On the Scalability of Open RAN's SMO Heartbeat Management with O-RAN SC

Mariano Moura, Glauco Gonçalves, Ilan Correa, Silvia Lins and Aldebaro Klautau

Abstract— **The Open RAN specification breaks vendors' lock-in promoting software defined networking principles, which establish the centralization of some network functions, particularly, the failure management (a.k.a., Heartbeat Service). However, centralization may introduce performance bottlenecks. Thus, this paper evaluates the scalability of the Heartbeat Service implemented by the O-RAN Software Community. Through measurements, we analyse how this service behaves when the infrastructure grows. Our results show the service reaches an average capacity about 45 messages/s. Moreover, we verified that the message sending period caused the biggest impact on performance, showing that operators must carefully configure this parameter to avoid false failure detection.**

Keywords— **Heartbeat Microservice, scalability, performance analysis, Service Management and Orchestration.**

I. Introduction

Current increasing demand for improved connectivity, coverage, and quality of mobile networks pushed the mobile telecommunication industry towards the development of the 5G technology, which stimulates innovation by leveraging new applications, business models and use cases. Some examples of new possibilities allowed by the 5G technology are micro operators providing solutions to specific vertical sectors [1], intelligent automation in industries [2], and broadband connectivity for rural areas [3].

However, this innovation results in the initial costs to establish new 5G setups that should be considered in the update from 3G/4G technologies to 5G. By comparing the adoption of Long-Term Evolution (LTE) base stations to cover a certain area, it can be noted that 5G networks demand three times more base stations to cover the same area (since 5G network uses high frequencies). Moreover, 5G base stations are four times more expensive than LTE ones, which can be explained by deployment of advanced technologies in the hardware, such as transmission and reception of signals with a massive number antennas[4].

To cope with this cost, one solution is the rearrangement of the 5G Radio Access Network (RAN) through the usage of virtualisation and cloud computing techniques for promoting flexible, scalable, and elastic 5G system. The Open RAN (O-RAN) specification (promoted by the O-RAN Alliance) aims to define the next generation RAN breaking vendors' lock-in through the definition of a common RAN architecture and open interfaces [5]. The specification also defines three independent RAN control loops operating at different timescales, which leverage the usage of recent machine learning techniques for telecommunications.

Based on common practices for software defined networking, the O-RAN architecture specifies a functional component named Service Management and Orchestration (SMO), which acts integrating data for the global optimization and control of the O-RAN setup. Amongst its responsibilities, the SMO uses the so-called O1 interface and heartbeat messages to monitor the status and availability of other O-RAN components.

The O-RAN Software Community[1] (O-RAN SC), an open source software implementation of O-RAN developed by O-RAN Alliance and Linux Foundation, implements the O1 interface and an specific centralized microservice (named Heartbeat Service) for coping with heartbeat messages. But, commonly, centralized systems making heartbeats may become bottlenecks when the managed infrastructure grows [6], [7].

This paper conducts a performance experiment of the Heartbeat Microservice component of the O-RAN SC in order to evaluate the current strategies available in this component to cope with performance bottlenecks. Also, to the best of our knowledge, this analysis is not found in literature about O-RAN SC performance, in other words, this paper is the first to carry out a performance evaluation of such component.

This paper is organized as follows: Section II details the architecture, components, and management flows of the O-RAN Software Community; Section III presents the methodology considered to perform our experiment; Section IV presents the scenarios and results; finally, Section V concludes this paper and presents some future work.

II. O-RAN Software Community

The O-RAN SC is composed of three fundamental blocks, which are shown at Figure 1: the Service Management and Orchestration (SMO) component, the Near-Real Time RIC (Radio Intelligent Controller), and the Non-Real Time RIC [8]. These components manage the RAN network functions, which are denoted by the Open Central Unit (O-CU), the Open Distributed Unit (O-DU), and the Open Radio Unit (O-RU). The O-DU and O-CU can be implemented as Virtual Network Functions (VNF) or as Physical Network Functions (PNF),

Mariano Moura, Glauco Gonçalves, Ilan Correa and Aldebaro Klautau are with LASSE - Telecommunications, Automation and Electronics Research and Development Center, Belém-PA, Brazil. Silvia Lins is with the Innovation Center, Ericsson Telecomunicações S.A, Brazil. E-mails: mariano.moura@itec.ufpa.br, {glaucogoncalves, ilan, aldebaro}@ufpa.br, silvia.lins@ericsson.com. This work was supported by the Innovation Center, Ericsson Telecomunicações S.A. and CNPq, Brazil.

[1] `https://oran-osc.github.io/`

where VNFs implements specifics hardware PNFs inside of a computing platform.

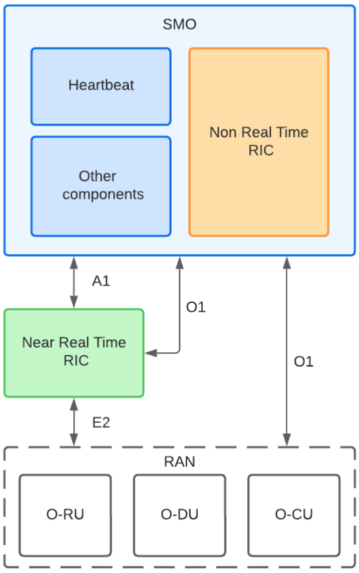

Fig. 1. O-RAN-SC architecture simplified.

In the O-RAN SC, excepting the Non-Real Time RIC, other internal components of the SMO are implemented by the Open Network Automation Platform (ONAP) software[2]. The ONAP software is an open source platform, developed and maintained by the Linux Foundation, for designing, deploying, and monitoring VNFs. In the O-RAN SC architecture, ONAP makes activities related to the management of O-RAN components through O1 interface. There is a wide range of management tasks that are delegated to ONAP components, such as: failure detection, resource inventory, network configuration, and service discovery are just some of these.

The SMO, Near-Real Time RIC, and Non-Real Time RIC perform specific tasks. For example, SMO is responsible for collecting data from RAN, modifying specific parameters of the RAN network functions, RAN health check, etc [9]. Because of the necessity to verify if all RAN components are in a healthy state, the SMO has a Heartbeat Microservice. This component receives periodic messages from the RAN network functions and, based on the quantity of messages received, decides if they are in an working or a failure state.

The message flows, from the network functions to the Heartbeat component, are shown in Figure 2. First, we can see that the messages from the VNFs are received by the VES (VNF Event Streaming) collector, this component is responsible for receiving and validating messages incoming from the RAN against a data model before sending them to DMaaP (Data Movement as a Platform), which is, in turn, a universal message bus inside the SMO. This service routes messages to other components according to the data received. Finally, if the message sent from the VNFs present the status of healthy, they are routed to the Heartbeat component for processing.

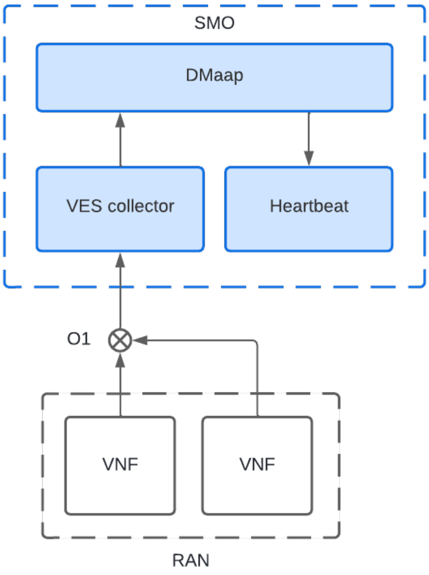

Fig. 2. Heartbeat message flow.

III. Heartbeat performance evaluation methodology

The method used to evaluate the heartbeat message output rate relies on verifying how many of the input messages are processed by the Heartbeat component. To evaluated that, we can vary the input rate and establish a relation between output and input rates, detecting possible bottlenecks in the service.

To produce a variable number of input messages, it was necessary to create multiple VNFs, where each of them had a period to send messages to the Heartbeat component. This was done by simulating the network functions with the Network Topology Simulator (NTS)[3], a VNF simulator provided by O-RAN-SC, and integrating the VNFs with the SDN controller. Thus, we can vary the input rate modifying the number of VNFs created by NTS or changing the period of messages sent by the network function to the Heartbeat component, this modification of the period is done by an SDN controller interface. Also, to convert the input messages sent from VNFs in input rate r_i, in number of messages per second, we use the following equation

$$r_i = \frac{n}{T} \tag{1}$$

where n is the number of VNFs and T is the heartbeat messages period established to all of them.

As O-RAN SC uses the same default message format to communicate with different types of VNFs, the heartbeat message sent from NTS is the same from any VNF that can establish communication with the Heartbeat component, but, the content of some fields can be slightly different. Thus, the average message size and processing time of 574 bytes and 500 microseconds, respectively, are the same to any network function deployed, also, the standard deviation of message size due to content is approximately 2.83 bytes.

[2] https://github.com/onap

[3] https://docs.o-ran-sc.org/projects/o-ran-sc-sim-o1-interface/en/latest/overview.html

The output messages were collected from the Heartbeat Microsevice logs, where we can see the messages processed at the last 20 seconds by the component. Then, the number of processed messages (m) was converted to the output rate (r_o), using the following equation 2.

$$r_o = \frac{m}{20} \tag{2}$$

Input characteristics were choosen in a certain way that the number of VNFs vary from 2 to 76 VNFs assuming only even values and the heartbeat period varies from 1 (the minimum value allowed) to 30 assuming only values divisible by 5. This input range was defined to cover a sufficient large range of input rates, enabling r_i to vary from 0.066 to 76 messages/s without stressing the Heartbeat component beyond its operating region.

This way we have a two-factor experiment with several levels, in a total of 266 experiments, but without replication. In each experiment we observe r_o for about 240 seconds, in order to the system enters in stationary phase. Figure 3 shows the typical behaviour of r_o during an experiment. Please note that r_o initiates in a transient phase and tends to the expected r_i value. Transient r_o samples were discarded and the average of the remaining samples is taken. We denote $\bar{r}_o$ as the average output rate and associate this value to each specific experiment.

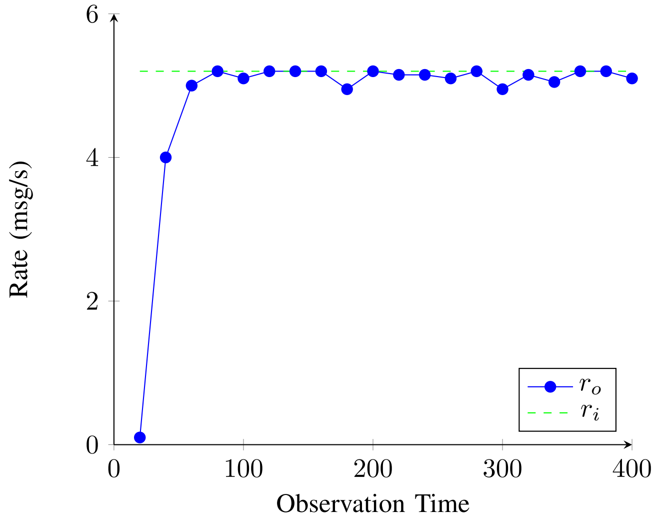

Fig. 3. Raw output rates r_o (blue line) and expected rate r_i (green dashed line) measured during the experiment with $n = 26$ and $T = 5$.

In this work we are also interested in understanding the impact that each factor (n and T) has over r_o. Particularly, we are interested in the metric $d = r_i - r_o$, which is the difference between the expected and the measured rate. It captures if there are bottlenecks in the system, thus when $d > 0$ we can suppose that the system is under stress. To evaluate the impact of the factors over d we use the Two Factor analysis of variance (ANOVA) without Replication hypothesis test [10]. In our analysis, we assume a confidence level of 99%.

Our testbed for this experiment is a cluster composed of two servers whose hardware configuration is shown at Table I. The software stack runs distributed over this cluster, it uses ONAP (Istanbul version) for SMO's tasks and the Dawn version of the O-RAN SC. For RAN simulation we use the E version of NTS, which runs on the Cluster Leader.

TABLE I
HARDWARE DESCRIPTION

	RAM	***HD***	***CPU cores***	***SO***
Cluster leader	64GB	1TB	16	Ubuntu 18.04.6 LTS
Cluster auxiliary	16GB	1TB	6	Ubuntu 18.04.6 LTS

IV. RESULTS

Figure 4 shows as the average output rate behaves for each input rate. As expected from the performance analysis theory, we have an output rate that follows the input rate until it reaches systems's capacity limit (the "knee") and the output rate starts to decay [11]. Thus, the $\bar{r}_o$ follows the red line, that represents the ideal output, with a normalized mean square error (NMSE) of 0.185 until it reaches the input rate of 52 messages/s. From that point on, suddenly the output rate goes to 35 msg/s with an NMSE of 12.5, between the knee and the input rate of 76 messages/s.

The output rate fall suddenly after the input rate reaches 52 msg/s because from this point on, the Heartbeat's processing rate starts to oscillate. In other words, the average number of processed messages varies between 1000 and 400 messages due to message queuing. Thus, the average output rate ($\bar{r}_o$) goes to 35 msg/s. This oscillation effect, and so the output rate limitation, occurs due to the very functioning of the heartbeat component when the input rate is to high.

The usable capacity of the Heartbeat component is obtained when the load is working on knee, and it has a value of 45.57 msg/s. therefore, the optimal point of Heartbeat component operation is on the knee, as predicted from performance theory, once it has the highest output rate with a low NMSE, estimate in 0.8 on this point.

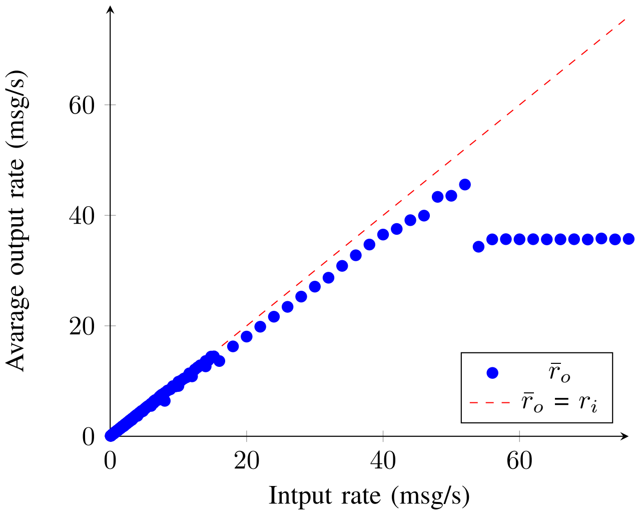

Fig. 4. Average output rates measured $\bar{r}_o$ (blue dots) and expected output rate $\bar{r}_o = r_i$ (red dashed line)

Output rate information presents an overview of the system behaviour, but in order to estimate how the number of VNFs and the Heartbeat message period, our experiment factors, affect the component response, we use the ANOVA hypotheses test, considering the difference between input and output rate as our metric, as explained in Section III.

Table II summarizes the results of the ANOVA test. The column ***df*** exposes the degree of freedom of each factor. The ***Sum Sq*** indicates the variance explained by each factor. The ***Mean Sq*** exhibit the sample variance. The **F value** is the ratio between the factor variance and the residual variance. Finally, the **p-value** determines the statistical relevance of the previous parameter.

TABLE II

TWO FACTOR ANOVA TABLE

	df	*Sum Sq*	*Mean Sq*	F value	p-value
Number of VNFs	37	960.98	25.97	1.06	0.393
Heartbeat Period	6	4072.79	678.8	27.56	0.0
Residuals	222	5466.99	24.63		

The residuals line in the Table II is related to the noise in the metric. Thus, the ***Mean Sq*** parameter of this factor indicates the noise variance over the metric. Then, if the ratio presented in **F value** is close to 1 we cannot establish that the factor affects the metric, because the variance seen is actually result of noise.

Evaluating the **F value** in ANOVA results, we can determine that the Heartbeat period have a significant impact over the difference of rates, so, Heartbeat Period is the main factor. Another way to visualize this result is through the **p-value**. Thus, at a 99% confidence level, we can reject the null hypothesis that the Heartbeat period does not impact in the difference of rates ($0.0 < 0.01$). In the other hand, we cannot reject the null hypothesis in case of the Number of VNFs factor, once $0.393 > 0.05$.

To conclude, in Figure 5 we can see the d metric for each Heartbeat message period (each line) and each level of the Number of VNFs. First, we can ratify that Heartbeat message period is the main factor impacting in the output rate. Moreover, this graph shows that the major difference in the input and output rates occurs for the cases when the Heartbeat period was small ($T = 1$), which causes a higher input rate. In this case, the output rate is consistently lower than the input rate as the Number of VNFs grows. Particularly, one can see that the rate difference suddenly grows when using 54 VNFs.

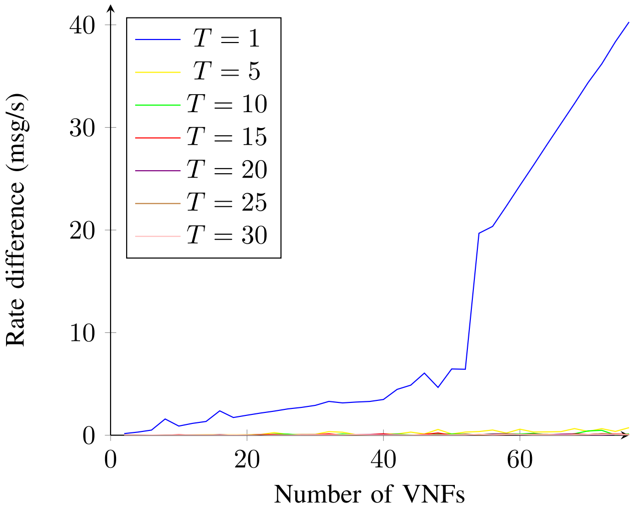

Fig. 5. Rate difference between throughput and load d, for each heartbeat message period

V. CONCLUSION

This article presented a performance analysis of the scalability of the Heartbeat Microservice, which is specified in the Open RAN architecture. We used the open source O-RAN SC implementation. Specifically we analysed how the Heartbeat component respond to the growth in the RAN infrastructure, i.e., when the number of clients connected to the SMO is increased. In our testbed, we showed that the Heartbeat component reached a usable average capacity about 45 msg/s. For more intense input rates, the output rate of the Heartbeat component decays. In such case, some heartbeat messages can be delayed and even lost, causing false failure alarms.

Our experiment also pointed that the factor that most impacted the Heartbeat component response is the Heartbeat period established to the RAN element sending messages. We have showed that when the Heartbeat period of one second caused the biggest impact on the output rate. This result shows that, when the RAN expands, operators must carefully configure the Heartbeat period to avoid data loss in this component.

As future work, we intend to enrich our evaluations incorporating other higher level metrics, as the rate of false failure alarms. Moreover, we want to evaluate how to explore redundancy and load balance strategies for improving the performance of the Heartbeat component.

REFERENCES

[1] P. Ahokangas, M. Matinmikko-Blue, S. Yrjölä, *et al.*, “Business models for local 5G micro operators,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 730–740, 2019.

[2] M. Attaran, “The impact of 5G on the evolution of intelligent automation and industry digitization,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–17, 2021.

[3] A. M. Cavalcante, M. V. Marquezini, L. Mendes, and C. S. Moreno, “5G for Remote Areas: Challenges, Opportunities and Business Modeling for Brazil,” *IEEE Access*, vol. 9, pp. 10 829–10 843, 2021.

[4] G. E. Gonçalves, G. L. Santos, L. Ferreira, *et al.*, “Flying to the Clouds: The Evolution of the 5G Radio Access Networks,” in *The Cloud-to-Thing Continuum*, Palgrave Macmillan, Cham, 2020, pp. 41–60.

[5] A. Garcia-Saavedra and X. Costa-Perez, “O-RAN: Disrupting the virtualized RAN ecosystem,” *IEEE Communications Standards Magazine*, 2021.

[6] R. Shi, Y. Gan, and Y. Wang, "Evaluating scalability bottlenecks by workload extrapolation," in *2018 IEEE 26th international symposium on modeling, analysis, and simulation of computer and telecommunication systems (MASCOTS)*, IEEE, 2018, pp. 333–347.

[7] Z. Hou, Y. Huang, S. Zheng, X. Dong, and B. Wang, "Design and implementation of heartbeat in multi-machine environment," in *17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003.*, IEEE, 2003, pp. 583–586.

[8] A. Akman, S. Gu, J. Huang, I. Wong, and more, "O-RAN Minimum Viable Plan and Acceleration towards Commercialization," *O-RAN Alliance white paper*, 2021.

[9] R. Niklasson, R. Bhyrraju, K. Thakar, D. Espadas, G. Hylander, and J. Paul, "An intelligent platform: The use of O-RAN's SMO as the enabler for openness and innovation in the RAN domain," *Ericsson white paper*, 2021.

[10] D. C. Montgomery, *Design and analysis of experiments*. John wiley & sons, 2017.

[11] R. Jain, *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.* Ser. Wiley professional computing. Wiley, 1991, pp. I–XXVII, 1–685, ISBN: 978-0-471-50336-1.