Latency Evaluation of a Virtualized MME

Jonathan Prados-Garzon, Juan J. Ramos-Munoz, Pablo Ameigeiras, Pilar Andres-Maldonado, Juan M. Lopez-Soler
Department of Signal Theory, Telematics and Communications
University of Granada
Granada, Spain
Emails: jpg@ugr.es, jjramos@ugr.es, pameigeiras@ugr.es, pam91@correo.ugr.es, juanma@ugr.es

***Abstract*—Network Virtualization is one of the key technologies for developing the future mobile networks. In this paper we propose an LTE *virtualized Mobility Management Entity* queue model to evaluate its service time for a given signaling workload. Additionally, we provide a compound data traffic model for the future mobile applications, and we predict theoretically the control workload that it will generate. Finally, we evaluate the *virtualized Mobility Management Entity* overall mean delay by simulation, providing insights for selecting the number of processing instances for a given number of users.**

***Index Terms*—virtualized MME, queue model, NFV, LTE.**

I. Introduction

Nowadays, telecom industry is regarding Network Virtualization as one of the key technologies in the future cellular networks. Network Functions Virtualization (NFV) offers the operators the possibility of running the network functions on industry standard high volume servers instead of using expensive, special purpose, and vendor-dependent hardware [1]. Concretely, NFV promises to enable organizations to: i) reduce capital and operational expenditures, ii) accelerate time-to-market of new services, iii) deliver agility and flexibility, and iv) scale up services on demand [1].

The present work focuses on estimating how the signaling plane workloads expected for the near future affects the service time of a *virtualized Mobility Management Entity* (vMME) which can scale its resources. The contribution of this paper is threefold. First, we propose a queue model based on [2] of a vMME in a datacenter, and we compute experimentally the service rates of the vMME processes. Second, we characterize theoretically and by simulation the control messages rate generated by the users's activity. Third, we characterize the service time of a vMME for different control plane workloads. As a result, we provide the estimation of the mean system delay depending on the number of users and vMME processing instances.

II. System Model

In this work, we assume a general LTE/EPC network architecture [3] based on NFV, with a logically centralized vMME, which runs in a cloud computing facility.

The User Equipments (UEs) run the users' applications that generate or consume network traffic, which, in turn, trigger the LTE network control procedures. These signaling procedures allow the control plane to manage the UE mobility and the data flow between the UE and PDN-GW. We only consider the ones that generate most signaling load [1], e.g., Service Request (SR), Service Release (SRR), and X2-Based Handover, which generate 3, 3, and 2 control messages to be processed by the vMME, respectively [3].

The eNodeBs (eNBs) receives the UEs signaling, and forwarding messages to the vMME. Each eNB contains an *user inactivity timer*, with an expiration time of T_I, to detect the users' inactivity and release network resources.

Regarding the vMME implementation, we consider the *1:N mapping* architectural option [4]. Thus, the vMME is split into 3 logical components: front end (FE), MME service logic (SL), and state database (SDB). The FE acts as a communication interface with other entities of the network and balance the load among several MME SLs, which implement the processing of the different control messages. The SDB stores the user session state making the MME SLs be stateless. Therefore, the number MME SLs can grow without affect on in-session users. Moreover, the vMME is seen like a single component from the rest of the network. Whenever the vMME processing capacity cannot withstand with the current control load, a new MME SL instance must be instantiated and a new processor is added to the processing resources pool. We assume that every processor in the data center facility provides the same computational power.

III. Application Traffic Models

Let us define a *session* as the UE activity elapsed between the instant the user launches a network application and the time instant he closes or stops using it. Likewise, *Application Activity Period (AAP)* is defined as the time interval in which the application sends or receives all necessary data to perform a single task, such as download a web page. A session consists of N $AAPs$ of length T_{on} separated by $N-1$ time intervals known as *reading times* of length D.

Let us define *Inter Arrival Session Time* $IAST$ as the time interval between the start of two consecutive *sessions*. And let's T_{sst} denote the *session standby time*, i.e., the time elapsed from the end of a session to the beginning of the next one. Let $\overline{T}_{sd} = \overline{N} \cdot \overline{T}_{on} + (\overline{N}-1) \cdot \overline{D}$ be the average session duration, then, $\overline{T}_{sst} = \overline{IAST} - \overline{T}_{sd}$. We suppose that T_{sst} is exponentially distributed and $\overline{IAST} = 1200$ seconds.

When a session begins, the UE chooses a certain application with a given probability P_{app}. Three types of applications are

TABLE I. Traffic models characterization

Traffic Type	Parameters	Statistical Characterization
Web browsing (HTTP) $P_{app} = 0.74$	Main Object Size	Truncated Lognormal Distribution: μ=15.098 σ=4.390E-5 min=100 B max=6 MB
	Embedded Object Size	Truncated Lognormal Distribution: μ=6.17 σ=2.36 min=50 Bytes max=2 MBytes
	Embedded Objects per Page	Truncated Pareto Distribution: mean=22 shape=1.1
	Parsing Time	Exponential Distribution: mean=0.13 seconds
	Reading Time	Exponential Distribution: mean=30 seconds
	Pageviews per session	Geometric Distribution: p=0.893 mean=9.312
HTTP progressive video $P_{app} = 0.03$	Video Encoding Rate	Uniform distribution with ranges: $(2.5, 3.0)$ Mbps / (4.0,4.5) Mbps / $(12.5, 16.0)$ Mbps / $(20.0, 25.0)$ Mbps, for equiprobable itags: 137 / 264 / 266 / 315 respectively.
	Video Duration	Distribution extracted from [5]
	Reading Time	Exponential Distribution: mean=30 seconds
	Videoviews per session	Geometric Distribution: p=0.6 mean=2.5
Video calling $P_{app} = 0.23$	Call Holding Time	Pareto Distribution: k=-0.39 s=69.33 m=0
	Calls per session	Constant = 1

considered in this work: i) web browsing, ii) HTTP progressive video and iii) video calling. The statistical characterization of these application models are summarized in Table I.

IV. Signaling procedures rate characterization

In this section we derive mathematical expressions to predict the rate of control procedure requests to the vMME.

An SR procedure occurs when a UE application is going to start an AAP without having network resources assigned. We can compute the mean arrival rate of SR procedures, λ_U^{SR}, as:

$$\lambda_U^{SR} = \lambda_S \cdot ((\overline{N} - 1) \cdot P(D > T_I) + P(T_{sst} > T_I)) \quad (1)$$

where $\lambda_S = 1/\overline{IAST}$ denotes the session rate, and $P(D > T_I)$ and $P(T_{sst} > T_I)$ are the probabilities that the user inactivity timer expires during any reading time and session standby time, respectively. Let X denote the time interval between the end of two consecutive $AAPs$, regardless these activity periods belong to the same session or not. If $X \geq T_I$, the SRR procedure is triggered. Since each SR have a corresponding SRR, the mean SRR arrival rate $\lambda_U^{SRR} = \lambda_U^{SR}$.

A HR is used to hand over a UE from a source eNB to a target eNB. Let P_{UA} be the probability that a user is active at a given time, and let CCR denote the mean user cell crossing rate. Then the mean HR arrival rate per user is:

$$\lambda_U^{HR} = CCR \cdot P_{UA} \quad (2)$$

Assuming that each user moves according the fluid-flow mobility model, i.e., at constant speed with a random direction uniformly distributed between $[0, 2\pi)$, it holds that

$$CCR = \frac{\overline{v} \cdot B}{\pi \cdot S} \quad (3)$$

where $\overline{v}$ is the mean user speed and B is the perimeter of the cell coverage area S. To compute P_{UA}, let us define T_{ua} as: $T_{ua} = X$ if $X \leq T_I$ and $T_{ua} = T_I$ otherwise. Thereby, the expected value of T_{ua} can be computed as:

$$\overline{T}_{ua}(X) = T_I \cdot P(X > T_I) + \int_0^{T_I} x \cdot f_X(x)\, dx \quad (4)$$

Finally, P_{UA} is λ_S times the amount of time that a user is active within a session:

$$P_{UA} = \lambda_S \cdot (\overline{N} \cdot \overline{T}_{on} + (\overline{N} - 1) \cdot \overline{T}_{ua}(D) + \overline{T}_{ua}(T_{sst})) \quad (5)$$

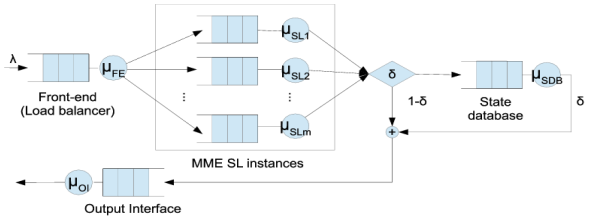

Fig. 1: Queue model of the vMME.

V. Queuing Model

To simulate the system architecture described in section II, we provide a queue model which considers the layout of 1:N mapping approach for *Virtualized Network Functions* (*VNFs*).

In our model, the SDB and the FE which balances the control requests among the SL processors are modeled with a single processor queue (Figure 1). The SL pool is modeled by a set of queues and processors to allow the parallel processing of the control messages. The SDB is accessed during each transaction with a probability δ. In our vMME implementation every transaction requires querying the database, $\delta = 1$.

VI. Numerical Results

A. Simulation Scenario

The simulation scenario is based on the dense urban information society scenario of the METIS project [6]. It is composed of 12 access points distributed regularly in a $4x3$ grid over a rectangular area of size $387m\,x\,552m$. The users move across the area following a fluid-flow mobility model. The user speed is uniformly distributed between 0 and $4.2m/s$. All users have a constant uplink and downlink data rate of $300Mbps$ [6]. The traffic models setup is found in Table I.

The service rates of our model for the FE, database and output interface are 120000 packets per second, 100000 transactions per second and 5000000 packets per second, respectively. They are based on *Amazon Elastic Compute Cloud (EC2)* [7].

B. NVF Processing Time Estimation

To calculate the system delay, we need to estimate the time a MME SL instance spends processing each control message.

Message	SR_1	SR_2	SRR_1	SRR_2	SRR_3	HR_1	HR_2
Inst. (M)	1.45	1.07	1.07	1.07	1.06	1.07	1.07
PT (μs)	127.4	94.0	94.0	94.0	93.2	94.0	94.0

TABLE II. Processing times (PTs) for the number of instructions measured, considering the *m3.xlarge* instance.

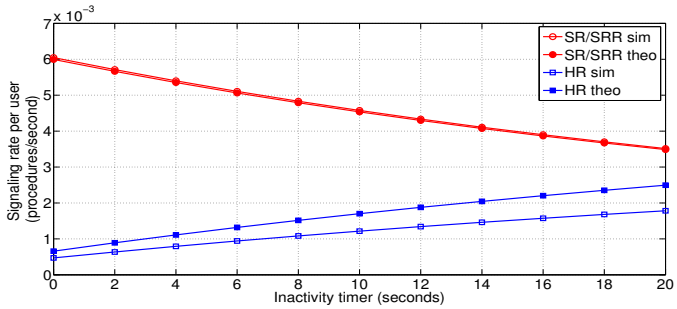

Fig. 2: Signaling arrival rates versus user inactivity timer.

This value depends on the type of control procedure served. Given a CPU processing capacity, we can estimate the delay of processing a message by assessing the average number of CPU instructions required for running a particular procedure.

We implemented in C the code of the functions which are invoked in the MME SL for each procedure. We measured the number of CPU instructions executed for every procedure by means of profiling tools. The delays was calculated for *EC2 m3.xlarge* instance which has an average computing capacity of $11.38 \cdot 10^9$ float operations per second [7] (see Table II).

C. Signaling Procedures Rate

To characterize the control messages arrival rate, we generated a signaling trace for 20000 users. Additionally, we validate the theoretical expressions (Equations 1 and 2) with these results. The results show that the SRs and SRRs rates decreases with T_I (see Figure 2). That is because the higher the timer value, the smaller the probability the timer runs out while the user is not within an AAP, avoiding the need for triggering news SR and SRR procedures. Conversely, the HRs rate increases with T_I, since the user remains active longer after an AAP. Consequently, there is a higher chance that a user is active when a cell crossing event takes place. The root-mean-square errors between the experimental and theoretical rates for SR and HR procedures ($4.07 \cdot 10^{-5}$ and $5.0 \cdot 10^{-4}$, respectively) demonstrate that the analytical expressions proposed are well fitted to the experimental data.

D. System Delay

In order to evaluate the delay of our system, we generated a signaling trace for 1200000 users and $T_I = 10$ seconds. The system delay grows exponentially with the number of users (see Figure 3). There is a point where the number of MME SL instances cannot withstand the control messages arrival rate and the system delay shoots up. At this point, a new MME SL instance must be added.

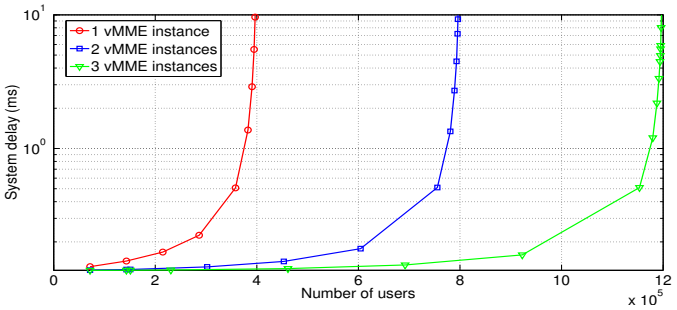

Fig. 3: Mean overall system delay.

From Figure 3, we could derive a criterion to calculate how many MME SL instances are needed to maintain the overall latency below a given threshold in this scenario. Assuming a system delay budget of 1 ms, we can predict the number of MME SL instances m given a number of users u as $m(u) = \lceil 2.50 \cdot 10^{-6} \cdot u + 6.36 \cdot 10^{-2} \rceil$. Please note that other traffic and processing times parameters may need a different equation.

VII. Conclusions

In this paper we propose a queue model of a vMME in a datacenter, estimating its processing time for several types of control procedures. Additionally, we have developed analytical expressions to predict the rate of UE signaling events for a given application traffic model. The accuracy of the proposed expressions has been verified by simulation. Using this framework we have characterized the service delay of the control signaling of a vMME which serves the traffic workloads expected in future mobile networks. Experimentally, we have obtained that, given a mean processing delay threshold of 1 ms, 3 MME SL instances are able to cope with the signaling control traffic generated by more than 1170000 users in a datacenter with nowadays processing power.

Acknowledgment

This work is partially supported by the Spanish Ministry of Economy and Competitiveness (project TIN2013-46223-P), FEDER and the Spanish Ministry of Education, Culture and Sport (FPU grant 13/04833).

References

[1] B. Hirschman *et al.*, "High-performance evolved packet core signaling and bearer processing on general-purpose processors," *Network, IEEE*, vol. 29, no. 3, pp. 6–14, May 2015.

[2] J. Vilaplana *et al.*, "A queuing theory model for cloud computing," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 492–507, 2014.

[3] *General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access*, 3GPP TS 23.401 Rel 12, 2014.

[4] T. Taleb *et al.*, "Ease: Epc as a service to ease mobile core network deployment over cloud," *Network, IEEE*, vol. 29, pp. 78–88, March 2015.

[5] P. Ameigeiras *et al.*, "Analysis and modelling of youtube traffic," *Trans. on Emerging Telecommun. Technologies*, vol. 23, pp. 360–377, June 2012.

[6] METIS, "Simulation guidelines (Deliverable D6.1)," Tech. Rep., 10 2013.

[7] A. Iosup *et al.*, "Performance analysis of cloud computing services for many-tasks scientific computing," *Parallel and Distributed Systems, IEEE Trans. on*, vol. 22, no. 6, pp. 931–945, 2011.