Performance Impact of RLC Discards and PDCP Reordering in 5G: A Measurement Study

MohammadBagher Tavassoli
Inria, Université Côte d'Azur
Sophia Antipolis, France
mtavasso@zimbra.inria.fr

Chadi Barakat
Inria, Université Côte d'Azur
Sophia Antipolis, France
chadi.barakat@inria.fr

Thierry Turletti
Inria, Université Côte d'Azur
Sophia Antipolis, France
thierry.turletti@inria.fr

Walid Dabbous
Inria, Université Côte d'Azur
Sophia Antipolis, France
walid.dabbous@inria.fr

***Abstract*—Fifth-generation mobile networks increasingly rely on software-based protocol stacks, where interactions between protocol layers can strongly affect end-to-end performance. In this paper, we experimentally study how packet drops below PDCP, and in particular after PDCP sequence number assignment, can affect downlink performance in a 5G RAN. We show that when packets are discarded in the RLC layer after PDCP sequence numbers have been assigned, the resulting gaps in the sequence numbers can trigger packet reordering timeouts and bursty delivery at the receiver, thereby overflowing limited UE buffers and leading to significant throughput degradation and increased latency, even when the radio link remains stable. Using OpenAirInterface and commercial UEs, we trace this behavior across the RAN protocol stack and show that the degradation is directly related to the PDCP reordering timer. Finally, we evaluate a simple sender-side mitigation that drops packets before PDCP sequence number assignment, preventing sequence gaps and stabilizing downlink performance.**

***Index Terms*—5G Networks; PDCP Reordering; RLC Discards; Performance Evaluation**

I. Introduction

Fifth generation mobile networks support a wide range of services, including video calls, cloud gaming, monitoring and control traffic for industrial and edge systems. These services differ in traffic pattern, latency target, and reliability requirement. Some, such as interactive audio and video, demand low delay and can tolerate limited loss. Others, such as file transfer and web traffic, require steady delivery over longer time scales. Because these services differ in transport and delivery requirements, user plane behavior has a direct impact on application performance.

5G system architecture is also more complex than in previous generations. User plane functions are often implemented in software and may be split across several nodes instead of a single base station. This shift improves flexibility and makes deployment easier in cloud and edge environments. It also makes implementation choices more important across different vendors, as well as in open-source 5G stacks that are widely used for experimentation and analysis. For that reason, understanding how these stacks behave under stress is important to evaluate network performance and the extent to which it meets application requirements.

In the downlink direction, packets arrive at the gNB from the core network and pass through several protocol layers before they are transmitted over the air. SDAP (Service Data Adaptation Protocol) maps QoS flows to DRBs (Data Radio Bearers) [1], [2]. Once packets are mapped onto a DRB, they pass through PDCP (Packet Data Convergence Protocol) and RLC (Radio Link Control). PDCP is responsible for functions such as sequence numbering and security processing, and at the receiver it supports ordered delivery [3]. RLC sits below PDCP and provides buffering and segmentation, and in acknowledged mode it also provides retransmission [4]. If the flow of arriving packets exceeds the service rate available in the RAN, SDUs (Service Data Units) wait in the RLC buffer and may eventually be dropped when the buffer capacity is exceeded. Across this stack, coordination between PDCP and RLC is an important aspect of performance, in particular because ordered delivery at the receiver depends on PDCP sequence numbers. If packets arrive out of order for any reason, such as retransmissions or delays introduced at lower layers, PDCP holds later packets while waiting for missing ones. For this reason, the standard defines a reordering timer so that the receiver does not wait indefinitely for missing sequence numbers. The same specification also notes that discarding PDCP SDUs after sequence assignment may create sequence gaps, and that such gaps increase reordering delay [3]. This can happen when RLC discards packets that have already been numbered by PDCP, causing the PDCP layer in the mobile terminal to wait unnecessarily for packets that will never arrive, hence causing burstiness and impacting throughput and delay.

In this paper, we show that the impact of PDCP reordering extends beyond delay. While prior work treats PDCP reordering and RLC buffering primarily from a latency perspective, we show that, under overload, the resulting sequence gaps also trigger a significant throughput degradation – one that remains invisible to MAC-layer monitoring, where throughput appears unaffected. Our contributions are threefold. First, we characterize this degradation experimentally and trace it through the full cross-layer chain, from post-numbering discard at the gNB to receiver-side reordering and host-buffer overflow. Second, we propose and evaluate pre-SN (Sequence Numbering) drop, a mechanism that prevents these gaps at their source, in the monolithic deployment. Third, we compare this effect across UE architectures – a commercial smartphone and commercial off-the-shelf (COTS) modems – and ground our analysis in OpenAirInterface [5], one of the most widely used open-source

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

5G stacks.

II. System Setup and Problem Statement

We use controlled testbed experiments to assess the impact of uncoordinated packet management between the PDCP and RLC layers in the gNB. We focus on the downlink direction because limited operating system buffers in mobile devices make them highly vulnerable to bursty packet delivery.

We run our experiments in the R2lab [6] testbed, an open wireless experimentation platform designed for reproducible research that provides a stable and controllable radio environment, and part of the SophiaNode, a site of SLICES-RI [7]. We deploy a 5G standalone system with Open5GS as the core network and OpenAirInterface5G as the RAN. The core and the RAN run on two separate machines orchestrated with Kubernetes. On the radio side, a USRP N300 serves as the gNB software-defined radio. We use two kinds of commercial UEs: a Quectel RM500Q-GL modem connected to a Raspberry Pi 4B, and a Google Pixel 7 smartphone. We begin with a monolithic 5G standalone deployment and generate downlink UDP traffic from the core network toward the UE. Keeping all other parameters fixed, we gradually increase the sender rate to induce losses at the gNB. In this setting, a well-behaved system should exhibit graceful saturation: as the offered load increases, the receiver throughput should rise until it approaches the achievable air-interface capacity, and then remain near that level. Excess traffic may be delayed or dropped, but the delivered throughput should not degrade as long as the radio link stays available.

However, we observe a different behavior. As shown in Fig. 1, once the offered load crosses a threshold, the throughput measured at the UE drops sharply instead of saturating. This drop is not visible at the MAC layer, where throughput remains high. We repeated the experiment with both the Quectel modem and the Pixel smartphone and observed the same behavior. This result is unexpected because the radio link remains stable throughout the degradation. The MCS stays unchanged, HARQ activity does not increase, and the SNR remains in the same range as before the drop. The problem therefore does not stem from a simple radio failure, but from user-plane behavior under overload.

We then examined whether the receiver showed any sign of stress. Across repeated runs, the throughput degradation begins together with a rise in UDP receive-buffer errors on the UE host. We also captured packets on the UE host and measured packet arrival intervals. Many intervals were close to 100 ms, near the PDCP reordering timer configured in this gNB.

To confirm that this behavior is driven by PDCP reordering, we conducted an additional experiment with different PDCP reordering timer values. In this experiment, we injected UDP traffic at 150 Mbit/s, which exceeds the achievable air-interface rate of about 86 Mbit/s in our setup and therefore forces packet drops under overload. As shown in Fig. 2, both the measured UDP throughput and the frequency of receive-buffer errors vary directly with the configured timeout. This confirms that

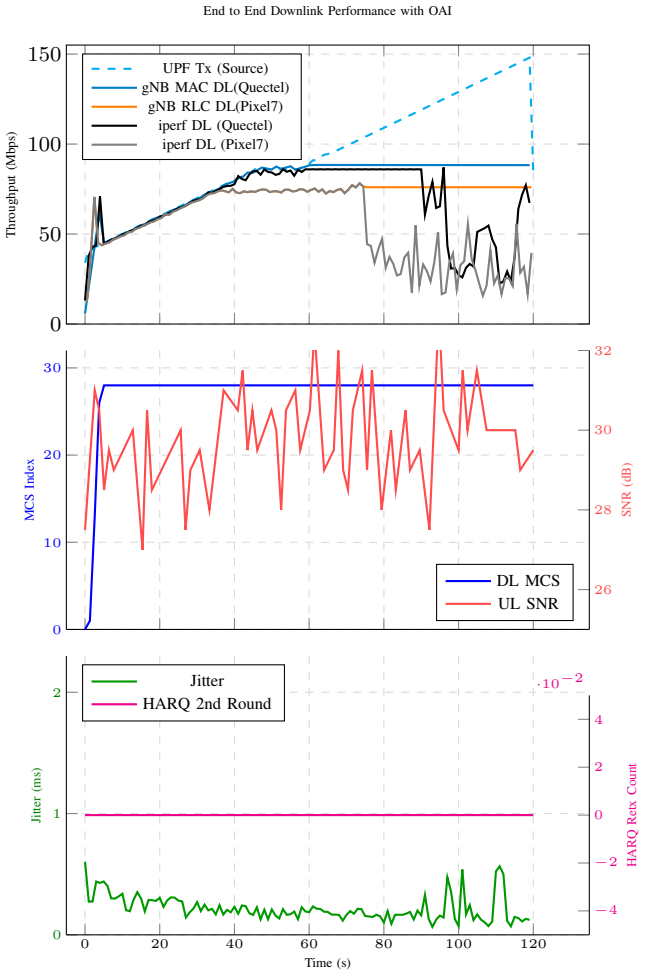


Fig. 1. Downlink overload experiment. Top: source rate and throughput measured at the gNB MAC, gNB RLC, and UE. Middle: DL MCS and UL SNR during the same interval. Bottom: jitter and HARQ second-round retransmissions. All panels share the same time axis.

the throughput degradation is a consequence of the PDCP reordering mechanism reacting to RLC drops.

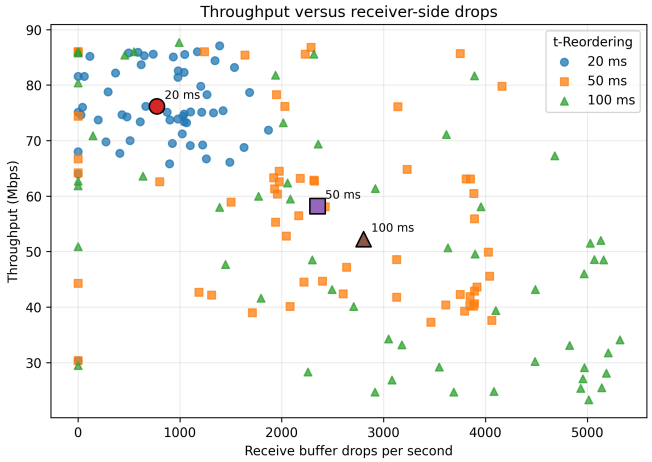


Fig. 2. UDP throughput and receive buffer errors based on different PDCP reordering timer.

The throughput drop can be explained as follows. Under overload, the gNB's RLC transmission buffer eventually fills and drops incoming packets that have already traversed PDCP and received sequence numbers. These RLC discards therefore create sequence-number gaps at the UE. Upon detecting a gap, the UE PDCP layer holds subsequent packets while waiting for the missing data. When the reordering timer expires, the accumulated data is flushed to the host operating system at once. The burst size scales with the downlink rate and waiting time, with $B \approx R \cdot T$, where R is the incoming rate to PDCP and T is the reordering waiting time. In our setup, the Quectel-UE host provides about 212 kB of receive buffering and the Pixel-UE about 313 kB, while an 86 Mbit/s flow held for 100 ms releases roughly 1 MB. This exceeds both receive buffers and causes internal host drops, leading to application-level throughput degradation.

III. Mitigating Throughput Degradation

The measurements above suggest three possible mitigation directions. The first is to increase the receive buffer at the UE host so that it can absorb larger bursts. This improves throughput for modem-based UEs where the host operating system can be tuned, but it is generally impractical for commercial smartphones and does not remove the underlying reordering behavior. Packets are still held by PDCP and released in bursts, so delay and jitter remain high.

The second option is to reduce the PDCP reordering timer. A smaller timer shortens the waiting time before buffered packets are released, reducing the burst size and the probability of receiver-side drops. However, this changes the receiver behavior for all losses and delays on the bearer. If packets are genuinely delayed by lower-layer recovery, a very small timer may expose out-of-order delivery to the transport layer and degrade TCP performance. Therefore, timer reduction is also a partial workaround rather than a general solution.

The third option addresses the root cause. In the observed failure mode, the problem originates when packets are dropped after PDCP sequence number assignment, creating sequence gaps at the receiver. A cleaner solution is therefore to reject packets earlier, before they are assigned a PDCP sequence number. In that case, no gap is created at the receiver, the PDCP reordering timer is not triggered by that packet, and the stall-and-burst pattern disappears.

In a monolithic gNB, PDCP and RLC run in the same node. The sender can therefore inspect the RLC transmit state before accepting a new SDU. We add a pre-SN drop check in the PDCP transmit path: before assigning a PDCP sequence number, PDCP queries the available RLC buffer space B_{avail} for the corresponding bearer. An SDU of size s is dropped if $B_{\text{avail}} < s + H$, where H is a small headroom margin used to account for enqueue delays. Otherwise, the SDU is accepted by PDCP and processed normally.

IV. Results

We evaluate whether pre-SN drop removes the throughput degradation observed under overload. The main experiment uses three Quectel UEs in a monolithic 5G SA deployment. UE1 starts at the beginning of the run, UE2 joins at 30 s, and UE3 joins at 60 s. Each UE receives 70 Mbps of downlink UDP traffic generated with iperf3. This progressively increases the offered load beyond the achievable air-interface capacity.

Fig. 3 shows the aggregate UDP throughput over all UEs. In the baseline, once the second UE joins, the offered load exceeds the radio capacity and the RLC queues start to build up. The arrival of the third UE further increases the pressure on the RLC buffer and pushes the system into the degraded regime. Instead of saturating near the available capacity, the delivered UE throughput becomes unstable and drops.

With pre-SN drop enabled, the system behaves differently. Packets that cannot be admitted by RLC are rejected before PDCP sequence number assignment, so they do not create sequence gaps at the UE. As a result, the PDCP reordering timer is not triggered by these local drops, the bursty delivery regime disappears, and the aggregate throughput remains stable under overload. A small reduction still appears after the third UE joins, because the MAC scheduler shares resources among more active UEs and the third UE has poorer radio conditions. This reduction reflects radio-resource sharing rather than the throughput collapse observed in the baseline.

Figure 4 shows the CCDF of UDP packet inter arrival times at the UE. In the baseline, the tail extends close to 100 ms, matching the PDCP reordering timer and indicating burst delivery. With pre-SN drop, this tail disappears, confirming that the mechanism prevents reordering timeout bursts caused by post-numbering drops.

To simulate realistic TCP congestion, we reduced the default RLC buffer size and used multiple parallel TCP Cubic connections across UEs with different radio conditions. Since TCP Cubic naturally adapts its sending rate to packet loss, the pre-SN drop mechanism mainly improves stability rather than raw throughput. As shown in Table I, pre-SN drop reduces the TCP reset rate by 47.4%, decreases the 95th percentile RTT by 12.1%, and slightly increases the median throughput by 2.59%.

These results confirm the diagnosis from Section II: the severe throughput degradation is not caused by a loss of radio capacity, but by the location of packet drops in the protocol stack. Moving the drop decision before PDCP sequence

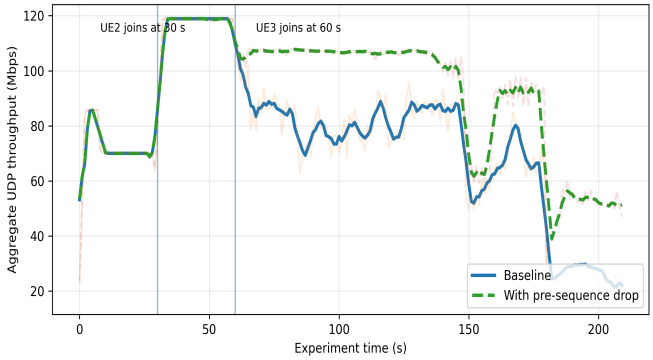


Fig. 3. Aggregate UDP throughput over all UEs: baseline vs. pre-SN drop.

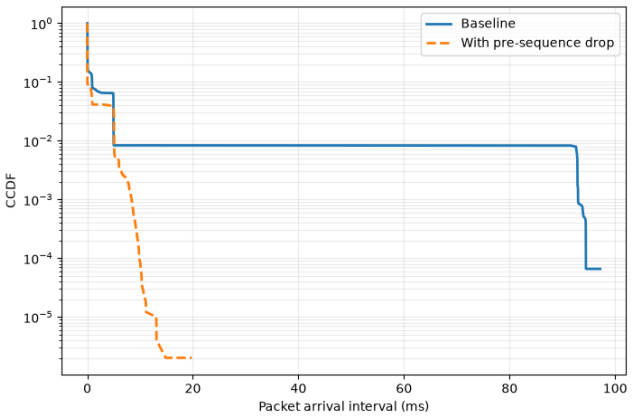


Fig. 4. CCDF of UDP packets arrival intervals.

TABLE I
TCP PERFORMANCE GAINS OF PRE-SEQUENCE DROPPING OVER UES.

Metric	UE1 gain	UE2 gain	Overall gain
Median throughput increase	1.90%	3.29%	2.59%
95th percentile TCP RTT reduction	14.1%	10.1%	12.1%
TCP RTT variation reduction	8.17%	34.1%	21.1%
TCP reset-rate reduction	57.6%	37.1%	47.4%
Out-of-order queue-rate reduction	1.31%	2.27%	1.79%

number assignment prevents sequence gaps at the receiver and stabilizes downlink throughput under overload.

V. Related Work

Prior work has studied buffer and queue management in the RAN and its impact on end-to-end performance. The loss-delay tradeoff in buffer sizing is well established [8], and several works have focused on controlling RLC buffering to reduce delay. Irazabal et al. show that the RLC buffer can become the effective bottleneck before the radio link and propose mechanisms such as e5G-BDP and EQP to bound RLC sojourn time at high utilization [9]. Other schemes dynamically adjust the admitted volume based on sojourn time, occupancy, or estimated capacity, while Kumar et al. size the shared RLC buffer per flow to reduce latency without degrading TCP throughput [10].

AQM has also been studied in softwarized and disaggregated RANs. TC-RAN proposes a programmable traffic-control pipeline after SDAP and before PDCP, using feedback from lower layers to pace or control traffic in split deployments [11]. Other work studies how delayed RLC queue information affects AQM decisions in CU-DU deployments and proposes prediction-based solutions [12]. These works control how much data enters or remains in the RAN queues, but they do not analyze how the position of a discard relative to PDCP sequence number assignment affects receiver-side reordering.

Closer to the protocol behavior, prior studies have shown that PDCP and RLC timers and buffers can strongly affect higher-layer performance [13]. However, they mainly tune timer or buffer values. In contrast, we isolate the drop point itself and show that dropping after PDCP sequence number assignment can turn a local RLC discard into a PDCP sequence gap, reordering stall, bursty delivery, and receiver-side loss.

VI. Conclusion and Future Works

In this paper, we showed that PDCP reordering, usually considered mainly as a source of latency, can also lead to throughput degradation under overload. When packets are dropped after PDCP sequence number assignment, the UE observes missing sequence numbers and holds later packets until the reordering timer expires. The resulting bursty delivery can overflow limited UE receive buffers, causing application-level throughput loss even when MAC-layer throughput and radio conditions remain stable. We validated this behavior experimentally in an open-source 5G RAN using commercial UEs. We also showed that dropping packets before PDCP sequence number assignment prevents sequence gaps at the receiver and stabilizes UDP and TCP performance under overload. These results highlight that the location of packet drops in the RAN protocol stack is an important performance factor, not only the amount of buffering or loss.

As future work, we plan to extend this pre-SN solution to disaggregated CU-DU deployments, where PDCP and RLC run in different nodes and the CU must rely on delayed feedback to estimate the DU buffer state.

VII. Acknowledgment

This work was supported by the ANR under the France 2030 program, grant ”NF-NAI: ANR-22-PEFT-0003”.

References

[1] 3GPP, “TS 37.324: E-UTRA and NR; Service Data Adaptation Protocol (SDAP) specification,” ETSI, 2024, release 18, version 18.0.0.
[2] ——, “TS 38.415: NG-RAN; PDU session user plane protocol,” ETSI, 2025, release 18, version 18.2.0.
[3] ——, “TS 38.323: NR; Packet Data Convergence Protocol (PDCP) specification,” ETSI, 2025, release 18, version 18.5.0.
[4] ——, “TS 38.322: NR; Radio Link Control (RLC) protocol specification,” ETSI, 2024, release 18, version 18.0.0.
[5] OpenAirInterface Software Alliance, “OpenAirInterface 5G Platform,” https://www.openairinterface.org/, accessed: 2026-05-29.
[6] “R2Lab: An open wireless testbed,” https://r2lab.inria.fr, accessed: 2026-09-01.
[7] SLICES-RI, “Scientific Large Scale Infrastructure for Computing/Communication Experimental Studies.” https://www.slices-ri.eu/, accessed: 2026-08-30.
[8] L. Sequeira, J. Fernández-Navajas, L. Casadesus, J. Saldana, I. Quintana, and J. Ruiz-Mas, “The influence of the buffer size in packet loss for competing multimedia and bursty traffic,” 2013.
[9] M. Irazabal, E. López-Aguilera, I. Demirkol, R. Schmidt, and N. Nikaein, “Preventing rlc buffer sojourn delays in 5g,” *IEEE Access*, vol. 9, pp. 39 466–39 488, 2021.
[10] R. Kumar, A. Francini, S. Panwar, and S. Sharma, “Dynamic control of rlc buffer size for latency minimization in mobile ran,” in *Proceedings of IEEE INFOCOM Workshops*, 2018.
[11] M. Irazabal, N. Nikaein *et al.*, “Tc-ran: A programmable traffic control service model for software defined ran,” *IEEE Communications Magazine*, 2023.
[12] A. Stoltidis, K. Choumas, and T. Korakis, “Active queue management in 5g and beyond cellular networks using machine learning,” *Computer Communications*, vol. 236, p. 108108, 2025.
[13] C. K. Sheemar, S. Kumar, J. Querol, and S. Chatzinotas, “Location dependent allocation of timers and buffers at layer-2 in leo based 5g-ntn,” 2023.