On the scalability of 5G Core network: the AMF case

Imad Alawe, Yassine Hadjadj-Aoul, Adlen Ksentini, Philippe Bertin, Davy Darche

On the scalability of 5G Core network: the AMF case

Imad ALAWE[1], Yassine HADJADJ-AOUL[1], Adlen KSENTINI[2], Philippe BERTIN[1], and Davy DARCHE[3]

[1]Firstname.Lastname@b-com.com, IRT b<>com, Rennes, France
[2]adlen.ksentini@eurecom.fr, EURECOM, Sophia-Antipolis, France
[3]davy.darche@tdf.fr, TDF, Paris, France

***Abstract*—One of the requirement of 5G is to support massive number of connected devices, considering many use-cases such as IoT and massive Machine Type Communication (MTC). While this represents an interesting opportunity for operators to grow their business, it will need new mechanisms to scale and manage the envisioned high number of devices and their generated traffic. Particularity, the signaling traffic, which will overload the 5G core Network Function (NF) in charge of authentication and mobility, namely Access and Mobility Management Function (AMF). The objective of this paper is to provide an algorithm based on Control Theory allowing: (i) to equilibrate the load on the AMF instances in order to maintain an optimal response time with limited computing latency; (ii) to scale out or in the AMF instance (using NFV techniques) depending on the network load to save energy and avoid wasting resources. Obtained results via computer system indicate the superiority of our algorithm in ensuring fair load balancing while scaling dynamically with the traffic load.**

***Keywords*—5G, Scaling, Load Balancing, AMF, NGC, Control Theory**

I. Introduction

Next mobile network generation (5G) is supposed to offer more new services, while supporting heterogeneous and high number of User Equipments (UE). Indeed, 5G should support not only human oriented devices, but also machine to machine devices; typically sensors and actuators. According to recent forecast, up to 28 billion devices will be connected to 5G by 2021, in front of 17 billion in 2016 [1].

To support the expecting high number of devices, 3GPP has rethought the Core network architecture aiming at being more flexible and scalable. The new architecture, namely New-Generation Core (NGC) architecture [2], addresses scalability and flexibility by introducing more modular Network Functions (NF) to compose the control plane service, which could also relies on network virtualization via Network Function Virtualization (NFV).

Our main focus in this paper concerns the Core Access and Mobility Management Function (AMF), the Session Management Function (SMF) and the Unified Data Management (UDM), which were combined in the LTE into a monolithic component named the Mobility Management Entity (MME). The separation of MME functions is interesting from the scalability point of view, where the UE states and the session states will be hosted respectively in the UDM and the SMF, while the AMF will be only dedicated for processing tasks. Moreover, this separation offers flexibility since UEs procedures are no more dedicated to only one AMF at a time. As UE contexts are hosted in the UDM, any UE procedure can be handled by any AMF connected to the UDM, where the UE context is hosted. Consequently, this separation offers the capability to dynamically dimension the AMF depending upon the network load to avoid network overhead and network congestion; hence managing the expected high number of devices in 5G.

In this paper, we propose a scaling algorithm based on Control Theory for the AMF instances, which allows: (i) controlling the load of each AMF by dispatching the requests based on each AMF load and the whole system load; (ii) taking benefit from NFV to dimension the system dynamically, by deploying new AMF instances (scale in) or remove AMF instances (scale out), according to the traffic load. The proposed solution is tested and verified using computer simulations, and its results are compared with a model using probabilistic procedure dispatching and Exponentially Weighted Moving Average (EWMA) for scaling out/in the AMF instances.

The remainder of this paper is organized as follows. Section II lists and analyses related work on legacy MME scaling and legacy MME load balancing in SDN and NVF paradigms. Section III presents the assumed architecture based on NGC and NFV, and then explains the interaction of the control model with the assumed architecture. Section IV describes the proposed analytical model for AMF balancing and scaling. Model evaluation and results analysis are presented in Section V. Finally Section VI concludes this paper with a summary recapping the main advantages and achievements of the proposed solution.

II. Related Work

In legacy 4G EPC, Mobile Network Operators (MNO) developed different strategies to load balance the MMEs in a way to avoid network congestions. Some, for example, deploy multiple MMEs for a granted zone, and use probabilistic distribution of UE arrival at the eNodeB level. Others may just deploy one over dimensioned MME for a given zone, and thus no load balancing procedure is needed at the eNodeB. However, experiences shown that even an over-dimensioned MME is subject to persistent overloads [3]. Besides load balancing, dynamic scaling, as offered by NFV, is not really

possible in 4G due to the difficulties to instantiate new MMEs on demand.

Nevertheless, in 5G, modern opportunities emerge to tackle this issue in an economic and intelligent manner. Indeed, With SDN and NFV the network control functions of the MME are virtualized and hosted in the cloud. Thus, the deployment of a new MME will be a matter of software deployment. This modern way of dimensioning the network is cheaper and faster than the classical solution that is based on hardware. In fact, many studies and reflections were conducted on the adaptation of SDN and NFV for 5G networks, i.e. in [4] and [5]. In [6] and [7], authors propose analytical models providing a quick way to help mobile operators to plan and design network optimization strategies without large-scale deployment, saving on cost and time. In [8], authors present an NFV-based traffic offloading framework architecture using virtual EPC (vEPC), aiming at enabling on-demand traffic offload when the legacy EPC network capacity is reaching an offload threshold. This can be considered as a transition solution from 4G to 5G.

Some other works have been conducted addressing the MME scaling and load balancing issues within the context of 5G. Authors in [9] propose a new model of state-full MME. They propose to split the MME into three parts: 1) the traffic sorter, 2) the processing functions and 3) the state database. This approach proposes creating groups of International Mobile Subscriber Identities (IMSI) based on hash value and assigning a processing function instance to a group of UEs. Based on the IMSI group, the traffic sorter will route UE requests to specific processing function. However, this limits the scalability of the processing functions, as there should always be at least one worker available to serve each group of UEs. Further, this work proposes only horizontal scaling (processing functions scaling), which is limited by the processing capability of the traffic sorter, as it is the single point of access to the MME node; hence, not addressing MME node scalability.

A distributed MME model is, also, proposed in [10] and [11]. In contrast with the approach proposed in [9], the MME model is stateless and based on an external users' state storage system. Thus, the migration between MMEs is limited only to UEs in an idle state. However, in the active state, UEs are attached to an MME instance. Therefore, another MME instance may receive a network event for that UE. This request has to be forwarded to the correct MME, hence increasing latency of EPC procedures.

Authors in [12] and [13] propose a stateless vMME that is split into three logical components: 1) front-end (FE), 2) MME service logic/Worker (SL), and 3) state database (SDB). The authors assume that MME SL can handle any request for any UE as they are stateless. However, as the users' sate database is local and is not shared with the other vMMEs, this limits requests' handling within the same MME. The authors, also, propose to use one FE element, which may represent a congestion point.

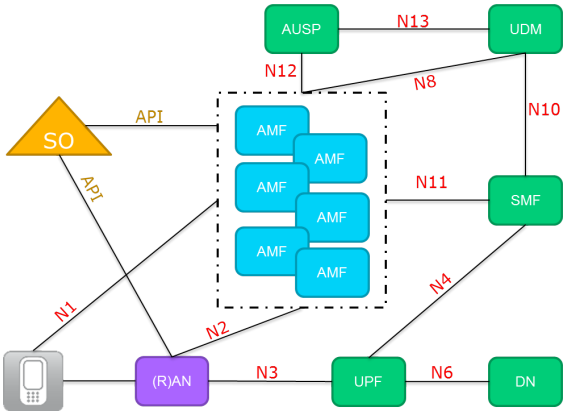

Fig. 1. Architecture proposal for 5G

III. Architecture

In this work, we assume a complete New-Generation Core (NGC) based upon the 3GPP work [2]. The overall architecture is depicted in Figure 1, which illustrates the new core Network Functions (NF), where some are the result of a split of the current EPC functions, such as the MME. The communication between those functions will be held by reference interfaces, noted as Nx as depicted in Figure 1. In this paper, we will direct our focus only on the Core Access and Mobility Management Function (AMF), the Session Management Function (SMF) and the Unified Data Management (UDM) as they are approximately the equivalent to the MME component in LTE. For further information concerning the other functions, please refer to [2]. In the NGC architecture, the legacy MME component is split into three NF: the AMF, the SMF and the UDM. The AMF handles only access control and mobility requests, the SMF will manage the UE sessions and the UDM will handle the UE contexts. Indeed, this approach offers more flexibility to the core network, but still the AMF may suffer from signaling traffic overload. Therefore, we propose, complimentary to the 3GPP work, a mechanism that addresses the AMF scalability by providing a novel scale in/out algorithm depending on the network load. Via this approach, a single UE procedure is processed by a unique AMF. The other procedures may use, however, a different AMF as the UEs states and the sessions are stored respectively in the UDM and the SMF.

To scale in/out the AMF, a Service Orchestrator (SO) is needed. It will be in charge of running the proposed control model, described below, and deploying new AMF instances when needed. The SO could take part to the NFV Orchestrator (NFVO), as defined by ETSI NFV model [14]. The SO will use management interfaces (or API) to communicate with the AMF and the (Radio) Access Network ((R)AN) functions (ex. eNodeB). Moreover, as specified by the NFV ETSI model, the SO should be connected to a VIM in order to add or remove virtual AMF instances, and to the VNF Manager (NFVM), in order to configure the new AMF instances.

A. Traffic steering

As the exchanged messages between the network functions, in the NGC architecture, are not defined yet by the 3GPP

group, we assume that the legacy messages will be adapted for the new entities. Indeed, in LTE, at the end of the association procedure between the eNodeB and the MME, the MME sends its relative capacity to the eNodeB. The relative capacity of an MME is a value between 0 and 255, which allows the eNodeB to balance the load between the MMEs in the same pool. In our solution, this message will be exchanged between the AMF and the (R)AN functions. Further, we propose adjusting the relative capacity field dynamically in order to give the (R)AN accurate information about the load of each AMF. The modification of this field uses "Configuration Update" message defined in 3GPP standard [15]. The value of the relative capacity of each AMF will be given by the control model, described below, depending on the system load and on the state of each AMF. Thus, each AMF will maintain an optimal load in order to process the requests with no additional latency.

B. Scale In/Out

As mentioned earlier, the paper contribution is a scale in and out algorithm for the AMFs depending on the overall load. In order to apply those services on the AMFs, we assume that the (R)AN is compatible with the DNS notification mechanism [16]. In that situation, the (R)AN will subscribe to the DNS list, and, thus, it will be notified of the creation and the deletion of a given AMF.

1) Scale out: When the control system, running in the SO, detects the need of a scale out for the AMF, the SO deploys a new AMF in the architecture. In addition, it notifies the DNS of the creation of the additional AMF. In its turn, the DNS pushes the new AMF IP to the (R)AN. In such a case, the control system push new relative capacity for each AMF if needed. Finally, the AMFs notifies the (R)AN about their new AMF relative capacity using the "Configuration Update message" defined in 3GPP standard as mentioned in Sub-Section III-A.

2) Scale In: When the control system detects a possibility of a scale in, the SO notifies the AMF in question. Once the AMF is notified, it pushes a relative capacity value equal to 0. Consequently, the (R)AN will not send any new procedure to this AMF instance. Following the "Configuration Update" Message, the AMF triggers an S1 release Request with the cause "load balancing". Logically, the other AMF instances will be notified about their new relative capacity following the decision taken for the scale in, so that in their turn, they will notify the (R)AN. Finally, the SO destroys the AMF instance when all procedures are done.

IV. AMF Balancing and Scaling Proposal

A. Model description

In this section, we describe our proposal featuring AMF load balancing and scaling. This model is triggered in order to split the load over the available AMFs in the NGC. When all AMFs are fully used, our proposal takes in command deploying a/multiple new one(s) to keep the access to the NGC and balance the load over all the operational AMF, in order to reduce latency. This model works, also, in the reverse way. When many AMFs are deployed and the overall system is underloaded, it will trigger a scale in procedure, in order to avoid wasting resources. Fig. 2 depicts the model of our AMF load balancing and scaling proposal. To start, for the sake of simplicity, we assume one (R)AN on the access side and multiple AMFs in the NGC. However, this model can be easily extended to consider the multiple (R)AN case.

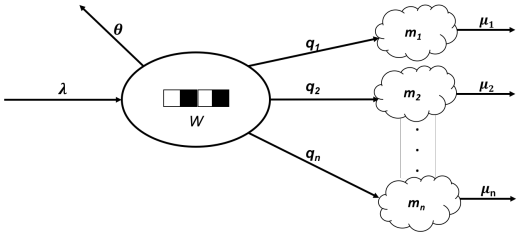

Fig. 2. System Model

The different parameters of the model are described below:

- λ: The number of arriving UEs request
- θ: The number of UE request that have not been satisfied and that need to be re-sent by the UEs
- w: The number of UE request in the buffer waiting to be dispatched over the deployed AMFs
- q_i: The number of UE request that are sent to AMF i for processing
- m_i: The number of UE request being processed by AMF i
- μ_i: The number of UE request satisfied by AMF i.

Following the generic system parameter description, the state evolution of our model is written below following the discrete-time system of equations:

$$\begin{cases} m_1(k+1) = m_1(k) + q_1(k) - \mu_1(k), \\ m_2(k+1) = m_2(k) + q_2(k) - \mu_2(k), \\ . \\ . \\ . \\ m_n(k+1) = m_n(k) + q_n(k) - \mu_n(k), \\ w(k+1) = w(k) + \lambda(k) - \theta(k) - \sum_{i=1}^{n} q_i(k), \end{cases} \tag{1}$$

Let $p_i(k) = q_i(k) - \mu_i(k)$ and $\gamma(k) = \lambda(k) - \theta(k) - \sum_{i=1}^{n} \mu_i(k)$. The system represented in (1) can be reformulated as below:

$$\begin{cases} m_1(k+1) = m_1(k) + p_1(k), \\ m_2(k+1) = m_2(k) + p_2(k), \\ . \\ . \\ . \\ m_n(k+1) = m_n(k) + p_n(k), \\ w(k+1) = w(k) + \gamma(k) - \sum_{i=1}^{n} p_i(k), \end{cases} \tag{2}$$

The model (2) can be written as a discrete-time linear system, in the form:

$$\begin{cases} X(k+1) = AX(k) + BU(k), \\ Y(k) = CX(k), \end{cases} \tag{3}$$

where, the state vector

$$X(k) = [M_n(k) - M_n^{ref}, w(k) - w^{ref}]^T$$

and

$$M_n(k) = [m_1(k), m_2(k), ..., m_n(k)]$$

The constant vector $M_n^{ref} = [m_1^{ref}, m_1^{ref}, \ldots, m_n^{ref}]$ represents the targeted load of the AMFs. This will avoid AMF overhead and guarantee that requests are satisfied with a limited latency. w^{ref} is the targeted overload allowing to detect if an AMF scale in/out is needed for the system to guarantee an optimal requests' dispatching and processing while minimizing resources' wastage.

The control vector $U(k)$ is defined as follows:

$$U(k) = [p_1(k), p_2(k), ..., p_n(k), \gamma(k)]^T$$

The remaining matrices are, thus, defined as follows:

$$A = C = I_{n+1},$$
$$B_{(n+1)\times(n+1)} = \begin{pmatrix} I_n & 0_{n\times 1} \\ -1_{1\times n} & 1_{1\times 1} \end{pmatrix}$$

The output $Y(k)$ of this system represents the load of each AMF and the state of the buffer at time step k.

It can be checked easily that all the eigenvalues of the matrix A do not satisfy the stability condition[1] cf. [17]. This means that the system described in (3) is unstable and do not converges to the desired state, if no control action is performed.

The controllability [18] of this model can be analyzed by calculating the controllability matrix, which is defined as follows:

$$C = [B \quad AB \quad A^2B \quad ... \quad A^{n-1}B]$$

To be controllable, the controllability matrix of the system should have a full row rank. Indeed, for our model, the controllability matrix has a full row rank equal to $n + 1$. It can, also, be checked that the system is observable [18].

B. Dynamic AMF Load Balancing

We focus in this part on the design of the regulator's model to stabilize the whole system by scheduling the UE requests to a given AMF, with the objective to efficiently use the available resources. Also, it will control the decision of AMF scale out in case where more resources are needed, or a scale in when fewer resources are used to reduce resources' wastage. It is worth recalling that the regulator's model is a function that runs at the SO.

Since the controller, following the requirements listed above, needs to dynamically and in real-time calculate m_i, θ and w, it will be based on the Linear Quadratic Regulator (LQR) [19]. More specifically on infinite-horizon, discrete-time LQR. In LQR model, there are two main characteristics: the performance index J and the feedback vector $U(k)$. The performance index is given as follows:

$$J = \sum_{k=0}^{\infty} [X(k)^T Q X(k) + U(k)^T R U(k)], \tag{4}$$

[1]All eigenvalues should be strictly smaller than 1.

where $X(k)$ and $U(k)$ are the state vector and the feedback (control) vector, respectively. The LQR aims to minimize the performance index in order to allow the system to converge to the goal with less controller action. Therefore, Q and R from 4 as well as the cost matrices should satisfy:

$$Q = Q^T \geq 0, R = R^T > 0 \tag{5}$$

Finally the feedback vector $U(k)$ can be written as the following:

$$U(k) = -KX(k) \tag{6}$$

where the matrix $K = [R + B^T S(k+1)B]^{-1} B^T S(k+1)A$ and the matrix $S(k)$ is the solution of the following Riccati difference equation [19]:

$$\begin{aligned} S(k) = Q + A^T S(k+1)A - \\ A^T S(k+1)B[R + B^T S(k+1)B]^{-1} B^T S(k+1)A. \end{aligned} \tag{7}$$

In steady state, $S(k) = S(k+1) = S$, thus, Riccati equation expressed in 7 can be written as:

$$S = Q + A^T SA - A^T SB(R + B^T SB)^{-1} B^T SA \tag{8}$$

The optimal control can thus be described by:

$$U(k) = -(R + B^T SB)^{-1} B^T SAX(k). \tag{9}$$

C. Dynamic AMF Scaling

Having detailed the AMF load balancing algorithm, we focus now on the AMF scaling process. Thanks to the control vector $U(k)$, the buffer $w(k)$ is calculated and updated in real-time. In addition to those two parameters, new parameters, inspired from [20], will be calculated. Starting by the Average Loss $aLoss$ that is determined as follows:

$$\textit{aLoss}(k+1) = \textit{wf} \times \textit{aLoss}(k) + (1 - \textit{wf}) \frac{\theta(k)}{\lambda(k)} \tag{10}$$

where *wf* is a weight factor. Once we have the *aLoss*, a Congestion probability is deduced as follows:

$$\textit{CongP} = \textit{aLoss}(k) + \beta \times \sqrt{\textit{aLoss}(k)} \times \gamma(k) - w^{\textit{ref}} \tag{11}$$

where β is a learning factor.

If *CongP* is higher than a fixed probability threshold, then a scale out is needed. Thus, a new AMF is deployed and integrated to the whole system. Otherwise, a scale in possibility is studied. For that, we need to check if the average load of the actual system can be supported if we remove one AMF. If it is the case a scale in procedure is triggered.

V. Model Evaluation and Results Analysis

A. Test Scenarios

As stated before, the first objective of our model is to dynamically scale out or in the AMF. The second one consists on dispatching the UE requests intelligently in order to have an optimal load in each AMF and, thus decreasing the response latency. To test and validate our model, four scenarios were implemented using Matlab, in addition to the probabilistic Exponential Weighting Moving Average model (EWMA). The EWMA model consists on dispatching the UE requests

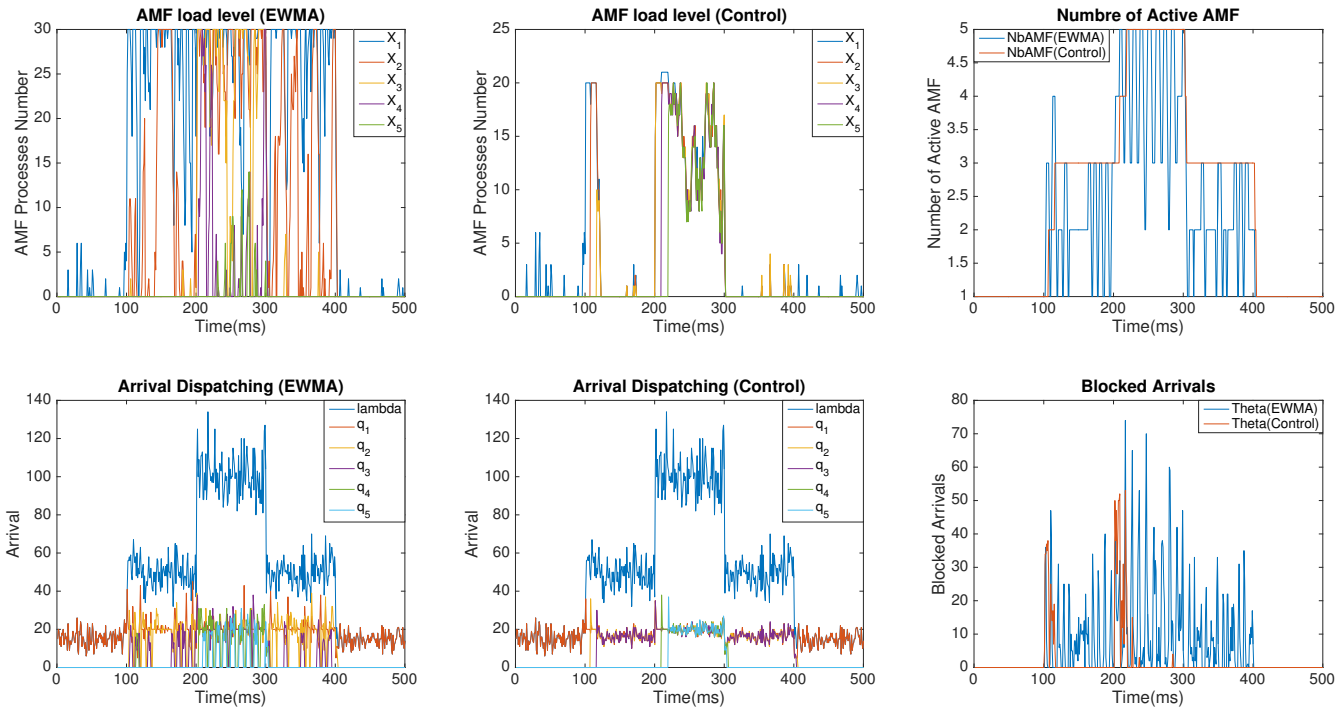

Fig. 3. Evaluation of the Adaptation Of the Control Model

randomly over the active AMFs. The envisioned scenarios will allow us to evaluate the performance of our model, which based on control theory, against the EWMA model.

For both, the control model and the EWMA model, we fixed the highest number of AMFs (5 AMFs) that can be deployed during the scenarios. Each AMF supports a maximum load up to 30 processes by time step. Above this highest rate, the UE arrivals will be blocked. We fixed also the optimum AMF load rate to 20 requests by time step. For this optimum rate, the AMF can process the requests with no latency. Above this rate, an additional latency, which may take very high values, is added to each request.

The first scenario implements different arrival's rate, based on Poisson distribution, over a given period of time. The arrival rate will be increased over the time until reaching the maximum capability of the system, and, then, it will be decreased until reaching a minimal arrival rate. This scenario allows to validate the dynamic adaptation of our model and its flexibility by scaling out the AMF NF when the arrival rate increases and scaling in when the arrival rate decreases. This scenario also will prove the stability of our model and will show the ability of our model to keep the AMF load around the optimal processing value.

The three other scenarios represent an underloaded system, a fully-loaded system and an overloaded system. Each scenario is repeated 30 times in order to compute confidence intervals. Those scenarios will allow to see the behavior of our model following different load patterns and to compare it to the behavior of the EWMA model.

B. Results Analysis

Following the scenarios' description, in this section we discuss the results of each scenario. The results of the first scenario are depicted in Figure 3. Following the arrival dispatching plots, we notice that the EWMA model schedules the requests randomly over the five AMFs (q_1 to q_5) independently from the arrival load ($lambda$), unlike the control model. From the AMF load level plots, we deduce that when the arrival rate increases, the EWMA model pushes the AMFs loads (X_1 to X_5) to their limits (30 process at a time), hence adding latency for each process. However, the control model dispatches the arrivals in a manner to have an optimal load (20 processes at a time) in each active AMF and thereby processing the requests with no additional latency. From the Number of Active AMF plot, we notice that the EWMA is not stable and falling in what is called a Zeno the phenomenon ("ping-pong"). The control model shows more stability and scales in/out the AMF instances accurately depending upon the needs.

The results of scenario 2 to 4 are depicted in Figure 4. Thanks to those scenarios we are able to compare the behavior of the control model, and the EWMA model in three traffic patterns: underloaded system;, fully-loaded system and overloaded system.

In the underloaded system plot, we notice that the EWMA model is using three AMFs as the control model. However, the EWMA load (X) is not balanced between the three AMFs, while the control model has an equivalent load over the three AMFs.

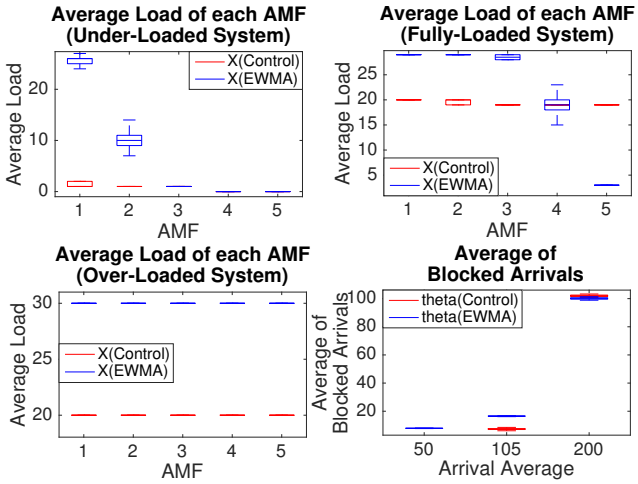

Fig. 4. Evaluation of the control model Behavior vs different arrival Load

In the fully-loaded scenario, we notice that the EWMA model scheduled randomly the arrivals over the five AMFs with an average of 28 processes at a time. Thus the EWMA model is loading the AMFs more than the optimal AMF load, and thereby adding latency while satisfying the arrivals. In contrast with the latter scenario, the control model transfers the arrivals to the five AMFs depending on their load in order to maintain the optimum AMF load as possible and so satisfying the requests with no additional latency. Despite the following scheduling solution, it is important to notice that the average number of blocked arrivals, in this category of tests, for the control model is lower than the EWMA model. Therefore, the control model is able to distribute load among the AMFs with neither additional latency nor blocked arrivals.

Finally, in case of overloaded conditions, unlike the EWMA model, the control model avoids fully loading the AMF, and tries to keep its load around the optimal AMF load. Thus, some additional arrival will be blocked as shown on the average of the blocked arrival's plot.

To summarize, the control model can schedule the arrivals depending on the AMFs load in order to maintain an optimal AMF load and to satisfy the request arrivals with no additional latency, especially that decreasing latency is an important requirement for 5G. Furthermore, the control model is designed to avoid resources' wastage by only activating (deploying) the exact number of needed AMF. Indeed, it can scale out the system in order to satisfy an arrival rate increase, if needed, while scale in when arrival rate decreases.

Unfortunately, reducing latency and scaling dynamically add some penalties on the system. As mentioned above, some sessions will be blocked when the system is overloaded or when scaling up. However, the system can remedy from this penalty by adding a prediction function allowing to deploy in advance a new AMF when arrival rate is predicted to increase.

VI. Conclusion

In this paper, we introduced an algorithm based on control theory in the scope of NGC work of 3GPP group. This algorithm allows to steer UE control traffic according to the AMF load, which maintain an optimal load (i.e. no UE blocked) in the AMFs' instances. Additionally, we showed that the design of this control helps saving resources by scaling out and in the AMF capacity as needed. Besides, procedures are proposed to be able to deploy the algorithm in the NGC architecture following the 3GPPP standards. Our future work is to extend this work and test it over a real platform in order to get practical results and validate its efficiency on the field.

Acknowledgement

EURECOM contribution has been partially funded by the European Framework Program under H2020 grant agreement No. 723172 5G!Pagoda project

References

[1] A. Nordrum, "Popular internet of things forecast of 50 billion devices by 2020 is outdated," IEEE Spectrum. [Online]. Available: https://goo.gl/5TS9s5

[2] 3rd Generation Partnership Project (3GPP), "System architecture for the 5g system." [Online]. Available: http://www.3gpp.org/ftp/specs/archive/23_series/23.501/

[3] D. Nowoswiat, "Managing lte core network signaling traffic." [Online]. Available: https://goo.gl/2pjygw

[4] A. S. Rajan and K. B. Ramia, "Application of nfv and sdn to 5g infrastructure," *Towards 5G: Applications, Requirements and Candidate Technologies*, pp. 408–420, 2016.

[5] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "Nfv: state of the art, challenges, and implementation in next generation mobile networks (vepc)," *IEEE Network*, vol. 28, no. 6, pp. 18–26, 2014.

[6] T. Phung-Duc, Y. Ren, J.-C. Chen, and Z.-W. Yu, "Design and analysis of deadline and budget constrained autoscaling (dbca) algorithm for 5g mobile networks," *arXiv preprint arXiv:1609.09368*, 2016.

[7] Y. Ren, T. Phung-Duc, J.-C. Chen, and Z.-W. Yu, "Dynamic auto scaling algorithm (dasa) for 5g mobile networks," in *GLOBECOM*. IEEE, 2016.

[8] S. Jeon, D. Corujo, and R. L. Aguiar, "Virtualised epc for on-demand mobile traffic offloading in 5g environments," in *CSCN*. IEEE, 2015.

[9] Y. Takano, A. Khan, M. Tamura, S. Iwashina, and T. Shimizu, "Virtualization-based scaling methods for stateful cellular network nodes using elastic core architecture," in *CloudCom*. IEEE, 2014.

[10] X. An, F. Pianese, I. Widjaja, and U. G. Acer, "Dmme: A distributed lte mobility management entity," *Bell Labs Technical Journal*, vol. 17, no. 2, pp. 97–120, 2012.

[11] A. Banerjee, R. Mahindra, K. Sundaresan, S. Kasera, K. Van der Merwe, and S. Rangarajan, "Scaling the lte control-plane for future mobile access," in *CoNEXT*. ACM, 2015, p. 19.

[12] J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado, and J. M. Lopez-Soler, "Modeling and dimensioning of a virtualized mme for 5g mobile networks," *arXiv preprint arXiv:1703.04445*, 2017.

[13] G. Premsankar, K. Ahokas, and S. Luukkainen, "Design and implementation of a distributed mobility management entity on openstack," in *CloudCom*. IEEE, 2015, pp. 487–490.

[14] *Network Functions Virtualisation (NFV); Management and Orchestration*. The European Telecommunications Standards Institute, December 2014.

[15] ETSI and 3GPP, "Ts 136 413." [Online]. Available: https://goo.gl/CK7qSG

[16] IETF, "DNS Push Notifications." [Online]. Available: https://goo.gl/GGuWeC

[17] M. C. de Oliveira, J. Bernussou, and J. C. Geromel, "A new discrete-time robust stability condition," *Systems & control letters*, vol. 37, no. 4, pp. 261–265, 1999.

[18] R. E. Kalman, "Contributions to the theory of optimal control," 1960.

[19] K. Zhou, J. C. Doyle, K. Glover *et al.*, *Robust and optimal control*. Prentice hall New Jersey, 1996, vol. 40.

[20] C. Wang, B. Li, Y. T. Hou, K. Sohraby, and Y. Lin, "Lred: a robust active queue management scheme based on packet loss ratio," in *INFOCOM 2004*, vol. 1. IEEE, 2004.