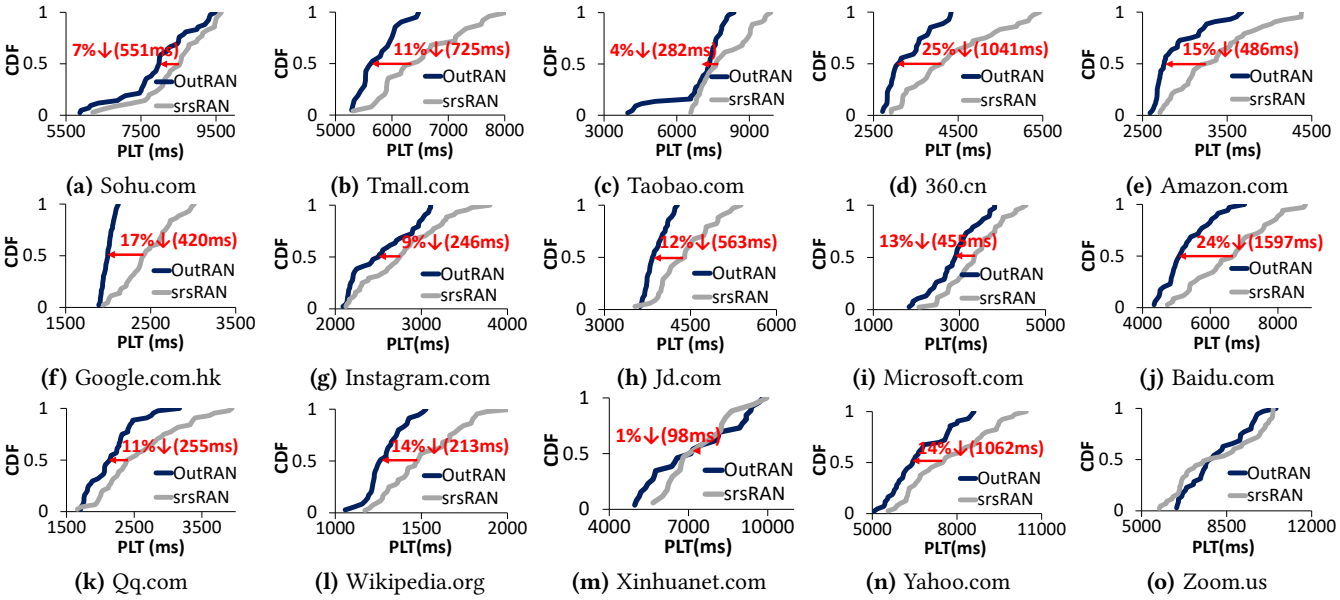
Scenario	Value	Load	Base Station	Overall Avg FCT (ms)	Short FCT (ms)	Short 95%-ile FCT (ms)	Middle FCT (ms)	Long FCT (ms)
Rome ID: 1018 (close, moderate)	ID: 1018 (close, moderate)	0.2	srsRAN	1370	880	4275	1375	6852
			OutRAN	836	242	1288	1093	6980
		0.4	srsRAN	2937	2019	6428	3542	12010
			OutRAN	1959	895	4441	3253	11291
		0.6	srsRAN	3918	2463	6729	5085	17880
			OutRAN	2495	809	3025	4759	16840
Boston ID: 1033 (close, fast)	ID: 1033 (close, fast)	0.2	srsRAN	855	457	1931	916	5196
			OutRAN	822	309	1820	1026	6161
		0.4	srsRAN	2054	1493	4894	2355	7734
			OutRAN	1347	676	2515	1550	8452
		0.6	srsRAN	3184	1913	9803	3874	16041
			OutRAN	2270	881	2951	3793	14783
POWDER ID: 1041 (medium, static)	ID: 1041 (medum, static)	0.2	srsRAN	411	202	1030	399	2786
			OutRAN	319	136	435	378	2245
		0.4	srsRAN	590	318	1145	607	3596
			OutRAN	406	135	683	449	3355
		0.6	srsRAN	1168	729	1889	1335	5747
			OutRAN	500	113	459	695	4446

Figure 19: FCT results of Experiments on Colosseum.



(a) Overall Average **(b)** Spectral efficiency and fairness

Figure 20: [NS-3 5G] [32] FCT across different cell load in 5G under 2019 mobile-app traffic [12].

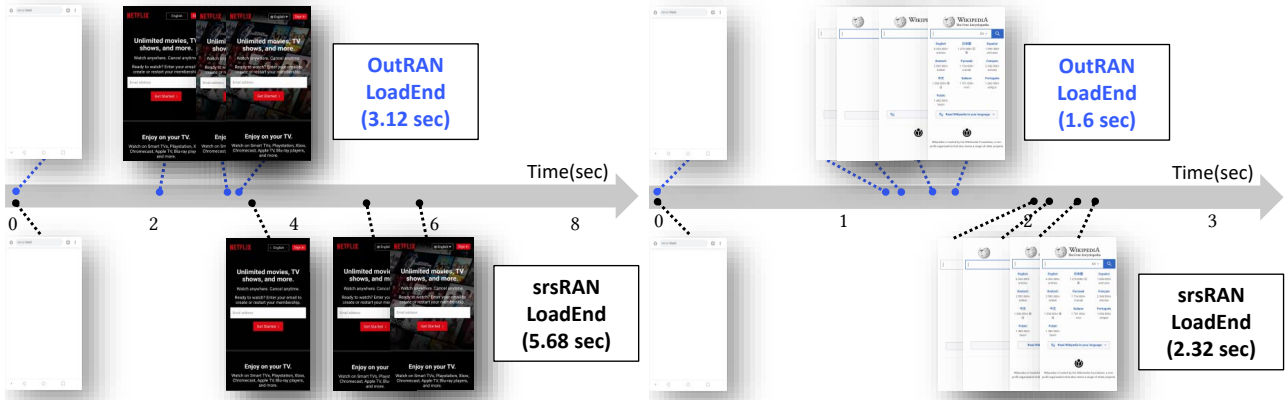


(a) Sohu.com (b) Tmall.com (c) Taobao.com (d) 360.cn (e) Amazon.com

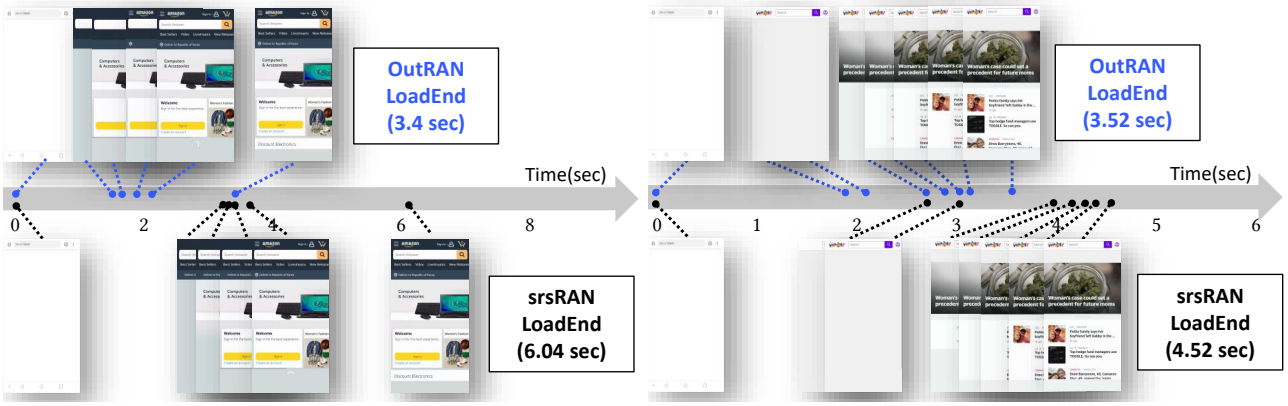
(f) Google.com.hk (g) Instagram.com (h) Jd.com (i) Microsoft.com (j) Baidu.com

(k) Qq.com (l) Wikipedia.org (m) Xinhuanet.com (n) Yahoo.com (o) Zoom.us

Figure 21: [Testbed] PLT across rest of the top 20 Webpages from Alexa.



(a) Netflix.com (b) Wikipedia.org



(c) Amazon.com (d) Yahoo.com

Figure 22: [Testbed] Comparison of Web Page Loading Time between OutRAN vs. srsRAN. The video is available at https://ina.kaist.ac.kr/projects/outran.