ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/323718822

Cloud-RAN Modeling Based on Parallel Processing

Article *in* IEEE Journal on Selected Areas in Communications · March 2018

DOI: 10.1109/JSAC.2018.2815378

CITATIONS
35

READS
329

2 authors:

Veronica Quintuna Rodriguez
Orange Labs Lannion
33 PUBLICATIONS **203** CITATIONS
SEE PROFILE

Fabrice Michel Guillemin
Orange Labs
256 PUBLICATIONS **2,616** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Flexible Network Control View project

Project Trafic prediction View project

All content following this page was uploaded by Veronica Quintuna Rodriguez on 29 May 2019.

The user has requested enhancement of the downloaded file.

Cloud-RAN modeling based on parallel processing

Veronica Quintuna Rodriguez, Fabrice Guillemin

***Abstract*—We consider in this paper the implementation of a Cloud Radio Access Network (C-RAN) on a centralized multi-core system supporting the base band processing of several distributed antennas. We present a parallel processing model based on both functional and data decomposition of virtualized Base Band Unit (BBU) functions in order to reduce their runtime. We study two scheduling strategies of parallel runnable BBU jobs, where computing resources can be allocated either per User Equipments (UE) or else per Code Blocks (CB). By using data obtained when running an open source RAN code (namely, OAI), we introduce a batch queuing model (the $M^{[X]}/M/C$ multi-service system) to assess the needed processing capacity in a data center while meeting tight latency requirements in the down-link and up-link directions. The proposed model is validated by simulation when processing a hundred LTE-cells in a multi-core system. Results provide valuable guidelines for sizing and deploying Cloud-RAN systems.**

***Index Terms*—Cloud-RAN, virtualization, NFV, VNF, queuing theory, parallel processing, batch model, $M^{[X]}/M/C$ system, scheduling, resource pooling, virtualized BBU, multi-core systems, channel coding, dimensioning.**

I. Introduction

The cloudification of mobile networks promises great economic savings as well as better management of radio resources. In addition, this emerging concept will enable network operators to deploy on-demand virtualized networks and offer innovative services. This goal is in line with the general framework of Network Function Virtualization (NFV) [1], which precisely consists of replacing network functions running on dedicated and proprietary hardware with open software applications running on shared Commercial off-the-shelf (COTS) servers in cloud platforms. Network operators are thus able to instantiate Virtualized Network Functions (VNFs) on the fly at various network locations to meet customer requirements [2].

The overarching principle of virtualization is to host network functions on one or more Virtual Machines (VMs) or containers. VNFs are deployed on top of a virtualized infrastructure, which may span over more than one physical location and even over a cloud computing infrastructure. Ideally, VNFs should be located where they are the most efficient in terms of performance and cost. VNFs can be hosted in data centers, network nodes or even in end user devices depending on the required performance (notably latency) and resources (bandwidth, storage and computing).

The cloudification and commoditization of network functions, however, brings new challenges, especially when virtualizing wireless access networks. This is notably the case of Cloud-RAN (C-RAN), which aims at implementing the whole base-band processing of radio signals in software. Several functional splits are currently under study; the most ambitious one is certainly the full centralization of base-band functions. This architecture relies on distributed antennas and Base Band Units (BBUs) grouped into a Central Office (CO). A BBU involves critical lower-layer functions such as channel encoding and decoding, modulation and demodulation as well as radio scheduling, radio link control, and data convergence.

A fully centralized architecture can achieve multi-site radio cooperation and interference management, which allows both better spectrum efficiency and user experience [3], [4]. However, the total centralization must meet strict latency requirements defined by Long Term Evolution (LTE) standards [5]. In fact, the base-band processing needs to be completed within 1 millisecond in the down-link direction and 2 milliseconds in the up-link. As a consequence, highly performing processors and their efficient utilization are mandatory.

Many efforts have already been devoted to the implementation of traditional Radio Access Network (RAN) by means of virtualized functions. Some studies reveal that functions belonging to the physical layer, especially the channel coding function, consume the largest amount of processing time and computing resources; see for instance [6]. Thus, besides the requirement for high-performance processors, parallel programing techniques can be considered to ensure the agility and flexibility promised by NFV [7], [8]. Moreover, one main performance problem of C-RAN is due to the non-deterministic behavior of the channel coding function, i.e., the variability of runtime that is required by the encoding and decoding processes. Much of the variability is due to radio channel conditions of each User Equipment (UE) attached to the eNodeB (eNB), the required data rate per UE, as well as the amount of traffic in the cell.

The above observations raise fundamental questions with regard to dimensioning the required computing capacity under non-deterministic conditions. To solve this problem, we adopt in this paper a probabilistic modeling approach. We consider a number of BBU hosted in a multi-core system. We model how the different tasks of BBU processing are invoked and we use data observed by running the Open Air Interface (OAI) open source RAN code to adjust the parameters of our model. It turns out that the $M^{[X]}/M/C$ queuing system [2] can be used to determine the required computing capacity (i.e., the number of processing units) when parallelizing BBU functions and to establish dimensioning rules for a C-RAN implementation.

This paper is organized as follows: Related work and the most popular C-RAN scenarios are presented in Section II. A detailed analysis of parallel runnable virtual RAN functions is presented in Section III. In Section IV, we introduce the modeling assumptions and the queuing system formulation. The theoretical analysis of the C-RAN model is exposed in

Manuscript received September 30, 2017; revised March 1, 2018. The authors are with Orange Labs, Lannion, France (e-mails: veronicakarina.quintunarodriguez@orange.com, fabrice.guillemin@orange.com).

Section V. Some numerical experiments are given in Section VI, where we also validate the model by emulating a real C-RAN system. Finally, in Section VII, we present the main conclusions of this study.

II. Related Work

C-RAN, also referred to as Centralized-RAN, was first introduced by China Mobile Research Institute in 2010 [9]. Since then, various studies and test-bed platforms have appeared in the literature. Certainly, the main challenge of this promising software based approach is the required real-time behavior of virtual RAN functions. This problem has been largely studied and analyzed by the industry [4], [10] and network operators [3], [7], [11], as well as academic researchers, notably via the development of several open-source or even proprietary solutions such as OAI [12] and Amarisoft [13].

The performance analysis of virtualized RAN functions when using OAI and several virtualization environments such as Kernel-based Virtual Machine (KVM), Docker, and Linux Containers (LXC) has been presented in [6]. Results show that the up-link direction is the dominant processing load and requires to be split and/or accelerated. Coding and decoding functions are the most time consuming with high variability. It is also found that container-based approaches provide slightly better performance than hypervisor-based approaches.

First efforts for reducing the runtime of RAN functions were presented in [14]. The authors propose a framework that split the set of BBUs into groups that are simultaneously processed on a shared homogeneous compute platform. This work shows that the centralized architecture can potentially result in savings of at least 22% in computing resources by exploiting the variations in the processing load across eNBs (BBUs). The performance gain when performing resource sharing and statistical multiplexing principles is also shown in [15].

Note that real-time and parallel processing systems have been widely studied in the literature for many years. Most efforts have been devoted to the improvement of scheduling strategies in both uniprocessor and multiprocessor models, e.g., Earliest Deadline First (EDF) and Processor Sharing (PS) models. Nevertheless, the only way to improve the C-RAN performance in terms of latency while avoiding overhead is decomposing heavy long tasks in parallel runnable small jobs [8], [11].

The most popular use cases of C-RAN rely on areas with huge demand, such as high density urban areas with macro and small cells, public venues, etc. The following ones are currently considered by academia and industry:

- *Network slicing* offers a smart way of segmenting the network and of supporting customized services, e.g, a private mobile network. Slices can be deployed with particular characteristics in terms of Quality of Service (QoS), latency, bandwidth, security, availability etc. This scenario and the strict C-RAN performance requirements are studied in [16].
- *Multi-tenancy 5G networks* handle various Virtual mobile Network Operators (VNOs), different Service Level Agreements (SLAs) and Radio Access Technologies (RATs). A global 5G C-RAN scenario is given in [17]. Authors propose a centralized Virtual Radio Resource Managements (VRRMs) so-called "V-RAN enabler" to orchestrate the global environment. The management entity estimates the available radio resources based on the data rate of different access technologies and allocate them to the various services in the network by the OAI scheduler.
- *Coexistence of heterogeneous functional splits* for supporting fully or partially centralization of RAN network functions. An in-depth analysis of the performance gain when performing different functional splits is presented in [10]. Results show that the performance decreases when lower-layer RAN functions are kept near to antennas. This work recommends full centralization to take advantage of the "physical" inter-cell connectivity for deploying advanced multi-point cooperation technologies to improve the Quality of Experience (QoE).
- *Intelligent Networks* which enable the automated deployment of eNBs for offering additional capacity in real time [17]. These intelligent procedures bring an enormous economic benefit for network operators, which today massively invest to extend their network capacity.

Other rising Cloud-RAN applications have been summarized in [10]. It includes massive Internet of Things (IoT) applications, broad band communications, which are delay-sensitive (e.g., virtual reality, video replay at stadiums, etc.), low-latency and high-reliability applications such as assisted driving and railway coverage, since C-RAN enables fast handover for UEs moving with high speed.

III. Cloud-RAN system

C-RAN aims at centralizing the base-band processing of radio signals coming from various antennas in a CO or more generally in the cloud. In other words, C-RAN dissociates antennas (Radio Remote Heads (RRHs)) and signal processing units (BBUs). C-RAN can be seen as a BBU pool, which handles tens or even hundreds of cell sites (eNBs). A site is typically composed of 3 sectors, each of them equipped with an RRH. The RRH has two RF paths for down-link and up-link radio signals, which are carried by fiber links to the BBU-pool.

A. Virtual functions

A C-RAN's VNF is nothing else but a virtualized BBU (vBBU), which implements in software all network functions belonging to the three lower layers of the LTE protocol stack. These functions mainly concern PHY functions as signal generation, IFFT/FFT, modulation and demodulation, encoding and decoding; radio scheduling; concatenation/segmentation of Radio Link Control (RLC) protocol; and encryption/decryption procedures of Packet Data Convergence Protocol (PDCP), for the down-link and up-link directions [18]. The challenge is to execute virtualized BBU functions sufficiently fast so as to increase the distance between RRHs and BBU functions (namely, the BBU-pool) and

thus to improve the concentration level of BBUs in the CO for CAPEX and OPEX savings.

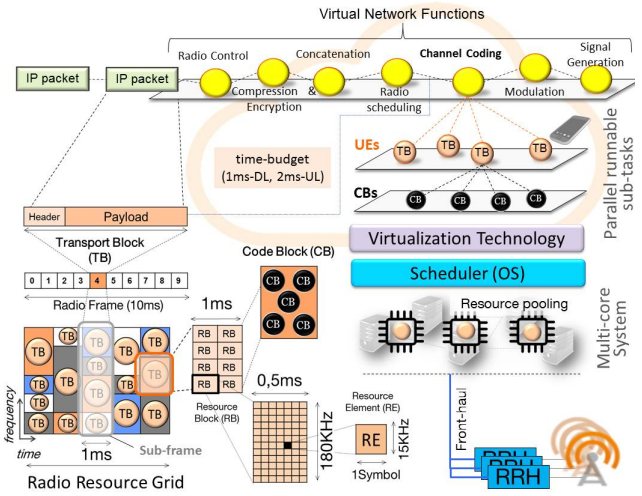


Fig. 1. Cloud-RAN architecture.

As shown in Figure 1, a vBBU can then be instantiated on the top of the cloud infrastructure and modeled by means of a forwarding graph of sub-functions, which in turn can be divided into parallel runnable tasks or jobs. BBU's jobs can be executed on a multi-core platform according to a scheduling strategy, that resides in the kernel of the Operating System (OS) of the host. In the proposed C-RAN architecture, we take advantage of the performance provided by containers, which in contrary to VMs, hold a single OS [19], i.e., all cores are controlled by a global scheduler.

When a UE requires either data transmission or reception, a vBBU instance is invoked. As a consequence, various instances of the virtualized BBU run simultaneously in the computing platform. In LTE-based cellular systems, the transmission data unit (namely, the sub-frame) can be composed of data of various UEs. The whole base-band processing of a sub-frame must be performed within 2 milliseconds and 1 millisecond in the up-link and down-link direction, respectively. Because a sub-frame is generated every millisecond in both directions (see Figure 2 for an illustration), the base-band processing of all cells belonging to a Cloud-RAN system very likely requires high-performance parallel computing.

When executing BBU functions on a multi-core system, various scheduling strategies can be envisaged such as scheduling per LTE sub-frame [14] or at finer granularity, for instance per UE or per Code Block (CB) [7], [8]. The scheduling strategy must improve the performance of the BBU-pool in terms of latency. As observed in [7], [8], the performance gain is more important when the scheduler deals with short jobs (e.g., processing CBs) instead of dedicating computing resources to heavy tasks (e.g., processing sub-frames) [11].

B. Parallel processing

The general philosophy of parallel computing consists of splitting large tasks into smaller parallel runnable sub-tasks, which can be executed on multi-core systems. The parallel

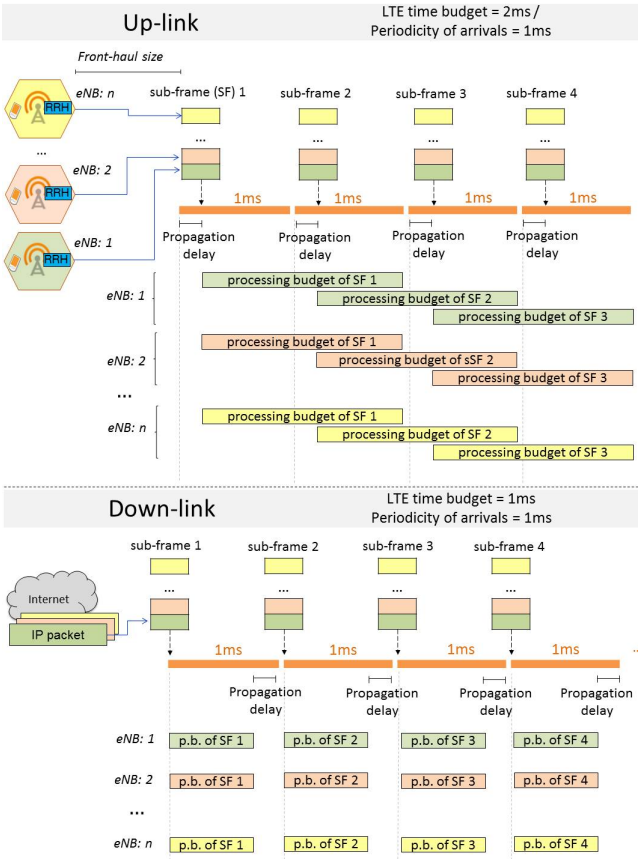


Fig. 2. Processing budget of LTE sub-frames [11].

execution of sub-tasks allows the runtime of the whole task to be shortened. In a Cloud-RAN system, the processing of the channel coding function is the most resource consuming and has furthermore a non-deterministic behavior [6]. In the following, we focus on the parallel processing of the channel coding function, which can be performed either per UEs (i.e., per Transport Blocks (TBs)) or else per CBs in parallel.

To be more specific, we use the fact that in LTE, when the size of a TB is too big, it is segmented into shorter data units, referred to as CBs, before being processed by the encoding/decoding sub-function. A CB represents the smallest processing unit, which can be executed in parallel.

It is worth noting that radio signal transmission between a UE and the eNB generates a TB every millisecond. The processing time of a TB depends on the radio channel conditions, the data load per UE and the amount of traffic in the cell. The radio scheduler allocates a number of physical Resource Blocks (RBs) to each UE in function of the volume of traffic in the cell, and determines the Modulation and Coding Scheme (MCS) based on the radio channel quality. Both the Number of Resource Blocks (NRB) and the MCS determine the Transport Block Size (TBS), i.e., the useful data [11].

Figure 1 shows the Cloud-RAN data units (i.e., TBs, CBs, RBs and Resource Elements (REs)) and the proposed parallel execution [7], [8]. In the present work, we consider RBs with normal Cyclic Prefix (CP) (the most common in LTE

networks) [20]. Thus, a RB consists of 12 sub-carriers in the frequency domain and 7 symbols in the time domain. Hence, the smallest defined unit is a RE, which is composed of a single sub-carrier (15KHz) and one symbol.

We use parallel processing in a strict sense so that jobs are simultaneously executed on separate cores, thus avoiding time-sharing processors. In parallel computing, each job runs on a single core and only one at any instant [21]. Conversely, concurrent computing enables the simultaneous execution of jobs on a single core by overlapping time-periods; this leads to processor-sharing PS models [22]. However, the drawback of processor sharing is in that multitasking on the same core requires context switching and memory splitting, which can notably increase the latency.

When performing parallelism per UE, the number of parallel coding jobs is determined by the number of UEs scheduled in a LTE-sub-frame. When applying parallelism per CBs, the number of parallel coding jobs is given by the product of the number of scheduled UEs and the number of CBs for each of them. The number of CBs per UE is given by the larger integer of the TBS divided by the used Code Block Size (CBS), i.e., $N_{CB} = \lceil TBS/(6144-24) \rceil$ [7], where we use the fact that LTE specifies the minimum and maximum code block size equal to 40 and 6144 bits, respectively. The last 24 bits of each CB corresponds to Cyclic Redundancy Check (CRC) [5], [11].

IV. Modeling Principles

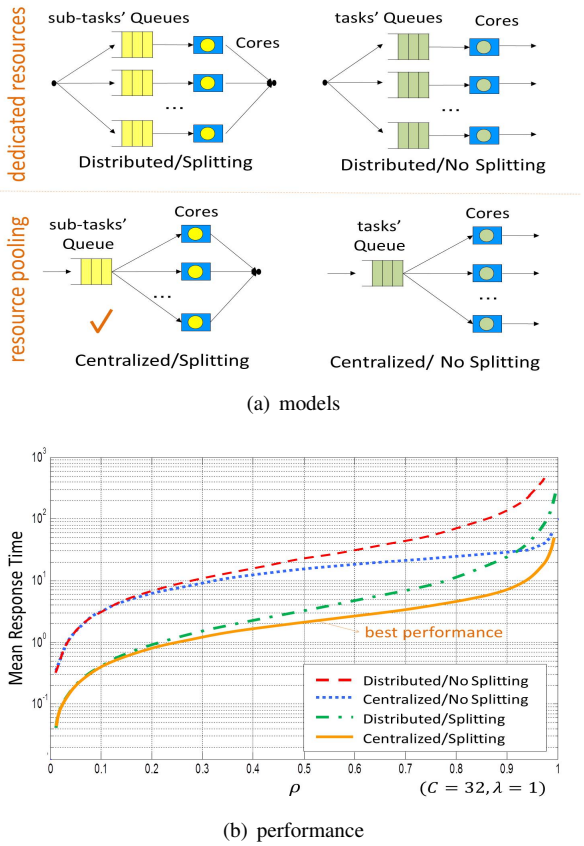
A. Modeling data processing

From a modeling point of view, each antenna (RRH) represents a source of jobs in the up-link direction, while for the down-link direction, jobs arrive from the core network, which provides connection to external networks (e.g., Internet or other service platforms). There are then two queues of jobs for each cellular sector, one in each direction. Since the time-budget for processing down-link sub-frames is half of that for up-link ones, they might be executed separately on dedicated processing units. However, dedicating processors to each queue is not an efficient way of using limited resources.

Nelson et al. in [23] evaluate the performance of different parallel processing models when considering "centralized" (namely, single-queue access on multi-core systems) and "distributed" architectures (namely, multi-queue access on multi-core systems). Parallelism (so-called "splitting") and no-parallelism (so-called, "no splitting") behaviors are also considered. Results show that for any system load (namely, ρ) the lowest (highest) mean job response time is achieved by the "centralized/splitting" ("distributed/no splitting") system, i.e., the best performance in terms of latency (response time) is achieved when processing parallel runnable tasks in a single shared pool of resources. See Figure 3 for an illustration.

In view of the above observations, we propose to use a single-queuing system with a shared pool of processors, namely a multi-core system with C cores. A global scheduler allocates computing resources to each runnable encoding (down-link) or decoding (up-link) job.

We assume that vBBUs (notably, virtual encoding/decoding functions) are invoked according to a Poisson process, i.e.,



(a) models

(b) performance

Fig. 3. Parallel processing systems [23].

inter-arrival times of runnable BBU functions are exponentially distributed. This reasonably captures the fact that there is a sufficiently great number of antennas, which are not synchronized. The occurrence of jobs then results from the superposition of independent point processes. This justifies the Poisson assumption. In practice, frames occur with fixed relative phases. The Poisson assumption is then in some sense a worst case assumption. Job arrivals are not synchronized because RRHs are at different distances of the BBU-pool. Furthermore, when considering no dedicated links, the front-haul delay (inter-arrival time) can strongly vary because of network traffic.

The parallel execution of encoding and decoding tasks on a multi-core system with C cores can be modeled by bulk arrival systems, namely, an $M^{[X]}/G/C$ queuing system[1]. We further consider each task-arrival to be in reality the arrival of B parallel runnable sub-tasks or jobs, B being a random variable. Each sub-task requires a single stage of service with a general time distribution. The runtime of each sub-task depends on the workload as well as on the network sub-function that it implements. The number of parallel runnable sub-tasks belonging to a network sub-function is variable. Thus, we

[1]By analogy, concurrent computing of VNFs can be formalized by a single server queuing system with a processor sharing discipline and batch arrivals, referred to as $M^{[X]}/G/1-PS$, where the single service capacity is done by the addition of individual capacities of all cores in the system. Because switching tasks produces undesirable overhead, this approach is not further considered in the present study.

consider a non fixed-size bulk to arrive at each request arrival instant. The inter-arrival time is exponential with rate λ. The batch size B is independent of the state of the system.

In the case of Cloud-RAN, full functional parallelism is not possible since some base-band procedures (i.e., IFFT, modulation, etc.) require to be executed in series. However, data parallelism of BBU functions (notably decoding and encoding) promises significant performance improvements. These claims are thoroughly studied in [7], [8]. Results show that the runtime of BBU functions can be significantly reduced when performing parallel processing in a sub-frame, i.e., through the parallel execution either of UEs or even of smaller data units, so-called CBs. We present below a stochastic service model for each of the parallelization schemes in order to evaluate the performance of a Cloud-RAN system. See Figure 1 for an illustration.

B. Parallelism by UEs

In LTE, several UEs can be served in a sub-frame of 1 millisecond. The maximum and minimum number of UEs scheduled per sub-frame are determined by the eNB bandwidth. LTE supports scalable bandwidth of 1.25, 2.5, 5, 10 and 20 MHz. In a sub-frame, each scheduled UE receives a TB (namely, a group of radio resources in the form of RB) either for transmission or reception. For example, when considering an eNB of 20 MHz, 100 RBs are available. According to LTE [5], the minimum number of RBs allocated per UE is 6. Hence, the maximum number of connected UEs per sub-frame is given by $b_{max} = \lfloor 100/6 \rfloor$. The TBS is determined by the radio scheduler in function of the individual radio channel conditions of UEs as well as the amount of traffic in the cell.

From the previous section, the parallel base-band processing (notably channel coding) of LTE sub-frames can be modeled as an $M^{[X]}/G/C$ queuing system. When considering parallelization per UE, the number of jobs within a batch corresponds to the number of UEs scheduled in an LTE sub-frame, e.g., the number of decoding jobs per millisecond in an eNB of 20 MHz range from 1 to 16. A sub-frame then comprises a variable number of UEs, which is represented by the random variable B.

We further assume that the processing time of a job (namely that of a TB) is exponential. This assumption is intended to capture the randomness in the processing time of UEs due to non-deterministic behavior of the channel coding function. For instance, the decoding runtime of a single UE can range from a few tens of microseconds to almost the entire time-budget[2], i.e., 2000 microseconds [12]. In practice, this service time encompasses the response time of each component of the cloud computing system, i.e., processing units, RAM memory, internal buses, virtualization engine, data links, etc. In the following, we precisely assume that the service time of a TB (i.e., a job) is exponentially distributed with mean $1/\mu$. If we further suppose that the number B of UEs per sub-frame is geometrically distributed with mean $1/(1-q)$ (that is $\mathbb{P}(B=k) = (1-q)q^{k-1}$ for $k \geq 1$), the complete service time of a frame is then exponentially distributed with mean $1/((1-q)\mu)$.

The geometric distribution as the discrete analogue of the exponential distribution capture the variability of scheduled UEs in a sub-frame. The size B depends on both the number of UEs requiring service in the cell and the radio channel conditions of each of them. In addition, B is strongly related to the radio scheduling strategy (e.g, round robin, proportional fair, etc.). The number of UEs always varies from 1 to b_{max}, where this latter quantity is function of the eNB's bandwidth. In LTE, b_{max} is reached, when users experiment bad radio conditions, i.e., when using a robust modulation as QPSK and a high degree of redundancy. For average radio conditions and non saturated eNBs, it is more probable to have small-sized batches of UEs. The geometric distribution is intended to reflect the mix between radio conditions of UEs and their transmission needs.

With regard to the global Cloud-RAN architecture, the total amount of time t which is required to process BBU functions is given by $t = s + w$, where, s is the job's runtime and w is the waiting time of a job while there are no free processing units. The front-haul delay between RRHs and the BBU-pool is then captured by the arrival distribution. See Figure 4 for an illustration.

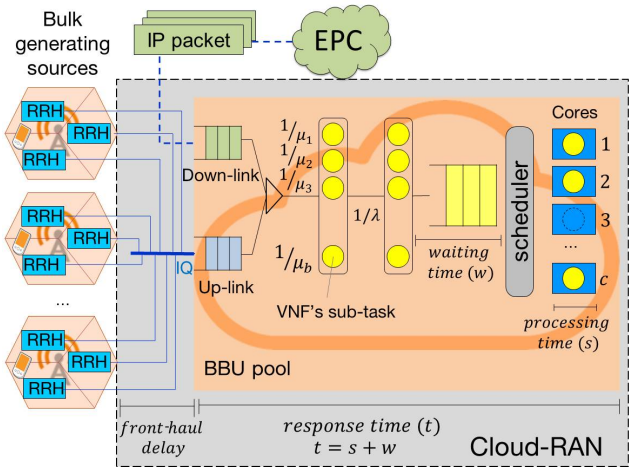


Fig. 4. Stochastic service system for Cloud-RAN.

When assuming that the computing platform has a non-limited buffer, the stability of the system requires:

$$\rho = \frac{\lambda \mathbb{E}[B]}{\mu C} < 1. \quad (1)$$

In the following, we are interested in the sojourn time of sub-frames (batches) in the system, having in mind that if the sojourn time exceeds some threshold (i.e., ≈ 1 millisecond for encoding and ≈ 2 milliseconds for decoding) the sub-frame is lost. If we dimension the system so that the probability for the sojourn time to exceed the threshold is small, we can then approximate the sub-frame loss rate by this probability. It is worth noting that in LTE, retransmissions and reception acknowledgments are handled per sub-frame by the Hybrid

[2]Runtime values are for reference and correspond to the execution of OAI-based channel coding functions on x-86-based General Purpose Processors (GPPs) of 2.6 GHz.

Automatic Repeat-Request (HARQ) process. When a TB is lost the whole sub-frame is re-sent.

C. Parallelism by CBs

In LTE, when a TB is too big, it is split into smaller data units, referred to as CBs. If we assume that the processing time of a CB is exponential with mean $1/\mu'$, we again obtain an $M^{[X]}/M/C$ model, where the batch size is the number of CBs in a TB. If this number is geometrically distributed, the service time of a TB is exponential, as supposed above. The key difference now is that individual CBs are processed in parallel by the C cores. The scheduler is able to allocate a core to each CB owing to the more atomic decomposition of sub-frames and TBs.

D. No parallelism

If the processing of TBs or CBs is not parallel, scheduling is based on sub-frames as presented in [14]. Still assuming a multi-core system, where sub-frames arrive according to a Poisson process, we are led to consider an $M/G/C$ queuing system. By making exponential assumptions for service times of CBs and TBs as well as supposing a geometric number of CBs per TB, we obtain an $M/M/C$ queue, which is well known in the queuing literature [22].

V. Batch model

From the analysis carried out in the previous section, the $M^{[X]}/M/C$ model can reasonably be used to evaluate the processing time of a sub-frame in a Cloud-RAN architecture based on a multi-core platform. While the sojourn time of an arbitrary job of a batch has been analyzed in [24], the sojourn time of a whole batch seems to have received less attention in the technical literature. In this section, we derive the Laplace transform of this last quantity; this eventually allows us to derive an asymptotic estimate of the probability of exceeding a large threshold.

Let us consider an $M^{[X]}/M/C$ queue with batches of size B arriving according to a Poisson process with rate λ. The service time of a job within a batch is exponential with mean $1/\mu$. We assume that the stability condition (1) holds so that a stationary regime exists. The number N of jobs in the system in the stationary regime is such that [24]

$$\phi(z) \stackrel{def}{=} \mathbb{E}\left(z^N\right) = \frac{\sum_{k=0}^{C-1}(C-k)p_k z^k}{C - \frac{\lambda}{\mu} z \left(\frac{1-B(z)}{1-z}\right)},$$

where $p_k = \mathbb{P}(N=k)$ and $B(z)$ is the generating function of the batch size B, i.e., $B(z) = \sum_{k=0}^{\infty} \mathbb{P}(B=k) z^k$. As explained in [24], the probabilities p_k for $k \geq 1$ satisfy the balance equations:

$$p_1 = \frac{\lambda}{\mu} p_0,$$

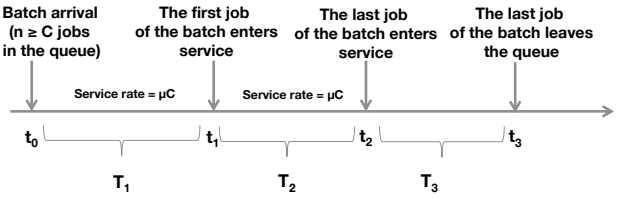
$$p_k = \left(1 + \frac{\lambda - \mu}{k\mu}\right) p_{k-1} - \frac{\lambda}{\mu k} \sum_{\ell=0}^{k-2} p_\ell b_{k-1-\ell} \quad 2 \leq k \leq C,$$

$$p_k = \left(1 + \frac{\lambda}{\mu C}\right) p_{k-1} - \frac{\lambda}{\mu C} \sum_{\ell=0}^{k-2} p_\ell b_{k-1-\ell} \qquad k \geq C,$$

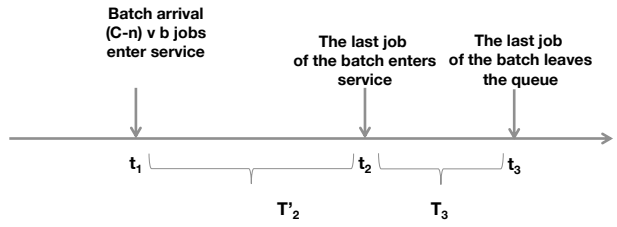
where b_ℓ is the probability that the batch size is equal to ℓ. We see in particular that the probabilities p_k for $k = 2, \ldots, C$ linearly depend on p_0, which can eventually be computed by using the normalizing condition $\sum_{k=0}^{C-1}(C-k)p_k = C(1-\rho)$.

We consider a batch of size b arriving at time t_0 and finding n jobs in the queue. We consider two cases (see Figure 5):

- **Case $n \geq C$:** In that case, the first job of the tagged batch has to wait before entering service.
- **Case $n < C$:** In that case, $b \wedge (C-n) \stackrel{def}{=} \min(b, C-n)$ jobs of the tagged batch immediately enter service; the $0 \vee (b+n-C) \stackrel{def}{=} \max(0, b+n-C)$ jobs have to wait before entering service.



(a) Case 1.



(b) Case 2.

Fig. 5. Two cases upon the arrival of a batch.

A. Analysis of the first case

In the case $n \geq C$, the tagged batch will have to wait for a certain time before the first job enters service. Let t_1 denote the time at which the first job of the tagged batch begins its service. We obviously have that $T_1 = t_1 - t_0$ is equal to the sum of $n-C+1$ independent random variables exponentially distributed with mean $1/(\mu C)$. The Laplace transform of T_1 is defined for $\Re(s) \geq 0$ by

$$\mathbb{E}_b\left(e^{-sT_1}\right) = \left(\frac{\mu C}{s + \mu C}\right)^{n-C+1},$$

where $\mathbb{E}_b$ is the expectation conditionally on the batch size b.

Let t_2 denote the time at which the last job of the batch enters its service. The difference $T_2 = t_2 - t_1$ is clearly the sum of $b-1$ independent exponential random variables with mean $1/(\mu C)$ (the quantity μC being the service rate of the system); the Laplace transform of this difference is

$$\mathbb{E}_b\left(e^{-sT_2}\right) = \left(\frac{\mu C}{s + \mu C}\right)^{b-1}.$$

To completely determine the sojourn time of the tagged batch, it is necessary to know the number y_b of jobs, which belong to this batch and which are in the queue when the

last job of the batch begins its service. Let $t_1 = \tau_1 < \tau_2 < \ldots < \tau_b = t_2$ denote the service completion times of jobs (not necessarily belonging to the tagged batch) in the interval $[t_1, t_2]$. (Note that the point t_1 corresponding to the time at which the first job of the tagged batch enters service is itself a service completion time of one customer present in the queue upon the arrival of the tagged batch.) By definition τ_n is the time at which the n-th job of the tagged batch enters service.

Let us denote by y_n the number of jobs belonging to the tagged batch at time τ_n^+. Then, the sequence (y_n) is a Markov chain for which the conditional transition probabilities can be expressed in terms of Stirling numbers of the second kind $S(n,k)$ [25] defined for $0 \leq k \leq n$ by

$$S(n,k) = \sum_{j=0}^{k} \frac{(-1)^{k-j}}{(k-j)!\, j!} j^n. \tag{2}$$

Stirling numbers are such that $S(n,n) = 1$ for $n \geq 0$, $S(n,1) = 1$ and $S(n,0) = 0$ for $n \geq 1$, and satisfy the recursion for $n \geq 0$ and $k \geq 1$

$$S(n+1,k) = kS(n,k) + S(n,k-1).$$

To formulate the results we alternatively use the polynomials $\mathcal{A}_{n,k}(x)$ defined by means of Stirling numbers as follows:

$$\mathcal{A}_{n,p}(x) = p! \sum_{j=0}^{n} \binom{n}{j} S(j,p) x^{n-j}. \tag{3}$$

The polynomials $\mathcal{A}_{n,p}(x)$ satisfy the recursion for $n, p \geq 0$

$$\mathcal{A}_{n,p}(x) = (x+p)\mathcal{A}_{n-1,p}(x) + p\mathcal{A}_{n-1,p-1}(x)$$

and $\mathcal{A}_{n,p}(0) = p!\, S(n,p)$.

We precisely have the following result.

Lemma 1: The conditional transition probabilities of the Markov chain (y_n) are given for $k \geq \ell$ by

$$\mathbb{P}(y_n = k \mid y_1 = \ell) = \frac{(C-\ell)!}{(C-k)!\, C^{n-1}} \sum_{m=0}^{n-1} \binom{n-1}{m} S(m, k-\ell) \ell^{n-1-m} = \frac{1}{C^{n-1}} \binom{C-\ell}{k-\ell} \mathcal{A}_{n-1,k-\ell}(\ell). \tag{4}$$

From the above Lemma, we deduce the identity

$$\frac{1}{C^{n-1}} \sum_{k=0}^{C} \binom{C-\ell}{k-\ell} k \mathcal{A}_{n-1,k-\ell}(\ell) = C\left(1 - \left(1 - \frac{\ell}{C}\right)\left(1 - \frac{1}{C}\right)^{n-1}\right), \tag{5}$$

where we have used the fact that

$$E(y_n \mid y_1 = \ell) = C\left(1 - \left(1 - \frac{\ell}{C}\right)\left(1 - \frac{1}{C}\right)^{n-1}\right).$$

With the above results, when the b-th job of the tagged batch enters service, there are y_b jobs of this batch in the queue. The time T_3 to serve these jobs is

$$T_3 = \mathcal{E}(y_b \mu) + \mathcal{E}((y_b - 1)\mu) + \ldots + \mathcal{E}(\mu),$$

where $\mathcal{E}(k\mu)$ for $k = 1, \ldots, y_b$ are independent random variables with mean $1/(k\mu)$. The Laplace transform of T_3 knowing y_b is

$$\mathbb{E}_b\left(e^{-sT_3} \mid y_b = k\right) = \frac{k!}{\prod_{\ell=1}^{k}\left(\frac{s}{\mu} + \ell\right)} = \frac{k!}{\left(\frac{s}{\mu} + 1\right)_k}, \tag{6}$$

where $(x)_k$ is the Pochammer symbol (a.k.a. rising factorial) defined by $(x)_k = x(x+1)\ldots(x+k-1)$. By using Lemma 1, it follows that the Laplace transform of the sojourn time T of a batch of size b in the system when there are $n \geq C$ customers in the queue upon arrival is

$$\mathbb{E}_b\left(e^{-sT} \mid N = n \geq C\right) = \frac{C!}{C^b}\left(\frac{\mu C}{s + \mu C}\right)^{n+b-C} \sum_{k=0}^{C} \frac{S(b,k)}{(C-k)!} \frac{k!}{\left(\frac{s}{\mu} + 1\right)_k}, \tag{7}$$

which can be rewritten by using the polynomials $\mathcal{A}_{n,p}(x)$ defined by Equation (3) as

$$\mathbb{E}_b\left(e^{-sT} \mid N = n \geq C\right) = \frac{1}{C^{b-1}}\left(\frac{\mu C}{s + \mu C}\right)^{n+b-C} \sum_{k=0}^{C} \binom{C-1}{k-1} \mathcal{A}_{b,k-1}(1) \frac{1}{\left(\frac{s}{\mu} + 1\right)_k}. \tag{8}$$

B. Analysis of the second case

When the number n of jobs in the queue is less than C upon the arrival of the tagged batch of size b, then $b \wedge (C-n)$ customers immediately begin their service. Let us first assume that $b + n > C$. Taking the tagged batch arrival as time origin, the last job of the tagged batch enters service at random time T_2' with Laplace transform

$$\mathbb{E}(e^{-sT_2'}) = \left(\frac{\mu C}{s + \mu C}\right)^{n+b-C}.$$

The number of jobs of the tagged batch present in the system when the last job enters service is Y_n such that

$$\mathbb{P}(Y_n = k) = \mathbb{P}(y_{b+n-C} = k \mid y_1 = C - n) = \frac{1}{C^{n+b-C-1}} \binom{n}{k+n-C} \mathcal{A}_{n+b-C-1,k+n-C}(C-n)$$

by using Equation (4). For a given value $Y_n = k$, the time T_3 needed to serve all jobs of the tagged batch has Laplace transform given by Equation (6). By using Lemma 1, we conclude that under the assumption $n < C$ and $b + n > C$, the sojourn time T of the tagged batch has Laplace transform

$$\mathbb{E}_b\left(e^{-sT} \mid N = n, b + N > C, N < C\right) = \left(\frac{\mu C}{z + \mu C}\right)^{n+b-C} \sum_{k=C-n}^{C} \mathbb{P}(Y_n = k) \frac{k!}{\left(\frac{s}{\mu} + 1\right)_k} \tag{9}$$

and hence

$$\mathbb{E}_b\left(e^{-sT} \mid N=n<C, b+N>C\right) = \left(\frac{\mu C}{z+\mu C}\right)^{n+b-C} \tau(n,b;s), \quad (10)$$

where

$$\tau(n,b,;s) = \frac{1}{C^{n+b-C-1}} \sum_{k=C-n}^{C} \binom{n}{k+n-C} \times \mathcal{A}_{n+b-C-1,k+n-C}(C-n) \frac{k!}{\left(\frac{s}{\mu}+1\right)_k}. \quad (11)$$

When $b+n \leq C$, all jobs of the tagged batch enter service just after arrival and the Laplace transform of the sojourn time is

$$\mathbb{E}_b\left(e^{-sT} \mid N=n, b+n \leq C\right) = \frac{b!}{\left(\frac{s}{\mu}+1\right)_b}. \quad (12)$$

C. Main result

By using the results of the previous sections, we determine the Laplace transform $\Phi(s) = \mathbb{E}(e^{-sT})$ of the sojourn time of a batch in the $M^{[X]}/M/C$ queue.

Theorem 1: The Laplace transform $\Phi(s)$ is given by

$$\Phi(s) = \beta(s)\left(\phi\left(\frac{\mu C}{s+\mu C}\right) - \phi_C\left(\frac{\mu C}{s+\mu C}\right)\right) + \mathbb{E}\left(\frac{B!}{\left(\frac{s}{\mu}+1\right)_B}\mathbb{P}(N \leq C-B)\right) + \sum_{n=0}^{C-1} p_n \mathbb{E}\left(\tau(n,B;s)\left(\frac{\mu C}{s+\mu C}\right)^{n+B-C}\right), \quad (13)$$

where

$$\beta(s) = \mathbb{E}\left(\frac{1}{C^{B-1}}\left(\frac{\mu C}{s+\mu C}\right)^{B-C} \sum_{k=0}^{C} \binom{C-1}{k-1} \frac{\mathcal{A}_{B,k-1}(1)}{\left(\frac{s}{\mu}+1\right)_k}\right), \quad (14)$$

the function $\phi_C(z)$ is given by

$$\phi_C(z) = \sum_{n=0}^{C-1} p_n z^n,$$

and $\tau(n,b;s)$ defined by Equation (11).

Proof: By conditioning on the batch size b, we have from the two previous sections

$$\mathbb{E}_b(e^{-sT}) = \beta_b(s) \sum_{n=C}^{\infty} p_n \left(\frac{\mu C}{s+\mu C}\right)^n + \sum_{n=0}^{C-1} p_n \tau(n,b;s)\left(\frac{\mu C}{s+\mu C}\right)^{n+b-C} + \frac{b!}{\left(\frac{s}{\mu}+1\right)_b}\mathbb{P}(N \leq C-b)$$

with

$$\beta_b(s) = \frac{C!}{C^b}\left(\frac{\mu C}{s+\mu C}\right)^{b-C} \sum_{k=0}^{C} \frac{S(b,k)}{(C-k)!} \frac{k!}{\left(\frac{s}{\mu}+1\right)_k}$$

and $\tau(n,b;s)$ is defined by Equation (11). Note that we use the fact that $\tau(n,b;s)=0$ if $b<C-n$ in the above equation. By deconditioning on the batch size, Equation (13) follows. ■

Following [24], let us define z_1 the root with smallest module to the equation

$$V(z) \stackrel{def}{=} C - \frac{\lambda}{\mu} z\left(\frac{1-B(z)}{1-z}\right) = 0;$$

the root z_1 is real and greater than 1. The negative real number

$$s_1 = -\mu C\left(1-\frac{1}{z_1}\right)$$

is the singularity with the smallest module of the Laplace transform $\Phi(s)$ if $s_1 > -\mu$ (namely, $z_1 < \frac{C}{C-1}$).

Corollary 1: If $s_1 > -\mu$, then when t tends to infinity

$$\mathbb{P}(T>t) \sim \frac{\mu C U(z_1)\beta(s_1)}{s_1 z_1^2 V'(z_1)} e^{s_1 t}, \quad (15)$$

where $U(z) = \sum_{k=0}^{C-1}(C-k)p_k z^k$. If $s_1 < -\mu$, then the tail of the distribution of T is such that when t tends to infinity

$$\mathbb{P}(T>t) \sim \kappa e^{-\mu t}, \quad (16)$$

where

$$\kappa = \mathbb{E}(B\mathbb{P}(N+B \leq C)) + C\mathbb{E}\left(\left(\frac{C}{C-1}\right)^B - 1\right)\sum_{n=0}^{\infty} p_{C+n}\left(\frac{C}{C-1}\right)^n + \sum_{n=0}^{C-1} p_n C\mathbb{E}\left(\mathbb{1}_{\{B>C-n\}}\left(\left(\frac{C}{C-1}\right)^{n+B-C} - \frac{n}{C-1}\right)\right).$$

Proof: When $s_1 > -\mu$, the root with the smallest module of the Laplace transform $\Phi(s)$ is s_1 and the estimate (15) immediately follows by using standard results for Laplace transforms [26].

When $s_1 < -\mu$, the root with smallest module is $-\mu$. We

have for s is the neighborhood of $-\mu$,

$$\beta(s) \sim \frac{\mu}{\mu+s} \times \mathbb{E}\left(\frac{1}{C^{B-1}}\left(\frac{\mu C}{C-1}\right)^{B-C}\sum_{k=0}^{C}\binom{C-1}{k-1}k\mathcal{A}_{B,k-1}(1)\right) = \frac{\mu}{\mu+s}C\mathbb{E}\left(\left(\frac{C}{C-1}\right)^{B}-1\right)\sum_{n=0}^{\infty}p_{C+n}\left(\frac{C}{C-1}\right)^{n},$$

where we have used Equation (5) for $\ell = 1$.

In addition, under the same conditions,

$$\mathbb{E}\left(\frac{B!}{\left(\frac{s}{\mu}+1\right)_B}\mathbb{P}(N \le C-B)\right) \sim \frac{\mu}{\mu+s}\mathbb{E}(B\mathbb{P}(N+B \le C))$$

and

$$\sum_{n=0}^{C-1}p_n\mathbb{E}\left(\tau(n,B;s)\left(\frac{\mu C}{s+\mu C}\right)^{n+B-C}\right) \sim \frac{\mu}{\mu+s}\sum_{n=0}^{C-1}p_n\mathbb{E}\left(\left(\frac{C}{C-1}\right)^{n+B-C}\frac{1}{C^{n+B-C-1}} \times \sum_{k=C-n}^{C}\binom{n}{k+n-C}k\mathcal{A}_{n+B-C-1,k+n-C}(C-n)\right) = \frac{\mu}{\mu+s}\sum_{n=0}^{C-1}p_nC \times \mathbb{E}\left(\mathbb{1}_{\{B>C-n\}}\left(\left(\frac{C}{C-1}\right)^{n+B-C}-\frac{n}{C-1}\right)\right),$$

where we have used Equation (5) for $\ell = C-n$. Gathering the above residue calculations yields Equation (16). ∎

Corollary 1 states that when the service capacity of the system is sufficiently large, the tail of the sojourn time of a batch is dominated by the service time of a single job. It is also worth noting that contrary to what is stated in [24], the same result holds for the decay rate of the sojourn time of a job in the system. Finally, when C is large and for moderate values of the load and the mean batch size, $\kappa \sim \mathbb{E}(B\mathbb{P}(N+B \le C)) \sim \mathbb{E}(B)$. This means that there is roughly the multiplicative factor $\mathbb{E}(B)$ between the tail of the sojourn time of a batch and that of a job.

When the batch size is geometrically distributed with mean $1/(1-q)$, we have $s_1 = -(1-q)\mu C(1-\rho)$ and

$$z_1 = \frac{C}{qC+\frac{\lambda}{\mu}} > 1 \text{ for } C > \frac{\lambda}{(1-q)\mu}. \tag{17}$$

We have $z_1 < \frac{C}{C-1}$ if and only if $\rho > 1 - \frac{1}{C(1-q)}$.

VI. Numerical experiments

In this section, we evaluate by simulation the behavior of a Cloud-RAN system hosting the base band processing of a hundred base-stations. The goal is to test the relevance of the $M^{[X]}/M/C$ model for sizing purposes and to derive dimensioning rules. C-RAN sizing refers to determining the minimum number of servers (cores), which are required to ensure the processing of LTE sub-frames within deadlines for a given number of base stations (eNBs), as well as the maximum front-haul distance between antennas and the BBU-pool.

In LTE, deadlines are applied to the whole sub-frame. For instance, when the runtime of the base-band processing of a sub-frame in the up-link direction exceeds 2 milliseconds, the whole sub-frame is lost and therefore retransmitted. In order to bring new perspectives for the radio channel efficiency, we also evaluate the loss of single users, so that RAN systems might hold less redundant data. The loss of sub-frames as well as UEs are captured in the $M^{[X]}/M/C$ model by the impatience of batches and customers, respectively.

A. Simulation settings

We evaluate a C-RAN system hosting 100 eNBs where each of them has a bandwidth of 20 MHz. All eNBs have a single antenna (i.e., work under Single Input Single Output (SISO) configuration) and use Frequency Division Duplex (FDD) transmission mode. Antennas (eNBs) are distributed around the computing center within a 100 km radius.

In the following, we focus our analysis on the decoding and encoding functions carried out during up-link and down-link processing, respectively, due to their non-deterministic behavior, as well as, because they are the greatest computing resource consumer of all BBU functions [8], [12]. To assess the runtime of decoding and encoding functions, we use OAI's code, which implements RAN functions in open-source software [12].

B. Model analysis

In order to represent the behavior of a Cloud-RAN system by using the $M^{[X]}/M/C$ model, we feed the queuing system with statistical parameters captured from the C-RAN emulation during the busy-hour; see Figure 6. We capture the behavior of the decoding function in a multi-core system performing parallelism by UEs. The obtained parameters are as follows:

- The mean service time of decoding jobs, $\mathbb{E}[S]$, is equal to 281 microseconds. Each decoding job corresponds to the data of a single UE.
- The mean number of decoding jobs requiring service at the same time, i.e, the mean batch size, is given by $\mathbb{E}[B] = 5$. As explained in Section III, the number of UEs scheduled per sub-frame can vary between 1 and 16 for an eNB of 20 MHz. This can be approximated by a geometric distribution with parameter $q = 0.8$ ($q = 1 - \frac{1}{\mathbb{E}[B]}$). Batch-sizes are in the interval $[1, 16]$ with probability equal to 0.97. Figure 6 gives the percentage of each of the sub-frame types in the system.
- The mean inter-arrival time of batches is 10 microseconds. Each eNB generates a bulk of decoding jobs (sub-frame) every millisecond. The mean inter-arrival time is computed by dividing the periodicity of sub-frames by the number of eNBs.

- The time-budget (deadline) for the up-link processing is given by $\delta = 2000$ microseconds.

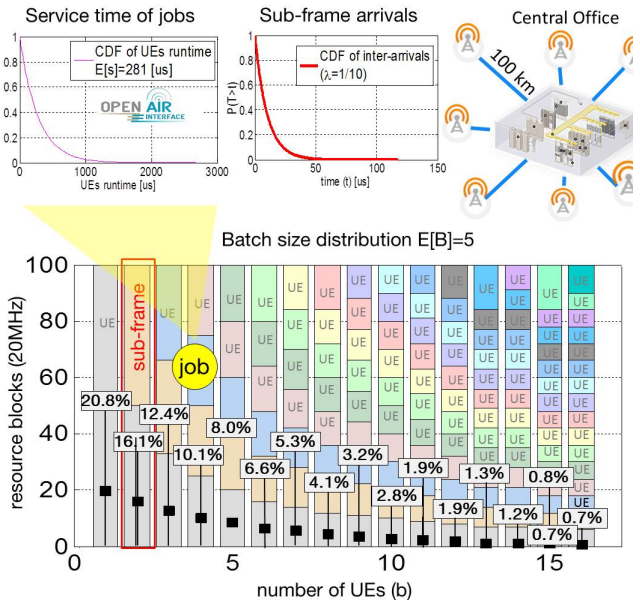


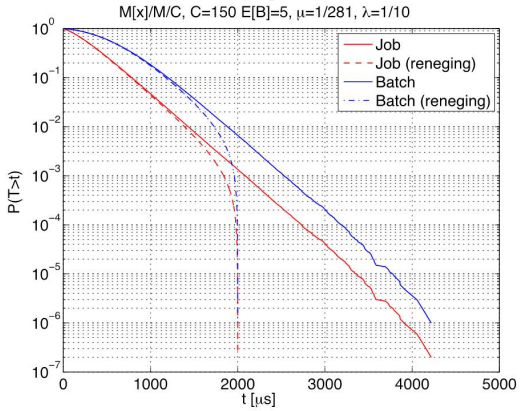
Fig. 6. Statistical parameters of Cloud-RAN.

We can then evaluate the $M^{[X]}/M/C$ model with the following parameters: $\mu = 1/281$ and $\lambda = 1/10$. By Equation (1) for $C = 150$, the load is $\rho = 0.9367$. The CDFs of the sojourn time of jobs and batches are shown in Figure 7(a). By using Corollary 1, we verify that if D is the sojourn time of a job in the $M^{[X]}/M/C$ queue, then $\mathbb{P}(T > t)/\mathbb{P}(D > t)$ tends to a constant when $t \to \infty$. It can also be checked that the slopes of the curves $-\log(P(D > t))/t$ and $-\log(P(T > t))/t$ for large t are both equal to μ.

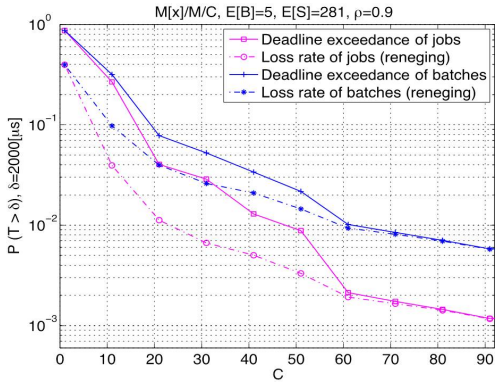
In practice, aborting the execution of sub-frames, which overtake deadlines, is highly desirable to save computing resources. We are then interested in the behavior of the $M^{[X]}/M/C$ with reneging of both customers and batches. A job (customer) leaves the system (even during service) when its sojourn time reaches a given deadline δ. In the case of reneging of batches, the sojourn time of a batch is calculated from the arrival until the instant at which the last job composing the batch is served. Results with impatient customers and batches are depicted in Figure 7(a).

With impatience, the loss rate of jobs and batches, is respectively 0.0013 and 0.0065. We observe that the gap between the two rates (i.e., $0.0065/0.0013$) is close to the mean batch size $\mathbb{E}[B]$. This is true when loss rates are at least of order 10^{-3}.

Due to the complexity of the theoretical analysis of impatience-based models, we choose to use the performance of an $M^{[X]}/M/C$ system without reneging for sizing a Cloud-RAN infrastructure. Since this model stochastically dominates the system with reneging, we obtain conservative bounds. As illustrated in Figure 7(b), we verify for both jobs and batches that the probability of deadline exceedance is always greater in a system without reneging, and moreover, these two probabilities are close to each other when C increases.



(a) Sojourn time of jobs and batches.



(b) Deadline exceedance of jobs and batches.

Fig. 7. M[x]/M/C behavior.

C. Cloud-RAN dimensioning

The final goal of Cloud-RAN sizing is to determine the amount of computing resources needed in the cloud (or a data center) to guarantee the base-band processing of a given number of eNBs within deadlines. For this purpose, we evaluate the $M^{[X]}/M/C$ model (without reneging) while increasing C, until an acceptable probability of deadline exceedance (say, ε). The required number of cores is then the first value that achieves $P(T > \delta) < \varepsilon$.

We validate by simulation the effectiveness of the $M^{[X]}/M/C$ model with the behavior of the real C-RAN system during the reception process (up-link) of LTE sub-frames. See Figure 8 for an illustration. Results show that for a given $\varepsilon = 0.00615$, the required number of cores is $C_r = 151$, which is in accordance with the real C-RAN performance, where the probability of deadline exceedance is barely 0.00018.

When C takes values lower than a certain threshold C_s, the C-RAN system is overloaded, i.e., the number of cores is not sufficient to process the vBBUs' workload; the system is then unstable. The threshold C_s can be easily obtained from Equation (1); for $\rho = 1$, $C_s = \lceil \lambda * E[B]/\mu \rceil = 141$ cores.

D. Performance Analysis

We are now interested in the performance of the whole Cloud-RAN system running in a data center equipped with

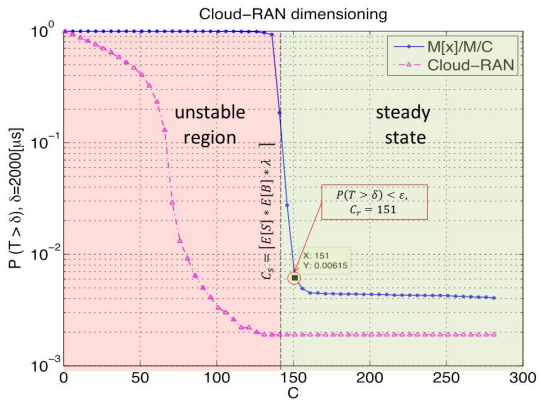
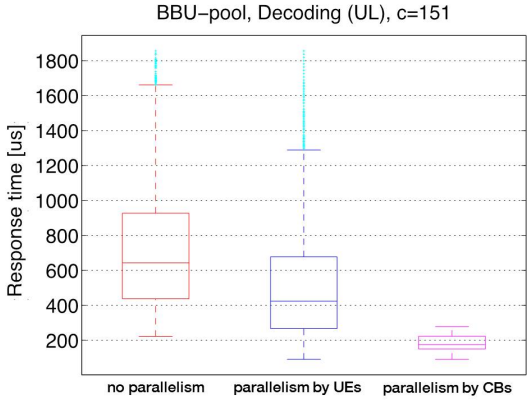
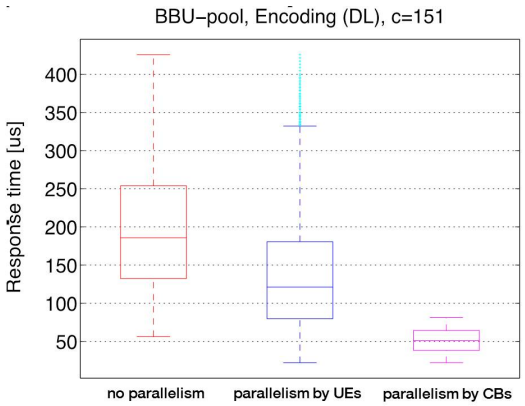


Fig. 8. C-RAN sizing when using the $M^{[X]}/M/C$ model.

151 cores. The system processes both up-link and down-link sub-frames belonging to 100 eNBs. Results show an important gain when performing parallelism per CBs during both reception (see Figure 9(a)) and transmission (see Figure 9(b)) processing.



(a) Up-link (Rx).



(b) Down-link (Tx).

Fig. 9. Cloud-RAN performance in terms of latency.

It is observed that more than 99% of sub-frames are processed within 472 microseconds and 1490 microseconds when performing parallelism by CBs and UEs, respectively. It represents a gain of 1130 microseconds (CB) and 100 microseconds (UE) with respect to the original system (non-parallelism). These gains in the sojourn time enable the operator to increase the maximum distance between antennas and the central office. Hence, when considering the light-speed in the optic-fiber, i.e., $2.25 * 10^8$m/s, the distance can be increased up to ≈ 250 km when running CBs in parallel. Figure 10 shows the CDF of the sojourn time of LTE sub-frames when performing parallel programing.

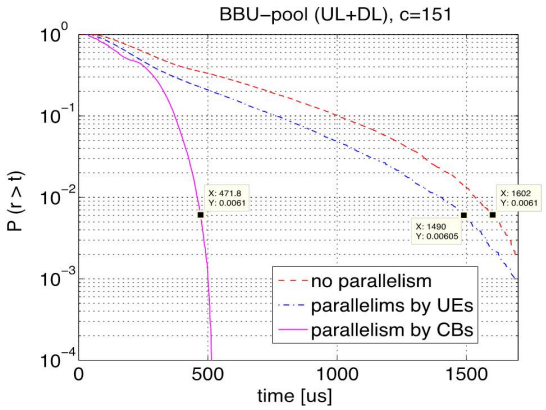


Fig. 10. CDF of the sojourn time of LTE-sub-frames.

VII. Conclusion

We have studied in this paper the performance of virtualized base-band functions when using parallel processing. We have concretely evaluated the processing time of LTE sub-frames in a C-RAN system. In order to reduce the latency, we have investigated the functional and data decomposition of BBU functions, which leads to batch arrivals of parallel runnable jobs with non-deterministic runtime. For assessing the required processing capacity to support a C-RAN system, we have introduced a bulk arrival queuing model, namely the $M^{[X]}/M/C$ queuing system, where the batch size follows a geometric distribution. The variability of the front-haul delay and jobs' runtime are captured by the arrival and service distributions, respectively. Since, the runtime of a LTE sub-frame becomes the batch sojourn-time, we have derived the Laplace transform of this latter quantity as well as the probability of exceeding certain threshold to respect LTE deadlines.

We have validated the model by simulation, when performing a C-RAN system with one hundred eNBs of 20 MHz during the busy-hour. We have additionally illustrated that the impatience criterion reflecting LTE time-budgets is not incident when the probability of deadline exceedance is low enough. Finally, once the C-RAN system is dimensioned, we have evaluated its performance when processing both up-link and down-link sub-frames. Results show an important gain in terms of latency when performing parallel processing of LTE sub-frames and the approach based on the batch model to sizing C-RAN systems.

List of Acronyms

BBU Base Band Unit. 1–3, 5

CB	Code Block. 3–6, 11
CBS	Code Block Size. 4
CO	Central Office. 1–3
COTS	Commercial off-the-shelf. 1
CP	Cyclic Prefix. 3
CRC	Cyclic Redundancy Check. 4
EDF	Earliest Deadline First. 2
eNB	eNodeB. 1–3, 5, 9–11
FDD	Frequency Division Duplex. 9
GPP	General Purpose Processor. 5
HARQ	Hybrid Automatic Repeat-Request. 5
IoT	Internet of Things. 2
KVM	Kernel-based Virtual Machine. 2
LTE	Long Term Evolution. 1
LXC	Linux Containers. 2
MCS	Modulation and Coding Scheme. 3
NFV	Network Function Virtualization. 1
NRB	Number of Resource Blocks. 3
OAI	Open Air Interface. 1, 2, 9
OS	Operating System. 3
PDCP	Packet Data Convergence Protocol. 2
PS	Processor Sharing. 2, 4
QoE	Quality of Experience. 2
QoS	Quality of Service. 2
RAN	Radio Access Network. 1
RAT	Radio Access Technologie. 2
RB	Resource Block. 3–5
RE	Resource Element. 3, 4
RLC	Radio Link Control. 2
RRH	Radio Remote Head. 2, 4, 5
SISO	Single Input Single Output. 9
SLA	Service Level Agreement. 2
TB	Transport Block. 3, 5, 6
TBS	Transport Block Size. 3–5
UE	User Equipment. 1–5, 9, 11
vBBU	virtualized BBU. 2–4
VM	Virtual Machine. 1, 3
VNF	Virtualized Network Function. 1, 2, 4
VNO	Virtual mobile Network Operator. 2
VRRM	Virtual Radio Resource Management. 2

REFERENCES

[1] GSNFV ETSI. 001:network functions virtualisation (nfv). *Architectural Framework*, 2013.

[2] Veronica Quintuna-Rodriguez and Fabrice Guillemin. On dimensioning cloud-ran systems. In *ValueTools, 11th EAI International Conference on Performance Evaluation Methodologies and Tools*. EAI, 2017.

[3] Ericsson Telefonica. Cloud-ran architecture for 5g. *White Paper*, 2015.

[4] Ericsson. Cloud-ran. *White Paper*, 2015.

[5] LTE, evolved universal terrestrial radio access, physical layer procedures (3GPP TS 36.213 version 12.4.0 release 12). Standard, European Telecommunications Standards Institute, 2015.

[6] Navid Nikaein. Processing radio access network functions in the cloud: Critical issues and modeling. In *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, pages 36–43. ACM, 2015.

[7] Veronica Quintuna Rodriguez and Fabrice Guillemin. Vnf modeling towards the cloud-ran implementation. In *Networked Systems (NetSys), 2017 International Conference on*, pages 1–8. IEEE, 2017.

[8] Veronica Quintuna Rodriguez and Fabrice Guillemin. Towards the deployment of a fully centralized Cloud-RAN architecture. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*, pages 1055–1060. IEEE, 2017.

[9] China Mobile Research Institute. C-RAN, the road towards green RAN. *White Paper*, 2011.

[10] Gabriel Brown. Cloud-RAN, the next-generation mobile network architecture. *White Paper*, 2017.

[11] Veronica Quintuna Rodriguez and Fabrice Guillemin. Performance analysis of VNFs for sizing cloud-RAN infrastructures. In *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on*, pages 1–6. IEEE, 2017.

[12] Navid Nikaein, Mahesh K Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. OpenAirInterface: A flexible platform for 5G research. *ACM SIGCOMM Computer Communication Review*, 44(5):33–38, 2014.

[13] Amarisoft. V-RAN lte, 2018.

[14] Sourjya Bhaumik et al. CloudIQ: A framework for processing base stations in a data center. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 125–136. ACM, 2012.

[15] Veronica Karina Quintuna Rodriguez and Fabrice Guillemin. Performance analysis of resource pooling for network function virtualization. In *Telecommunications Network Strategy and Planning Symposium (Networks), 2016 17th International*, pages 158–163. IEEE, 2016.

[16] Navid Nikaein, Eryk Schiller, Romain Favraud, Kostas Katsalis, Donatos Stavropoulos, Islam Alyafawi, Zhongliang Zhao, Torsten Braun, and Thanasis Korakis. Network store: Exploring slicing in future 5G networks. In *Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture*, pages 8–13. ACM, 2015.

[17] Sina Khatibi, Luísa Caeiro, Lúcio S Ferreira, Luis M Correia, and Navid Nikaein. Modelling and implementation of virtual radio resources management for 5G Cloud RAN. *EURASIP Journal on Wireless Communications and Networking*, 2017(1):128, 2017.

[18] Erik Dahlman, Stefan Parkvall, and Johan Skold. *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013.

[19] Rabindra K Barik, Rakesh K Lenka, K Rahul Rao, and Devam Ghose. Performance analysis of virtual machines and containers in cloud computing. In *Computing, Communication and Automation (ICCCA), 2016 International Conference on*, pages 1204–1210. IEEE, 2016.

[20] Syed Faraz Hasan. *Emerging trends in communication networks*. Springer, 2014.

[21] R Pike and A Gerrand. Concurrency is not parallelism. *Heroku Waza*, 2012.

[22] Leonard Kleinrock. *Queueing Systems*, volume II: Computer Applications. Wiley Interscience, 1976.

[23] Randolph Nelson, Don Towsley, and Asser N. Tantawi. Performance analysis of parallel processing systems. *IEEE Transactions on software engineering*, 14(4):532–540, 1988.

[24] M.V. Cromie, M.L. Chaudhry, and W.K. Grassman. Further results for the queueing systems $M^X/M/c$. *J. Opl Res. Soc.*, 30(8):755–763, 1979.

[25] John Riordan. *Combinatorial identities*. Wiley series in probability and mathematical statistics. Wiley, New York, 1968.

[26] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, January 1968.

Veronica Quintuna-Rodriguez received the Engineering degree in electronics from the Politecnica Salesiana University, Ecuador, in 2010. She received the M.Sc. degree in Telecommunications Systems Engineering from Telecom Bretagne, France, in 2015. From 2012 to 2013, she was member of the National Agency of Regulation & Control of Telecommunications, Ecuador. She is currently PhD Candidate at the University Pierre et Marie Curie, working as part of the French Institute for Research in Computer Science and Applied mathematics (Inria), Paris. She works in the field of Network Function Virtualization (NFV) at Orange Labs, France, since 2015.

Fabrice Guillemin graduated from Ecole Polytechnique in 1984 and from Telecom Paris in 1989. He received the PhD degree from the University of Rennes in 1992. He defended his habilitation thesis in 1999 at the University Pierre et Marie Curie (LIP6), Paris. Since 1989, he has been with Orange Labs (former CNET and France Telecom R&D). He has been involved in the standardization of ATM and IP traffic metrology. He is currently leading a project on the evolution of network control. He is a member of the Orange Expert community.

View publication stats