POLITECNICO DI TORINO
Repository ISTITUZIONALE

Characterizing Delay and Control Traffic of the Cellular MME with IoT Support

Original
Characterizing Delay and Control Traffic of the Cellular MME with IoT Support / Vitale, C., Chiasserini, C.F., Malandrino, F., Tadesse, S.S.. - In: IEEE TRANSACTIONS ON MOBILE COMPUTING. - ISSN 1536-1233. - STAMPA. - 20:4(2021), pp. 1325-1336. [10.1109/TMC.2020.2964677]

Availability:
This version is available at: 11583/2776776 since: 2021-03-06T12:37:30Z

Publisher:
IEEE

Published
DOI:10.1109/TMC.2020.2964677

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

12 October 2026

Characterizing Delay and Control Traffic of the Cellular MME with IoT Support

Christian Vitale, *Member, IEEE,* Carla Fabiana Chiasserini, *Fellow, IEEE,*
Francesco Malandrino, *Senior Member, IEEE,* and Senay Semu Tadesse, *Member, IEEE*

arXiv:2007.00107v1 [cs.NI] 1 Jan 2020

***Abstract*—One of the main use cases for advanced cellular networks is represented by massive Internet-of-things (MIoT), i.e., an enormous number of IoT devices that transmit data toward the cellular network infrastructure. To make cellular MIoT a reality, data transfer and control procedures specifically designed for the support of IoT are needed. For this reason, 3GPP has introduced the Control Plane Cellular IoT optimization, which foresees a simplified bearer instantiation, with the Mobility Management Entity (MME) handling both control and data traffic. The performance of the MME has therefore become critical, and properly scaling its computational capability can determine the ability of the whole network to tackle MIoT effectively. In particular, considering virtualized networks and the need for an efficient allocation of computing resources, it is paramount to characterize the MME performance as the MIoT traffic load changes. We address this need by presenting compact, closed-form expressions linking the number of IoT sources with the rate at which bearers are requested, and such a rate with the delay incurred by the IoT data. We show that our analysis, supported by testbed experiments and verified through large-scale simulations, represents a valuable tool to make effective scaling decisions in virtualized cellular core networks.**

I. Introduction

Massive Internet-of-things (MIoT) is an umbrella term for a fairly diverse set of applications, including smart factory, cloud robotics, automotive leveraging smart city sensors, and surveillance/security; as such, it represents one of the main motivations behind 5G [2]. For all these applications, service latency is a critical constraint, even more so than sheer network throughput. Also, IoT applications are characterized by a very high density of devices, up to 10,000 devices/km^2 [2, p. 6], and peculiar traffic patterns: devices may be inactive for a long time, and then multiple devices may transmit data in a (almost) synchronized manner.

Such traffic patterns are a poor match for the default procedures followed by the cellular core network and such a mismatch may jeopardize the application-latency requirements. Indeed, before a terminal can transmit data packets toward the cellular infrastructure, typically the following operations are required: authentication, identity verification, and bearer establishment. If the terminal remains silent longer than a timeout, the bearer is released and the whole procedure has to be performed again. Thus, for MIoT traffic, bearer instantiation (including bearer establishment and release) is one of the most critical tasks: using the default procedures would result in an exceedingly high latency and control overhead, compared to the data traffic generated by an MIoT device.

To cope with that, 3GPP has introduced a new standard [3], called Control Plane Cellular IoT Evolved Packet System (CIoT) optimization, which is already available in off-the-shelf products [4]. Such a standard (i) simplifies the procedures, roughly halving the associated overhead, (ii) uses the Mobility Management Entity (MME) of the cellular core network to forward user-plane traffic, and (iii) limits the involvement of MIoT sources in bearer establishment procedures, hence reducing the power consumption. Importantly, since under the CIoT optimization the MME is in charge of both control- and user-plane processing, it bears the brunt of MIoT traffic, thus becoming the pivotal component of the cellular core. It follows that the MME performance and the associated delay determine the ability of the network as a whole to support MIoT traffic.

Ensuring that the MME has sufficient computational capability to efficiently process the traffic load generated by MIoT sources becomes even more sensitive in the context of network *softwarization*. Such a paradigm refers to a global trend towards replacing special-purpose network equipment – including the entities of the cellular core [5] – with virtualized network functions (VNFs) running on general-purpose hardware. In the case of a virtual Evolved Packet Core (EPC) [6], the number of MME instances and their computational capability can be *scaled* to adapt to the variations in the current and expected MIoT traffic they must process. In particular, in the case of the MME, effective scaling requires:

- characterizing the relation between the number of MIoT sources and the arrival rate of bearer requests at the MME;
- modeling the impact of the MME capacity on the delay introduced by the bearer establishment procedure.

In this paper, we study both the above aspects with reference to the case where a network operating according to the CIoT optimization serves MIoT traffic. Specifically, our main contributions are as follows:

(i) We begin by characterizing analytically the time between consecutive bearer instantiation requests coming from MIoT sources, proving that it is well described by an exponential distribution;

(ii) By running and profiling the components of a real-world EPC implementation, we make some fundamental observations on the system that we then exploit to develop our analytical model;

A sketch of the present work was included in our poster presented at ACM Mobihoc 2019 [1].

C. Vitale is with KIOS Center of Excellence, Cyprus. C. F. Chiasserini and S. Tadesse are with Politecnico di Torino, Italy. C. F. Chiasserini and F. Malandrino are with CNR-IEIIT, Italy. C. F. Chiasserini and F. Malandrino are also with CNIT, Italy.

(iii) Leveraging the analytical results on MIoT traffic patterns and the experimental observations, we build an M/D/1-PS queuing model of the MME and study how the bearer instantiation time depends on (a) the traffic load, i.e., the arrival rate of bearer requests, and (b) the computational capability assigned to the MME itself. Importantly, we obtain a novel *closed-form* expression for the packet forwarding delay as a result of our analysis;

(iv) We show that the obtained analytical results represent a powerful tool to drive real-time scaling decisions in softwarized cellular networks when dealing with delay-sensitive applications;

(v) We validate our analysis through large-scale simulations using both a synthetic traffic model based on the 3GPP standard, and a real-world scenario including topology and mobility information from the city of Monte Carlo, Monaco.

The remainder of the paper is organized as follows. After introducing our system model in Sec. II, we present our analysis and a closed-form expression for the characterization of the bearer request arrival process in Sec. III. In Sec. IV, we run some experiments and make useful observations to develop our analytical model of the cellular core network. Furthermore, we characterize the core network delay performance. Through detailed simulations using both synthetic and real-world traffic, in Sec. V we show how our analysis can be used to effectively tune the computational capability of a vEPC. Finally, we review related work in Sec. VI and conclude the paper in Sec. VII.

II. System Model and Preliminaries

Here, we present the CIoT bearer instantiation procedure [3] and how the MIoT traffic is served when such a procedure is adopted. In particular, we consider the current cellular network, namely, the Evolved Packet Core (EPC), which is briefly introduced in Sec. II-A. Then we detail the CIoT bearer instantiation procedure and its relevance to the NB-IoT standard in Sec. II-B. Finally, we describe the model we adopt for the IoT traffic, in Sec. II-C.

A. Evolved packet core network

IoT cellular traffic has to traverse the EPC network, which includes four main components, as depicted in Fig. 1:

- the Serving Gateway (S-GW) mainly routes data traffic and acts as anchor point when User Equipments (UEs) move from one eNB to another;
- the PDN Gateway (P-GW) acts as ingress and egress point of the mobile access network; it is also the responsible for policy enforcement;
- the Mobility Management Entity (MME) is the termination point of UE control channels. The MME authenticates and tracks registered UEs and, most importantly, it handles bearer activation, i.e., it is the MME that creates a data path between the UEs and the P-GW. When CIoT optimization is in place, the data path between the UE and the P-GW includes the MME itself, since the MME is also responsible for relaying the traffic of the MIoT sources to the correct S-GW (see Fig. 1);

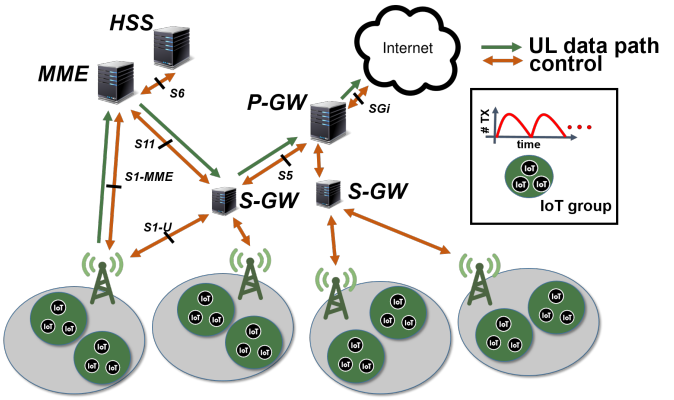

Fig. 1. EPC architecture.

- the Home Subscriber Server (HSS) is a central database where UE-related information is stored. The HSS assists the MME in UE authentication.

Note that the MME is connected to the S-GWs for bearer establishment and, under the CIoT optimization, it also performs packet decryption/forwarding, while the P-GW handles the data traffic to/from several S-GWs. Importantly, in the case of a vEPC, the MME, P-GW, and S-GW typically run on different (virtual) machines whose number and capability can be adjusted as needed.

B. CIoT bearer instantiation procedure

Exactly as any other cellular transmitter, an MIoT source sends or receives data traffic only if a logical connection with the corresponding P-GW is in place, i.e., if the MME has completed the bearer instantiation procedure. However, unlike the ordinary procedure, the CIoT optimization foresees that bearers are released immediately after packet transmission/reception, unless an MIoT source explicitly signals the presence of imminent traffic. As a consequence, an established bearer lasts for quite a short time and no handover procedure is typically required for MIoT traffic at the MME level. In the following, we therefore focus only on the performance of the MME when handling bearer instantiation procedures.

As depicted in Fig. 2, each time an MIoT source has to transmit a packet, five operations are performed: (i) authentication, (ii) identity verification, (iii) bearer establishment, (iv) forwarding (after data decryption and integrity check) of the data packets piggybacked by the MIoT source in the Radio Resource Control (RRC) Early Data Request message, and (v) bearer release. Specifically, hereinafter *bearer establishment* will refer to the set of operations comprised between step 1 and step 6 (included) in Fig. 2. We remark that such a procedure represents a crucial contribution to the data forwarding latency, and it cannot be overlooked in the MIoT data delay computation. Indeed, the time needed to complete a bearer establishment also corresponds to the delay incurred by the data transfer within the EPC.

Finally, it is worth remarking that CIoT well pairs up with the Narrowband IoT (NB-IoT) standard – both being specifically designed to support massive IoT traffic, taking,

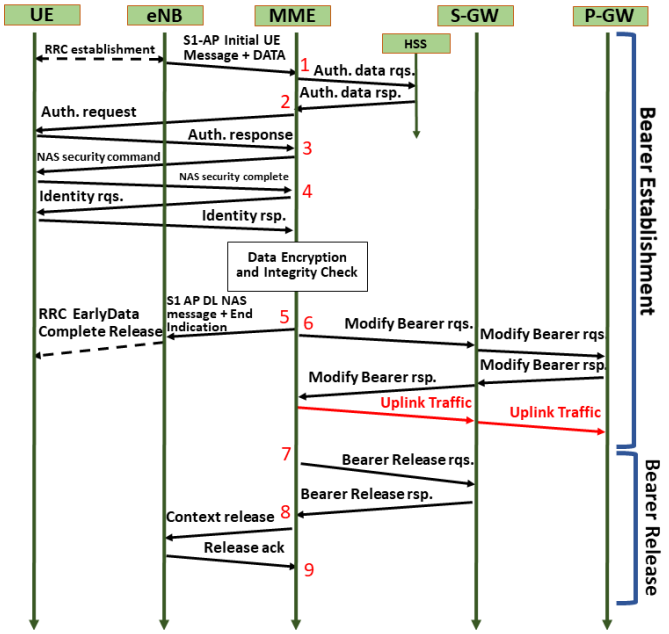

Fig. 2. CIoT bearer instantiation procedure and uplink data transmission.

respectively, the core network side and the radio access perspective. Indeed, NB-IoT is an IoT system built from existing LTE functionalities, which aims to support over 50,000 low-data rate stationary devices within a cell-site sector [7]. Beside defining an energy-efficient and robust physical layer for enhanced indoor coverage, NB-IoT also effectively addresses cell search, synchronization, and random access for initial link establishment. Specifically, according to NB-IoT, the RRC establishment on the top-right of Fig. 2 summarizes the following steps [8]: (a) the UE transmits a random access preamble; (b) the eNB replies with a random access response including a timing advance command and which uplink resources are assigned the UE to perform (c); (c) the UE transmits its identity; (d) the eNB transmits a message to resolve any contention due to multiple UEs accessing the channel (in step (a)) using the same preamble.

C. IoT traffic model

As mentioned above, after data transmission/reception, the MIoT source's bearer is released and a new bearer has to be established if later on the MIoT source has some more traffic to send/receive. Intuitively, depending on the IoT traffic pattern, the time between subsequent data packets may vary significantly, and so does the rate of bearer instantiation requests of an MIoT source. In order to characterize the arrival process of bearer requests, in the following we consider the traffic model described by the 3GPP standard [9] for machine type communications.

In [9], MIoT sources are organized in groups. Reflecting real-world operation conditions, [9] envisions quasi-synchronous packet transmissions within a group. This represents, for example, a group of sensors monitoring a geographical area, programmed to raise an alarm when a specific event occurs. After the occurrence of the event of interest, e.g., a gas leak, the closest sensors to the event raise the alarm. Sensors neighbouring the area where the event occurred react to the event subsequently, with a delay due to the propagation of the phenomenon. Such an effect triggers alarms from all the sensors belonging to the group, with a peak of alarms (hence, of traffic) roughly at the center of a period and an aggregate traffic distribution over time that follows a Beta(3,4). In [9], the events, and the related group transmissions, occur in subsequent periods of duration T, each of which with an aggregate traffic distribution over time following a Beta(3,4).

Notice that this model is quite general. Indeed, modifying T allows us to account for different aggregate transmission rates, while setting the group size to 1 allows us to represent MIoT sources behaving independently from each other. Furthermore, as we explain later in Sec. III, the model can be easily adapted to include data aggregators (a.k.a. gateways) that, as often envisioned in sensor network applications, collect and forward the data packets generated within groups of MIoT devices.

The traffic model specified by the 3GPP standard represents the aggregate behavior of a set of MIoT sources. However, we are interested in characterizing the latency of the data transfers by individual sources, each of which requires a bearer instantiation. In order to address this issue, we leverage the data generation model for an individual IoT device presented in [10], which results in an aggregate group traffic that still matches the Beta(3,4) distribution specified by the 3GPP standard.

In [10], each MIoT source is modeled as a Markov chain including two states, named *regular operation* and *alarm*, and hereinafter denoted with R and A, respectively. The period T of the IoT traffic pattern is divided into an arbitrary number N of slots, each of duration δ. In state A, the MIoT source sends packets according to a Poisson process with mean λ_A, which, without loss of generality and consistently with [10], we set equal to 1 packet/slot, i.e., an MIoT source successfully transmits at least one data packet with probability $(1-e^{-1})$. Introducing such a probability of transmitting (at least) one packet while being in state A allows capturing communication aspects that may arise in real-world IoT scenarios, e.g., transceiver failure or harsh propagation conditions due to the unfavourable IoT location. In state R, instead, the MIoT source transmits packets with an arbitrary small rate ϵ, representing, e.g., keep-alive or synchronization messages (in [10] the average transmission rate in state R is set to λ_R=0.0005 packet/s).

In each slot n ($n = 1, ..., N$) within a period T, the MIoT source may move from one state to the other. When in A, the source moves to R in the next time slot with probability 1. When in R, the source moves to A in time slot n with probability mass function (pmf)[1] $f_b(n)$, which depends on the considered slot in the period. As shown in [10], $f_b(n)$ can be

[1]The pmf of a discrete random variable x at n will be denoted by $f_x(n)$. The evaluation at n of the pmf of x conditioned to the random variable y, when $y = m$, will be denoted by $f_x(n|y{=}m)$ Also, we will indicate with $\mathbb{P}(X)$ the probability of a specific event X. The probability density function (pdf) and the cumulative density function (CDF) of a continuous random variable x will be denoted by $f_x(y)$ and $F_x(y)$, respectively.

obtained from the sampling of the Beta(3,4) shape[2], as follows:

$$f_b(n) = \text{Beta}\left(\frac{n\delta}{T}\right)\frac{\delta}{T} = 60\left(\frac{n\delta}{T}\right)^2\left(1-\frac{n\delta}{T}\right)^3\frac{\delta}{T}. \quad (1)$$

In summary, given the above IoT traffic model, on average, an MIoT source visits state A once every T seconds, and therein it transmits a packet with probability $(1-e^{-1})$. Instead, when an MIoT source sojourns in state R, it transmits a packet every $1/\epsilon$ s. In the following, we assume that ϵ is small enough so that we can neglect the occurrence of a packet transmission in state R (this holds, e.g., setting ϵ to the value suggested in [10]).

Finally, from (1), we observe that an MIoT source moves from state R to state A with a probability that only depends on slot n, not on the past, nor on the activity of other MIoT sources. It is important to point out two aspects that justify the use of such a model:
(i) even if in the model the single IoT traffic does not depend on the past, the obtained overall aggregate traffic of a group still follows the Beta(3,4) distribution suggested by the 3GPP specification, i.e., the group aggregate traffic follows the typical pattern of a set of sensors reacting to a specific event;
(ii) from the perspective of the core network, it does not matter which IoT sensor within a group triggers an alarm; indeed, each IoT transmitter within a group performs the same bearer instantiation procedure and introducing spatial correlation between IoT sensors activity does not have any impact on the MME load and the resulting distribution of the packet forwarding latency.

TABLE I
TABLE OF NOTATIONS

Symbol	Variable
δ	slot duration
T	time between events monitored by MIoT groups
N	number of slots in a period T
Q	number of MIoT sources served by the EPC
$f_b(n)$ ($f_b(n)$)	probability of transition from R to A in slot n (time t)
β	time between bearer requests at the EPC
s	slot (time) of the last bearer request
s_q	slot (time) of the last bearer request by source q
α	time between the last bearer request and the next transition to A by any source
α_q	time between the last bearer request and the next transition to A by source q
ω_q	offset of the time reference of source q with respect to source 0
$E(z,\lambda_\alpha)$	Erlang CDF with shape z and rate λ_α
λ_x	rate of the exponential random variable x
O_X	number of CPU operations per bearer procedure for EPC entity X
C_X	capacity, in CPU operations/s, of entity X
d	time between a bearer request and its completion,
v	delay due to the MME of the bearer establishment procedure
K	constant delay due to all EPC entities, other than the MME, in bearer establishment

[2]Note that the Beta(3,4) distribution is only defined in $[0,T]$.

III. IoT Control Traffic Characterization

To evaluate the delay performance of the MME when the CIoT optimization is supported, we first prove that the arrival process of the bearer instantiation requests at the MME follows a Poisson distribution. To this end, in this section we derive $F_\beta(\cdot)$, the cumulative distribution function (CDF) of the time interval between subsequent bearer instantiation requests at the MME. The steps we perform are summarized below:
(i) we observe that, under the CIoT optimization, every time an MIoT source has a new packet to transmit, the MME has to establish a new bearer and forward the packet to the right S-GW. Thus, the time interval between subsequent bearer instantiations by the MME corresponds to the time interval between packet transmissions by any of the MIoT sources served by the MME;
(ii) we then derive $F_\alpha(\tau|s=t)$, the distribution of the time interval between a packet transmission by any source in the system and the subsequent visit to state A by any, potentially different, MIoT source;
(iii) for $\delta \to 0$, we prove that such a distribution does not depend on the time of the last transmission in the system and turns out to be exponential. Furthermore, the result holds also for the time interval between subsequent packet transmissions, i.e., the inter-arrival time of bearer requests at the MME.

All notations we adopt are summarized in Table I; we also mention that the term "packet transmission" is often used interchangeably with "bearer request".

A. Inter-arrival time between bearer requests

In the following, we consider Q MIoT sources served by the same MME, generating traffic according to the 3GPP model described in Sec. II-C. As the first step, we fix to k the time slot at which the last transmission in the system occurred and we compute $f_\alpha(m|s{=}k)$, i.e., the probability density function (pdf) of the time interval between k and the slot in which the first device, among the Q MIoT sources, moves to state A. It is easy to see that $f_\alpha(m|s{=}k)$ can be written as the minimum over the time intervals between k and the first visit to A of the Q MIoT sources, i.e.,

$$f_\alpha(m|s{=}k) = f_{\min(\alpha_q)}(m|s{=}k). \quad (2)$$

In the above expression, α_q is the time interval between k and the transition to state A of the MIoT source q, and $f_{\min(\alpha_q)}(\cdot)$ is the pdf of the minimum over the α_q's.

Using (2) and considering the fact that in the adopted MIoT traffic model, MIoT packet transmissions are independent of each other, the CDF $F_\alpha(m|s{=}k)$ can be obtained as the minimum among random variables:

$$F_\alpha(m|s{=}k) = 1-\prod_{q=1}^{Q}\left(1-F_{\alpha_q}(m|s{=}k)\right). \quad (3)$$

As already mentioned, an MIoT source moves from state R to state A with a probability that depends only on the slot within period T corresponding to time k, i.e., on k only and not on the past. Thus, in the following proposition, we can prove that (3) can be computed as if any MIoT source q transmitted

its last packet in k, i.e., denoting with s_q the slot of the last packet transmission[3] by q, $F_{\alpha_q}(m|s{=}k) = F_{\alpha_q}(m|s_q{=}k)$, $\forall q$. This is an important property, which allows us to greatly simplify the subsequent derivations.

Proposition 1. *Denote with s the variable representing the slot in which the last packet transmission in the system by any of the MIoT sources occurred, and with α_q the time interval between slot s and the subsequent transition of the q-th MIoT source to state A. Since the IoT traffic does not depend on the past, α_q can be computed as if the last packet transmission in the system was by q. Denoted with s_q the slot of the last packet transmission by q, for a sufficiently large number of slots in period T, i.e. for a small δ/T, we get:*

$$F_\alpha(m|s{=}k) = 1{-}\prod_{q=1}^{Q}\left(1{-}F_{\alpha_q}(m|s_q{=}k)\right). \tag{4}$$

Proof. The proof of the proposition can be found in Appendix A in the Supplemental Material. □

The above proposition tells us that $F_\alpha(m|s{=}k)$ can be derived by analyzing the dynamics of the individual MIoT sources separately, i.e., through the CDF of the time interval between the last transmission by q and the subsequent visit to state A of q itself, which is significantly easier to compute than using $F_{\alpha_q}(m|s{=}k)$.

Next, we rewrite $F_{\alpha_q}(m|s_q{=}k)$ accounting for the time reference of source q. To this end, we recall that each source belongs to a specific group and it is quasi-synchronized only with the IoT sources belonging to that group, while different groups may exhibit a temporal offset with respect to each other [4]. By taking as global reference the time of source 0, we denote with $\omega_q \in \{0, ..., N{-}1\}$ the time offset between source 0 and the q-th source ($q = 1, \ldots, Q-1$). Then (4) can be rewritten as:

$$F_\alpha(m|s{=}k) = 1{-}\prod_{q=1}^{Q}\left(1{-}F_{\alpha_q}(m|s_q(k,\omega_q))\right), \tag{5}$$

where $s_q(k,\omega_q){=}\mathrm{mod}(k{+}\omega_q, N)$ and

$$F_{\alpha_q}(m|s_q(k,\omega_q)) = \sum_{x=1}^{m} f_b(\mathrm{mod}(s_q(k,\omega_q){+}x, N)) \cdot \prod_{y=s_q(k,\omega_q)+1}^{s_q(k,\omega_q)+x-1}\left[1{-}f_b(\mathrm{mod}(y,N))\right], \tag{6}$$

with $f_b(n)$ being the transition probability from R to A given in (1). In (6), $F_{\alpha_q}(m|s_q(k,\omega_q))$ has been derived considering the probability that the following sequence of events takes place: no transition into state A for $m{-}1$ slots, and a transition into state A, exactly m slots after $s_q(k,\omega_q)$.

We now switch to continuous time and evaluate the system dynamics when the slot duration δ tends to 0. We recall that the duration of slot δ is arbitrary and it only affects the number of slots within a period of activity of a group, without affecting the MIoT traffic model. Let us denote with t the reference time of MIoT source 0, with $s_q(t,\omega_q)$ the time instant of MIoT source q in its period corresponding to t, i.e., $s_q(t,\omega_q){=}\mathrm{mod}(t{+}\omega_q, T)$, and with τ the interval from the last transmission in the system to the time of the first transition from R to A by any of the MIoT sources.

First, for $\delta \to 0$, we rewrite (5) and (6), respectively, as,

$$F_\alpha(\tau|s{=}t) = 1{-}\prod_{q=1}^{Q}\left(1{-}F_{\alpha_q}(\tau|s_q(t,\omega_q))\right) \tag{7}$$

and

$$F_{\alpha_q}(\tau|s_q(t,\omega_q)) = \int_0^{\tau} f_b\left(\mathrm{mod}(s_q(t,\omega_q){+}x, T)\right) \cdot \prod_{y=s_q(t,\omega_q)}^{s_q(t,\omega_q)+x}\left(1{-}f_b(\mathrm{mod}(y,T))\,dx, \right. \tag{8}$$

where $f_b(x)$ can be obtained directly from (1) as,

$$f_b(x) = \frac{60\left(\frac{x}{T}\right)^2\left(1-\frac{x}{T}\right)^3}{T} \tag{9}$$

Looking at (7), one can see that, when the number of MIoT sources in the system grows, the time interval between a packet transmission and the subsequent visit to state A by any MIoT source decreases dramatically, since the minimum over a large number of positive random variables should be considered. Consequently, it is enough to provide an expression for $F_{\alpha_q}(\tau|s_q(t,\omega_q))$ that is accurate for small values of τ; given that, we can assume: $s_q(t,\omega_q){+}\tau < T, \forall\, s_q(t,\omega_q) \in [0, ..., T]$. Then a good approximation of $F_{\alpha_q}(\tau|s_q(t,\omega_q))$ for Q large, hence τ small, is given by:

$$F_{\alpha_q}(\tau|s_q(t,\omega_q)) = \int_0^{\tau} f_b(s_q(t,\omega_q){+}x) \prod_{y=s_q(t,\omega_q)}^{s_q(t,\omega_q)+x}\left(1{-}f_b(y)\right) dx\,. \tag{10}$$

Interestingly, the product form in (10) is the Volterra's product integral. Using such an integral expression in (10), we obtain:

$$F_{\alpha_q}(\tau|s_q(t,\omega_q)) = \int_0^{\tau} f_b\left(s_q(t,\omega_q){+}x\right) e^{\int_{s_q(t,\omega_q)}^{s_q(t,\omega_q)+x} -f_b(y)dy} dx. \tag{11}$$

Replacing (9) in (11) and solving both integrals, we get:

$$\begin{aligned} F_{\alpha_q}(\tau|s_q(t,\omega_q)) &= 1 - e^{-\frac{\tau}{T}\left(60\left(\frac{s_q(t,\omega_q)}{T}\right)^2\left(1-\frac{s_q(t,\omega_q)}{T}\right)^3\right)} \cdot e^{o(\tau^2)} \\ &\overset{(a)}{\approx} 1 - e^{-f_b(s_q(t,\omega_q))\tau}\,, \end{aligned} \tag{12}$$

where (a) holds for τ small. As a result, for Q large, $F_{\alpha_q}(\tau|s_q(t,\omega_q))$ follows an exponential distribution with rate parameter $f_b(s_q(t,\omega_q))$. Substituting (12) in (7), we obtain:

$$\begin{aligned} F_\alpha(\tau|s{=}t) &\approx 1{-}\exp\left(-\sum_{q=1}^{Q} f_b(s_q(t,\omega_q))\tau\right) \\ &= 1{-}e^{-\lambda_{\alpha|t}\tau}\,, \end{aligned} \tag{13}$$

[3]Since different groups are not syncronized with each other, i.e., time k corresponds to different slots within the period of different MIoT sources, $F_{\alpha_q}(m|s_q{=}k)$ depends on MIoT source q.

[4]Sources belonging to the same group have zero offset relatively to each other.

which states that, when Q grows large, $F_\alpha(\tau|s{=}t)$ follows an exponential distribution with rate $\lambda_{\alpha|t} = \sum_{q=1}^{Q} f_b(s_q(t,\omega_q))$.

Interestingly, under the above conditions, we can write:

$$\begin{aligned}\lambda_{\alpha|t} &\approx Q\int_0^T \mathbb{P}(s_q(t,\omega_q))f_b(s_q(t,\omega_q))d\omega_q \\ &= Q\int_0^T \frac{f_b(s_q(t,\omega_q))}{T}d\omega_q \\ &= \frac{Q}{T} \\ &\triangleq \lambda_\alpha\,, \end{aligned} \tag{14}$$

where we considered that:

- exploiting the law of large numbers, the experienced $\lambda_{\alpha|t}$ is approximated accurately by its average value;
- for high values of Q, also the number of IoT groups served by the MME grows large, hence the offsets ω_q can be assumed to be random variables uniformly distributed in $[0,T]$;
- such an observation holds also for $s_q(t,\omega_q)$, $\forall t$, since, by definition, $s_q(t,\omega_q) = \mathrm{mod}(t{+}\omega_q, T)$;

Note that (14) not only states that $F_\alpha(\tau|s{=}t)$ follows an exponential distribution, but also that such a distribution does not depend on t, i.e., $F_\alpha(\tau|s{=}t){=}F_\alpha(\tau)$.

We now use this result to compute the CDF of the inter-arrival time between subsequent bearer instantiation requests at the MME, i.e., $F_\beta(\tau)$. We account for the fact that not all transitions to state A by an MIoT source lead to a packet transmission: after a transition in state A by an MIoT source, the probability of transmitting at least a packet is equal to $1{-}e^{-1}$. Thus, we compute $F_\beta(\tau)$ considering that two subsequent transmissions in the system are separated by $z-1$ transitions to state A without any transmission. Given the fact that the time for a transition to state A is well described by an exponential distribution ($F_\alpha(\tau)$), we compute $F_\beta(\tau)$ as a sequence of z i.i.d. exponentially distributed time intervals, i.e., an Erlang(z,λ_α) distribution, weighted by the probability that two subsequent transmissions in the system are separated exactly by z transitions to state A. Denoting the Erlang(z,λ_α) distribution with $E(z,\lambda_\alpha)$, we write:

$$F_\beta(\tau) = \sum_{z=1}^{\infty} F_{E(z,\lambda_\alpha)}(\tau)(1-e^{-1})(e^{-1})^{z-1}. \tag{15}$$

In (15), we remark once again that the probability of transmitting at least one packet in state A is an input data to the model, and the specific value in [10] can be substituted with any arbitrary value (even 1, assuming that packets are always sent successfully upon visiting state A). Using the above results, we can prove the theorem below.

Theorem 1. *When the IoT group offsets are independent of each other and Q grows large, $F_\beta(\tau)$ is given by:*

$$F_\beta(\tau){=}1{-}e^{-\lambda_\beta\tau}$$

with rate parameter $\lambda_\beta = \frac{Q(1-e^{-1})}{T}$.

Proof. The proof of the theorem can be found in Appendix B in the Supplemental Material. □

The above result states that the inter-arrival time between bearer establishment requests at the MME follows an exponential distribution, which implies that *the number of requests that the MME, hence the EPC, receives in a time interval follows a Poisson distribution*. This is a key result that allows us to characterize first the control overhead due to bearer establishment and forwarding, and then the delay performance of the EPC. Note that the above result holds also in more general scenarios where there are aggregators relaying the data packets generated by the MIoT sources (and requesting for bearer instantiations) to the MME.

IV. EPC Model and Analysis

In this section, we begin by showing the results of our experimental study, which highlight the following facts: (i) the bearer establishment takes a deterministic amount of processing, (ii) the variation in the delay of EPC entities other than the MME is negligible, (iii) a PS well mimics the MME serving policy. To perform our validation, we run and profile the components of a real-world EPC implementation called OpenAirInterface (OAI) [11], as described in Sec. IV-A. Then, based on the above key observations, we analytically characterize the EPC control overhead and, using a M/D/1-PS model, we derive an expression for the delay experienced by the MIoT traffic within the EPC.

A. Understanding the EPC through the OpenAirInterface implementation

The OAI EPC is an implementation of the cellular core network where the MME and the HSS are implemented as separate entities, while the S-GW and the P-GW as a single unit (called SPGW). To investigate the interaction between the EPC and the IoT sources, we connected the OAI EPC to a software simulator of the Radio Access Network (RAN), called Open Air Interface Simulator (OAISIM). Herein, UEs and eNBs communicate with the OAI EPC through an Ethernet cable, sending and receiving control messages as if a real RAN was in place.

The use of OAISIM implicitly creates some limitations to our experimental results, the most important one the fact that OAISIM supports a maximum of 3 simulated UEs in our setting. Nevertheless, we use OAI EPC and OAISIM for our study because it is an open-source controlled environment where the behavior of the EPC and the UEs can be controlled at the millisecond time-scale. Also, importantly, OAI EPC is compliant with Release 10 functionalities, and off-the-shelf smartphones can connect to the OAI EPC. Finally, we mention that, even if OAI EPC implements the standard bearer establishment procedure, which includes a superset of the messages exchanged between the EPC entities during the CIoT bearer instantiation procedure, below we report the results considering only the messages included in the CIoT procedure, as depicted in Fig. 2.

The total number of CPU operations for each EPC entity, obtained by profiling the OAI EPC with the Callgrind tool from the Valgrind suite [12], is depicted in Fig. 3. Therein, the number of users attached to the EPC varies from 1 to

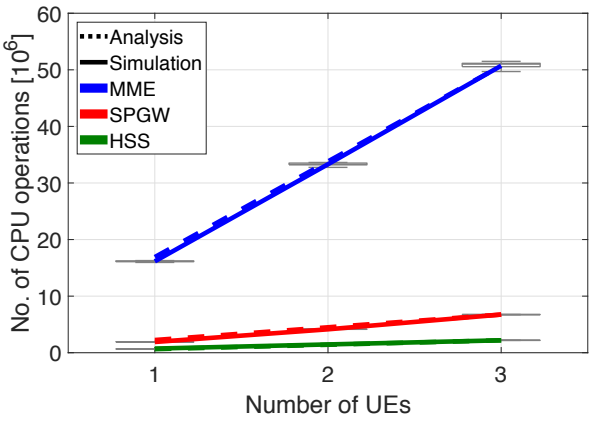

Fig. 3. Number of CPU operations required by bearer establishment vs. number of UEs.

3, and each data point has been obtained using 20 runs, resulting in a 95% confidence interval of up to $\pm 0.7\%$ of the plotted percentile values. Note also that, in each run, every UE performs one bearer establishment.

Fig. 3 demonstrates that the job size associated with a bearer establishment procedure is fixed and deterministic. This is shown by two facts: (1) the number of operations required for a bearer establishment procedure grows linearly with the number of bearer instantiation requests, and (2) the variation of the number of CPU operations required by the EPC entities across different runs is negligible. The former is further highlighted in the plot by the excellent match between the solid line, showing the experimental values, and the dotted line, which represents a linear fit whose slope is forced to the average number of CPU operations required by a single bearer instantiation. The latter fact, instead, can be observed from the boxplots in Fig. 3, representing the 10-th and 90-th percentile of the CPU operations distribution: the variance is very small in all analyzed cases and for any of the EPC entities in the system.

The second important observation we can make by looking at Fig. 3 is that the MME is the dominant component of the performance for the EPC: the number of operations required by any other entity is at most the 13% of those needed by the MME. Given the fact that typical EPC implementations include entities with similar computational capability [13], as the traffic load changes, it is fair to neglect the variations in the delay introduced by entities of the EPC other than the MME.

We also performed dedicated experiments to grasp some insights on the policy used by the OAI EPC implementation to serve packets that are simultaneously queued at the different entities. In this set of experiments, in order to avoid interference, we isolate MME, SPGW, and HSS, assigning to each entity a dedicated CPU core of the PC acting as EPC. Note that all the UEs emulated through OAISIM make a bearer request almost simultaneously and it is not possible to determine beforehand their time of attachment. Therefore, users contend for the same resources during nearly the whole duration of the bearer establishment. The first and second column of Table II show the average (over 20 runs) and the standard deviation of the time elapsing between the first and

TABLE II
BEARER ESTABLISHMENT TIME. 1 UE VS. 2 UES

1 UE Bearer Time Average	2 UEs Bearer Time 1^{st} User - Avg.	2 UEs Bearer Time 1^{st} User - PS Avg.
0.84 ± 0.02 s	1.63 ± 0.03 s	0.87 ± 0.02 s

the last packet processed by the MME with one UE and two UEs, respectively. In the case of two UEs, we only consider the data relative to the bearer establishment of the first UE. In the third column, we demonstrate that it is fair to assume that a PS policy is in place. Indeed, considering the time in the second column, and halving the time in which the procedures of the two UEs overlap, we obtain a value that is very close to the one in the first column.

Finally, we argue that our experimental results, although obtained for 3 users, have general validity. Indeed,

- the MME uses a PS policy to serve the incoming traffic and the number of served users does not have any impact on the service policy of the system. Note that such an observation is consistent with the fact the PS policy closely emulates the behaviour of a multi-threaded application running on a virtual machine instantiated on commodity hardware;
- the number of CPU operations required by a bearer establishment procedure using CIoT optimization is deterministic at any entity (as also described in, e.g., [14]), and does not depend on the number of on-going procedures;
- the MME is the computational bottleneck of the EPC, which is also evident given the load and the capacity values assigned to the EPC entities [13].

It is therefore fair to consider that the assumptions we make, based on our experimental findings, still hold as the number of IoTs grows.

B. Control Overhead and EPC Delay Characterization

As discussed above, the bearer establishment procedure in Fig. 2 requires a deterministic number of CPU operations. Then, at every entity X involved in the procedure, each bearer instantiation is characterized by a fixed number of CPU operations O_X, which is the sum of the CPU operations required by the messages in Fig. 2. It follows that the mean number of CPU operations per second that entity X has to perform is given by: $\mathbb{E}[\mu_X] = \lambda_\beta O_X$.

Next, we derive the pdf, $f_d(\tau)$, of the interval between a bearer request and its completion, i.e., the time passing from the first to the last message in Fig. 2. To this end, we exploit the fact that the inter-arrival time of bearer requests at the MME follows an exponential distribution, as well as the observations set out below, which have been derived through the experimental measurements.

(a) The MME is the main bottleneck of the control plane. As shown experimentally in Sec. IV-A, the computational load requested to the MME for a single bearer implementation is roughly one order of magnitude larger than the computational load requested to any other entity. This implies that the CPU utilization of entities other than the MME is very low and variations of the control

message processing times can be neglected, i.e., they can be considered as constant.

(b) As shown above, the MME serving policy can be modeled through a PS discipline.

(c) It is fair to assume that the duration of a specific bearer instantiation procedure is very short compared to the timescale at which the MME load varies. When the utilization of the MME is high, and making scaling decisions is critical, the number of competing messages at the MME is high as well. Thus, during the short time-scale of a bearer instantiation (in the order of milliseconds), the difference between the number of incoming and outgoing messages at the MME is negligible if compared to the number of queued messages. It follows that the fraction of capacity assigned, according to the PS policy, to an MIoT bearer request does not vary throughout a bearer instantiation procedure and the processing time of each message belonging to the same bearer instantiation is roughly the same (as in an M/D/1-PS queue). Thus, each bearer request can be considered as a single job, even if composed of multiple subsequent messages, with a computational load equal to O_{MME}.

Given the above observations and the result in Theorem 1, we model the MME as an M/D/1-PS queue, where the deterministic service time depends on the capability of the MME, while the rate of arrivals of the bearer instantiation requests is equal to λ_β, as reported in Theorem 1. Then $f_d(\tau)$ can be written as,

$$f_d(\tau) = f_v(\tau) + K, \tag{16}$$

where:

- $f_v(\tau)$ is the pdf of the time spent by a bearer instantiation at the MME, i.e., the sojourn time of a job in the M/D/1-PS queue;
- K is the constant delay due to entities other than the MME (see our observation *(a)* above), which can be computed as:

$$K = \frac{O_{UE}}{C_{UE}} + \frac{O_{eNB}}{C_{eNB}} + \frac{O_{HSS}}{C_{HSS}} + \frac{O_{S-GW}}{C_{S-GW}} + \frac{O_{P-GW}}{C_{P-GW}}, \tag{17}$$

where O_X is the total number of CPU operations that entity X has to perform for each bearer establishment, while C_X is the computational capability of entity X, expressed in CPU operations per second.

To derive $f_v(t)$, we leverage the results in [15], which, owing to the complexity of computing such a distribution, provides the following approximation for the CDF:

$$F_v(\tau) \approx \psi e^{-\gamma\tau}\,. \tag{18}$$

In the above equation, ψ is given by [15]:

$$\psi = \frac{(1-\rho)(\lambda_\beta - \gamma)}{2\lambda_\beta(1-\rho) - \gamma\rho(2-\rho)}\,,$$

where $\rho = \lambda_\beta D$ is the control traffic load at the MME, with $D = \frac{O_{MME}}{C_{MME}}$ being the deterministic service time of the bearer instantiation at the MME, and γ is the only positive solution of [15, Eq. (3.2)].

We remark that, given the pdf of the time interval between a bearer instantiation request and its completion (i.e., $f_d(\tau)$), we can compute the pdf of the delay that the control plane introduces in handling data packet forwarding at the MME when the CIoT optimization is supported. The derivation of the latter pdf implies considering only the messages in Fig. 2 that are exchanged till the data packet transmission is completed. Then, based on our earlier observation *(c)* and given the number of CPU operations required by each message, we can obtain the pdf of the MIoT traffic latency by properly scaling $f_d(\tau)$.

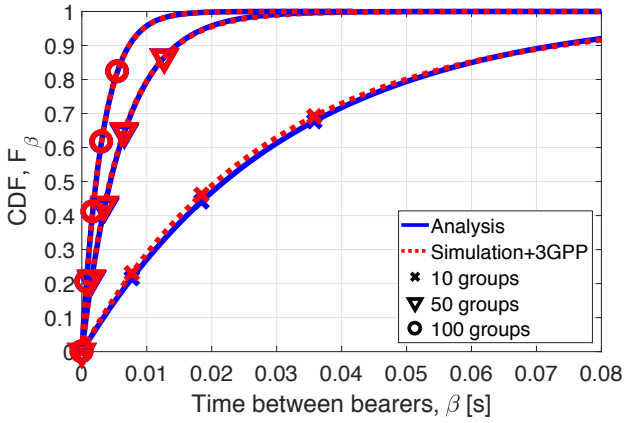

Fig. 4. Inter-arrival time distribution of bearer requests: analysis vs. simulation using the 3GPP traffic model.

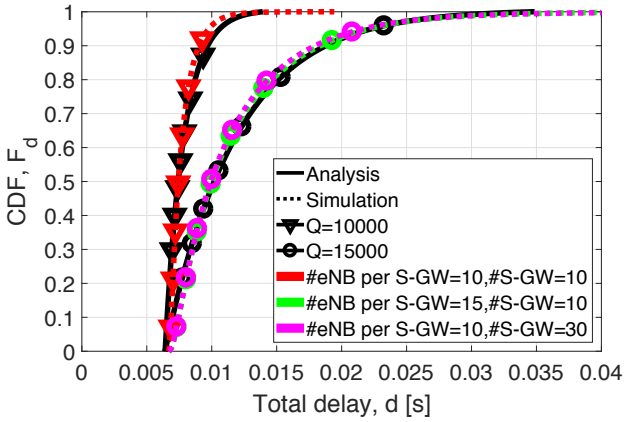

Fig. 5. Delay distribution: analytical vs. simulation results, using the 3GPP traffic model.

V. Model Validation and Exploitation

In the following, we show how the behaviors – inter-arrival times and bearer instantiation delays – predicted by our analysis match those yielded by extensive simulations, using both synthetic traffic models [3] (Sec. V-A) and real-world mobility traces (Sec. V-B). Furthermore, we demonstrate how our model can be leveraged in the dimensioning and management of vEPC networks handling MIoT traffic.

A. 3GPP synthetic traffic

We developed a Matlab simulator that accurately implements the 3GPP traffic model described in Sec. II-C. The

parameters we used are as follows: $T = 10\,\text{s}$ (as specified in [3]), $\delta = 10\,\mu s$, and group size equal to 50 MIoT sources.

Model validation. Here we validate the approximations introduced in our analysis as well as our main result in Sec. III (i.e., the inter-arrival time of bearer requests is exponentially distributed). To compare $F_\beta(\tau)$, computed as in Theorem 1, to the CDF of the inter-arrival time of bearer requests at the MME in simulation, we performed extensive experiments, varying the number of groups in the scenario and the offsets between them ω_q. In Fig. 4, we present the results obtained with a specific set of offsets, as the number of groups served by the MME varies; however, similar results have been also obtained changing the ω_q values.

With as few as 10 groups served by the MME, Fig. 4 highlights that simulation and analytical results closely match, thus showing that the exponential $F_\beta(\tau)$ captures very well the behavior of the 3GPP traffic model presented in Sec. II-C. Furthermore, as expected, the match between the two curves improves as the number of groups served by the MME grows.

We now validate our delay model presented in Sec. IV-B. We first remark that, for the analytical derivation of $f_d(\tau)$, we neglected the load due to the integrity check and decryption, at the MME. Indeed, while a single control message requires (roughly) one million floating-point operations [14], studies on commodity processors show that nowadays a 50-byte packet (as in the case of IoT applications) requires few hundreds of floating-point operations for encryption/decryption [16]. In our simulations, instead, we account for data encryption/decryption as well as integrity check at the MME. Second, to compute the constant delay component of the delay distribution, K, in (17), we proceed as follows:

- we obtained the number of CPU operations, O_X, required at the EPC entities by a bearer establishment through our experimental measurements described in Sec. IV-A, and
- we leveraged the work in [13], which provides the computational capability of the EPC entities, C_X, based on real-world data from a large mobile network operator.

Finally, in order to validate the analytical expression of $f_d(\tau)$, we extended our Matlab simulator to perform the whole procedure in Fig. 2, starting from the S1-AP Initial Message sent by the eNB. In our setup, all MIoT sources belonging to the same group, each containing 50 MIoT sources, are attached to the same eNB. Several eNBs may be attached to the same S-GW, while all S-GWs are attached to the same P-GW. Except for the RRC connection closing message sent by the eNB to the UE, all messages belonging to the same bearer instantiation travel sequentially between the involved entities, as foreseen by the CIoT optimization. Each entity is implemented as a PS server whose service rate matches the processing capability provided in [13].

Fig. 5 shows the analytical and experimental $F_d(\tau)$ in different scenarios. Specifically, we present the results of the CIoT optimization for two different values of traffic load, i.e., with $Q = 10,000$ and $Q = 15,000$. In the latter case, we also study two different configurations of the EPC to check whether changing the number of eNBs/S-GWs in the system has an impact on $F_d(\tau)$ or not.

First, we observe that the CDF of the bearer instantiation delay computed through (16)-(18) closely matches the experimental delay obtained via simulation – a fact that is especially evident looking at the tail of the CDFs. This result proves that considering the whole bearer establishment handshake as a single job at the MME, plus a constant delay due to the other entities, is a valid approximation. Small differences between the analytical and experimental CDFs for low values of delay, are mainly due to the model in [15], used to approximate the sojourn time in an M/D/1-PS queue. Indeed, due to the complexity of the M/D/1-PS characterization, [15] explicitly aims at modeling with higher accuracy the tail of the sojourn time CDF, which is what most matters in delay sensitive applications. Second, for $Q = 15,000$ the simulation results highlight that the two configurations with a different number of S-GWs provide exactly the same delay CDF, which validates our finding: $F_d(\tau)$ depends only on the number of MIoT sources in the scenario, and it is not affected by variations in the number of eNBs and S-GWs. This confirms that the MME delay contribution dominates that of the other EPC entities.

Model exploitation. We now show how our model can be used to develop efficient scaling algorithms for EPC networks serving MIoT traffic. Let us consider the following case, reflecting, e.g., a smart factory or cloud robotics application [17], where the delay introduced by the EPC should be less than $0.1\,\text{s}$ with 0.99 probability. Since the delay performance depends on the number of MIoT sources served by the EPC and on the capability of the EPC entities, we need an algorithm that, given the IoT traffic, scales the capability of the EPC entities according to the number of MIoT sources in the system. Such an algorithm can leverage the analytical expression of $F_d(\tau)$ we obtained.

As an example, we considered a simple threshold-based algorithm, which, as the number of active IoT sources grows, increases the computation capability of the EPC entities by 100%, and then by 150%, with respect to the initial value, depending on the MME delay predicted by our model (note that increasing the EPC capability by 100% can be realized by creating a new instance of its components). As shown in Fig. 6, such an algorithm meets the target performance. The figure also reports the delay corresponding to the cases when the capability values C_X are fixed to the initial value provided in [13], and to such a value increased by 100% or by 150%. Although more advanced scaling algorithms may be designed, we remark that, thanks to our model, even a simple threshold-based algorithm is able to meet the target delays and that our analysis, coupled with off-the-shelf virtualization tools like OpenStack, can be a key enabler to the support of IoT applications with delay guarantees.

B. Real-world trace

We now consider a real-world setting and leverage a large-scale mobility trace generated accounting for the MoST scenario [18]. The MoST scenario is a highly detailed representation of the mobility in the Monte Carlo urban area, including: (i) a multi-layered road topology, with tunnel and bridges; (ii) multi-modal mobility, e.g., users driving to a parking lot

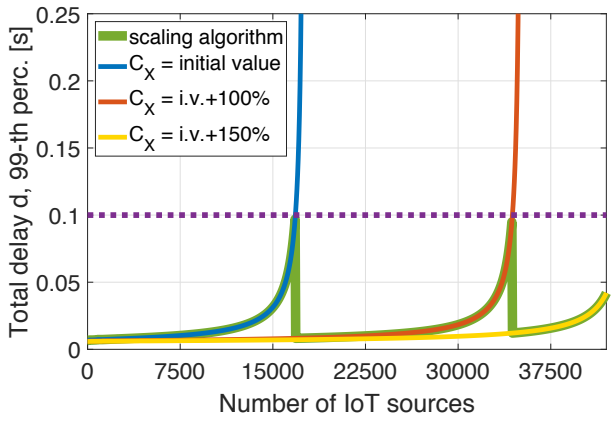

Fig. 6. Analysis exploitation: 99−th percentile of the bearer instantiation delay vs. number of MIoT sources. Performance obtained with: the initial computational capability (C_X) of the EPC entities set to the value provided in [13] (blue line), capability increased by 100% (red line), capability increased by 150% (yellow line), and capability determined through the scaling algorithm (green line). Dashed, purple line: target value of the 99−th percentile.

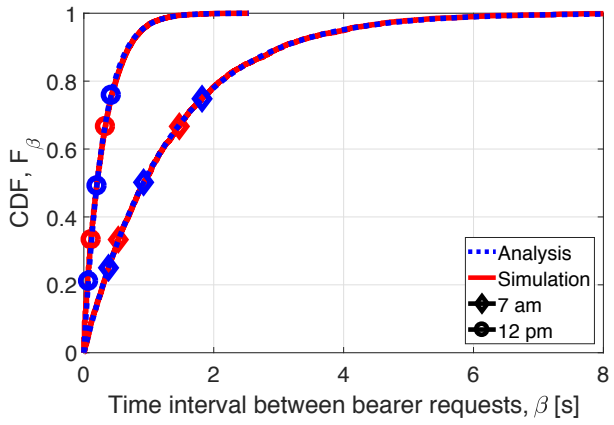

Fig. 7. MIoT trace: Comparison between analytical and empirical distribution of the inter-arrival time of bearer requests at the MME, for two representative time periods.

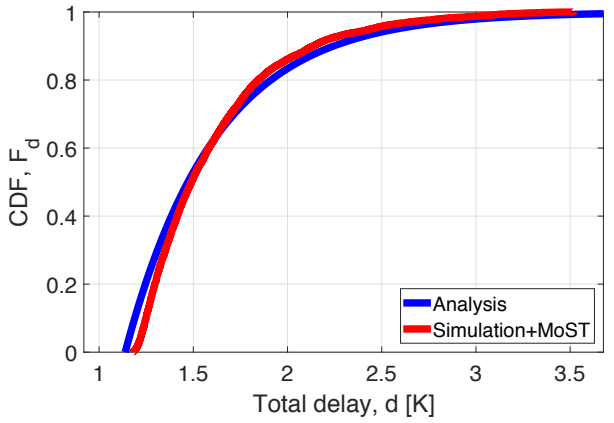

Fig. 8. Delay distribution of sensor traffic: Analysis vs. simulation, using the MoST trace.

and riding public transportation thence; (iii) multiple types of coexisting users, e.g., commuters and tourists. The scenario models the mobility of a total of $27,967$ users throughout an 8-hour period from 5 AM to 1 PM, and includes a total of 607 tagged points of interest (POIs) such as offices, restaurants, and tourist attractions. We assume that every time a user visits or stops at one of the POIs, a sensor, e.g., an identity-recognizing device, is triggered, resulting in a packet transmission, hence, a bearer instantiation request towards the MME serving the area.

Model validation. Our first objective is to establish whether the inter-arrival time between bearer requests obtained experimentally matches the exponential distribution $F_\beta(\tau)$ obtained through our analysis. To this end, we divided the time into 8 periods of one hour each, and computed the empirical distribution of the inter-arrival time of bearer instantiation requests in the MoST trace in every time period. The average arrival rates of bearer requests in the the various periods are very different, reflecting the daily fluctuations in mobility. Nevertheless, as exemplified in Fig. 7, the match between the analytical and the empirical distribution is excellent for all the time periods, proving that the inter-arrival time of the bearer requests obtained from the MoST trace follows an exponential distribution as well. This confirms that our analysis holds also for applications that do not follow explicitly the 3GPP traffic model described in Sec. II-C.

Then we used the bearer requests obtained from the MoST trace to evaluate if the delay distribution of the bearer request procedures can be approximated with $f_d(\tau)$ (as in (16)) also in this realistic IoT scenario. We fed to the previously mentioned Matlab simulator the time instants of the bearer requests by sensors in the MoST trace and, since the number of bearer requests is rather small even in the rush hour, we reduced the EPC entities capability of one order of magnitude. The analytical and simulation results for the rush hour are compared in Fig. 8, where the bearer establishment procedure delay is normalized by K (with K given in (17)). Given the fact that the largest difference between the two CDFs (which happens for low values of delay) is very small, we can conclude that our analysis well approximates the behavior of the EPC when serving MIoT sources, also in the case of a realistic scenario as the one of the MoST trace.

Model exploitation. We now present how our analytical results can be exploited under the MoST scenario. Fig. 9 shows the time evolution of the 99−th percentile of the bearer instantiation delay, for different values of capability C_X. Considering a target performance of 0.1 s, we observe that the simple scaling algorithm employed when deriving Fig. 6 (and which exploits our analytical results) successfully meets the delay requirement even under a sudden and very significant surge in the bearer request rate (see the black line in Fig. 9).

VI. Related Work

IoT support through cellular networks has recently attracted significant attention by both the scientific community and the standardization fora.

A first body of works deal with the requirements posed by IoT scenarios, and how 5G networks can cope with them.

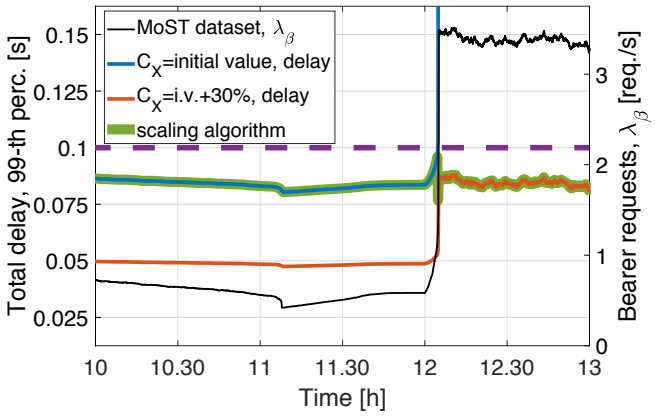

Fig. 9. Analysis exploitation: 99−th percentile of the bearer instantiation delay vs. time, when the MoST trace is used. Black line: bearer request arrival rate from the trace; dashed, purple line: target value of the 99−th percentile. Performance obtained with: the initial computational capability of the EPC entities (blue line), capability increased by 30% (red line), capability determined through the scaling algorithm (green line).

As an example, [19] quantifies the latency and capability requirements for the main IoT scenarios, from factory automation to parking machines, and discusses the improvement needed to both the radio access and core networks. The CIoT optimization and the role of MME are, however, not accounted for in [19], which mainly focuses on the SGW/PGW gateways. [17] has a narrower focus, namely, cloud robotics, and presents a working prototype; however, the latency introduced by the core networks and the entities therein is not taken into account. [20] and [21], instead, present the impact on the radio access, e.g., on Cloud-RAN, of IoT-specific physical layers, such as NB-IoT. Note that these works tackle the IoT support in cellular networks from the perspective of centralized IoT transmission coding/decoding, but they do not take into account IoT optimized control procedures, such as the CIoT.

Among the studies that do account for the core network, most, including [22], [23], envision a virtualized network, where network functions are implemented through VNFs. Unlike our work, [22] does not specifically target EPC or any of its entities. The authors of [23], focusing on multicast traffic in IoT scenarios, specifically study the MME delay. Their proposed solution is to endow the SGW with some of the MME tasks, the opposite of the CIoT optimization we consider in our study.

The use of NFV and SDN, for the implementation of the EPC under massive IoT traffic conditions, has been discussed in [24], while enhancements to the standard EPC can be found, e.g., in [25]. That work introduces new entities in the network architecture, which are specifically devoted to the IoT support. Importantly, although such solutions yield a remarkable performance improvement, they inevitably involve significant changes to the standard.

Analytical models of IoT systems have been developed for specific application use cases, like management [26], opportunistic crowd sensing in vehicular scenarios [27], or ambient backscatter devices [28]. Other works have presented theoretical models for the study of networking aspects such as the performance of middleware protocols [29], implemented between the application and the transport layer, or of the random access procedure in NB-IoT [30], [31]. Note, however, that none of the above works investigates the critical role of the EPC control plane in IoT-based systems; indeed few studies exist on the characterization of the overhead and service delay when the EPC handles massive IoT data traffic. In this context, the studies that are the most relevant to ours are [13] and its extension [32], which present a scheme for aggregating multiple IoT bearers and analyze the gain that is obtained with respect to the standard procedure. Such works, however, are based on deterministic inter-packet transmission time for MIoT sources and do not address the most recent and efficient 3GPP specifications for IoT support. Likewise, the study in [14] analytically evaluates the EPC control procedures (not specific to CIoT optimization) considering a simple IoT traffic model, coexisting with other cellular traffic sources. Furthermore, unlike our work, both [14] and [32] derive only the *average* processing latency of standard bearer establishments. A more comprehensive study on EPC control procedures has been presented in our conference paper [33], which, however, does not address any delay analysis.

To the best of our knowledge, our work is the first one presenting the delay distribution under the CIoT optimization and deriving an exponential inter-arrival time for MIoT bearer establishments at the MME. It is important to stress that, in our paper, exponential inter-arrival times are not an assumption, but the result of the analysis in Sec. III, which is based on the 3GPP MIoT traffic model [9] and has been validated in Sec. V. A sketch of our work has been presented in our poster publication [1]. Finally, we remark that the goal of our work differs significantly from that of [10], which presents the Markov Modulated Poisson Process (MMPP) model for individual IoT sources that we adopt to develop our analysis and that is in accordance with the 3GPP traffic model for IoT. Indeed, [10] investigates large-scale IoT scenarios via simulation only: it does not present any analytical model of the EPC or of its control procedures under IoT traffic support.

VII. Conclusions

Effectively dimensioning the cellular core network and evaluating its latency performance are crucial tasks for the support of massive IoT applications. Observing that the MME has a major impact on the control-plane latency, we characterized the statistics of the latency it introduces. In particular, we derived closed-form expressions that link the number of IoT devices to the inter-arrival times of bearer instantiation requests. Then, leveraging these results and an M/D/1-PS queue model of the MME, we characterized the latency experienced by IoT traffic in the cellular core. Importantly, our analysis is also based on findings obtained by measuring and profiling the performance of a real-world EPC implementation. Furthermore, using both the 3GPP traffic model and a real-world, large-scale mobility trace, we validated via extensive simulations the distribution of request inter-arrival times and of the IoT traffic latency.

We demonstrated that our model and results can be exploited to dimension the computation capabilities of the entities of

the virtualized cellular core as the offered IoT traffic load varies. Future work will leverage our results to design more advanced algorithms for the scaling of the resources allocated to virtualized EPC entities as the traffic load varies, and assess their performance using real-world EPC implementations.

ACKNOWLEDGMENTS

This work was supported by the European Commission through the H2020 project 5G-EVE (Project ID 815074). The work of Christian Vitale was also supported by the European Unions Horizon 2020 Research and Innovation Programme under Grant 739551 (KIOS CoE) and from the Republic of Cyprus through the Directorate General for European Programmes, Coordination, and Development.

REFERENCES

[1] C. Vitale, C. F. Chiasserini, F. Malandrino, and S. Tadesse, "Characterizing delay and control traffic of the cellular MME with IoT support," in *ACM MobiHoc Posters & Demos*, July 2019, pp. 1–2.

[2] M. Maternia, S. E. El Ayoubi, M. Fallgren, P. Spapis, Y. Qi, D. Martín-Sacristán, Ó. Carrasco, M. Fresia, M. Payaró, M. Schubert *et al.*, "5G PPP use cases and performance evaluation models," *see https://5g-ppp. eu/wp-content/uploads/2014/02/5G-PPP-use-cases-and-performance-evaluation-modeling_v1. 0. pdf*, 2016.

[3] 3GPP. (2017) Specification: 23.401; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/23_series/23.401/

[4] "Small data over NAS, S11-U and SGi interfaces," https://www.cisco.com/c/en/us/td/docs/wireless/asr_5000/21-3_N5-5/Ultra_IoT_CSGN/21-3-Ultra-IoT-CSGN-Guide/21-3-Ultra-IoT-CSGN-Guide_chapter_0111.pdf, accessed: 2018-21-12.

[5] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization," in *IEEE NetSoft*, 2015.

[6] H. Hawilo, L. Liao, A. Shami, and V. C. M. Leung, "NFV/SDN-based vEPC solution in hybrid clouds," in *IEEE Middle East and North Africa Communications Conference (MENACOMM)*, April 2018, pp. 1–6.

[7] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "NB-IoT system for M2M communication," in *2016 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, April 2016, pp. 428–432.

[8] Y. . E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3GPP narrowband internet of things," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 117–123, March 2017.

[9] 3GPP. (2014) Specification: 37.868; RAN improvements for Machine-type Communications. [Online]. Available: http://www.3gpp.org/ftp//Specs/archive/37_series/37.868/

[10] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," in *IEEE ISWCS*, 2013.

[11] "OpenAirInterface, 5G software alliance for democratising wireless innovation," http://www.openairinterface.org, accessed: 2018-14-12.

[12] Valgrind. (2018) The Valgrind tool suite. [Online]. Available: http://www.valgrind.org/info/tools.html

[13] G. Hasegawa and M. Murata, "Joint bearer aggregation and control-data plane separation in LTE EPC for increasing M2M communication capacity," in *IEEE GLOBECOM*, 2015.

[14] 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," *IEEE Transactions on Vehicular Technology*, 2017.

[15] R. Egorova, B. Zwart, and O. Boxma, "Sojourn time tails in the M/D/1 processor sharing queue," *Probability in the engineering and informational sciences*, 2006.

[16] "Encryption performance on commodity processors," https://calomel.org/aesni_ssl_performance.html.

[17] F. Voigtländer, A. Ramadan, J. Eichinger, C. Lenz, D. Pensky, and A. Knoll, "5G for robotics: Ultra-low latency control of distributed robotic systems," in *IEEE ISCSIC*, 2017.

[18] L. Codecá and J. Härri, "Towards multimodal mobility simulation of C-ITS: The Monaco SUMO traffic scenario," in *IEEE VNC*, 2017.

[19] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, 2017.

[20] Y. D. Beyene, R. Jantti, O. Tirkkonen, K. Ruttik, S. Iraji, A. Larmo, T. Tirronen, and J. Torsner, "NB-IoT technology overview and experience from cloud-RAN implementation," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 26–32, 2017.

[21] D. Darsena, G. Gelli, and F. Verde, "Cloud-aided cognitive ambient backscatter wireless sensor networks," *IEEE Access*, vol. 7, pp. 57 399–57 414, 2019.

[22] R. Vilalta, A. Mayoral, R. Casellas, R. Martínez, and R. M. noz, "SDN/NFV orchestration of multi-technology and multi-domain networks in cloud/fog architectures for 5G services," in *IEEE OECC*, 2016.

[23] M. Condoluci, G. Araniti, T. Mahmoodi, and M. Dohler, "Enabling the IoT machine age with 5G: Machine-type multicast services for innovative real-time applications," *IEEE Access*, 2016.

[24] A. Jain, N. Sadagopan, S. K. Lohani, and M. Vutukuru, "A comparison of SDN and NFV for re-designing the LTE packet core," in *IEEE NFV-SDN*, 2016.

[25] V. Nagendra, H. Sharma, A. Chakraborty, and S. R. Das, "LTE-Xtend: scalable support of M2M devices in cellular packet core," in *ACM SIGCOMM ATC Workshop*, 2016.

[26] Y. Zhang, W. Wang, N. Wu, and C. Qian, "IoT-enabled real-time production performance analysis and exception diagnosis model," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 3, pp. 1318–1332, July 2016.

[27] V. Petrov, A. Samuylov, V. Begishev, D. Moltchanov, S. Andreev, K. Samouylov, and Y. Koucheryavy, "Vehicle-based relay assistance for opportunistic crowdsensing over narrow IoT (NB-IoT)," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3710–3723, Oct 2018.

[28] D. Darsena, G. Gelli, and F. Verde, "Modeling and performance analysis of wireless networks with ambient backscatter devices," *IEEE Transactions on Communications*, vol. 65, no. 4, pp. 1797–1814, April 2017.

[29] G. Bouloukakis, I. Moscholios, N. Georgantas, and V. Issarny, "Performance modeling of the middleware overlay infrastructure of mobile things," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[30] R. Harwahyu, R. Ch eng, and C. Wei, "Investigating the performance of the random access channel in NB-IoT," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1–5.

[31] R. Harwahyu, R. Cheng, W. Tsai, J. Hwang, and G. Bianchi, "Repetitions versus retransmissions: Tradeoff in configuring NB-IoT random access channels," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3796–3805, April 2019.

[32] S. Abe, G. Hasegawa, and M. Murata, "Design and performance evaluation of bearer aggregation method in mobile core network with C/U plane separation," in *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 2017, pp. 1–8.

[33] C. Vitale, C. F. Chiasserini, and F. Malandrino, "On the impact of IoT traffic on the cellular EPC," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 1–6.