Statistical Multiplexing of Computations in C-RAN with Tradeoffs in Latency and Energy

Kalør, Anders Ellersgaard; Agurto Agurto, Mauricio Ignacio; Pratas, Nuno; Nielsen, Jimmy Jessen; Popovski, Petar

Published in:
IEEE International Conference on Communications (ICC), 2017

Publication date:
2017

Document Version
Peer reviewed version

Statistical Multiplexing of Computations in C-RAN with Tradeoffs in Latency and Energy

Anders E. Kalør*, Mauricio I. Agurto*, Nuno K. Pratas†, Jimmy J. Nielsen†, Petar Popovski†
Department of Electronic Systems, Aalborg University, Denmark
*{akalar12,magurt15}@student.aau.dk, †{nup,jjn,petarp}@es.aau.dk

***Abstract*—In the Cloud Radio Access Network (C-RAN) architecture, the baseband signals from multiple remote radio heads are processed in a centralized baseband unit (BBU) pool. This architecture allows network operators to adapt the BBU's computational resources to the aggregate access load experienced at the BBU, which can change in every air-interface access frame. The degree of savings that can be achieved by adapting the resources is a tradeoff between savings, adaptation frequency, and increased queuing time. If the time scale for adaptation of the resource multiplexing is greater than the access frame duration, then this may result in additional access latency and limit the energy savings. In this paper we investigate the tradeoff by considering two extreme time-scales for the resource multiplexing: (i) *long-term*, where the computational resources are adapted over periods much larger than the access frame durations; (ii) *short-term*, where the adaption is below the access frame duration. We develop a general C-RAN queuing model that describes the access latency and show, for Poisson arrivals, that long-term multiplexing achieves savings comparable to short-term multiplexing, while offering low implementation complexity.**

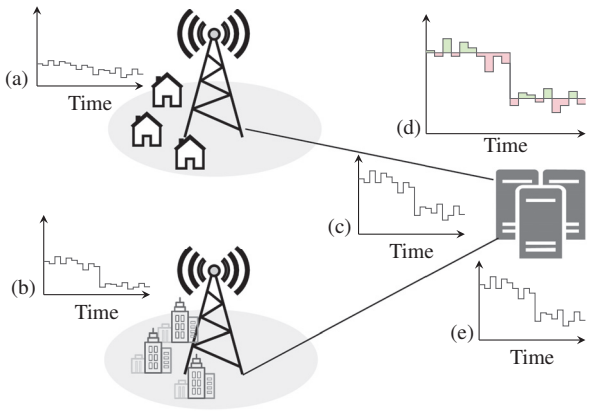

Fig. 1. A C-RAN deployment with 2 RRHs and the application of BBU resource multiplexing to short-term (in frames) and long-term (in hours) load fluctuations. (a) and (b) Load at the individual RRHs. (c) Aggregate load at the BBU. (d) Long-term multiplexing. (e) Short-term multiplexing.

I. Introduction

In the Cloud Radio Access Network (C-RAN) architecture, the Remote Radio Heads (RRHs) are connected through low latency and high capacity front-haul links to a central pool of virtual Base Band Units (BBUs), as illustrated in Fig. 1. This architecture enables the baseband signals from spatially distributed RRHs, to be partially or fully processed in the BBUs [1], allowing for a high level of synchronization and coordination between the RRHs. This ultimately enables spectral efficiency enhancements brought by cooperative techniques, such as coordinated multipoint (CoMP) [2], [3].

The traditional cellular network architecture, denoted as Distributed RAN (D-RAN), has the functionalities of the RRH and BBU concentrated at the base stations. This is neither resource nor cost efficient, as the processing resources at these base stations are dimensioned to handle peak access loads. Furthermore, the only way to reduce the energy and operating expenditures is to turn off partially or completely the base station during periods of low access load. In a C-RAN architecture, the number of processing resources at the BBU can be chosen to take advantage of the load fluctuations across the RRHs, with the goal of reducing energy and operating costs. Access load fluctuations are both slow and fast. Taking a Poisson arrival perspective, *slow fluctuations* refer to the average arrival rate changing over the course of the day, i.e. $\lambda(t)$ which changes in the minutes to hours scale—the so-called *tidal effect* [1]. The *fast fluctuations* refer to the instantaneous realization of the arrival process, i.e. in the milliseconds to seconds scale. Thus, long-term multiplexing refers to adapting the resources to $\lambda(t)$, while the short-term multiplexing refers to the capability to adapt to fast fluctuations. In a C-RAN setting, the adaptation to slow fluctuations is achieved by enabling/disabling RRHs and the associated processing resources according to the current needs. While this also exists in traditional radio access networks, the multiplexing gain becomes more significant when multiple RRHs share the same computational resources, as in the BBU pool. Ideally, fast fluctuations can be taken advantage of by occupying or freeing up computational resources for the BBU pool at high frequency in an elastic cloud environment.

Since long-term multiplexing adapts to the slow fluctuations in the load, there may be periods where the load is higher than what can be served by the allocated resources, as shown in the red area in the curve in Fig. 1(d). These periods introduce queuing in the system and hence higher latency for the users. Similarly, there are periods where the load is lower than what the system can serve (the green area). In this region and depending on the amount of queued arrivals, some resources may be unused, thereby creating a potential for savings which the long-term multiplexing cannot achieve. These fluctuations can be exploited by the short-term multiplexing which operates at a much higher frequency than long-term multiplexing, see Fig. 1(e). In this case, the number of servers follows the

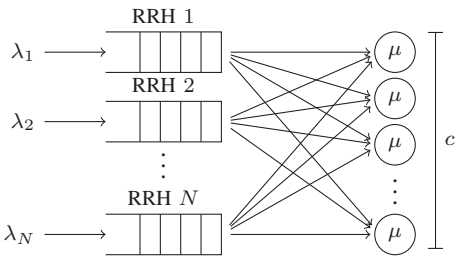

Fig. 2. The transfers from each RRH are queued before they are handled by one of the c shared servers in the BBU pool.

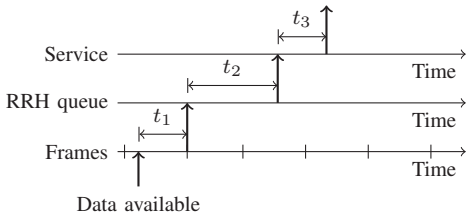

Fig. 3. Transfer delays in the system.

load in every frame. Since the number of active resources is limited only by the air-interface, no additional queuing latency is introduced. However, this requires quick adaptation of resources to the fluctuations, which may be very difficult in practice. Nevertheless, it serves as reference when measuring the potential computational resource savings.

In this paper, we analyze and evaluate the system latency introduced by the multiplexing of computational resources in the BBU pool in a frame-based admission setting, where the amount of resources is determined by the current load. We consider two time-scales for the resource multiplexing: (i) long-term multiplexing, where the computing resources are adapted to the access load in an interval much higher than the air-interface frame duration; (ii) short-term multiplexing, where the number of computing resources can adapt faster than the frame duration. We define a general queuing system model to study resource multiplexing in terms of the tradeoff between latency and resource/energy savings. We also identify the regions where the multiplexing gains at different scales are identical in terms of the achieved savings.

Latency in C-RAN has been previously investigated [4], [5], but only few have studied the latency incurred by computational resource multiplexing in the BBU pool and the effect of frame-based admission. Other studies [6], [7] consider the probability of missed deadlines in the case of BBU sharing, where users are served without queuing. A scheduling framework for long-term load multiplexing is proposed in [8], where both energy savings and the probability of missing a deadline are characterized.

The remainder of the paper is organized as follows. Section II defines the system model and in Section III we analyze the dynamics of the system latency. Numerical results are presented and discussed in Section IV and finally the paper is concluded in Section V.

II. System Model

We consider N RRHs connected to a shared BBU pool via a front-haul link, as presented in Fig. 1. Since we are interested in the latency caused by resource multiplexing, we assume that the front-haul link is dimensioned to meet the latency constraints required by the system. The air-interface at each RRH follows a time and frequency based frame structure, such as in LTE [9], of duration F. The maximum number of supported concurrent user transactions in an air-interface frame is denoted by L. The BBU pool shared between the N RRHs consists of c servers, each able to handle one user transaction at a time (Fig. 2). Under this scheme, at most L servers can serve transfers from the same RRH concurrently. Since no more than $L \cdot N$ transfers can be active at the same time, we only consider the cases where $c \leqslant L \cdot N$. Each RRH has its own queue of transfers; and the available servers in the BBU are assigned to these queues in a round-robin fashion to provide fairness between the RRHs.

Within the time scale of an air-interface frame, we treat c as constant. In the case of long-term resource multiplexing, c will only change after a large number of frames have elapsed (large enough to assume stationary conditions). In the short-term resource multiplexing, c is adapted at the beginning of each frame according to the access load.

We assume that a user transaction is composed of several uplink and downlink exchanges; corresponding to the user connection establishment to the network, the Scheduling Request (SR), the subsequent data exchanges and release of the network connection. A user transaction is completed only after all its uplink and downlink transmissions have been completed. We model the user transaction arrivals at the jth RRH as a Poisson arrival process with intensity λ_j, where the arrivals can only enter the system at the beginning of each frame. Upon the connection establishment, the scheduling request of a user is queued at the RRH until resources become available to initiate the transfer and users are instantaneously informed when they are assigned resources.

The latency experienced by a user is illustrated in Fig. 3. The delay t_1 corresponds to the time from when data is available to be transferred at the user device until the scheduling request is transmitted in the beginning of the following frame. t_2 corresponds to the time spent in the RRH queue, i.e. the time from reception of the scheduling request until the first resource is granted. The last delay, denoted as t_3, is the service time, i.e. the time it takes for the user to transmit its data. We assume that the amount of requested resources in a scheduling request follows an exponential distribution with rate μ [10]. In this way the amount of requested resources may exceed one frame, in which case the transfer will span multiple frames.

A. Resource/Energy Savings

In long-term multiplexing, the servers are adapted to the mean arrival rate. We assume that some target delay τ exists with a certain reliability ζ (e.g. less than 1 frame period queuing time with probability 99%) and that the minimum

number of servers required to fulfill this requirement are allocated.

For short-term multiplexing, we assume that servers can be turned off when they are idle, and are only turned back on when needed. If servers are turned on and off instantaneously and at any time, the achievable savings would be equal to the mean idle time, $1-\rho$. However, we consider a more realistic scenario, where servers can be turned on and off only in the beginning of each frame. Specifically, the number of transfers in the system is observed immediately after the arrival of SRs in the beginning of each frame, and the servers required to serve them are allocated while the remaining servers are turned off. The number of allocated servers in the BBU is given by

$$c = \sum_{j=1}^{N} \min(L, l_j) \tag{1}$$

where l_j is the number of transfers in the system (ongoing and queued) from RRH j.

III. Analysis

In this section we obtain the probability distributions describing the latency in the system. Based on these distributions we find expressions for potential energy/resource savings. We start by defining the conditions required to ensure stability in the network. Then we characterize the long-term resource multiplexing through an approximate Markov chain model which describes the metrics of interest. Finally, we present the short-term resource multiplexing, where the number of servers c at the BBU pool is dynamically adopted in each frame to the fast fluctuations of the arriving traffic.

A. Stability Conditions

The stability condition for the described system is given in terms of the utilization ρ. Since the utilization is both limited by the total number of servers in the BBU, c, and the number of concurrent transactions, L, there are two conditions that must be satisfied for stability. First, the total number of arrivals must be less than what the c servers in the BBU pool can handle (2). Second, the arrivals at each RRH must be below what is supported by the air-interface (3):

$$\rho_{\text{BBU}} = \frac{1}{c\mu F} \sum_{j=1}^{N} \lambda_j < 1, \tag{2}$$

$$\rho_{\text{RRH}j} = \frac{\lambda_j}{L\mu F} < 1, \quad \forall\, j = 1 \dots N. \tag{3}$$

When $\rho_{\text{BBU}} \geqslant 1$ or $\rho_{\text{RRH}j} \geqslant 1$, the queue grows to infinity.

B. Long-term Resource Multiplexing

In this subsection, we present a Markov chain model which approximates the latencies introduced by queuing in the system under stationarity. Motivated by the round-robin scheduling, we analyze the RRHs individually with a fixed number of servers proportional to the arrival rates. The number of servers used for analyzing RRH j is given as:

$$\hat{c}_j = \left\lfloor \frac{\lambda_j}{\sum_n \lambda_n} c \right\rfloor \tag{4}$$

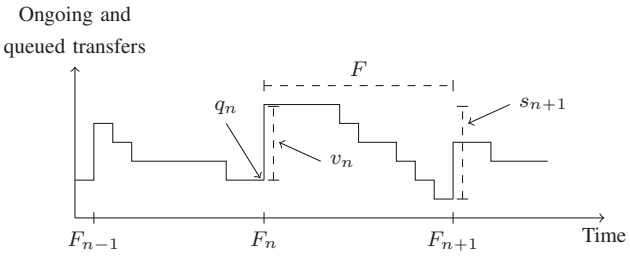

Fig. 4. The notation used in the Markov chain model.

where c is the number of servers in BBU pool and $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. To simplify the notation, we shall refer to $\hat{c}_j$ simply as c and λ_j as λ in the remainder of this section and refer to Fig. 4 for the Markov chain symbol notation.

We consider a discrete-time Markov chain that is observed in the beginning of each frame, immediately prior to the arrival of SRs, denoted as $\{F_n\}$. The states are indexed by the number of transfers (ongoing and queued) in the system q_n. Let v_n denote the number of SRs arriving in frame n specified by the Poisson probability mass function

$$p_{v_n}(v_n) = \frac{\lambda^{v_n} e^{-\lambda}}{v_n!}, \qquad v_n \geqslant 0, \tag{5}$$

and let s_{n+1} be the number of transfers completed between F_n and F_{n+1} (Fig. 4). We form the basic relation

$$q_{n+1} = q_n + v_n - s_{n+1}.$$

We seek the transition probabilities $p_{i,j}$ in the Markov chain such that

$$p_{i,j} = \Pr(q_{n+1} = j | q_n = i).$$

For convenience, we analyze the transition probabilities conditioned on the number of arriving SRs in a frame since this quantity is independent of the system state:

$$\psi_{i+k,j} = \Pr(q_{n+1} = j | q_n = i, v_n = k).$$

From $\psi_{i+k,j}$ we obtain $p_{i,j}$ by marginalizing over the number of arrivals v_n:

$$p_{i,j} = \begin{cases} \sum_{k=0}^{\infty} \psi_{i+k,j} p_{v_n}(k) & i+k \geqslant j \geqslant 0, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

We consider the following three cases of $\psi_{i+k,j}$:

1) $i + k \leqslant c$ corresponding to when all arriving SRs are immediately served.
2) $i + k > c$, $j \geqslant c$ corresponding to when some of the arriving SRs are placed in the queue, and all servers remain busy during the entire frame.
3) $i + k > c$, $j < c$ corresponding to when some of the arriving SRs are placed in the queue, but at least one server is idle in the beginning of the succeeding frame.

When $i + k \leqslant c$ all k arriving SRs are immediately served. Exactly $i + k - j$ transfers will complete service within the frame period, or equivalently, exactly j out of $i + k$ will *not*

complete transfer. Since the service time is exponential, the probability that a transfer will not complete service within the frame period is $e^{-\mu F}$, and hence we obtain

$$\psi_{i+k,j} = \binom{i+k}{j} e^{-\mu F j}(1-e^{-\mu F})^{i+k-j}, \qquad i+k \leqslant c.$$

For case 2, where $i+k > c, j \geqslant c$, exactly $i+k-c$ SRs are placed in the queue and all c servers will remain busy throughout the frame period during which $i+k-j$ transfers will complete service. Since all servers are busy in the entire frame period, the number of completed transfers is Poisson distributed with rate $c\mu F$:

$$\psi_{i+k,j} = \frac{(c\mu F)^{i+k-j} e^{-c\mu F}}{(i+k-j)!}, \qquad i+k > c,\ j \geqslant c.$$

When $i+k > c, j < c$, one or more servers are idle prior to the subsequent frame. As in the previous case, $i+k-c$ SRs are initially placed in the queue and all servers are busy, but after $i+k-c+1$ transfers have been completed, some servers remain idle for the remaining frame.

Let $t \leqslant 1$ denote the time (in frame durations) until $i+k-c+1$ transfers have completed. t is a sum of $i+k-c+1$ independent and identically distributed exponential random variables and hence is Erlang distributed [11] with density function

$$p(t) = \frac{(c\mu F)^{i+k-c+1} t^{i+k-c} e^{-c\mu F t}}{(i+k-c)!}.$$

The probability that $j < c$ transfers remain in service after ξ frame periods, assuming the total number of transfers in the system is less than c is given by

$$p(j|\xi) = \binom{c-1}{j} e^{-\xi\mu F j}(1-e^{-\xi\mu F})^{c-j-1}.$$

Marginalizing over $\xi = 1-t$ we obtain the final expression for $\psi_{i+k,j}$:

$$\begin{aligned}\psi_{i+k,j} =& \int_0^1 \binom{c-1}{j} e^{-(1-t)j\mu F}(1-e^{-(1-t)\mu F})^{c-j-1} \\ & \frac{(c\mu F)^{i+k-c+1} t^{i+k-c} e^{-c\mu F t}}{(i+k-c)!}\, dt \\ =& \binom{c-1}{j} \frac{e^{-j\mu F}(c\mu F)^{i+k-c+1}}{(i+k-c)!} \\ & \int_0^1 e^{-t\mu F(j-c))}(1-e^{-(1-t)\mu F})^{c-j-1} t^{i+k-c}\, dt, \\ & \qquad i+k > c,\ j < c.\end{aligned}$$

To obtain the stationary queuing time distribution we first seek the stationary queue length distribution. Let $\boldsymbol{P} = \left[p_{i,j}\right]$ denote the transition matrix and $\boldsymbol{\pi} = \left[\pi_i\right]$ be a vector of state probabilities. Since the Markov chain is irreducible and aperiodic, the stationary state distribution $\boldsymbol{\pi}$ is given by

$$\boldsymbol{\pi} = \lim_{n\to\infty} \boldsymbol{\pi}^{(0)} \boldsymbol{P}^n$$

where $\boldsymbol{\pi}^{(0)}$ is the initial state distribution. We obtain $\boldsymbol{\pi}$ by imposing a finite queue length M and multiplying iteratively by $\boldsymbol{P}$ until convergence. Equation (6) then becomes:

$$p_{i,j} = \sum_{k=0}^{\infty} p_{v_n}(k)\psi_{i+\min(k,c+M-i),j}, \qquad i+k \geqslant j \geqslant 0,\ i \leqslant c+M.$$

From $\boldsymbol{\pi}$ we may obtain the distribution of the queuing time t_2 under stationarity. Let $q'_n = q_n + v_n$ be the number of transfers in the system immediately *after* arrival of SRs. Since we assume stationarity, we omit the time index and write $q' = q + v$. We may factorize the queuing time distribution as

$$p_{t_2}(t_2) = \sum_q \sum_{q'} \sum_l p_{t_2|l}(t_2|l)\Pr(l|q,q')\Pr(q'|q)\pi_q, \qquad t_2 \geqslant 0,\ c+M \geqslant q' \geqslant q \geqslant 0,$$

where $p_{t_2|l}(t_2|l)$ is the density function of the queuing time conditioned on an SR arriving to state l and $\Pr(l|q,q')$ is the probability of arriving to state l given that the newly arrived SRs occupy states $q+1,\ldots,q'$. Two cases of $p(t_2|l)$ exists: when the SR arrives to an idle server ($l \leqslant c$), and when it arrives to the queue ($l > c$). In the former case, the SR is immediately served and the queuing time is 0. In the latter case, the queuing time is Erlang distributed with parameters $l-c$ and $c\mu$:

$$p_{t_2|l}(t_2|l) = \begin{cases} \frac{(c\mu)^{l-c} t_2^{l-c-1} e^{-c\mu t_2}}{(l-c-1)!} & l > c, \\ \delta(t_2) & \text{otherwise} \end{cases}$$

where $\delta(x)$ is the Dirac delta function. It is equally likely for an SR to arrive to any of the states between $q+1$ and q', hence $\Pr(l|q,q')$ is a discrete uniform distribution between $q+1$ and q', i.e. $\Pr(l|q,q') = (q'-q)^{-1}$. $\Pr(q'|q)$ is obtained by truncating the Poisson distribution at the queue size limit M,

$$\Pr(q'|q) = \begin{cases} p_{v_n}(q'-q) & q' < M, \\ 1 - \sum_{n=q}^{M-1} p_{v_n}(n-q) & q' = M, \\ 0 & \text{otherwise.} \end{cases}$$

We may obtain the system time, t_2+t_3 by the convolution of the queuing time and the service time density functions. Similarly, the transfer time $t_1+t_2+t_3$ can be obtained by convolution of the densities for t_2+t_3 and t_1. Since t_1 is uniformly distributed in the range $[0,F]$ we obtain

$$p_{t_2+t_3}(t) = \int_0^t p_{t_2}(x)\mu e^{-\mu(t-x)}\, dx,$$

$$p_{t_1+t_2+t_3}(t) = \frac{1}{F}\int_0^F p_{t_2+t_3}(t-x)\, dx.$$

The normalized savings (server-hours) in the long-term multiplexing scheme, as defined in II-A, is expressed as

$$S_{\text{LT}} = \frac{1}{L\cdot N} \min\{c : \Pr(t_2 < \tau) \geqslant \zeta\} \tag{7}$$

where $L \cdot N$ is the maximum number of servers in the BBU pool and τ and ζ are design parameters.

TABLE I
PARAMETERS CONSIDERED IN THE EVALUATION

Parameter	Symbol	Value
Frame duration	F	10
Maximum concurrent transactions per RRH	L	25
Number of RRHs	N	2
Mean number of requested resources	$1/\mu$	5

C. Short-term Resource Multiplexing

Recall that in the short-term multiplexing we assume instant adaptation to the active and queued transfers in the beginning each frame (see II-A). The number of transfers in the system immediately after arrival of the scheduling requests in the beginning of the frame is given by

$$\Pr(q') = \sum_{q} \Pr(q'|q)\pi_q.$$

As we assume that we can instantly switch servers on and off, we may obtain the expected number of active servers by marginalizing over q'. By further using the fact that at most c servers can be active at the same time, and normalizing by c, the expected savings of short-term multiplexing are,

$$\mathbb{E}[S_{\text{ST}}] = 1 - \frac{1}{c}\sum_{q'} \min(q', c)\Pr(q') \tag{8}$$

One case of c which is particularly interesting is where the maximum number of servers is used in the BBU pool and the RRHs have equal arrival rates, i.e. $c = L$. Since we are adapting in the beginning of each frame, the frame length has high impact on the savings. Specifically, when the frame length is short, we can adapt more often and the servers will be inactive for shorter time. This is also clear from (8) where q' will be lower (in a stochastic ordering sense) when the frame length is shorter due to fewer arrivals per frame.

IV. NUMERICAL RESULTS

In this section we present the numerical results of the long-term and short-term resource multiplexing approaches. We consider a system with the parameters specified in Table I. We study the case with two RRHs with equal arrival rates $\lambda_1 = \lambda_2$, as this is sufficient to show the dynamics of the resource multiplexing.

A. Resource Multiplexing Savings

The plot presented in Fig. 5 shows the 99-percentile queuing time experienced in a system with long-term multiplexing for a different number of available servers and arrival rates. The horizontal dashed line indicates $\tau = 1$. The intersection between the curves and $\tau = 1$ corresponds to the number of servers required to achieve a 99-percentile queuing time of $\tau = 1$, i.e. $1/10$ frame duration.

The lower bound of the queuing time, for all considered λ, occurs when $c = 50$. On the other hand, there is a minimum number of servers required to keep the system stable that obeys the condition in eq. (2). This point is reflected in the vertical asymptotes in the plot. The fact that the queuing time only decreases slightly when the number of servers increases indicates that savings can be done with only limited increased latency. This motivates the long-term multiplexing scheme where the number of servers in the BBU pool is adapted to the slowly varying mean arrival rate. For instance, in the considered case with arrival rates $\lambda_1 = 10, \lambda_2 = 10$, only 20 servers are required to provide 99-percentile queuing time $\tau = 1$, which introduces considerable savings when compared to the case where all the servers are active (i.e. $c = 50$). Hence, long-term multiplexing provides high savings in this case.

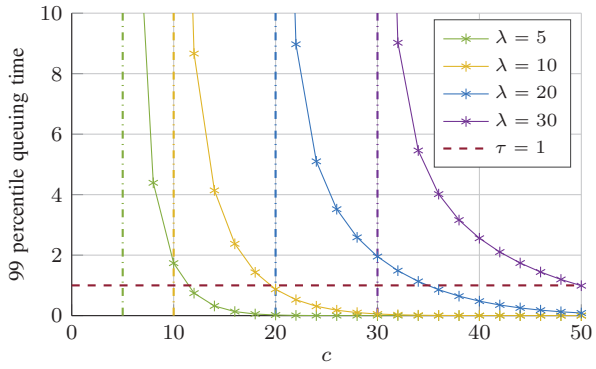

Fig. 5. The 99-percentile queuing delay vs. the number of servers for different arrival rates, where $\lambda = \lambda_1 = \lambda_2$. The dashed horizontal line indicates $\tau = 1$, and the dash-dotted lines show the asymptotes.

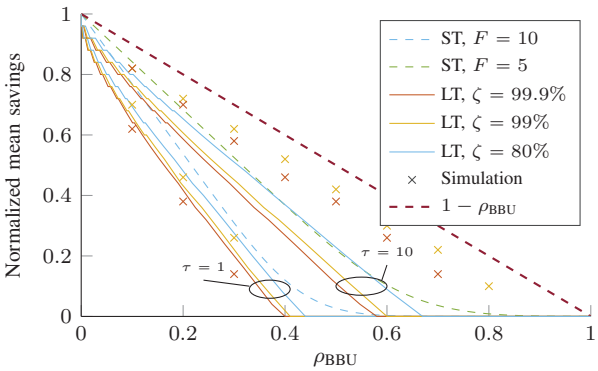

Fig. 6. Long-term (LT) and short-term (ST) server-hour savings for different mean arrival rates (reflected in ρ_{BBU}) and ζ-percentile queuing times. $1-\rho_{\text{BBU}}$ for the case with $c = 50$ provides an upper bound on the savings. The simulation results correspond to the 99- and 99.9-percentile cases.

The savings (server-hours normalized by the server-hours in the baseline case with 50 servers) which can be achieved using long-term and short-term multiplexing are illustrated in Fig. 6. For long-term multiplexing we consider the minimum number of servers required to provide a queuing delay of $\tau = 1$ and $\tau = 10$ at different percentiles ζ. We obtain this number by calculating the queuing time distribution functions for different values of c and choose the minimum that satisfies the expression in (7).

The simulation results, shown for the 99- and 99.9-percentiles, reveal that the derived analytical model fits well

for $\tau = 1$ but overestimates the number of servers required in the case of $\tau = 10$. This effect comes from the analysis considering each RRH queue separately, which leads to a lower statistical multiplexing gain compared to the actual system where high queuing delays are less likely.

As shown in Fig. 6, the case with $\tau = 10$ allows for higher savings since we allow the queue to be larger and hence can reduce the number of active servers. Likewise, a low percentile allows for higher savings since we allow the queuing time to exceed τ with higher probability. This reflects that high savings come at the cost of an increased queuing delay. However, increasing the percentile only leads a to minor decrease in savings, which indicates that significant savings can be achieved with long-term multiplexing while maintaining a very low latency.

The long-term savings approach the upper bound provided by the normalized idle time, $1 - \rho_{\text{BBU}}$, as $\tau \to \infty$. Independently of ζ and τ, a low utilization ρ_{BBU} (i.e. a low arrival rate) also leads to higher savings since less servers are required to provide the desired queuing delay. Even by slightly increasing the queuing time, notably $\tau = 1$, it is possible to achieve significant savings when the utilization is low.

The savings achieved by short-term multiplexing are also shown in Fig. 6 for frame durations of $F = 10$ and $F = 5$. Interestingly, we see that if the delay requirements are not too strict and $\tau = 10$ can be accepted, then the long-term multiplexing provides higher savings than the short term multiplexing with $F = 10$, which is the same frame length as in LTE. Even if only $\tau = 1$ can be accepted, the savings of long-term multiplexing are only slightly lower than the short-term multiplexing for $F = 10$. If a reduced frame length of F is used, larger savings can be achieved; approaching the upper bound $1 - \rho_{\text{BBU}}$ as $F \to 1$.

We note that savings comparable to the short-term multiplexing can be achieved using long-term multiplexing, while still offering guarantees of low latency. With the outlook to smaller frame sizes in 5G [12] it is unlikely that servers can be turned on and off fast enough to enable short-term multiplexing. Even if it became possible, short frames and faster resource adaptation causes high levels of signaling overhead and high complexity, meaning that long-term multiplexing would anyways be preferred. An exception is the support of ultra-reliable low latency communications (URLLC), where latency violations are not acceptable. However, URLLC is not well suited for round-robin scheduling, as considered in this paper, and will likely require latency sensitive schedulers.

V. Conclusion

This paper studies the latency and energy tradeoffs in computational multiplexing in C-RAN. We identify two multiplexing time-scales: (i) long-term multiplexing, where the mean arrival rate varies over the time of day; and (ii) short-term multiplexing where the statistical multiplexing in the arrivals within each frame is exploited. The long-term multiplexing introduces additional queuing delay, but has low implementation complexity. Short-term multiplexing does not add queuing delay, but is difficult to realize in practice since it requires switching resources on and off at a very high frequency. We propose a general system model where user transfers are modelled as jobs in a queuing system with servers shared among several RRHs.

We show that both multiplexing schemes can provide significant resource savings. Furthermore, it is possible to achieve long-term multiplexing savings that are comparable to those in the short-term while still maintaining a low queuing latency in high percentiles. This suggests that long-term multiplexing provides a good tradeoff between resource savings and realization complexity.

Acknowledgment

This work was performed partly in the framework of H2020 project FANTASTIC-5G (ICT-671660) and partly by the European Research Council Consolidator Grant Nr. 648382. The authors acknowledge the contributions of the colleagues in FANTASTIC-5G.

References

[1] China Mobile, "C-RAN: The road towards green RAN," *White Paper*, 2011.

[2] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks—a technology overview," *IEEE Communications surveys & tutorials*, vol. 17, no. 1, pp. 405–426, 2015.

[3] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 148–155, 2012.

[4] Q. Han, C. Wang, M. Levorato, and O. Simeone, "On the effect of fronthaul latency on ARQ in C-RAN systems," *arXiv preprint arXiv:1510.07176*, 2015.

[5] U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier, "Quantitative analysis of split base station processing and determination of advantageous architectures for LTE," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 105–128, 2013.

[6] J. Liu, S. Zhou, J. Gong, Z. Niu, and S. Xu, "On the statistical multiplexing gain of virtual base station pools," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 2283–2288.

[7] T. Werthmann, H. Grob-Lipski, and M. Proebster, "Multiplexing gains achieved in pools of baseband computation units in 4G cellular networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2013, pp. 3328–3333.

[8] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, and T. Woo, "CloudIQ: A framework for processing base stations in a data center," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 125–136.

[9] G. C. Madueño, J. J. Nielsen, D. M. Kim, N. K. Pratas, Č. Stefanović, and P. Popovski, "Assessment of LTE wireless access for monitoring of energy distribution in the smart grid," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 675–688, 2016.

[10] 3rd Generation Partnership Project (3GPP), "TS 36.213 - Further Advancements for E-UTRA Physical layer Aspects," 2010.

[11] L. Kleinrock, *Queueing Systems*. Wiley Interscience, 1975, vol. I: Theory.

[12] H. Tullberg, Z. Li, A. Hoglund, P. Fertl, D. Gozalvez-Serrano, K. Pawlak, P. Popovski, G. Mange, and O. Bulakci, "Towards the METIS 5G concept: First view on horizontal topics concepts," in *Networks and Communications (EuCNC), European Conf. on*. IEEE, 2014, pp. 1–5.