



HAL
open science

Processing Time Evaluation and Prediction in Cloud-RAN

Hatem Khedher, Sahar Hoteit, Patrick Brown, Ruby Krishnaswamy, William
Diego, Véronique Vèque

► **To cite this version:**

Hatem Khedher, Sahar Hoteit, Patrick Brown, Ruby Krishnaswamy, William Diego, et al.. Processing Time Evaluation and Prediction in Cloud-RAN. ICC 2019 - 2019 IEEE International Conference on Communications (ICC), May 2019, Shanghai, France. pp.1-6, 10.1109/ICC.2019.8761870. hal-02345783

HAL Id: hal-02345783

<https://hal.science/hal-02345783v1>

Submitted on 11 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Processing time evaluation and prediction in Cloud-RAN

Hatem Khedher*, Sahar Hoteit*, Patrick Brown, Ruby Krishnaswamy[†], William Diego[†] and Véronique Vèque*

*Laboratoire des Signaux et Systèmes, Université Paris Sud-CNRS-CentraleSupélec, Université Paris-Saclay, France,
Emails: hatem.ibnkheder@l2s.centralesupelec.fr, sahar.hoteit@u-psud.fr, veronique.veque@u-psud.fr.

[†]Orange Labs, Chatillon, France

Emails: {ruby.krishnaswamy, william.diego}@orange.com

Email: brown.patrick2@gmail.com

Abstract—Cloud RAN (C-RAN) is a very promising architecture for future mobile network deployment, where the cloud-centric approach is useful in improving total processing load. In this context, radio and baseband network functions processing pose interesting problems that we try to expose and solve in this paper. A novel architecture for C-RAN and a first modeling of the system are proposed. Furthermore, we study the impact of many radio parameters on the processing time. Moreover, a mathematical and a deep learning model are proposed and evaluated for processing time prediction. Results show the feasibility of the proposed approaches.

I. INTRODUCTION

The virtualization of compute resources has been widely used recently in many network and service architectures. The basic tasks of the virtualization/cloudification are enabling new network functions, migration and switching [1]. They strongly depend on the underlying network configuration and topology in a way that makes them tributary to network conditions. In other terms, sometimes it is not possible or even not recommended to accomplish some virtualization tasks if the network (or the system) does not present the minimum requirements. This aspect brings back many questions to networks theory but the answers to these questions require an understanding of the new context. Most of the virtualization architectures such as Network Function Virtualization (NFV) included in the Software Defined Network (SDN) paradigm rely on these tools to implement their technical solutions. In this paper, we consider the cloudification of radio network functions and we study the problems that arise from this dynamic process. We present a use case for C-RAN based services.

Cloud Radio Access Network, commonly known as C-RAN, is a novel architecture for future mobile network operators infrastructure. It is composed of three main components as shown in Fig. 1:

- BBU pool: a centralized based-band unit pool regrouping many Base Band Units (BBUs). Its role is to dynamically allocate resources to the remote radio head (RRH) networks based on current network needs.
- RRH network: a wireless network that connects wireless devices. It is similar to access points or towers in traditional cellular networks.
- Fronthaul/transport network: that provides the links between BBUs and a set of RRHs. High link capacities are

needed to address higher bandwidth requirements between BBUs and RRHs. Optical fiber is usually used to handle these requirements.

Decoupling baseband processing from the radio elements, C-RAN leads to multiple advantages in terms of optimizing CAPEX and OPEX of network operators, which allows implementing interference management mechanisms, and hence improves user experience. In this paper, we focus on the cloudification and the virtualization of BBUs, that consists in performing signal processing remotely at the BBUs' level. This study proposes a first characterization of processing time as a function of different radio parameters and a prediction model for processing time in C-RAN.

The softwarization of these functions raises new problems with respect to response times, but in return offers opportunities for flexibility, intelligent management and multi-layer optimization. Intuitively, the most time-requiring task will be the decoding of information from user equipment (i.e., in the uplink direction) [2] [3]. As a matter of fact, the decoding time strongly increases with the density of the coding. The latter is expressed by an integer noted Modulation and Coding Scheme (MCS). However, the decoding must be done typically within 2 ms of the reception time otherwise the information has to be re-transmitted (and the corresponding bandwidth will be lost) [4]. In addition, after a certain number of re-transmissions, the information is lost (at RAN level). In this context, the modeling

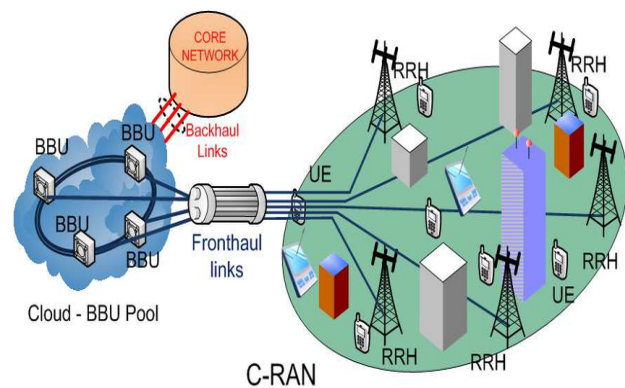


Figure 1: C-RAN architecture

and the analysis of the decoding functions (timing) permits to characterize the impact of different radio parameters on decoding time and to take (scheduling) decisions accordingly. For instance, within a single frequency band (carrier), a RRU can receive up to several hundred blocks of resources, each encoded with a certain MCS. A high MCS value indicates an increasing encoding density. The relationship between MCS and other important parameters such as Signal to Noise Ratio (SNR) is studied in this paper. We use an open-source software defined radio (SDR) called OpenAirInterface (OAI) [5] to understand the impact of system (CPU) and radio parameters (e.g., MCS, SNR, etc.) on decoding time and other BBU processing functions. Then, we use a mathematical and a deep learning technique to predict decoding time. The rest of this paper is organized as follows: Section II highlights recent C-RAN architectures and the state of the art. Section III presents our C-RAN architecture and a first modeling of the system. Section IV discusses the impact of different parameters on the processing time. In Section V, we propose our computational model for prediction purpose. We conclude the paper and presents our future work in Section VI.

II. STATE OF THE ART

In this section, we present the main use cases highlighted by the ETSI standard [6].

- 1) Virtualization of mobile base stations: In this approach, the digital functions of the radio run on a pool of virtualized resources named BBU at a distance from the underlying antenna hardware which is distributed in the Remote Radio Units (RRU). The virtualization can be done in a data center that communicates with the distributed RRU through an optical back-haul networks (optical fiber) in order to respect latency constraints. In this context, authors in [3] propose a framework that splits the set of BBUs into groups that are simultaneously processed on a shared compute platform and show that the centralized architecture can potentially result in savings at least 22 % of compute resources by exploiting the variations of the processing load across base stations.
- 2) Virtualization of the home network: The virtualization of the home network includes the virtualization of the two main components: Residential Gateway and Set-Top Boxes that offer home services (internet access, multimedia service, etc.) to end users. This approach is based on implementing virtualized and programmable software based NFV solution such as: firewalls, DHCP servers, VPN Gateways, DPI Gateways. These functions are moved to data centers in order to decrease the cost of devices and to improve the QoS.
- 3) Virtualization of Evolved Packet Core (EPC): the EPC is the 5G Mobile Core Network. In this use case, the virtualization targets several functions such as: SGW (Serving Gateway), PGW (PDN Gateway providing mobile

equipment connectivity to external packet data networks), MME (Mobility Management Entity), HSS (Home Subscriber Server central database containing subscription-related information), and PCRF (Policy and Charging Rules Function) [7]. The virtual EPC includes all the above functions as software based NFV solutions moved into a cloud EPC. Using this approach, we can reduce the network control traffic by 70 % according to [7].

The main related work on C-RAN architectures and business models are quoted in this section.

In [4], authors study the critical issues of C-RAN which are the fronthaul capacity and BBU latency which is closely related to operating systems virtualization and their real-time behavior. They use OpenAirInterface (OAI) to characterize the baseband processing time under different conditions. Using OAI, the authors propose a BBU processing model that computes the total uplink and downlink processing time focusing on the number of physical resources blocks (PRB) (related to packet size), the modulation and coding scheme (MCS), and the virtualization environments (VE). However, important network parameters such as SNR and the uplink block error rate (UPBLER) are not considered in their model.

In [8], authors propose a virtual RAN (vRAN) architecture that includes three main types of actors: mobile virtual network operators (MVNOs) that request a RANaaS (Radio Access Network as a service), a physical network that provides the RANaaS, and end users that request real time services (e.g., IoT client) or best effort services to MVNOs. The authors define an optimization problem and divide the problem into two sub-problems; the estimation of the available resources (PRB) as a function of SINR (Signal-to-interference-plus-noise ratio) and then their allocation.

Authors in [9] propose a probabilistic approach to solve C-RAN dimensioning and modeling. The authors propose to increase parallelism on certain 5G functions to reduce latency. They apply queuing theory on 5G datasets and consider fixed computing capacity under non-deterministic conditions (i.e., channel decoding function). The results show that there is a high variability of runtime in coding/decoding functions comparing to FFT or demodulation functions. In their parallelism-based work, BBU tasks are decomposed into small runnable jobs or subtasks/threads. However their work does not address real time programming architecture issues.

Literature lacks a detailed characterization of processing time as a function of different radio parameters. Moreover, to the best of our knowledge, there is no paper in the literature that tries to predict processing time using deep learning techniques. In this paper, we try to fill out this gap. The results of this study might be very useful for future studies (e.g., scheduling, resource allocation, etc.).

III. SYSTEM DESCRIPTION

A. Proposed C-RAN architecture

ETSI-MANO standardizes a framework for deploying different Virtual Network Functions (VNFs). In our case, RAN

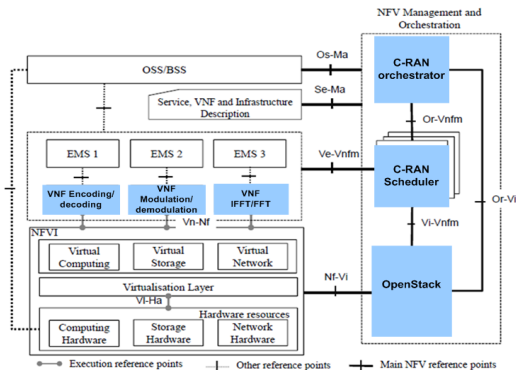


Figure 2: C-RAN reference architecture with respect to ETSI standard

is the target VNF. In this paper, we follow this standard and propose our specific design and architecture for RAN that can be used in 5G deployment. This latter (C-RAN or vRAN) extends the virtual network functions (VNFs) to FFT (or Inverse FFT (IFFT)), demodulation (or modulation), decoding (or coding). The VNFs are controlled by a Virtual Network Function Manager (VNFM).

NFVI is composed of three domains: i) virtual computing domain, ii) virtual storage domain, and iii) virtual networking domain. NFVI is managed by cloud management platform (e.g., OpenStack) which corresponds to the Virtual Infrastructure Manager (VIM) that manages VMs or dockers used in the virtualization process. The proposed C-RAN architecture has a global orchestration that manages and orchestrates the C-RAN VNFMs (if there are many of them), OpenStack, and OSS/BSS. This latter manages QoS, network failure, and security in 5G radio access. We propose a simplified C-RAN architecture with respect to the ETSI-MANO standard as depicted in Fig. 2.

B. BBU modeling

At present, virtual BBU (vBBU) resource allocation in virtual platforms have a somehow static behavior; virtual machines or containers are reserved for a specific VNF (e.g., vBBU) even though a VNF is sporadically invoked. As a consequence, efficiency in resource utilization is not achieved, since computing resources are frozen but no used. It would thus be better to perform statistical multiplexing on computing resources. More precisely, in the BBU model, we assume that a set of cores is available to execute vBBU components with dynamic resource orchestration. Cores are allocated for the execution of a vBBU when that function is invoked.

In this work, we assume that a vBBU is composed of sub-functions, each of them being executed on the multi-core platform as shown in Fig. 3. The goal of this work is to evaluate the processing time of each of these sub-functions, the results can be used later for scheduling and for improving the vBBU performance.

In this context, the functional dis-aggregation in the virtualization process should take into account the correspondence between sub-functions, because it determines the behavior

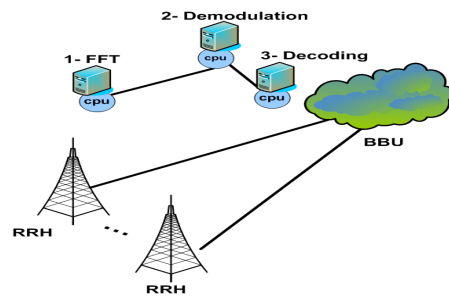


Figure 3: BBU service function graph

TABLE I: Key Features of Modulation and Coding Schemes (MCS)

<i>CQI</i>	<i>Modulation</i>	<i>MCS</i>	<i>SINR(dB)</i>
1-6	QPSK	0 - 9	< 3
7- 9	16-QAM	10 - 16	< 9
10- 15	64-QAM	17 - 28	< 20

in the execution process. Then, a particular vBBU could be viewed as a process flow with sub-functions either running in sequence or being executed in simultaneous threads (in parallel). We consider a vBBU as a sequence of executable processes. Each of them executes a specific sub-function in such a way that the global BBU VNF can be realized. On the one hand, some processes have to be executed in series (i.e., they can start only when the output of the previous one is available), as the Fast Fourier Transform (FFT) and demodulation functions. On the other hand, other tasks can run in parallel (e.g., uplink decoding) even if the subsequent task can be executed only when the output of all parallel processes is available.

We propose a BBU processing time model that gives an average processing time (i.e., decoding time) considering heterogeneous configuration inputs of MCS and SNR. In our model, SNR values are coupled with CQI (Channel Quality Indicator) values where the CQI is an indicator, sent by user equipment (UE) carrying the information on how good/bad the communication channel quality is. We present hereafter the proposed method that maps MCS indexes with SNR values. First, we filter OAI measurements to convert the SNR to CQI. The exact mapping between SNR and CQI is not specified by the 3GPP standard, therefore each device manufacturer implements it according to its own criteria. In this work, we use the current implementation of OAI that proposes such a mapping ¹ [10]. Secondly, CQI is mapped to MCS according to the emulated radio conditions. The mapping table is shown in Table I. It is worth mentioning that for a given CQI value, a MCS lower than or equal to the value indicated in the table is allowed [11].

¹<https://gitlab.eurecom.fr/oai/openairinterface5g/blob/master/openair1/PHY/>

IV. RESULTS

In this section, we study the decoding/encoding processing time by using the receiver/transmitter part of OAI ulsim/dlsim with a fixed CPU frequency that is equal to 3.408273 GHz (performance governor mode). In particular, we are interested in the subframe decoding time of turbo decoder algorithm. We highlight the impact of CPU and other radio parameters in the processing of BBU components.

A. CPU performance analysis

CPU frequency is an important performance parameter in C-RAN architectures. Both uplink and downlink directions are quantified using such a parameter.

In downlink direction, physical procedures include many sub-functions. For instance, in LTE, these sub-functions include Orthogonal frequency division multiplexing modulation (OFDM mod), Downlink Shared Channel modulation (DLSCH mod), Downlink Shared Channel encoding (DLSCH enc) and Scrambling². The Downlink Shared Channel encoding is composed of three main sub-functions which are Rate-matching³, turbo encoding and Sub-block interleaving. We summarize these timing compositions as follows:

$$t_{\text{PHY proc tx}} = t_{\text{OFDM mod}} + t_{\text{DLSCH enc}} + t_{\text{DLSCH mod}} + t_{\text{Scrambling}} \quad (1)$$

$$t_{\text{DLSCH enc}} = t_{\text{Rate-matching}} + t_{\text{Turbo enc}} + t_{\text{Sub-block interleaving}} \quad (2)$$

In uplink direction, the main functions are OFDM demodulation (*OFDMdemod*), Uplink Shared Channel decoding (ULSCH dec) and demodulation (ULSCH demod). The ULSH decoding process includes the Turbo decoder (Turbo dec), Rate-matching, demultiplexing (Demul), and Sub-block interleaving sub-functions. We summarize these timing compositions below:

$$t_{\text{PHY proc rx}} = t_{\text{OFDM demod}} + t_{\text{ULSCH dec}} + t_{\text{ULSCH demod}} \quad (3)$$

$$t_{\text{ULSCH dec}} = t_{\text{Rate-matching}} + t_{\text{Turbo dec}} + t_{\text{Demul}} + t_{\text{Sub-block interleaving}} \quad (4)$$

In Fig. 4a, we use a specific configuration in order to assess the impact of CPU frequency on different BBU baseband processing times in the downlink direction (from the cloud server to end-user equipment). The configuration input is represented as a vector of the MCS index (64-QAM modulation type), a fixed resource grid (25 PRB), and a time series of 100 subframes. We clearly see that Turbo encoding algorithm is the most time-consuming in the total subframe encoding process. Moreover, we notice that increasing CPU frequency reduces processing time and hence decreases decoding failure cases. Recall here that the processing time is plotted per subframe which does ensure its linearity.

²Scrambling process is used for protection against burst errors.

³The main task of the rate-matching is to extract the exact set of bits to be transmitted within a given Time Transmission Interval (TTI). The rate-matching for Turbo coded transport channels is defined for each code block (CB).

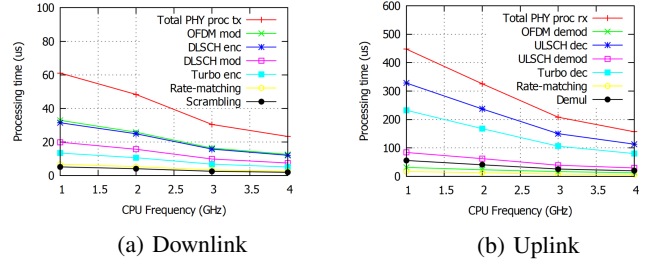


Figure 4: CPU frequency analysis in both downlink and uplink direction

TABLE II: Simulation parameters

Parameter	Value
MCS Indexes	0, 4, 9 for QPSK modulation 10, 13, 16 for 16-QAM modulation 17, 22, 27 for 64-QAM modulation
Physical Resource Block	15 that corresponds to a bandwidth =3 MHz 100 that corresponds to a bandwidth =20 MHz
Link direction	Uplink

Fig.4b, shows the results in the uplink direction. As in the downlink direction, we notice that the subframe processing time decreases with CPU frequency. Moreover, we clearly see that decoding process is the most time-consuming function.

B. BBU processing time versus MCS

In this section, our main objective is to model the BBU processing time using different configuration inputs of MCS, SNR, CQI, and PRBs (physical resource blocks). As the uplink subframe decoding requires more processing time, we aim at studying the impact of MCS on the average decoding time while varying the potential SNR values as explained in Section III-B. As previously mentioned, we use OAI simulator to assess C-RAN implementation. Indeed, we launch different simulations on a single PC using the simulation parameters shown in Table II.

In our experiments, we use box-plots to show, for each MCS value, the statistical distribution of the median latency obtained through 10000 runs. It is worth mentioning that for each MCS, a range of SNR is allowed. The boxplots show the quartiles (Q1 and Q3), the median value as well as the min-max values presented through the ends of the whiskers. Fig. 5, shows that for the case of $PRB = 100$, the decoding time is more significant than with a $PRB = 15$. Moreover, the variation of the decoding time for a $PRB = 100$ is very negligible (i.e. the median latency over different SNRs values does not vary much for a fixed MCS value) when compared to that of a $PRB = 15$. These results favor the use of the 20 MHz bandwidth (i.e., the one that corresponds to a $PRB = 100$) in future 5G networks when computing latency variance is an issue.

After choosing the most suitable bandwidth (which is $PRB = 100$), we focus into finding the accurate decoding time

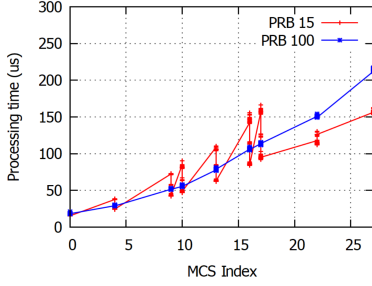


Figure 5: ULSCH decoding times for different PRBs.

for each MCS value. We plot in Fig. 6a, the boxplots of the median as well as the first and third percentile decoding time for each MCS index. As clearly seen in the figure, *PRB100* suffers from high variability in the first and third percentile decoding times while the median decoding time (*DEC_MEDIAN*) presents a low variance. Therefore, median values may be used to model decoding time.

As previously explained, in uplink direction, BBU executes three main components: FFT, ULSCH demodulation, and channel decoding. For that, we study the impact of MCS Indexes on the total BBU processing time that includes all the physical layer procedures. From Fig. 6b, we notice that the BBU software can be classified into two main classes:

- FFT & ULSCH demodulation that does not depend on MCS indexes.
- ULSCH decoding that is function of MCS index.

We clearly see that the channel decoding requires the highest processing time and it increases with the increase of MCS index.

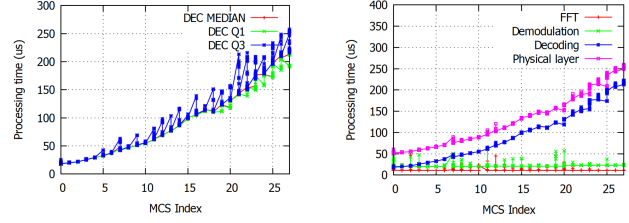
C. BBU processing time versus SNR

In order to assess the impact of SNR on the subframe decoding time, we plot in Fig. 6c the median decoding times as a function to SNR for each MCS value. Results give the gain in processing time when gNB MAC scheduler decides to reduce modulation type (moving from a higher MCS to a lower one). Moreover, for each MCS index, we show that all the SNR values in our proposed method can be accepted.

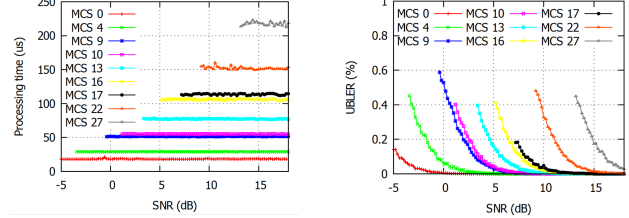
In Fig. 6d, we analyze the block error rate (BLER) in uplink direction which helps network operator to choose relevant SNRs. Results show that high MCS values are decoded at higher SNR and present a significant UBLER. For low values of SNR, we can deduce that a right selection of MCS can play a role in saving bandwidth (when minimizing the number of re-transmissions).

V. FROM BBU PROCESSING TIMES TO COMPUTATIONAL MODELS

As clearly seen in Fig. 6b, we can confirm that the processing time of FFT and demodulation functions are independent of MCS values. However decoding time strongly depends on MCS value. In this section, we present mathematical models as well as a machine learning-based model for median decoding time prediction.



(a) Decoding time as a function of (b) Processing time of BBU functions



(c) Decoding time as a function of (d) Uplink BLER as a function of SNR

Figure 6: OAI-based BBU processing time measurement

A. Mathematical models for decoding time prediction

Using OAI simulator, we create a real dataset that contains the processing time of the main BBU sub-functions. We use two mathematical models that map MCS values to the median decoding time:

- Linear interpolation: In some non-critical 5G scenarios, where significant *UBLER* can be admitted by network operators (especially in Internet of Things (IoT) and smart-grid use cases), linear models are highly recommended to be used in order to have an idea on how the decoding times will increase in the future. In Eq. (5), the estimated median decoding time, $\tilde{t}_{\text{ULSCH dec}}^l$, is formulated as a function of MCS values, i_{MCS} :

$$\tilde{t}_{\text{ULSCH dec}}^l(i_{\text{MCS}}) = a \times i_{\text{MCS}} + b, \quad (5)$$

where $a = 6.4622$ and $b = 3.6835$.

- Quadratic interpolation (Eq. (6)), $\tilde{t}_{\text{ULSCH dec}}^q$: In this interpolation a smooth function is constructed that approximately fits the created dataset.

$$\tilde{t}_{\text{ULSCH dec}}^q(i_{\text{MCS}}) = a \times i_{\text{MCS}}^2 + b \times i_{\text{MCS}} + c, \quad (6)$$

where $a = 0.1842$, $b = 2.2882$, and $c = 17.365$.

The mean-squared error (*MSE*) for each interpolation is presented in Table III where we observe that the quadratic interpolation produces notably better predictions.

B. Machine learning based model for decoding time prediction

Machine learning has addressed different networking scenarios such as resource allocation, mobility prediction, traffic classification, etc.

In C-RAN, the processing of BBU functions is executed in cloud data centers where it is possible to use big data analysis tools such as Hadoop and Spark. Then, recent machine learning techniques such as (un)supervised learning (e.g., network

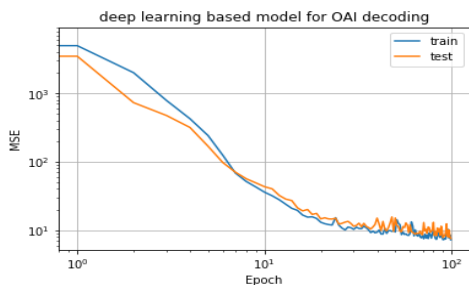


Figure 7: Deep learning-based prediction model.

TABLE III: Comparison

Model	MSE
Linear	133.86
Quadratic	8.52
DNN	7.56

metrics approximation, classification/prediction, and clustering) and reinforcement learning (network flow in motion and resource allocation/management) may be enabled to facilitate decision making and to provide recommendations. Therefore, in this paper, we explore the use of a deep neural network (DNN) approach for the evaluation of C-RAN processing time.

It is worth mentioning that predicting decoding time is useful in order to avoid subframe re-transmission and to enhance end-user quality of experience as well as the overall network performance. The proposed neural network aims at finding the relationship (mapping) between median decoding times and relevant network parameters (MCS and SNR). Then, it tries to predict future inputs. To evaluate our prediction model, we use the mean squared error (MSE) metric (measured over the number of predictions n) to detect the amount of divergence between the predicted and the actual median decoding time as calculated in Eq. 7:

$$MSE = \frac{1}{n} \sum (t_{ULSCH\ dec}^{pred} - t_{ULSCH\ dec}^{act})^2, \quad (7)$$

where $t_{ULSCH\ dec}^{pred}$ and $t_{ULSCH\ dec}^{act}$ represent the predicted and the actual decoding time respectively.

In Fig. 7, we show that, using 67% of the dataset as teaching set and 33% as testing set, the MSE metric decreases to a limit value. In Table III we compare the final MSE for the three decoding time prediction methods. We can clearly see that the DNN method performs better than linear and quadratic estimators. It has the best approximation to the actual decoding times, presenting the lowest prediction error.

VI. CONCLUSION

In this paper, we have highlighted the impacts of the main network parameters on real-time C-RAN processing. We detailed system and network issues of BBU processing in different scenarios. We concluded that network functions should be processed with specific configurations for full BBU processing

under deadlines. Cloudified BBU network functions can be dynamically processed and scheduled in a virtual environment but they need high requirements to achieve full processing performance and full virtualization. We explained also different computational models for C-RAN processing times according to OAI simulations. Different recommendations using the output measurements are proposed to help network operators managing 5G network resources in large scale. As a future work, we plan to propose a novel C-RAN scheduling strategy that schedules BBU subframes in a multi-core cloud platform.

ACKNOWLEDGMENT

This research work has been carried out in the framework of IRT SystemX, Paris-Saclay, France, and therefore granted with public funds within the scope of the French Program Investissements d'Avenir.

REFERENCES

- [1] H. Ibn-Khedher, E. Abd-Elrahman, A. E. Kamal, and H. Afifi, "Opac: An optimal placement algorithm for virtual cdn," *Computer Networks*, vol. 120, pp. 12 – 27, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128617301391>
- [2] K. C. Garikipati, K. Fawaz, and K. G. Shin, "Rt-opex: Flexible scheduling for cloud-ran processing," in *Proceedings of the 12th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '16. New York, NY, USA: ACM, 2016, pp. 267–280. [Online]. Available: <http://doi.acm.org/10.1145/2999572.2999591>
- [3] S. Bhaumik, S. Preeth Chandrabose, M. Kashyap Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, and T. Woo, "Cloudiq: A framework for processing base stations in a data center," 08 2012.
- [4] N. 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*, ser. MCS '15. New York, NY, USA: ACM, 2015, pp. 36–43. [Online]. Available: <http://doi.acm.org/10.1145/2802130.2802136>
- [5] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "Openairinterface: A flexible platform for 5g research," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 33–38, Oct. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2677046.2677053>
- [6] E. G. N. V1.1.1. (2013) Network functions virtualization (nfv); use cases. [Online]. Available: <http://www.etsi.org/deliver/>
- [7] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, Feb 2015.
- [8] S. Khatibi, L. Caeiro, L. S. Ferreira, L. M. Correia, and N. Nikaein, "Modelling and implementation of virtual radio resources management for 5g cloud ran," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, p. 128, Jul 2017. [Online]. Available: <https://doi.org/10.1186/s13638-017-0908-1>
- [9] V. Q. Rodriguez and F. Guillemin, "Cloud-ran modeling based on parallel processing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 457–468, March 2018.
- [10] M. Kawser, N. Imtiaz Bin Hamid, M. Nayeemul Hasan, M. Shah Alam, and M. Musfiqur Rahman, "Downlink snr to cqi mapping for different multiple antenna techniques in lte," *International Journal of Information and Electronics Engineering*, vol. 2, pp. 756–760, 09 2012.
- [11] G. Piro, N. Baldo, and M. Miozzo, "An lte module for the ns-3 network simulator," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '11. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 415–422. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2151054.2151129>