A Quasi-Birth-Death model for Functional Split in 5G Controllers

Luis Diez, Cristina Hervella, Ramón Agüero
Communications Engineering Department
University of Cantabria
{ldiez, ramon}@tlmat.unican.es, cristina-aurora.hervella@alumnos.unican.es

***Abstract*—It is broadly accepted that network function virtualization will play a key role to meet the stringent and heterogeneous requirements of 5G networks. Although fully centralized approaches were initially proposed, they may impose unfeasible requirements over fronthaul links. Consequently, flexible functional split solutions are being fostered, where a central controller adapts the centralization level to current circumstances. In spite of the growing interest in this type of solutions, most of existing works focus on real implementation, while little attention has been paid so far to performance modeling. In this paper we propose a Markov Chain based controller model, which boils down to a Quasi-Birth-Death process. Under reasonable assumptions, this model provides expected values of buffer occupancy and the time frames would spend in the controller. In this sense, it aims to be a tool to support the allocation of computational resources of the virtualized entities. We validate the proposed model by comparing its results with those obtained by simulation, evincing an almost perfect match between both approaches.**

***Index Terms*—Functional split, 5G, Markov Chain, Quasi-Birth-Death**

I. Introduction

In the last years, communication networks are undergoing a profound transformation, to bring the capabilities promised by 5G technology. It is believed that 5G communications will span to all domains, fostering the development of highly heterogeneous services. To this end, future networks will offer increased communication capacity to the enhanced Mobile Broadband (eMBB), allowing a large number of connected devices to enable massive Machine Type Communications (mMTC), and providing Ultra Reliable Low Latency Communications (URLLC) for critical services. To enable such capabilities, improvements are required both in the radio technology and at the network architecture.

In this sense, one of the architectural evolutions that will characterize 5G networks comes from the centralization of network functions. This is achieved by exploiting Software Defined Networking (SDN) techniques, leading to the so-called Network Function Virtualization (NFV) paradigm, where some traditional Base Station (BS) functions are virtualized and centralized, while the remaining ones stay closer to the antenna. This way, the traditional BS is split into a Central Unit (CU), placed at the controller, which manages a Distributed Unit (DU).

Initially, fully centralized architectures, Cloud RAN (C-RAN) [1], [2], were proposed where the DU, known as Remote Radio Head (RRH), performs only basic Radio Frequency (RF) functions. However, this approach demands high communication fronthaul capacities between the DU and CU, which may not be feasible [3]. In order to solve this limitation, academia, industry, and standardization bodies [4], [5] are working together to define solutions that allow the selection of different splits in the base station [6], leading to the so-called functional split.

When virtualizing network functions, one key decision that needs to be taken is the functional split to be used, that is to say, the particular functions that are placed in the distributed and central units. Recently, an increasing number of works propose flexible functional split solutions [7], [8] where functions are dynamically shifted. In this case, a central controller selects the virtualization level of one or many DUs, according to service requirements. In addition, the controller also needs to manage virtualization resources, such as processing or memory, so that the chosen splits are manageable.

In this work we propose a controller model based on Markov Chain theory, which brings expected performance in terms of delay and buffer occupancy in the DUs and CUs. It is worth noting that the proposed model does not aim to provide a split selection algorithm, but to predict the controller performance once a particular policy is applied. This way, the model presented in this work can be afterwards used as a tool to dimension controller capabilities. To the best of our knowledge, this is the first work that proposes a model to analyze controller performance in flexible functional split networks. We assess its validity by means of an extensive simulation campaign, which was carried out over a proprietary event-driven simulator, which was also exploited to broaden the analysis. The code developed for the model validation has been made available in a public repository[1] to ease results replication.

The rest of the paper is structured as follows: in Section II we discuss related research, and we point out the main differences with the work we present herewith. Section III depicts the proposed controller model, which is based on Markov Chain theory and Quasi-Birth-Death processes. We discuss how performance parameters can be analytically obtained. In Section IV we validate the model, by means of an extensive experiment campaign over an event-driven simulator. Finally,

[1] https://github.com/ldiez/5GvRanController_QBD

we conclude the paper in Section V, which also provides an outlook of our future work.

II. Related work

Several studies have looked at optimizing the performance of the fronthaul network by assuming functional split. Most of these works propose novel routing solutions, which seek to minimize the delay. Worthy of mention is [9] where the worst case delay is minimized or [10], which employs machine learning techniques to minimize the fronthaul delay. However, these proposals do not consider dynamic split selection.

On the other hand, some studies aim to optimize the split selection policy in a dynamic fashion. In this regard, Martinez Alba and Kellerer [11] analyzed the convergence time requirements of split selection algorithms, thus setting the base to benchmark different techniques. Other works have proposed split selection algorithms in a variety of scenarios. For instance, jointly optimization of split selection and content caching is addressed in [12], while energy efficiency along with split selection, in a scenario with Unmanned Aerial Vehicles (UAVs), is considered in [13]. Other works vary in the constraints that are assumed, like energy [14], [15] or delay [16].

Other group of works foster a more holistic solution, putting together routing and split selection. In [17], Abdullaziz *et al.* propose a framework that integrates generic heuristic solutions to optimize energy-efficient flow routing, allowing reallocation of virtual functions, and a particular heuristic is proposed in [18]. The joint optimization of split selection, routing and Mobile Edge Computing (MEC) services is proposed in [19], while the number of active nodes is also considered in [20].

Although these works are related to the one presented here, the scope is rather different. The aforementioned proposals would yield split selection policies to be implemented by a central controller. On the other hand, as mentioned earlier, we propose a model that captures the controller performance upon a particular policy.

In this sense, a few works have addressed the performance analysis of split selection from a practical perspective. In [21] a flexible functional split implementation road-map is presented, describing the guidelines to follow. Others works, such as [22] or [23], have implemented functional split selection solutions and analyzed their performance. In the same way, the impact of split on the fronthaul capacity using different packetization [24] and scheduling options [25] has been studied using open Software Defined Radio (SDR) solutions.

The implementations and analyses yielded by these works might be useful to validate the goodness of our model. Similarly, some practical metrics could be also used to realistically configure it.

III. CU Controller Model

We consider a controller with a single CU equipped with a given amount of computation and memory resources. The CU has S different possible splits, $1 \dots s$, each of them having a service rate μ_k (ms^{-1}) which would depend on the packets length and computation resources. Frames arrive at a rate λ (ms^{-1}), and we also assume that the CU has enough memory capacity to keep frames waiting before they can be served, so (provided that the system operates at a stable regime) all incoming frames will be eventually served. In addition, First-Come First-Served (FCFS) queue policy is assume for the buffer.

The controller implements a particular split selection policy or algorithm. In order to model changes on the functional split used by the controller, it is assumed that at a given rate, γ (ms^{-1}), the CU goes into a stand-by situation. Frames could still arrive, and the CU would not serve them, but otherwise keep them within the buffer. From such stand-by operation, the CU would shift to another split configuration, at a rate ξ (ms^{-1}). We assume that whenever the CU leaves the stand-by operation, the k^{th} split configuration is selected with a probability α_k, and that $\sum_{k=1\dots s} \alpha_k = 1$.

We define the state the CU is currently operating as (i, j), where i is the current number of frames at the controller, either at the processor or at the waiting buffer, and j is the current split. If j equals 0, the CU would be at the stand-by configuration. Based on these states we build the Markov Chain that we describe below.

A. CU Markov Chain

If we assume that all service rates are exponentially distributed, and that the arrival process can be considered as Poisson, we can use the 2 dimensional *Markov* chain depicted in Figure 1 to model the behavior of the CU. As can be seen, each row corresponds to a particular split. In this sense, when a frame arrives, it increases the state rightward, and when a frame is served (it exits the CU), there is 1-state transition (leftward). Hence, if the split does not change, all transitions occur within the same row.

It is assumed that whenever the CU changes its current split configuration, it first goes to the standby operation, which is captured by the lower row in Figure 1. As can be observed, if the CU is at any state (i, j), with $j = 1 \dots s$, it can go to state $i, 0$, at a rate γ (ms^{-1}). Hence, we model the time the CU stays at a particular split configuration with an exponential random variable, whose average value equals $\frac{1}{\gamma}$ ms. Note that we have used the same value for all splits, but this could be easily changed to capture a different operation. Once the CU is in such stand-by state, no frames can be served and so state transitions just occur rightward, whenever a new frame arrives, as can be seen in Figure 1.

The time the CU stays in the stand-by situation is also modeled by means of an exponential random variable, with mean $\frac{1}{\xi}$ ms. As mentioned earlier, the split configuration is randomly selected (it goes to split j with probability α_j), and so the rate from $(i, 0)$ to (i, j) equals $\xi \cdot \alpha_j$ (ms^{-1}).

The defined model corresponds to a Quasi-Birth-Death (QBD) process. The reader can refer to the seminal work of Neuts [26], or complete books by Neuts himself [27] or Latouche and Ramaswami [28], for a more thorough discussion of the corresponding theoretical framework, in particular

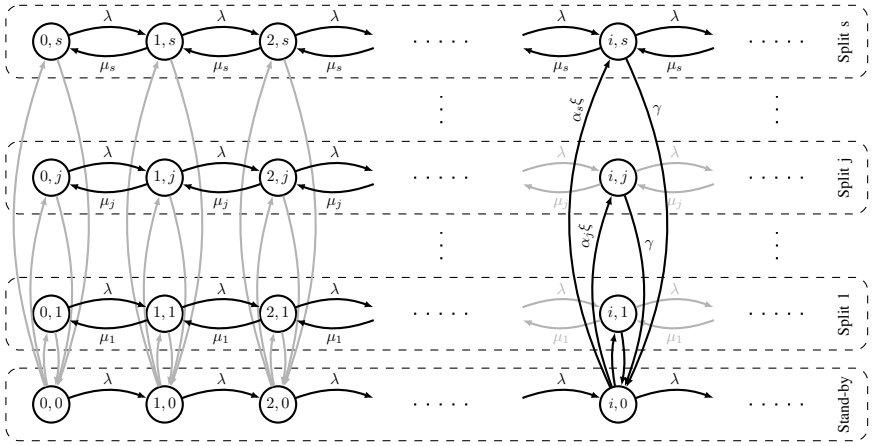


Fig. 1: Markov Chain to model the operation of the Base-Band Unit (BBU)

the so-called Matrix Geometric Method, which we will use hereinafter to analyze the behavior of the CU.

B. State Probabilities and Average CU Time

We define $\mathcal{Q}$ as the infinitesimal generator matrix of the corresponding QBD Process:

$$\mathcal{Q} = \begin{bmatrix} L_0 & F & 0 & 0 & \cdots \\ B & L & F & 0 & \cdots \\ 0 & B & L & F & \cdots \\ \vdots & & \ddots & & \ddots \end{bmatrix} \tag{1}$$

Where each element corresponds to a $(s+1) \times (s+1)$ matrix, which are given in (2). F is a diagonal matrix with all elements in its diagonal being λ, while B is also a diagonal matrix (but its first element, which is zero), with values (from the second one onward) $\mu_1 \dots \mu_s$. L_0 can be straightforwardly calculated as the sum of L and B.

We define the stationary distribution of the system as: $\Pi = [\pi_0, \pi_1, \dots]$, where each π_i is a column vector of length $s+1$, so that $\pi_i(j)$ is the probability of having i packets in the CU, configured in the j^{th} split (if $j = 0$, the CU would be on stand-by).

Provided the system has a stationary solution (stable operation regime), there exists a constant matrix R so that [27, Theorem 3.1.1]:

$$R^2 \cdot B + R \cdot L + F = 0 \tag{3}$$

In addition, there is a unique positive solution to the finite system of equations:

$$\begin{aligned} \pi_0^\intercal \left(L_0 + RB\right) &= \mathbf{0}^\intercal \\ \pi_0^\intercal \left(I - R\right)^{-1} \mathbf{1} &= 1 \end{aligned} \tag{4}$$

where $\mathbf{0}$ and $\mathbf{1}$ are column vectors of appropriate length $(s+1)$, with all their elements 0 and 1, respectively.

Then, $\mathbf{\Pi} = [\pi_0, \pi_1, \dots]$ is given by:

$$\pi_i^\intercal = \pi_0^\intercal \cdot R^i \tag{5}$$

Since there is not a straightforward closed solution for the quadratic equation in (3), an iterative method can be used to find R[2].

Once we have the stationary probability distribution, we can easily obtain the average number of frames in the CU:

$$\overline{N} = \left\| \frac{\pi_1}{(I-R)^2} \right\|_1 = \left\| \frac{\pi_0^\intercal \cdot R}{(I-R)^2} \right\|_1 \tag{6}$$

Then, applying Little's Law (λ is constant) we can finally establish the average time a frame stays at the CU, both waiting and at the processor:

$$\overline{T} = \frac{\overline{N}}{\lambda} \tag{7}$$

Based $\mathbf{\Pi}$, we could also find the average waiting or processing times, but this is left out due to lack of space.

C. Stability condition

The QBD process has a stationary solution if and only if:

$$\eta B \mathbf{1} > \eta F \mathbf{1} \tag{8}$$

where η is the stationary probability vector of matrix $A = B + L + F$, and $\mathbf{1}$ is a column vector, of length $s+1$.

By solving $\eta A = 0$, we obtain that: $\eta_0 = \frac{\gamma}{\gamma+\xi}$ and $\eta_j = \frac{\alpha_j \xi}{\gamma+\xi}, \forall j = 1 \dots s$. Then, by substituting into (8), we can finally establish the stability condition of the CU.

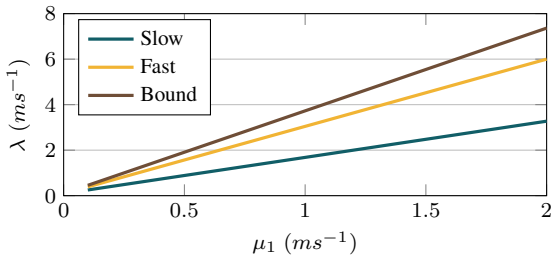
$$\lambda < \frac{\gamma + \xi \sum_{k=1}^{s} \alpha_k \cdot \mu_k}{\gamma + \xi} \tag{9}$$

In some cases, it is not possible knowing split probabilities[3], so it might be handy having a bound for (9). Since $\sum_{k=1}^{s} \alpha_k \cdot \mu_k \leq \max_k \mu_k$, we can establish that:

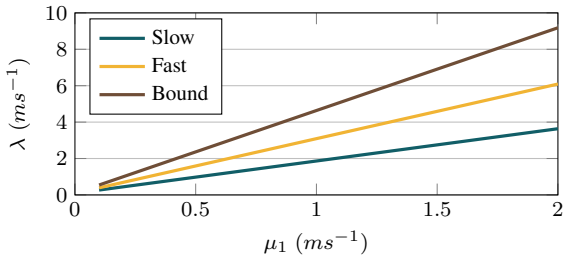
[2]In Matlab 2018, for an error of $\epsilon = 10^{-10}$, and four splits, it takes approximately 2500 iterations to yield R, in less than 40 ms over a laptop with an i7 Intel processor

[3]This would depend on the particular split selection policy

$$F = \begin{bmatrix} \lambda & 0 & \cdots & 0 \\ 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & \mu_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_s \end{bmatrix} \quad L = \begin{bmatrix} -(\lambda+\xi) & \alpha_1\xi & \cdots & \alpha_s\xi \\ \gamma & -(\lambda+\gamma+\mu_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma & 0 & \cdots & -(\lambda+\gamma+\mu_s) \end{bmatrix} \quad (2)$$



(a) 2 Splits



(b) 4 Splits

Fig. 2: Comparison of the λ bound with stability conditions of fast and slow CU setups

$$\lambda < \frac{\gamma + \xi \left\|\mu\right\|_\infty}{\gamma + \xi} \quad (10)$$

In order to assess the looseness of the previous bound, we have made two experiments, using the CU configurations that will be used afterwards. We assume that $\lambda = \xi = 1\ ms^{-1}$, and $\gamma = \frac{1}{10}\ ms^{-1}$. Then, we use two CU configurations, one having two splits: $\mu = [1\ 4]$, and another one having four splits $\mu = [1\ 1.5\ 3\ 5]$. For each of these configurations, we have two different α vectors, so that both the fastest and slowest service rates have a larger probability, leading to fast and slow setups. We increase the value of μ_1, keeping the ratio of the other service rates. Results are shown in Figure 2. As can be observed, when μ_1 is large, the bound behaves worse. On the other hand, for lower service rates, the bound provides a rather good approximation (in particular for the fast configuration) to find the maximum arrival rate that could be accepted.

IV. Results

In this section we use a proprietary event-driven simulator that has been implemented (*C++*) with the goal of both ascertaining the validity of the proposed model, as well as complementing its results.

We will use the CU configurations that are depicted in Table I. In all the experiments, $\xi = 1\ (ms^{-1})$. In addition, we will assume that $\gamma = \frac{1}{10}\ (ms^{-1})$. We thus consider that the time the CU requires to leave the stand-by is much lower than the time it stays at a particular split configuration. For each CU configuration, there are two cases, depending on whether the fastest or slowest configuration is more probable.

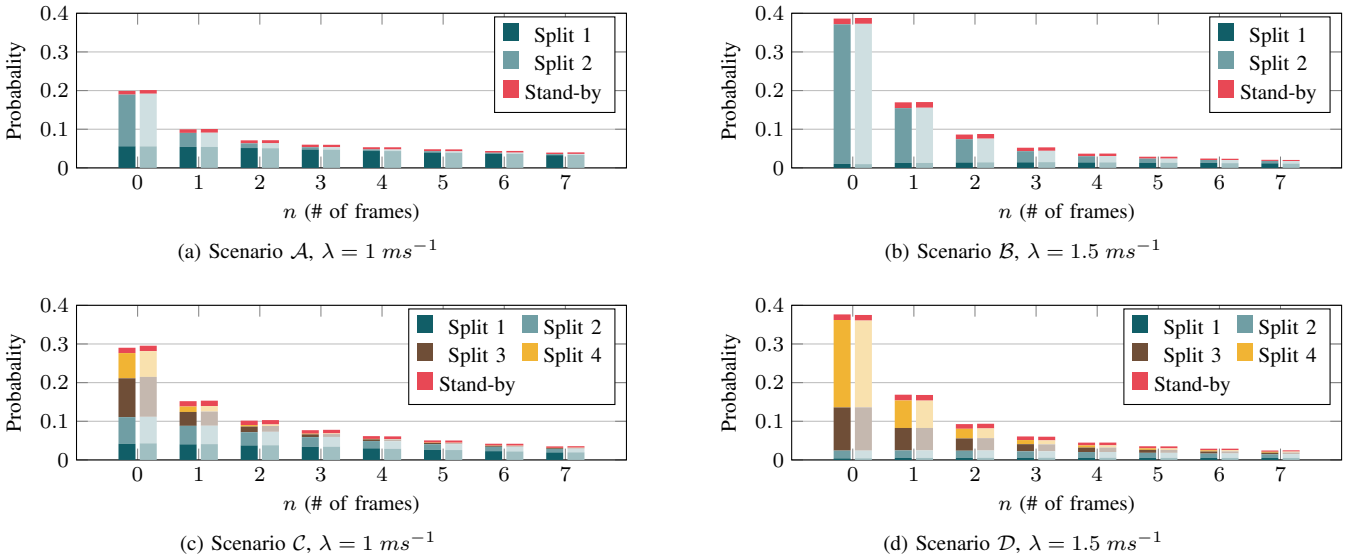
TABLE I: Configuration of analyzed scenarios

Scenario	#Splits	α	$\mu\ ms^{-1}$	$\gamma\ ms^{-1}$
$\mathcal{A}$	2	$[0.75, 0.25]$	$[1, 4]$	0.1
$\mathcal{B}$	2	$[0.25, 0.75]$	$[1, 4]$	0.1
$\mathcal{C}$	4	$[0.4, 0.3, 0.2, 0.1]$	$[1, 1.5, 3, 5]$	0.1
$\mathcal{D}$	4	$[0.1, 0.2, 0.3, 0.4]$	$[1, 1.5, 3, 5]$	0.1

First, Figure 3 shows the state probabilities. We use stacked bars, so the overall length of a bar reflects the probability of having n frames (x-axis) in the CU, and the different colors represent how such probability is divided between the various split configurations. Solid colors correspond to the analytical results, and shaded colors (right side bars) are the values obtained with the simulator. We made 1 experiment for each configuration, with 200000 frames, so as to ensure statistically tight results. In scenarios $\mathcal{A}$ and $\mathcal{C}$, where the slowest split is more probable, $\lambda = 1\ ms^{-1}$, while in $\mathcal{B}$ and $\mathcal{D}$ we increase λ to $1.5\ ms^{-1}$. First of all, we can see an almost perfect match between the analytical and simulated results, which validates both the proposed model and the simulator. In addition, when the fastest configuration is more probable (scenarios $\mathcal{B}$ and $\mathcal{D}$), the probability of working at the slower split configurations strongly decreases.

Figure 4 depicts the average time at the CU for the different configurations. Solid lines are the results obtained with the proposed analytical model, while markers are the average value of 100 independent simulations, each of the them encompassing 10000 frames. In addition, dashed lines were the results obtained with the simulator, but having a constant service time (instead of the exponentially distributed one). Again, we can see an almost perfect match between the analytical results and the values yielded by the simulator. Interestingly, it can be seen that the results obtained with the constant service times are very similar to those observed for the exponentially distributed ones. The model could thus shed light on the maximum reasonable arrival rate that could be accepted at a CU, without hindering the stringent 5G delay requirements. As can be seen, the larger λ, the more time spent at the controller.

Besides the average time that frames would spend at the controller, it is also of utter relevance to study its variance. The analytical model does not allow us to characterize this,



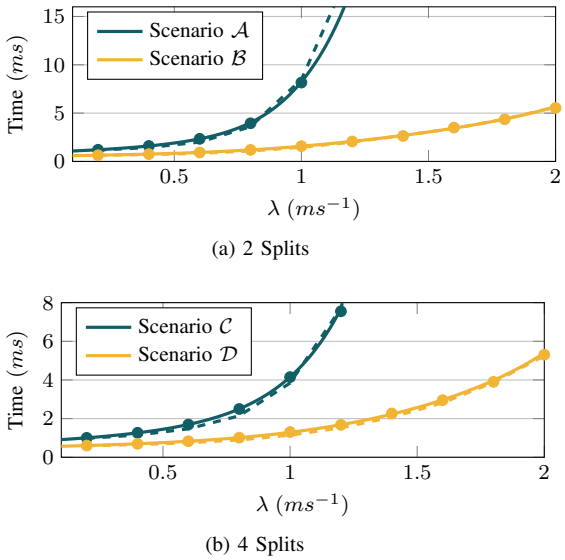
(a) Scenario $\mathcal{A}$, $\lambda = 1\ ms^{-1}$

(b) Scenario $\mathcal{B}$, $\lambda = 1.5\ ms^{-1}$

(c) Scenario $\mathcal{C}$, $\lambda = 1\ ms^{-1}$

(d) Scenario $\mathcal{D}$, $\lambda = 1.5\ ms^{-1}$

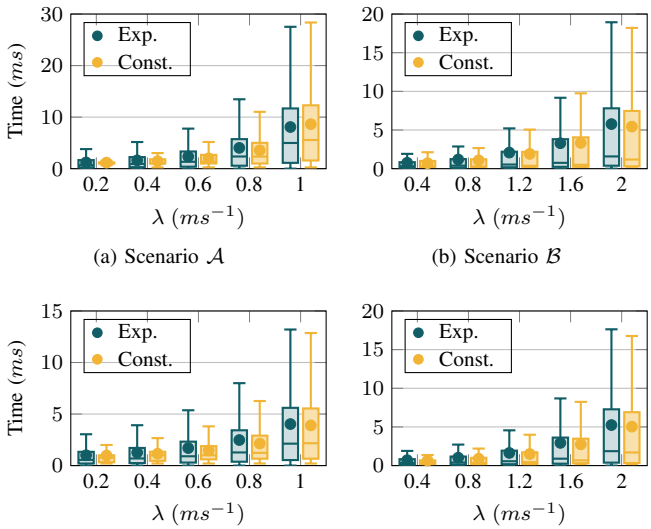
Fig. 3: Probability of having n packets in the controller. Analytical and simulation results are shown in solid and shaded colors, respectively



(a) 2 Splits

(b) 4 Splits

Fig. 4: Total time Vs. arrival rate (λ). Model is shown with solid lines. Simulation with exponential and constant service time are shown with markers and dashed lines, respectively



(a) Scenario $\mathcal{A}$

(b) Scenario $\mathcal{B}$

(c) Scenario $\mathcal{C}$

(d) Scenario $\mathcal{D}$

Fig. 5: Whisker plot of simulated total time Vs. arrival rate (λ) for exponential and constant service times

so we will use the simulator. Figure 5 shows the corresponding results. We increase the arrival rate, and for each configuration we carry out one long experiment, encompassing 200000 frames. We then use a whisker plot to represent statistical distribution of the time spent at the CU. Each plot shows the median (horizontal line within each box), the 0.25 and 0.75 percentiles (box limits), as well as the 0.95 and 0.05 percentiles. In addition, we have included a marker (solid circle), which corresponds to the average value (i.e. results previously shown in Figure 4). We also use the exponentially distributed and the constant service times. We can see that times are rather tight around their average value when the arrival rate is low, but the variance increases for greater λ values. In addition, it can be observed that both exponential and constant service time configurations present similar distributions.

V. Conclusion

We have proposed a model that captures the behavior of Functional Split controllers, which will play a key role in forthcoming 5G communications. It is based on a Quasi-Birth-

Death process, captured by a 2-dimensional Markov Chain, which can be solved by means of Matrix-Geometric solutions. To our best knowledge, this is the first attempt to have a mathematical model that allows studying the performance of Central Units in virtualized 5G architectures.

We have ascertained the validity of the proposed model by means of an extensive simulation campaign, carried out over a proprietary event-driven simulator, which has been also exploited to broaden the analysis. These tools would allow establishing a limit on the arrival rate (load) that is admissible in 5G controllers, respecting the stringent delay requirements that characterize 5G communications. In this sense, we have studied both the average time spent at the controller, as well as its corresponding variance, for two service time distributions. The implementation that we have used to obtain the results of this paper, Matlab scripts and the event-driven simulator (C++), has been made available to the scientific community.

In our future work, we plan to exploit the simulator to carry out more detailed analysis on the performance of 5G controllers. We will consider different arrival rates (i.e. not Poisson), and we will also include different policies to change the split configuration. On the other hand we will also broaden the model so as to consider buffer-limited situations, where frames could be discarded. Last, we are also looking at using dynamic control techniques, which could be used to study the performance of various scheduling and split configuration policies, based on the initial results that we have discussed in this paper.

ACKNOWLEDGMENT

This work has been funded by the Spanish Government (Ministerio de Economía y Competitividad, Fondo Europeo de Desarrollo Regional, MINECO-FEDER) by means of the project FIERCE: Future Internet Enabled Resilient smart CitiEs (RTI2018-093475-AI00).

REFERENCES

[1] 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, Firstquarter 2015.

[2] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): a primer," *IEEE Network*, vol. 29, no. 1, pp. 35–41, Jan 2015.

[3] G. O. Pérez, J. A. Hernández, and D. Larrabeiti, "Fronthaul network modeling and dimensioning meeting ultra-low latency requirements for 5G," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 6, pp. 573–581, June 2018.

[4] IEEE, "Next generation Fronthaul Interface," IEEE 1914 Working Group, Standard. [Online]. Available: https://sagroups.ieee.org/1914/

[5] 3GPP, "Study on new radio access technology: Radio access architecture and interfaces," 3rd Generation Partnership Project (3GPP), TR 38.801, 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3056

[6] C. I, Y. Yuan, J. Huang, S. Ma, C. Cui, and R. Duan, "Rethink fronthaul for soft RAN," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 82–88, Sep. 2015.

[7] P. Arnold, N. Bayer, J. Belschner, and G. Zimmermann, "5G radio access network architecture based on flexible functional control / user plane splits," in *2017 European Conference on Networks and Communications (EuCNC)*, June 2017, pp. 1–5.

[8] D. Harutyunyan and R. Riggio, "Flex5G: Flexible Functional Split in 5G Networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 961–975, Sep. 2018.

[9] Y. Nakayama, D. Hisano, T. Kubo, Y. Fukada, J. Terada, and A. Otaka, "Low-latency routing scheme for a fronthaul bridged network," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 1, pp. 14–23, 2018.

[10] Y. Nakayama, D. Hisano, T. Kubo, T. Shimizu, H. Nakamura, J. Terada, and A. Otaka, "Low-latency routing for fronthaul network: A Monte Carlo machine learning approach," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[11] A. Martinez Alba and W. Kellerer, "A Dynamic Functional Split in 5G Radio Access Networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[12] A. Sriram, M. Masoudi, A. Alabbasi, and C. Cavdar, "Joint Functional Splitting and Content Placement for Green Hybrid CRAN," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–7.

[13] L. Wang and S. Zhou, "Energy-Efficient UAV Deployment with Flexible Functional Split Selection," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[14] ——, "Flexible Functional Split in C-RAN with Renewable Energy Powered Remote Radio Units," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.

[15] D. A. Temesgene, M. Miozzo, and P. Dini, "Dynamic Functional Split Selection in Energy Harvesting Virtual Small Cells Using Temporal Difference Learning," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 1813–1819.

[16] A. Alabbasi, M. Berg, and C. Cavdar, "Delay Constrained Hybrid CRAN: A Functional Split Optimization Framework," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–7.

[17] O. I. Abdullaziz, M. Capitani, C. E. Casetti, C. F. Chiasserini, S. B. Chundrigar, G. Landi, X. Li, F. Moscatelli, K. Sakaguchi, and S. T. Talat, "Energy monitoring and management in 5G integrated fronthaul and backhaul," in *2017 European Conference on Networks and Communications (EuCNC)*, 2017, pp. 1–6.

[18] S. S. Tadesse, C. Casetti, C. F. Chiasserini, and G. Landi, "Energy-efficient traffic allocation in SDN-basec backhaul networks: Theory and implementation," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2017, pp. 209–215.

[19] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint Optimization of Edge Computing Architectures and Radio Access Networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2433–2443, 2018.

[20] F. Malandrino, C. F. Chiasserini, C. Casetti, G. Landi, and M. Capitani, "An Optimization-Enhanced MANO for Energy-Efficient 5G Networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1756–1769, 2019.

[21] A. M. Alba, J. H. G. Velásquez, and W. Kellerer, "An adaptive functional split in 5G networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 410–416.

[22] C. Chang, N. Nikaein, R. Knopp, T. Spyropoulos, and S. S. Kumar, "FlexCRAN: A flexible functional split framework over ethernet fronthaul in Cloud-RAN," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7.

[23] Y. Alfadhli, M. Xu, S. Liu, F. Lu, P. Peng, and G. Chang, "Real-Time Demonstration of Adaptive Functional Split in 5G Flexible Mobile Fronthaul Networks," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, March 2018, pp. 1–3.

[24] C. Chang, N. Nikaein, and T. Spyropoulos, "Impact of Packetization and Scheduling on C-RAN Fronthaul Performance," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7.

[25] C. Chang, R. Schiavi, N. Nikaein, T. Spyropoulos, and C. Bonnet, "Impact of packetization and functional split on C-RAN fronthaul performance," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7.

[26] M. Neuts, "Markov Chains with Applications in Queueing Theory, Which Have a Matrix-Geometric Invariant Probability Vector," *Advances in Applied Probability*, vol. 10, no. 1, pp. 185–212, 1978.

[27] ——, *Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach*. Johns Hopkins University Press, 1981.

[28] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Society for Industrial and Applied Mathematics, 1999.