pDCell: an End-to-End Transport Protocol for Mobile Edge Computing Architectures

Claudio Fiandrino, Joerg Widmer, Kirill Kogan
IMDEA Networks Institute
firstname.lastname@imdea.org

Antonio de la Oliva
Universidad Carlos III de Madrid
aoliva@it.uc3m.es

ABSTRACT

To deal with increasingly demanding services and the rapid growth in number of devices and traffic, 5G and beyond mobile networks need to provide extreme capacity and peak data rates at very low latencies. Consequently, applications and services need to move closer to the users into so-called edge data centers. At the same time, there is a trend to virtualize core and radio access network functionalities and bring them to edge data centers as well. However, as is known from conventional data centers, legacy transport protocols such as TCP are vastly suboptimal in such a setting.

In this work, we present pDCell, a transport design for mobile edge computing architectures that extends data center transport approaches to the mobile network domain. Specifically, pDCell ensures that data traffic from application servers arrives at virtual radio functions (i.e., C-RAN Central Units) timely to (i) minimize queuing delays and (ii) to maximize cellular network utilization. We show that pDCell significantly improves flow completion times compared to conventional transport protocols like TCP and data center transport solutions, and is thus an essential component for future mobile networks.

CCS CONCEPTS

• **Networks → Network architectures**; **Transport protocols**; *Data center networks*; Mobile networks;

1 INTRODUCTION

The fifth generation (5G) of cellular mobile networks aims at supporting high mobility, massive connectivity, extremely high data rates and ultra low latency [30]. While previous generation systems were optimized for a specific objective (e.g., voice or data), 5G networks need to support several or even all of the above requirements simultaneously. Enhanced mobile broadband (eMBB), massive machine type communications (mMTC) and ultra-reliable low-latency

communications (URLLC) type of services exhibit highly diverse requirements and traffic characteristics. For example, automation processes pose very stringent latency and high reliability requirements to complete actuation commands (< 1 ms latency and 10^{-9} packet loss rate) while communications between sensors employed for smart city services are infrequent and pose relaxed latency requirements [27]. To support such services, the architectural design of mobile networks needs to evolve. In particular, to empower URLLC in systems for 5G and beyond, applications and services need to move closer to the end users.

Mobile Edge Computing (MEC), Software-Defined-Networking (SDN) and Network-Function-Virtualization (NFV) are the main driving factors to enable 5G services. The MEC paradigm brings computing and storage resources closer to the end users. SDN allows to decouple control plane and user plane and NFV decouples network functions from dedicated hardware for their execution on commercial-off-the-shelf hardware. Future mobile networks will leverage these technologies to bring key core and radio functions near the user, in the so called edge data center. Virtual Central Units v(Cus) implementing parts of the air interface stack (following a C-RAN approach) will directly connect users to MEC applications running in the edge data center such as local Evolved Packet Cores (EPCs), keeping local traffic local, reducing the load of the transport network and reducing the latency experienced by the user.

The adoption of edge data centers also brings the whole transport, from application server to the mobile terminal, under the control of the Mobile Network Operator, which allows for performance optimizations. Indeed, conventional transport protocols applied to such an ecosystem would perform poorly. For example, cell load increase is known to limit user bandwidth availability [24] and the increased delay, due to large per-user queues at base stations, can reduce the precision of the TCP retransmission timeout estimation. Consequently, conventional TCP may experience unnecessary timeouts, causing retransmissions and slow start, and thus leading to poor link utilization. Transport protocols for MEC architectures, bridging data center and radio networks together, should take advantage of Cus pooling and information about feedback on channel quality that is available in MEC environments as part of the Radio Network Information Services [23]. Furthermore, they should also take advantage from technological advances in data center transport aiming at minimizing flow completion time (FCT) of small flows [6, 17, 22, 26]. Cross-domain transport optimization allows to fully benefit from the potential gains, e.g., better cell load management and enables just-in-time scheduling to ensure that MEC servers send traffic such that it arrives at the Cus exactly at the right point in time to be scheduled.

This work presents pDCell, a new transport design spanning from the data center domain to the mobile end users. pDCell takes
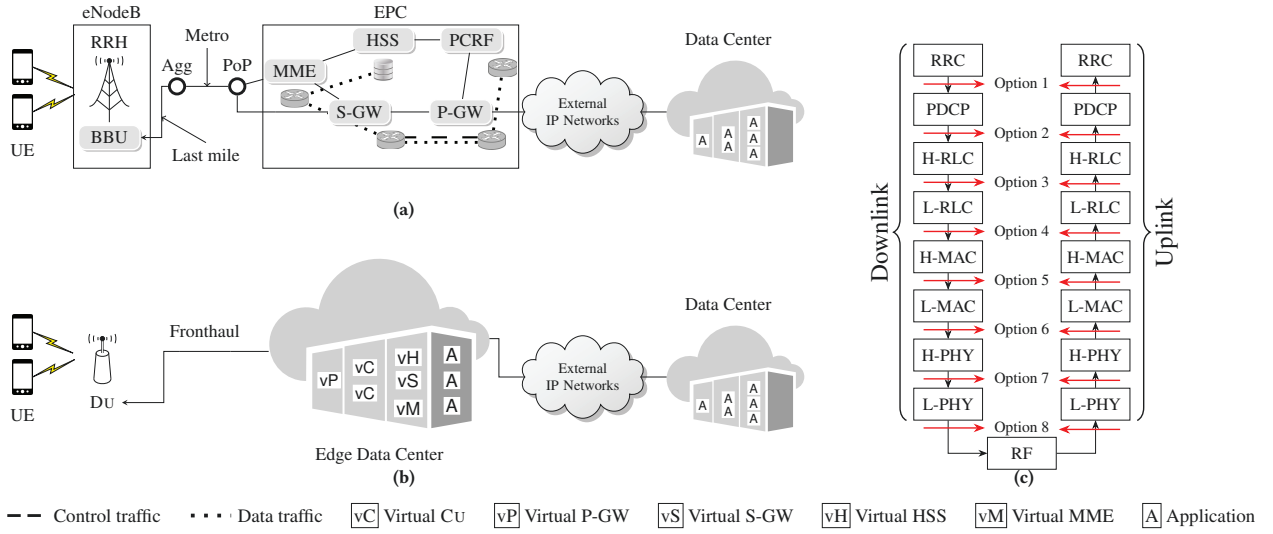
Figure 1: Architectures of a mobile network operator: (a) conventional and (b) MEC realization, and (c) functional split options

inspiration from recent advances in data center transport designs and extends the data center transport to seamlessly incorporate the requirements of the mobile domain. The main novelty of pDCell is to couple the transport protocol with the scheduler of the air interface, residing within the Cu at the edge data center. pDCell vastly improves transport efficiency and can be deployed in a way that is transparent to services and applications by exposing a TCP-compliant socket application interface at the server and mobile terminal side. By coupling the transport congestion control with the wireless scheduler, pDCell can immediately slow down a source sending traffic to a user when radio channel conditions worsen, rather than waiting for slow TCP end-to-end congestion control to adapt. Therefore, buffer sizes in the transport network can be substantially smaller, reducing overall latency.

Our main findings are as follows:

- pDCell significantly outperforms data center transport solutions that are not aware of the wireless domain.
- pDCell is scalable, as the increase in network load does not significantly affect the average FCT.
- pDCell operates with minimal queue occupancy and prevents unnecessary retransmissions to achieve ultra-low end-to-end latency.

2 MEC ARCHITECTURE AND MOTIVATING EXAMPLE

2.1 MEC Architecture

2.1.1 Centralized/Cloud-RAN. C-RAN systems split the air interface protocol stack into two separate parts, one that remains at the antenna site called Distributed Unit (Du) and a Centralized Unit (Cu) which is moved to centralized pools in nearby edge data centers. For the operators, C-RAN systems enable significant cost reduction to deploy and maintain antenna sites while boosting the network capacity, e.g., by enabling Cooperative Multi Point techniques with tight synchronization between the aggregated cells. The complexity of distributed and central units differs based on

the functions that are processed locally. The different points in the protocol stack where the separation of central and distributed functions can take place are called *functional splits* (see Fig. 1(c)) [8, 12]. The implementation of a given functional split uniquely defines the properties of the system design [2], and of the network connecting the Du and Cu, known as fronthaul (Fʜ). In the full C-RAN scenario, radio I/Q samples are transported in the fronthaul and only circuit-based fiber technology can support the massive capacity required (approximately 1.229 Gbps of CBR traffic per 20 MHz LTE channel, independent from the user traffic) and very low latency requirements (approximately 100 μs one way) [14]. With less demanding splits, such as Packet Data Convergence Protocol/Radio Link Control Protocol (PDCP/RLC) (see Fig. 1(c)), high throughput packet-based fronthaul technologies, such as millimeter-wave or Ethernet, can be used with the caveat that packetization delays need to be taken into account [11]. There is a trade-off between the benefits and drawbacks of using the different functional splits. The nearer to the physical layer the split, the more complex are the signal processing techniques that have to be performed in the Cu, and the higher the potential performance gains and cost savings. At the same time, such lower split options impose very stringent delay and rate requirements on the transport between the Du and Cu. Higher layer functional splits like PDCP/RLC provide some of the benefits of the lower layer splits, such as separation between control and user plane and the possibility of using some joint transmission features in order to boost the performance at a much lower transport cost. For the PDCP/RLC split, the transport requirements just include an overhead of around a 20% over the user traffic and easily achievable delays between Du and Cu in the order of ms.

2.1.2 Evolved Packet Core (EPC). The complete set of core functions specified by the 3GPP are called the Evolved Packet Core, which jointly with an EUTRAN (LTE RAN) forms the complete EPS (Evolved Packet System) or cellular network. The EPC consists of multiple components (see Fig. 1(a)), including Packet Data Network Gateway (P-GW), Serving Gateway (S-GW), Mobility Management Entity (MME) and Policy Charging Rule Functions (PCRF). Each
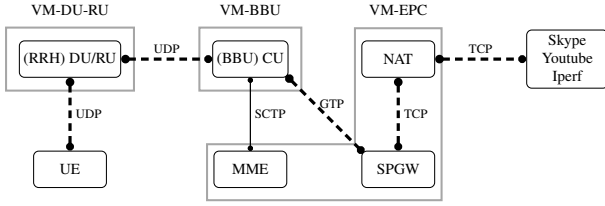
Figure 2: Setup OpenEPC

Table 1: Traffic Analysis on OpenEPC for higher (Hʟs - Option 2) and lower layer splits (Lʟs - Option 6)

Aᴘᴘʟɪᴄᴀᴛɪᴏɴ	Hʟs		Lʟs	
	Dᴀᴛᴀ	Cᴏɴᴛʀᴏʟ	Dᴀᴛᴀ	Cᴏɴᴛʀᴏʟ
Iperf	1720922	8	4349872	8
Skype Video Call	441532	6	331482	8
Skype Text	1872	8	2362	8
Youtube	122572	8	3192	8

component performs specialized functions that are unique to mobile networks and are typically implemented on proprietary hardware. Control and data plane are tightly coupled especially in P-GW and S-GW through a number of dedicated interfaces and protocols. In LTE systems, the GPRS Tunneling Protocol (GTP) is employed to carry data traffic to the end-users. With GTP, end-users maintain the same IP address while moving and traffic is routed between the P-GW and the eNodeB through the S-GW. The Stream Control Transmission Protocol (SCTP) is used for control traffic within the EPC and between the EPC and eNodeB. The recently specified 5G Core redesigns the EPC towards a more de-centralized architecture by instantiating virtual cores in edge data centers through MEC and NFV. In MEC architectures (see Fig. 1(b)), the aforementioned functionalities are virtualized and executed in edge data centers along with pools of Cᴜs performing the radio processing. Although in virtualized environments LTE control signalling can lead to significant overhead [28], in this work we focus on data delivery of small-size traffic typical in data center and mobile networks [9, 32] where LTE events are executed rarely. To understand traffic characteristics of MEC platforms, we experimentally perform traffic analysis on the OpenEPC platform by Core Network Dynamics[1]. OpenEPC implements a licensed version of the LTE EPC and it is configured as shown in Fig. 2. Several applications were selected for the experiments: bulk transfer through Iperf, Skype call, Skype chat and Youtube video streaming. All experiments run for 60 seconds with different split options, i.e., option 2 and 6 that implement a higher and lower layer functional split. Table 1 shows the traffic, in number of packets, in the mobile network after filtering out ARP, NTP, and DNS traffic. The amount of SCTP control traffic is always extremely low compared to data traffic (carried with TCP Cubic) and is due to HEARTBEAT messages between MME and Cᴜ and the corresponding acknowledgments. The employed functional split does not affect the amount of control traffic. For this reason, in our evaluation we do not take into account mobile network control traffic.

---

[1]Accessible from: https://www.openepc.com/

2.2 Motivating Example

Our goal is to design a transport protocol for edge data centers hosting C-RAN virtualized central units (Cᴜs), virtualized EPCs and MEC applications. To minimize end-to-end latency, a transport protocol is required such that packets generated by application servers avoid long queuing delays and arrive at the Cᴜ at the most appropriate point in time for scheduling. The challenge is to achieve simultaneously Gbps of user data rates while supporting low latencies. In mobile networks, the MAC layer (see Fig. 1(c)) schedules packets by allocating radio resources in the time and frequency domains. Resources are allocated based on the estimated channel quality of the users. This feedback, known as Channel Quality Indicator (CQI) reporting, drives the selection of the user bitrate at the Cᴜ. The CQI report has direct relation with the modulation and coding scheme and in turn the transport block size used to allocate physical resource blocks for data transmission to the mobile user. By allowing the Cᴜs to exploit such a feedback and couple it to the edge data center, it becomes possible for the data center transport to perform flow scheduling and to adjust the source sending rate to the amount of radio resources that will be allocated by the mobile network, achieving at the same time high data rates and low latency.

To verify the need for this new transport protocol, Fig. 3 presents the FCT performance for *i)* TCP New Reno, *ii)* a state-of-the-art transport for data center networks (DCNs), pHost [17], with infinite queue size and *iii)* pHost with limited queue size at the Cᴜ. The objective is to compare TCP and unmodified existing data center solutions applied to a MEC edge data center. While TCP runs end-to-end, pHost is terminated at the Cᴜ and from there to the UE, either UDP or TCP can be employed according to the split option (see § 2.1.1). pHost transport empowers end hosts to perform flow scheduling on the basis of grant assignments. Destinations can choose which source is entitled to transmit data packets by sending tokens. Sources decide to which destination to reply when receiving multiple tokens. pHost exploits packet-spraying to eliminate congestion by leveraging the property of DCN of full-bisection-bandwidth and avoiding explicit path-scheduling. The traffic flows from applications towards the EPC located in the MEC platform, which in turn forwards the traffic to the end user through the Cᴜ. For the cases of TCP and pHost with limited queue size, a per-user buffer size of a LTE (Long Term Evolution) UE category 3 is assumed [1]. The traffic trace used for this experiment is explained in § 4 and its flow size distribution is shown in Fig. 7. The FCT of pHost with finite queue shows moderate delays for all flows. pHost tries to fill the buffer of the Cᴜ as soon as possible and since it does not have any feedback about the wireless domain, packets are discarded once the buffer is full. pHost will respond to the loss of packets with retransmissions by reissuing grants. Considering the timeouts of pHost designed for DCN, this retransmission is extremely fast. The FCT of pHost with infinite queue shows the higher delays. When the buffer of the Cᴜ is assumed to be infinite, no packets are lost due to buffer overflow. For this reason, packets experience high queuing delays. Finally, TCP FCT shows an intermediate performance. Although the congestion control of TCP considers the delay and bandwidth of the channel, its reaction time
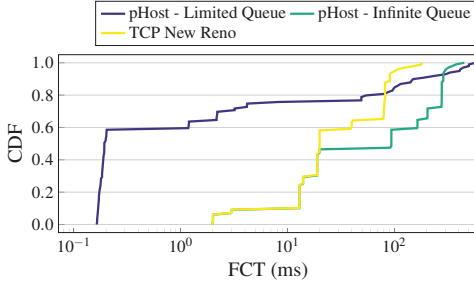
Figure 3: Performance (in term of FCT) of legacy transport protocols in the edge data center scenario

can be on the order of seconds as packets traverse and wait in long queues.

Given the above results, we argue that a new transport needs to be designed specifically for mobile edge data centers. We base this design on data center transport rationale, but parameterize and adapt it to the requirements of the mobile network. The new transport must incorporate specific aspects of the wireless domain and be flexible to adapt to latency and bandwidth variation, while maintaining the benefits of the DCN transport such as low queue occupancy, low FCT and low losses.

3 THE PDCELL TRANSPORT

This section discusses the design rationale of pDCell transport, illustrating how it integrates the data center and wireless domains.

3.1 Assumptions and Design Challenges

We consider an end-to-end architecture where each Du is controlled by one Cu and each mobile user is attached to a single Du at a time. We will discuss in § 5 the modification to the architecture layout necessary to relax such assumption. pDCell has to overcome a number of challenges:

(1) The RTTs of the two components of end-to-end system (mobile and data center networks) are different. While future 5G mobile networks should support millisecond RTTs, current RTTs are in the order of tens of milliseconds to seconds [32]. In contrast, the data center delay is in the order of microseconds [13]. Hence, pDCell should take into account this difference and prevent bursts of packets to be queued simultaneously at Cus that will increase latency. To this end, pDCell operates on flow scheduling and source sending rate adaptation in the data center to seamlessly incorporate the requirements of the mobile network.

(2) Multiple data center sources can send traffic to the same mobile user simultaneously. The size of such flows can vary and they have to be processed by the Cu hosting the baseband processing of the Du associated with the mobile user. Hence, these flows share the same buffer and compete with flows destined to other users. To minimize FCT, short flows should be prioritized over long ones. Moreover, while the LTE standard requires packet-level guarantees, data center networks only enforce flow level guarantees [28]. To this end, Cus schedule the processing order of incoming data center traffic with a newly defined scheduling policy, explained in § 3.2 and § 3.6.

(3) pDCell should ensure a clear separation between control and data traffic. In edge data centers, both the mobile network control traffic attributed to message exchange between the diverse functionalities of the EPC and the data center control traffic coexist.

(4) While in data centers packet losses are infrequent, in the wireless domain they are common, and hence pDCell should react accordingly depending on the nature of losses. Whereas in the wireless domain pDCell relies on existing mechanisms, in the data center environment pDCell operates on the buffer management level by defining an admission policy that prevents packets from being dropped and later retransmitted.

3.2 Architectural Design Overview

In this work, we consider a MEC architecture as in Fig. 1(b), implementing a functional split according to option 2 (see Fig. 1(c)). This split limits the multiplexing gains as it leaves a considerable part of the baseband processing at Dus. However, it poses much less stringent fronthaul requirements in terms of latency and rate [12, 18], allowing edge data centers and Dus to be interconnected with packet-based technology. To exploit available radio resources, each Du performs resource allocation based on CQI user feedback and takes care of retransmissions if needed (at the RLC and MAC layers). Note that implementing split option 8 would require CQI feedback to be propagated to the Cu, and thus the scheduling is based on less up-to-date channel state information. This is the second advantage of implementing split option 2. However, to solve challenge (1) and perform sophisticated scheduling taking into account simultaneously data center and mobile networks states, also the Cu needs to be aware of the user channel conditions, which the split option 2 does not provide. Thus, the challenge to solve is: how to propagate such information back to the data center? To resolve this challenge, we resort to buffer management and propose a new mechanism tailored to the requirements of the mobile network.

3.3 Buffer Management at Cus and Dus

At the Cu, an IP packet undergoes the PDCP protocol with associated robust header compression (see bottom of Fig. 4). There exists a direct mapping between a PDCP service data unit (SDU) and a protocol data unit (PDU), that are of a maximum size of 8188 Bytes, to also handle MTUs larger than those of typical IP packets. PDCP PDUs are then sent to the RLC layer at Dus. To better manage wireless resources, the association between RLC SDUs and PDUs is not univocal. The payload of RLC PDUs can contain multiple SDUs, e.g., combining new incoming packets from PDCP and retransmissions. A RLC SDU can also be split into multiple PDUs. At this stage, the information on original data center flows is no longer available, precluding the capability of performing scheduling with joint information on data center and mobile network and thus to meet objectives like FCT minimization. Hence, pDCell advocates the need for combined scheduling decisions at Cus by leveraging both flow and channel quality information. The total per-user buffer space allocated for RLC in acknowledge mode is given in number of SDUs, typically 1024 for a UE cat 3, i.e., 1.4 MB to accommodate 1500 Bytes long RLC SDUs. When TCP and URLLC traffic coexists,
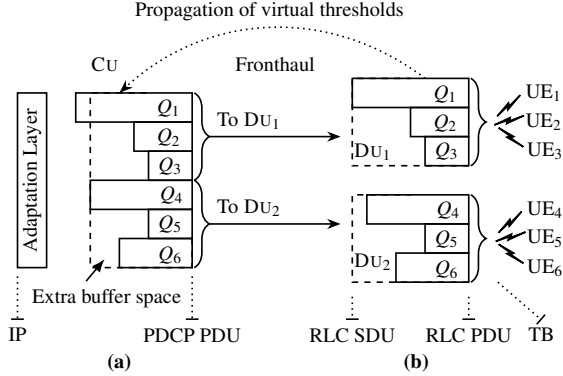
Figure 4: Buffering architecture of the wireless domain: (a) each Cu is a shared memory switch with a number of output ports equal to the number of associated Dus; (b) each Du is a shared memory switch with a number of output ports equals to the number of associated users.

the queuing delay can grow up to 540 ms, which is highly detrimental to the performance of interactive applications [21]. To prevent such behaviour, we advocate the a better used of buffer space and we implement the following mechanism for pDCell.

At the Cu level, one queue per user is necessary to control application sources in the data center. The allocation is performed by the Radio Resource Control (RRC) protocol when it detects the status of an user as active. The queue size is defined via a virtual threshold that changes based on: i) the feedback from the corresponding Du, ii) the current load at Cu level. Since the sum of the virtual thresholds of per-user queues could exceed the processing capacity of the Cu, the buffer space is shared among all per-user queues and the total per-user queue size is adjusted to the processing capacity of the Cu. The number of output ports in this shared memory switch is equal to the number of associated Dus, where outgoing traffic of each port is shaped to the Du processing capacity (see Fig. 4(a)). If for the same user the propagated value from a Du virtual threshold is smaller than the one currently available in the Cu, the latter is decreased. Otherwise, the value of the virtual threshold increases if the current buffer occupancy allows. Similarly to prior work [21], the shared memory switch operates at PDCP level, although for scalability reasons per-user and not per-flow queues are allocated.

The natural choice for buffer management in a shared memory switch is a Longest-Queue-Drop (LQD) [4] policy that drops already admitted packets from the longest queue in case of congestion. By design, LQD provides some level of fairness among users to access Cu processing capacity. Since the goal of the transport is to ensure reliability and minimize retransmissions, already admitted packets should not be dropped. Hence, the LQD behavior needs to be modified. When the Cu buffer is full, the virtual threshold of the currently longest queue Q is decreased by one unit without dropping packets in Q and another incoming packet currently evaluated for admission is simply not admitted. Hence, we guarantee that each admitted packet has buffer space reserved. Furthermore, incoming packets belonging to a user whose queue occupancy is equal to its virtual threshold are also not admitted, regardless of the buffer occupancy. In § 3.7 we will show how to handle retransmission of non-admitted packets to limit non-in-order arrivals and consequent
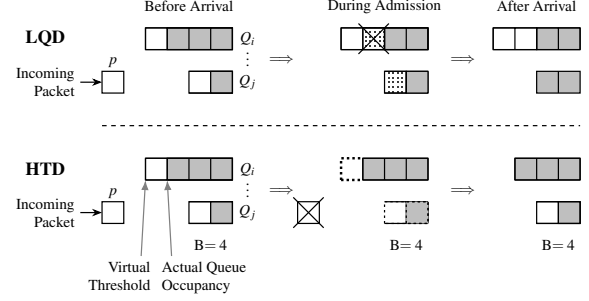

Figure 5: Admission of an incoming packet at Cu. Difference between LQD and the proposed HTD before, during and after admission of a single packet. The size of the Cu buffer B is 4.

time-consuming reordering. We call this buffer management policy Highest-Threshold-Decrease (HTD).

Fig. 5 highlights the difference between the LQD and the proposed HTD admission controls. Before the new packet p arrival, the virtual thresholds of the first and the second queues (Q_i and Q_j) are 4 and 2, respectively. The actual queue occupancy of Q_i and Q_j are 3 and 1, respectively. Since the size of the buffer is 4, newly arriving packet p to Q_j causes congestion in the Cu buffer. LQD drops one already admitted packet from Q_i as it is the longest queue, and admits p to Q_j. The virtual thresholds for both queues remain the same. In contrast, HTD never drops an already admitted packet, hence, HTD reduces by one unit the virtual threshold of Q_i and prevents p from being admitted. Note that HTD reacts more slowly than LQD, but satisfies the requirement that admitted packets are never dropped (challenge (4)).

Since each mobile user is controlled by a single Du, each Du maintains a single queue per user (see Fig. 4(b)). In stationary regime, the queue status at $t + 1$ is given as follow:

$$Q(t + 1) = Q(t) + A(t) + R(t) − D(t), \qquad (1)$$

where $Q(t)$ is the queue status at time t, $A(t)$ denotes incoming packets from Cu accepted by the policy HTD, $R(t)$ denotes PDUs to be retransmitted and $D(t)$ corresponds to successfully acknowledged PDUs from the mobile user that can be removed from the buffer. Users experiencing good channel quality report a high CQI index, which in turn allows the MAC scheduler to employ higher modulation and coding scheme and transport block size. Hence, the component $D(t)$ drains the queue fast and the component $R(t)$ is marginal. In contrast, for users with a low channel quality, the MAC scheduler uses a more robust modulation and coding scheme which reduces the number of bits per resource block. Consequently, the components $D(t)$ and $R(t)$ may cause Q to grow, limiting the space for new incoming packets. Then Dus propagate $A(t)$ to Cu (using uplink channels) and the corresponding virtual threshold increases if there is room. The propagation of $A(t)$ makes the Cus *indirectly* aware of the channel conditions that split option 2 precludes in order to limit/increase the sending rate of data center sources with the adaptation layer (see § 3.5). Note that the size of Q cannot grow indefinitely, i.e., the sum of all allocated per user queues cannot exceed the overall Cu buffer space.

Because of the propagation delay between Cu and associated Dus, the values of per-user virtual thresholds in the Cu may be

outdated, and the Cu may transmit more traffic than the associated Dus can accept. To accommodate this excess traffic, one solution would be to add extra buffer space, which is however detrimental to interactive applications. Therefore, to avoid allocating this extra buffer for every queue, all virtual per-user queues in Du share the same physical buffer space.

3.4 pDCell Design

The transport design of the mobile network and the data center are transparent to each other. The latter, however, requires to incorporate mechanisms to reflect specific properties of the wireless environment. In the context of edge data centers, maintaining the data center transport simple to avoid delays introduced by scheduling is imperative as timeouts and retransmissions negatively impact the overall FCT. Consequently, for pDCell, we rely on transports where flow scheduling, congestion control and reliability are performed by the end hosts, following a minimalistic approach with requests-to-transmit (RTS) and credit assignment. Both are exchanged by end data center servers through credit packets as detailed in § 3.5. The design of pDCell is inspired by previous credit-based transports pHost [17] and ExpressPass [13], and shares the fast start approach common in other recent works [20], i.e., to ensure low-latency, pDCell sends data packets without completing an handshake procedure by probing for available bandwidth. Specifically, pDCell allows a flow to start by sending data packets along with a requests-to-transmit packet.

In contrast to protocols like pHost and ExpressPass that are specific to data center applications, pDCell is adapted to the unique features of a mobile network as follows. Conventional data center control traffic (credit packets) is exchanged between end hosts and the arrival order of credits at the bottleneck link schedules the subsequent arrival of data packets, which helps to avoid the incast problem. pDCell inherits this property as well, as shown in § 4.1. In MEC architectures, the data traffic needs to traverse Epc hosts before being processed at Cus (challenge (3)). pDCell supports this feature as illustrated in § 3.5.

3.5 The Design of the Adaptation Layer

pHost performs transmission control through grant exchange. Each credit packet informs the sender of the next packet (on a packet by packet basis) to be sent for each flow. Hence, pHost effectively performs scheduling at a per-packet granularity. pDCell incorporates this design. Similar to pHost and PIAS [7], pDCell makes use of the limited number of priority queues in commodity switches (4-8 per port [7]) to prioritize credit over data packets. Fig. 6 shows the workflow of pDCell for two sources and two destinations and highlights the traffic exchange between all segments of the considered architecture. A data center source sends an requests-to-transmit packet to inform the adaptation layer at Cu about the estimated flow size along with a small number of data packets. Although some works criticize the availability of precise information on flow sizes [7], it can be estimated [5]. The adaptation layer processes the requests-to-transmit request and admits arriving data packets into the corresponding per-user queue. Implicitly, this will preallocate buffer space in the Cu buffer and will allow to allocate credits on the basis of the difference between the current queue occupancy and
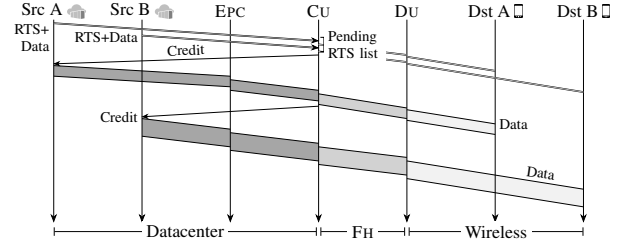

Figure 6: Example of pDCell workflow

the virtual threshold. Such difference defines the current capability of the mobile user to receive data according to the channel conditions. Obviously, the current queue occupancy cannot be larger than the virtual threshold and when the two values are equal, no more credits are generated. Note that this is a conservative approach as more credits could be generated because of the queue turnaround time. Each credit has two associated timers t_s and t_e: the first prevents sources to transmit before the timer expiration and the second prevents sources to transmit data packets after its expiration. If credits expire, i.e., the source does not use them in due time, they are simply re-sent by the destination. By default, we set $t_s = 0$ and $t_e = t_s + 1.5\times$ the MTU-sized packet transmission time. Only Cus can update t_s, to ensure that only mobile traffic benefits from the adaptation required to solve challenge (1). Congestion is detected by Cus when no data packets are received as a reply to a credit packet. Three are multiple possible reasons: i) data packets have not been admitted at the Cu, ii) data packets have been discarded in the network fabric, or iii) the sender has not utilized the token given by credit packets before the expiration of t_e. Consequently, pDCell adjusts t_s as $t_s = \text{TTI} + \alpha$, where TTI is the Transmission Time Interval and α is the time necessary to transmit $A(t)$ packets at the lowest modulation and coding scheme. In this way pDCell prevents the data center source from sending traffic to the mobile user and this backoff period is set to be sufficient to absorb the already queued traffic.

3.6 Wireless Scheduling Algorithms

Although Dus are in charge of *actual* packet scheduling and mapping to transport blocks, to pursue specific objectives like FCT minimization the Cus need to select the processing order of incoming flows. A FIFO approach would naturally process packets according to the arrival time, which can be detrimental for the FCT of small flows. Hence, each t_a interval equal to a frame duration of 1 TTI, the Cus read packets from the buffer B and construct virtual LTE frames to emulate the actual schedule at Dus. Specifically, to each per-user queue (see § 3.3) is assigned a weight and the scheduler visits queues with high-priority first. This weight is defined as the time it takes to transmit the current flow with the current modulation and coding scheme of the mobile user. The higher the time, the lower the weight.

To assess the impact of the wireless scheduler in the overall framework of pDCell, we propose a new scheduling algorithm and compare its performance against the state-of-the-art:

- Minimum Remaining Flow Size (MRFS) schedules flows according to the remaining time required by the flow to complete. It considers the bandwidth available to the user in

the wireless domain and the estimated number of packets remaining in the flow. This algorithm optimizes the FCT, especially for small flows and users with good channel quality, providing an advantage to interactive applications. The details of the scheduling algorithm are given in Algorithm 1.

- Proportional Fair Scheduler (PFS) preferentially schedules users with good channel quality (while maintaining proportional fairness). PFS is commonly implemented in current mobile networks [10] and does not consider the characteristics of the flows to be scheduled. This algorithm ensures fairness among the users and exploits the wireless medium efficiently, hence we employ it as baseline for comparison with MRFS.

Algorithm 1 Minimum Remaining Flow Size (MRFS)

1: $queue \leftarrow$ array of all per-user queues
2: $F_a \leftarrow$ array of flows at queue ordered by minimum completion time ($size_{flow}/bw_{user}$)
3: $N_Q \leftarrow$ number of per-user queues
4: $sf \leftarrow$ sub-frame
5: **while** ($size(sf) > 0 \,||\, size(F_a) > 0)$) **do**
6:     **for all** f in $\{0, size(F_a)\}$ **do**
7:         $f_{id} = F_a[f].id$
8:         **for all** q in $\{1, N_Q\}$ **do**
9:             **while** $find(queues(q), f_{id})\, != \,NULL$ **do**
10:               $pk = find(queues(q), f_{id})$
11:               **if** $size(sf) >= size(pk)$ **then**
12:                   $Sf = [Sf \leftarrow pk]$
13:                   $F_a[f].remaining_{pk}--$
14:               **end if**
15:             **end while**
16:         **end for**
17:     **end for**
18: **end while**

The choice of the employed scheduling algorithm allows to fulfill specific goals. To understand the rationale behind the MRFS design, let us consider the following example. Two sources in the data center S_1, S_2 send two flows each to two mobile users UE_1 and UE_2 served by the same Cu. Assume $CQI_{UE_1} > CQI_{UE_2}$ and the paths between S_2 and Cu to experience congestion while the paths between S_1 and Cu do not. Consider $\mathcal{F}_{S_1UE_1} < \mathcal{F}_{S_1UE_2}$ and $\mathcal{F}_{S_2UE_1} > \mathcal{F}_{S_2UE_2}$ with $\mathcal{F}_{S_1UE_1}$ and $\mathcal{F}_{S_2UE_2}$ to be of equal size. Then, to minimize overall FCT, the transport should schedule in order $\mathcal{F}_{S_1UE_1}$, $\mathcal{F}_{S_2UE_2}$, $\mathcal{F}_{S_1UE_2}$ and $\mathcal{F}_{S_2UE_1}$.

Given the preference that MRFS gives in scheduling first short flows, long ones can be starved in the short term. By penalizing fairness and giving high priority to short flows, we ensure to meet latencies required for URLLC type of traffic. Note that in the long term, the long flows get *actually* scheduled because, as their remaining flow size becomes smaller, in turn their scheduling priority increases.

3.7 Handling Packet Losses

Although the credit-based scheme is designed to minimize packet losses and has been proven to ensure almost zero losses in data

centers, in an end-to-end architecture this is not true anymore. In mobile networks, packet losses are more frequent because of the inherent characteristics of the wireless channel. Consequently, pDCell considers the different nature of packet losses and provides different reaction mechanisms.

In the data center, HDT (see § 3.2) ensures reliability by not dropping already admitted packets. When data packets are not admitted at the Cu or lost, credit packets, which include the ID of the next data packet to be sent, are issued again. This also limits out-of-order arrivals and the consequent reordering that is detrimental for interactive applications.

In the wireless domain, pDCell relies on existing reliability mechanisms implemented in mobile networks, namely the Automatic Repeat Request (ARQ) and the Hybrid ARQ (HARQ) at the RLC and MAC layer respectively. Hence, the mobile users do no propagate explicit per-packet feedback to the Cus, which would require either modifications to the existing protocols through explicit signaling over uplink control channels or adopting a technology- and split-dependent eCPRI solution that estimates HARQ timeouts. Furthermore, because of the combined effect of HDT and MRFS, mobile users with bad channel quality signal implicitly this status to data center sources, which in turn slow down and reduce the sending rate to prevent that significant amount of traffic to incur in high loss probability.

4 EVALUATION

This section evaluates pDCell for a MEC setting composed of a edge data center and wireless domain by means of simulations. Our evaluation setup builds on the one used for previous works [6, 17] by extending it to incorporate the mobile network component.

4.1 Evaluation Methodology

Network Topology: The network topology spans from the DCN to the cellular network. The latter consists of LTE cells configured with a 20 MHz channel and 4×4 MIMO, providing up to 100 Mbps of user-plane data. A set of 100 LTE category 3 UEs are randomly located in the area of coverage. The channel model and the UE link quality measurement report follow [15]. The functional split is according to the 3GPP functional split 2. The fronthaul interface is modeled as a serial line of 10 Gbps. For CQI reports, we employ traces reported in [16] and assume a reporting interval of 5 ms, which is the typical value used in LTE to carry the CQI feedback over the Physical Uplink Shared Channel under fading conditions [3].The CQI feedback is used by the Du to determine the modulation and coding scheme for data transmissions to the end user.

As in prior work [6, 17], the DCN is a two-tier leaf-spine topology comprising 144 servers grouped into 9 racks. Unlike large cloud data centers hosting hundreds of thousands of servers, edge data centers for baseband processing are expected to be of a smaller size [31]. Interconnectivity between the computing servers and the 4 core switches is provided by 9 top-of-rack (TOR) switches through a full mesh. As a result, the data center network is a non-oversubscribed topology where each TOR has 16 10 Gbps links towards the servers and 4 40 Gbps towards the core switches. Both core and TOR switches implement packet spraying and are
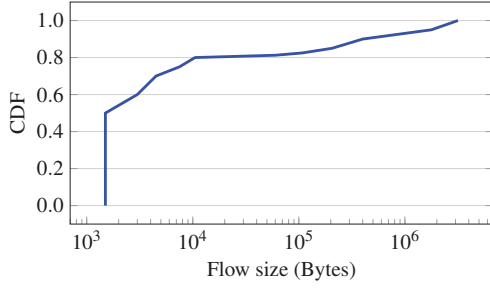
C. Fiandrino, A. de la Oliva, J. Widmer, and K. Kogan

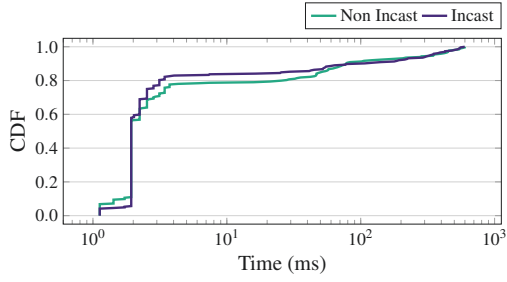Figure 7: Distribution of flow sizes of data center traffic to the Cᴜ

Figure 8: FCT with different traffic matrices

equipped with tiny per-port buffers of 36 KB each. The DCN includes as hosts Cᴜ and distributed EPC to be compliant with § 2.1.1 and the latest technology developments [19].

The wireless scheduler decides which packets are transmitted to the Dᴜ and it works on the time scale of frequency-division duplex (FDD) sub-frames (1 ms).

Workloads: The data center workloads used in the evaluation exploit a flow size distribution based on measurements in a production data center with similar topology as in our study and from a cluster running data mining type jobs [9]. Fig. 7 shows the CDF of the flow size. Most of the flows are small ($\leq$ 10 KB), while a limited number of long flows contain most of the traffic (the largest flow is 3 MB and 1 GB for production data center and cluster respectively). The flow size distribution is consistent with measurements of cellular network traffic [25]. To test the impact of background traffic (i.e., traffic generated by other applications hosted in the data center) on mobile traffic, we perform the following experiment. Similar to previous works [6, 17], the flow arrival follows a Poisson process with a rate that ensures a given target network load set by default to 0.6. As explained in § 3.4, pDCell inherits the incast avoidance of credit exchange based DCN transport protocols, i.e., no throughput degradation is perceived when multiple sources make concurrent requests to send data to the same receiver simultaneously. To evaluate this property, we compare the CDF of the flow completion time under two different traffic matrices. In the *non-incast* one, 20% of the flows are destined to a single Cᴜ server, while the remaining flows are exchanged between random hosts in the data center. In the *incast* traffic matrix, all sources send traffic to the same Cᴜ server. Fig. 8 compares the completion time of the flows. The CDF shows negligible differences for the two traffic matrices. Consequently, in the remainder of the paper, we present results for a single Cᴜ, which can be extrapolated to scenarios where several Cᴜs are located in the same edge data center.
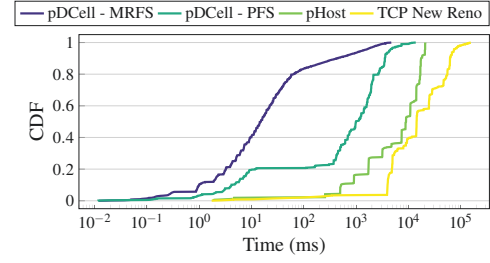

Figure 9: Distribution of FCT for pDCell, pHost and TCP

Performance Metrics: To evaluate pDCell, we adopt FCT as main performance metric, i.e., the time between the first packet being sent and the last one being received of a given flow. In the rest of the paper the FCT is measured end-to-end and not only within the data center. Hence, lower values of the slowdown indicate better system performance. We assess and discuss the effect of buffer size at the Cᴜ and its implication on FCT. To evaluate the impact of the protocol in the wireless domain, we analyze the FCT of flows of users with different channel quality, e.g., users located near the edge of the cell versus users located near the base station.

4.2 Results

The first experiment compares the FCT of pDCell against solutions that are not aware of the wireless domain requirements at the congestion control protocol level. Specifically, Fig. 9 compares performance of pDCell with the different scheduling algorithms illustrated in Section 3.6 against TCP New Reno and legacy pHost in the data center domain, both employing the PFS scheduler. We observe that pDCell, being aware of the wireless network status, outperforms significantly the other solutions regardless the employed scheduling algorithm. Focusing on pDCell performance, the MRFS scheduling algorithm consistently reduces FCT with respect to PFS. In particular, small flows for users with good quality benefit the most from MRFS as they are scheduled immediately in the next frame after arrival at Cᴜ. PFS instead, treats all users and flows the same way, using sub-optimally the bandwidth of the cellular network and increasing the overall FCT for all flows.

The following experiment aims at verifying the effectiveness of channel quality awareness by the congestion control protocol and compares the performance of the wireless schedulers. Fig. 10(a) analyzes the effect of the CQI reporting. For the experiment, we compare the cases where the CQI report remains fixed during the simulation period and when it varies. In the former case, we assign to each user a CQI value uniformly distributed in the range $[0 - 15]$. Compared to PFS, MRFS provides a significant advantage with trace-based CQI reports, as it adjusts the scheduling according to channel quality and prioritizes the schedule of short flows. Hence, when the CQI is fixed, MRFS can only perform decisions taking into account flow size and performs sub-optimally. PFS performance is similar with fixed and trace-based CQI as scheduling decisions do not take into account the flow characteristics. For the next experiment, we focus on small flows, i.e., the < 10 KB flows that represent nearly 80% of the traffic. Fig. 10(b) shows the FCT grouped according to the users' channel quality. Specifically, users with CQI values in the range $1 \leq CQI < 6$ experience a bad channel quality and use QPSK modulation, users with $7 \leq CQI < 10$ use 16QAM, while

(a) Effect of CQI reports on MRFS and PFS

(b) FCT of short flows
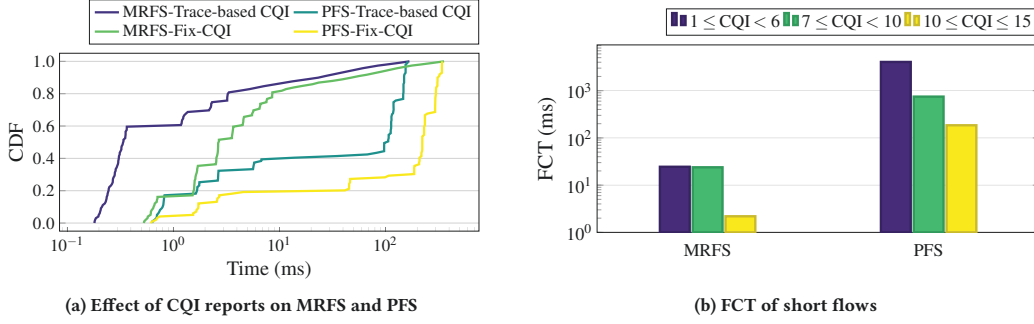
Figure 10: Analysis of FCT for users experiencing different channel quality

users experiencing good channel quality with $10 \leq CQI \leq 15$ use 64QAM. MRFS with its awareness of channel quality statistics and remaining flow size significantly outperforms PFS. In particular, the FCT achieved by users experiencing bad channel quality improves by 2 orders of magnitude compared with PFS. Interestingly, while the dependence between the FCT and user CQI of PFS is linear, in MRFS is not. The reason is that, by design, MRFS always chooses to schedule first flows with minimum remaining size of users in favorable channel conditions.

The next set of experiments measures FCT performance of pDCell while varying the amount of buffer space at the Cu for both schedulers. To achieve ultra-low latency, it is essential that flows encounter small queuing delays along the path to the end user. Fig. 11(a) shows the percentage of non-admitted packets and compares the performance of pDCell and pHost for different buffer sizes. The former, by leveraging information on user-perceived channel quality, slows down application sources by reducing the fraction of non-admitted packets that have to be rescheduled. In contrast, pHost application sources sending rates do not adapt to the user-perceived channel quality and keep pushing traffic to Cus' buffers, who fill up quickly and maintain a high percentage of non-admitted packets. This negatively impacts the FCT. Fig. 11(b) analyzes buffer utilization after having set the maximum Cu buffer size at 3 MB. Note that the Cu buffer size is shared among all the users and flows, hence the 3 MB buffer corresponds to an average per flow buffer of only two packets. pHost reaches quicker than pDCell the full buffer occupancy which it then maintains for the entire simulation period. pDCell after the initial period where the queue fills up, achieves lower full buffer utilization. Hence, it reduces the fraction of non-admitted packets and improves consequently FCT. Fig. 11(c) shows that FCT values remain very similar when the buffer size varies from 3 to 30 MB for both schedulers. These results show that the performance of pDCell remains very similar for a wide range of values and can operate at very low buffer sizes, independently of the choice of scheduler.

5 PERSPECTIVES FOR IMPLEMENTATION IN 5G SYSTEMS

The system enhancements proposed and evaluated so far are applicable to MEC architectures for current LTE mobile networks and next 5G systems with users attached to one Du at a time. Next, we show that minimal modifications need to be performed to the buffering architecture to support multi-connectivity.

In 3GPP Release 12, the concept of dual-connectivity was introduced. A UE can be simultaneously connected to two eNodeBs, one master (MeNB) and a secondary (SeNB). The MeNB handles both control and user plane, while the SeNB typically handles the user-plane only by providing additional resources. For 5G systems, the concept of dual-connectivity evolved into multi-connectivity where a UE can be simultaneously attached to a LTE eNodeB and a 5G next generation eNodeB (gNB). The 3GPP Release 15 defines the concepts of 5G new radio (NR) and provides the basis for inter-networking among LTE and NR [29]. Leveraging the PDCP/RLC split, user-plane traffic can be split or duplicated with packet duplication at the aggregation point (the PDCP layer) in the Cu [27]. Then, traffic undergoes separate procedures on the lower-layer protocols, and UEs can individually benefit from separate scheduling, transmission on different frequencies, and reliability schemes.

The buffer management architecture in § 3.2 can support multi-connectivity with minimal modifications. The number of per-user queues at the Cu is not longer one per Du (see Fig. 4), but the RRC protocol will allocate two queues to each active user, one for MeNB and a second one for the SeNB. Since traffic for each Du is scheduled on a different output port of the shared memory switch and multiple per-user queues are allocated at the Cu, the update of virtual thresholds at the Du is driven by both the CQI from the user side and feedback from the Cu.

6 CONCLUSION

MEC architectures leverage edge data centers to perform processing of core and baseband functions of mobile networks. While considerable efforts have been devoted to the analysis of architectural requirements, transport protocol design has remained largely an open research area. In this work, we present pDCell, the first transport for edge data centers hosting MEC applications, which is specifically adapted to cover the whole domain from the data center to the end users attached to base stations. To this end, pDCell incorporates requirements of the wireless domain into the data center transport, including a novel congestion control protocol coupled with a wireless scheduler. Our performance evaluation highlights that pDCell significantly outperforms data center domain transport solutions which are agnostic of the wireless domain. Specifically, it vastly reduces FCT, which makes our solution a prerequisite for ultra-low latency applications.
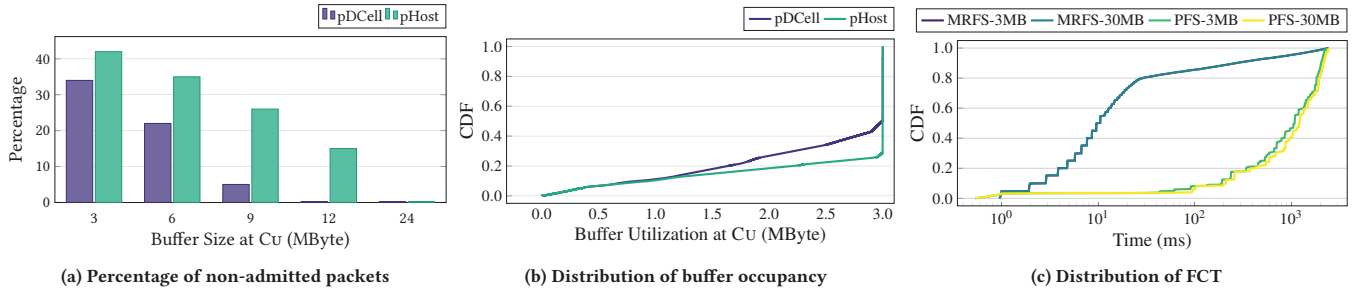
(a) Percentage of non-admitted packets   (b) Distribution of buffer occupancy   (c) Distribution of FCT

Figure 11: Analysis on buffer size at Cᴜ

ACKNOWLEDGMENT

REFERENCES

[1] 3GPP RAN3. 2016. TS 36.306.
[2] 3GPP RAN3. 2017. TR 38.801 V14.0.0.
[3] 3GPP RAN3. 2018. TS 36.521-1 V15.2.0.
[4] William Aiello, Alexander Kesselman, and Yishay Mansour. 2008. Competitive buffer management for shared-memory switches. *ACM Trans. Algorithms* 5, 1 (2008), 1–16. https://doi.org/10.1145/1435375.1435378
[5] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, and Amin Vahdat. 2010. Hedera: Dynamic Flow Scheduling for Data Center Networks. In *Proc. USENIX NSDI*. 281–296.
[6] Mohammad Alizadeh, Shuang Yang, Milad Sharif, Sachin Katti, Nick McKeown, Balaji Prabhakar, and Scott Shenker. 2013. pFabric: Minimal Near-optimal Datacenter Transport. In *Proc. ACM SIGCOMM*. 435–446. https://doi.org/10.1145/2486001.2486031
[7] W. Bai, L. Chen, K. Chen, D. Han, C. Tian, and H. Wang. 2017. PIAS: Practical Information-Agnostic Flow Scheduling for Commodity Data Centers. *IEEE/ACM Trans. on Networking* 25, 4 (Aug 2017), 1954–1967. https://doi.org/10.1109/TNET.2017.2669216
[8] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, and G. Fettweis. 2015. Fronthaul and backhaul requirements of flexibly centralized radio access networks. *IEEE Wireless Communications* 22, 5 (Oct 2015), 105–111. https://doi.org/10.1109/MWC.2015.7306544
[9] Theophilus Benson, Aditya Akella, and David A. Maltz. 2010. Network Traffic Characteristics of Data Centers in the Wild. In *Proc. ACM IMC*. 267–280. https://doi.org/10.1145/1879141.1879175
[10] 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 and Tutorials* 15, 2 (Second 2013), 678–700. https://doi.org/10.1109/SURV.2012.060912.00100
[11] C. Y. Chang, N. Nikaein, and T. Spyropoulos. 2016. Impact of Packetization and Scheduling on C-RAN Fronthaul Performance. In *Proc. IEEE GLOBECOM*. 1–7. https://doi.org/10.1109/GLOCOM.2016.7841885
[12] A. Checko, A. P. Avramova, M. S. Berger, and H. L. Christiansen. 2016. Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings. *Journal of Communications and Networks* 18, 2 (Apr 2016), 162–172. https://doi.org/10.1109/JCN.2016.000025
[13] Inho Cho, Keon Jang, and Dongsu Han. 2017. Credit-Scheduled Delay-Bounded Congestion Control for Datacenters. In *Proc. ACM SIGCOMM*. 239–252. https://doi.org/10.1145/3098822.3098840
[14] A. de la Oliva, J. A. Hernandez, D. Larrabeiti, and A. Azcorra. 2016. An overview of the CPRI specification and its application to C-RAN-based LTE scenarios. *IEEE Communications Magazine* 54, 2 (Feb 2016), 152–159. https://doi.org/10.1109/

[15] Ayman Elnashar and Mohamed A El-Saidny. 2013. Looking at LTE in practice: A performance analysis of the LTE system based on field test results. *IEEE Vehicular Technology Magazine* 8, 3 (2013), 81–92. https://doi.org/10.1109/MVT.2013.2268334
[16] Jose Oscar Fajardo, Ianire Taboada, and Fidel Liberal. 2015. Analysis of CQI Traces from LTE MIMO Deployments and Impact on Classical Schedulers. In *Wired/Wireless Internet Communications*. 60–73.
[17] Peter X. Gao, Akshay Narayan, Gautam Kumar, Rachit Agarwal, Sylvia Ratnasamy, and Scott Shenker. 2015. pHost: Distributed Near-optimal Datacenter Transport over Commodity Network Fabric. In *Proc. ACM CoNEXT*. 1–12. https://doi.org/10.1145/2716281.2836086
[18] A. Garcia-Saavedra, J. X. Salvat, X. Li, and X. Costa-Perez. 2018. WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul. *IEEE Trans. on Mobile Computing* (2018). https://doi.org/10.1109/TMC.2018.2793859
[19] Fabio Giust, Gianluca Verin, Kiril Antevski, Joey Chou, Yonggang Fang, and Walter Featherstone et al. 2018. MEC Deployments in 4G and Evolution Towards 5G. ETSI White Paper.
[20] Mark Handley, Costin Raiciu, Alexandru Agache, Andrei Voinescu, Andrew W. Moore, Gianni Antichi, and Marcin Wójcik. 2017. Re-architecting Datacenter Networks and Stacks for Low Latency and High Performance. In *Proc. ACM SIGCOMM*. 29–42. https://doi.org/10.1145/3098822.3098825
[21] R. Kumar, A. Francini, S. Panwar, and S. Sharma. 2018. Dynamic Control of RLC Buffer Size for Latency Minimization in Mobile RAN. In *Proc. IEEE WCNC*. 1–6. https://doi.org/10.1109/WCNC.2018.8377190
[22] Y. Lu, G. Chen, L. Luo, Y. Xiong, X. Wang, and E. Chen. 2017. One More Queue is Enough: Minimizing Flow Completion Time with Explicit Priority Notification. In *Proc. IEEE INFOCOM*. 1–9. https://doi.org/10.1109/INFOCOM.2017.8056946
[23] Mobile Edge Computing (MEC) ETSI Industry Specification Group (ISG). 2017. Radio Network Information API.
[24] 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 *Proc. ACM All Things Cellular*. 41–46. https://doi.org/10.1145/2627585.2627594
[25] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das. 2011. Understanding traffic dynamics in cellular data networks. In *Proc. IEEE INFOCOM*. 882–890. https://doi.org/10.1109/INFCOM.2011.5935313
[26] Jonathan Perry, Amy Ousterhout, Hari Balakrishnan, Devavrat Shah, and Hans Fugal. 2014. Fastpass: A Centralized "Zero-queue" Datacenter Network. In *Proc. ACM SIGCOMM*. 307–318. https://doi.org/10.1145/2619239.2626309
[27] J. Rao and S. Vrzic. 2018. Packet Duplication for URLLC in 5G: Architectural Enhancements and Performance Analysis. *IEEE Network* 32, 2 (March 2018), 32–40. https://doi.org/10.1109/MNET.2018.1700227
[28] M. T. Raza, D. Kim, K. H. Kim, S. Lu, and M. Gerla. 2017. Rethinking LTE network functions virtualization. In *Proc. IEEE ICNP*. 1–10. https://doi.org/10.1109/ICNP.2017.8117554
[29] J. Sachs, G. Wikstrom, T. Dudda, R. Baldemair, and K. Kittichokechai. 2018. 5G Radio Network Design for Ultra-Reliable Low-Latency Communication. *IEEE Network* 32, 2 (Mar 2018), 24–31. https://doi.org/10.1109/MNET.2018.1700232
[30] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder. 2017. 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *IEEE JSAC* 35, 6 (Jun 2017), 1201–1221.
[31] Chanh Nguyen Le Tan, C. Klein, and E. Elmroth. 2017. Location-aware load prediction in Edge Data Centers. In *IEEE FMEC*. 25–31. https://doi.org/10.1109/FMEC.2017.7946403
[32] Yasir Zaki, Thomas Pötsch, Jay Chen, Lakshminarayanan Subramanian, and Carmelita Görg. 2015. Adaptive Congestion Control for Unpredictable Cellular Networks. In *Proc. ACM SIGCOMM*. 509–522. https://doi.org/10.1145/2785956.2787498