Media Query Processing For The Internet-of-Things: Coupling Of Device Energy Consumption And Cloud Infrastructure Billing

Francesco Renna, Joseph Doyle, Vasileios Giotsas and Yiannis Andreopoulos *Senior Member, IEEE*

***Abstract*—Audio/visual recognition and retrieval applications have recently garnered significant attention within Internet-of-Things (IoT) oriented services, given that video cameras and audio processing chipsets are now ubiquitous even in low-end embedded systems. In the most typical scenario for such services, each device extracts audio/visual features and compacts them into feature descriptors, which comprise media queries. These queries are uploaded to a remote cloud computing service that performs content matching for classification or retrieval applications. Two of the most crucial aspects for such services are: *(i)* controlling the device energy consumption when using the service; *(ii)* reducing the billing cost incurred from the cloud infrastructure provider. In this paper we derive analytic conditions for the optimal coupling between the device energy consumption and the incurred cloud infrastructure billing. Our framework encapsulates: the energy consumption to produce and transmit audio/visual queries, the billing rates of the cloud infrastructure, the number of devices concurrently connected to the same cloud server, the query volume constraint of each cluster of devices, and the statistics of the query data production volume per device. Our analytic results are validated via a deployment with: *(i)* the device side comprising compact image descriptors (queries) computed on Beaglebone Linux embedded platforms and transmitted to Amazon Web Services (AWS) Simple Storage Service; *(ii)* the cloud side carrying out image similarity detection via AWS Elastic Compute Cloud (EC2) instances, with the AWS Auto Scaling being used to control the number of instances according to the demand.**

***Index Terms*—visual search, internet-of-things, cloud computing, analytic modeling**

I. Introduction

Most of the envisaged applications and services for wearable sensors, smartphones, tablets or portable computers in the next ten years will involve analysis of audio/visual streams for event, action, object or user recognition, recommendation services and context awareness, etc. [1]–[8]. Examples of early commercial services in this domain include Google Goggles, Google Glass object recognition, Facebook automatic face tagging [9], Microsoft's Photo Gallery face recognition, as well as technology described in recent publications from Google, Siemens and others [1].

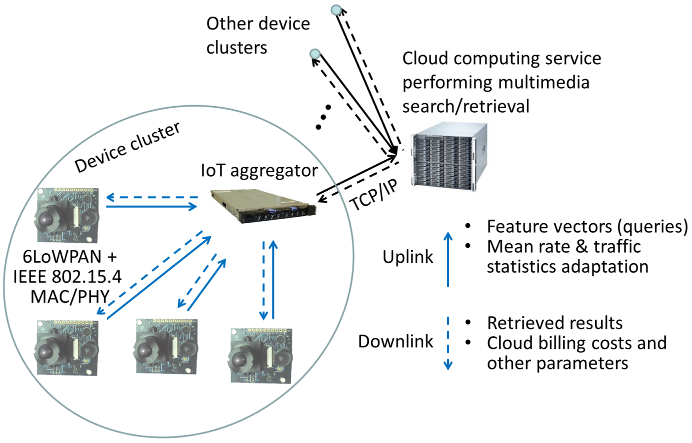
Other device clusters
Cloud computing service performing multimedia search/retrieval
Device cluster
IoT aggregator
TCP/IP
6LoWPAN + IEEE 802.15.4 MAC/PHY
Uplink
Feature vectors (queries)
Mean rate & traffic statistics adaptation
Downlink
Retrieved results
Cloud billing costs and other parameters

Figure 1. System hierarchy for a media search application within an IoT context. Low-power devices send query data to an IoT aggregator using low-power protocols for the physical, medium access control and network layer, such as IEEE 802.15.4 MAC/PHY and 6LoWPAN. The IoT aggregator sends aggregated query volumes to the cloud-computing service using TCP/IP.

Figure 1 presents an example of how such applications can be deployed in practice within an Internet-of-Things (IoT) context. Energy-constrained devices capture and extract audio/visual features from audio and/or image streams and compact such features into feature-descriptor vectors [8], [14]–[16]. Such feature vectors can be seen as *queries* in a multimedia search application [7], [14]. For example, Serra *et. al.* [8] propose beat and tempo feature extraction for cover song identification. A similar service is now widely deployed by Shazam. In the visual search domain, several approaches produce image salient points and then compact their associated features into vectors of 64~8192 elements [15], [16]. All such feature vectors can be matched to equivalent vectors of very large content libraries via a cloud-based deployment within the context of classification, retrieval and similarity identification for, so-called, "big data" applications. Devices of the same type running the same application software can be partitioned into "device clusters" (Fig. 1). Within each cluster, devices can be further subdivided into several classes based on the mean volume of query data produced within a certain time

F. Renna is with the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WA, UK (e-mail: fr330@cam.ac.uk)

J. Doyle and Y. Andreopoulos are with the Electronic and Electrical Engineering Department, University College London, Roberts Building, Torrington Place, London, WC1E 7JE, UK (e-mail: {j.doyle, i.andreopoulos}@ucl.ac.uk)

V. Giotsas is with Dithen Ltd., www.dithen.com, 843 Finchley Road, London NW11 8NA, UK (e-mail: v.giotsas@dithen.co.uk)

This work was supported in part by the European Union (Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 655282 – F. Renna), EPSRC (grants EP/M00113X/1 and EP/K033166/1) and Innovate UK (project ACAME, grant 131983).

[1] See "A Google Glass app knows what you're looking at" MIT Tech. Review (Sept. 30, 2013) and EU projects SecurePhone [10], [11] and MoBio [12], [13].

interval (e.g., “low”, “average” and “high” volume of queries within a 24-hour interval). For instance, when cameras are deployed in a broad square, the data generation volume for some regions will be higher than others, depending on the expected activity of each region [17]. An IoT aggregator can be used to aggregate traffic from each device cluster and upload it to a remote cloud computing service that carries out the search operations for recognition and retrieval purposes [2]–[4], [18].

In this paper, we consider the energy consumption and billing costs incurred by such IoT applications in a holistic, system-oriented, manner. Specifically, we derive a parametric model that allows for the coupling of the energy consumption and cloud billing costs in function of the system and query volume constraints for each cluster. A key aspect of our model is the derivation of the *optimal balancing* between:

1) *idle time*, where device energy consumption or cloud billing cost is incurred for no useful output, e.g., image acquisition and processing or buffering/standby time on the device that does not lead to query generation, or cloud servers idling due to small volumes of queries being submitted;
2) *active time*, where, despite resource consumption being incurred for useful output, one does not want to exceed certain limits in order not to cause excessive energy consumption in the device or excessive billing costs from the cloud infrastructure provider.

Another key aspect of our work in comparison to previous work on optimal energy management [17]–[22] is that, beyond establishing the configuration that minimizes the device energy consumption metric of interest, we also derive closed-form expressions for the corresponding minimum cloud billing cost, as well as the corresponding number of devices that can be admitted by the service.

In order to validate our analytic derivations, we utilize a proof-of-concept image similarity identification application, deployed via: *(i)* running the image feature extraction and query generation and transmission on a Beaglebone Linux embedded platform; *(ii)* implementing the back-end query processing for similarity identification and retrieval on Amazon Web Services Elastic Compute Cloud (AWS EC2) on-demand instances. Our results illustrate how the proposed model can be applied to real-world IoT-oriented media query retrieval systems in order to establish the desired operational parameters with respect to energy consumption and cloud infrastructure billing. More broadly, the experimental results reported in this paper exemplify the efficacy of our framework for feasibility studies on energy consumption and billing cost provisioning in cloud-based IoT query processing applications, prior to time-consuming testing and deployment.

The remainder of the paper is organized as follows. In Section II, we present the system model corresponding to the application scenarios under consideration. The analytic derivations characterizing energy-constrained feature extraction are presented in Section III, where we also derive the optimal coupling with the utilized cloud-computing service under four widely-used statistical characterizations for the query production rate. Section IV presents experimental results and Section V concludes the paper.

II. System Model

Within the system hierarchy of Fig. 1, each device connects to a “repository” service of a cloud provider, which represents the collecting unit, i.e. a cloud storage service like AWS Simple Storage Service (S3). This is where all device queries are uploaded to be processed by the back-end search mechanism of the service. As shown in Fig. 1, an IoT aggregator can be present in-between IoT clusters of the same type and the cloud repository, in order to: *(i)* reshape the IoT query traffic volume before uploading it to the cloud; *(ii)* carry out other device-specific and service-specific operations[2]. The figure shows that the essentials of the problem boil down to the analysis of the interaction between each mobile device node and its corresponding IoT aggregator and cloud computing service.

A. System Description

We assume that the mobile application is running continuously for a “monitoring” interval of T seconds. This interval corresponds to the typical device usage per day, or in-between battery recharging periods, e.g., $T \in [60, 18000]$ seconds per day. The activation, processing and transmission is either triggered by the user, or by external events at irregular times throughout the application’s running time T. Examples are: user-triggered audio or visual feature extraction by recording a particular content segment (e.g., as in the Shazam, Google Voice or Google Goggles services), or motion-activated frame capturing and processing within an audio/visual surveillance application [23]–[26]. We therefore assume that the query data volume during T seconds is modeled as a random variable.

The on-board battery of each device can last for prolonged intervals of time (e.g. tens or hundreds of hours, as it is typical in most audio/visual sensors and mobile devices). Therefore, the battery capacity can be assumed to be infinite compared to the energy budget spent by the media search application within each interval of T seconds [27]–[29]. Hence, issues such as leakage current and battery aging do not need to be considered. Nevertheless, it is well known that prolonged power surges from applications have inadvertent effects on a mobile device, such as faster-than-expected battery drainage or device overheating [27], [29]. Therefore, in our analysis we shall be considering constraints on the mean energy consumption during the monitoring interval, as well as the one-sided deviation from the mean. Finally, we remark that the query data production and transmission and the cloud billing are not strictly continuous processes. However, given that we are focusing on large monitoring intervals comprising tens or hundreds of seconds, they can be seen as continuous processes.

B. Definitions

We now present the key concepts behind our analytic framework. The nomenclature summary of our system model is given in Table I.

[2] Depending on the exact application, the IoT aggregator may carry out authentication or encryption of queries, reformatting of the retrieved results from the cloud service so that they display correctly on the particular devices, application/collection of device metadata for service statistics and advertising, etc. We do not discuss these aspects as they are out of the scope of this paper.

1) Query data production: The query data production and transmission by each device is a non-deterministic process, because it depends on the frequency of the application invocation (or on event-driven activation alerts), as well as on the query size, which may vary, depending on the media search application. Therefore, the query data volume (in bits) for each time interval of T seconds of each device is modeled by random variable (RV) Ψ_{e} with probability density function (PDF) $P(\psi_{\mathrm{e}})$. More broadly, given that devices may be monitoring multiple event (or "activity") zones, e.g., low, medium and high activity, we consider A activity zones with corresponding data volumes (in bits) for each time interval of T seconds modelled by A random variables with PDFs $P_a(\psi_{\mathrm{e}})$, $1 \leq a \leq A$. The statistical modeling of data volumes for each activity zone can be gained by observing the occurred processing and analyzing the behavior of each device when it captures image or audio data and produces query bits to be transmitted to the IoT aggregator. Alternatively, the query data production and transmission volume can be controlled (or "shaped") by the system designer in order to achieve a certain goal, such as limiting the occurring latency or utilizing inactivity periods of other applications running concurrently on the mobile device. Examples of systems with variable query data production and transmission rates include visual sensor networks transmitting image features [30]–[33], as well as activity recognition networks where the data acquisition is irregular and depends on the events occurring in the monitored areas [34]–[36].

Beyond individual devices, the query volume uploaded from each IoT aggregator to the cloud service is modelled by random variable Ψ_{b} with PDF $P(\psi_{\mathrm{b}})$. The distributions $P_a(\psi_{\mathrm{e}})$ and $P(\psi_{\mathrm{b}})$ will be of the same type (the latter will be a scaled version of the former) if the IoT aggregator shapes its uploaded traffic in the manner it receives it. Alternatively, if no traffic shaping is performed and the processing latency at the aggregator is fixed, combining $n_1, \ldots, n_A$ devices producing queries from A activity zones leads to:

$$P(\psi_{\mathrm{b}}) = \underbrace{P_1(\psi_{\mathrm{e}}) \star \ldots \star P_1(\psi_{\mathrm{e}})}_{n_1 \text{ times}} \star \cdots \star \underbrace{P_A(\psi_{\mathrm{e}}) \star \ldots \star P_A(\psi_{\mathrm{e}})}_{n_A \text{ times}}, \quad (1)$$

i.e., the PDF characterizing the uploaded traffic is the result of simple addition of the RVs modelling the data volumes received by all $n_{\mathrm{tot}} = \sum_{a=1}^{A} n_a$ devices.

Since the query data production volume may be non-stationary, we assume its marginal statistics for $P_a(\psi_{\mathrm{e}})$, $1 \leq a \leq A$, and $P(\psi_{\mathrm{b}})$, which are derived starting from a doubly stochastic model for these processes. Specifically, such marginal statistics can be obtained by [37], [38]: *(i)* fitting PDFs to sets of past measurements of query production volumes, with the statistical moments (parameters) of such distributions characterized by another PDF; *(ii)* integrating over the parameter space to derive the final form of $P_a(\psi_{\mathrm{e}})$ and $P(\psi_{\mathrm{b}})$. For example, if the query data production is modeled as a Half-Gaussian distribution with variance parameter that is itself exponentially distributed, by integrating over the parameter space, the marginal statistics of the query data volume become exponential [37], [38]. The disadvantage of using marginal statistics for the query data production volume is the removal of the stochastic dependencies to transient properties of these quantities. However, in this work we are interested in constraining the first and second moments of the energy consumption, as well as minimizing the expected cloud billing cost, over a lengthy time interval (e.g. several minutes or hours) and not in the instantaneous variations of these figures of merit over short time intervals. Thus, a mean and variance-based analysis using the marginal statistics is suitable for this purpose.

2) Energy and cloud infrastructure billing parameters: We assume that, on average, the production and transmission of one query bit incurs energy consumption rate of g_{e} Joule-per-bit (J/b). This rate incorporates the audio or visual capturing, the feature extraction and compaction process to produce the compacted feature vector, and the transmission of the feature vector to the IoT aggregator. For example, under a visual search application, this would incorporate the energy consumption for the image acquisition, the processing to extract salient point descriptions, the compaction process to produce a 256-element feature vector comprising 32-bit numbers (visual query) corresponding to each image [16], and the transceiver energy consumption to transmit this 8192-bit stream to the aggregator. Assuming that the entire process requires on average 10^{-5} J on the mobile device under consideration, this leads to $g_{\mathrm{e}} \cong 1.2 \times 10^{-9}$ J/b. However, given the time-varying nature of the query data production per device, we also encounter the case where the device is consuming energy to run the application (and possibly capture images or audio) in the background without producing any queries. This corresponds to "idle" energy consumption by each device with average rate i_{e} Joule-per-bit (i.e., i_{e} Joule for the time interval corresponding to the production and transmission of one query bit). We assume that the application goes in idle mode during time intervals where the amount of produced query bits is below $c_{\mathrm{e}} E[\Psi_{\mathrm{e}}]$ b, with $E[\Psi_{\mathrm{e}}]$ the statistical expectation of Ψ_{e}. The value of c_{e} depends on the processing and transmission capabilities of the device, as well as on the specifics of the application, e.g., the size of the feature vector per query, the manner in which query generation is activated, etc. For instance, regular query generation (e.g., once per second) will correspond to lower value of c_{e} in comparison to motion-activated query generation, as the motion detection requires continuous capturing and processing of data that corresponds to higher percentage of "idle" energy consumption, i.e., energy consumption that does not lead to query data generation. For small time intervals, the energy consumption rates g_{e} and i_{e} and the value for c_{e} may fluctuate depending on the device and network state (e.g., when memory paging, caching or other operating system tasks are carried out, or when transmission is hampered by high interference levels). However, given we are considering long periods of time for each monitoring interval T, we assume g_{e}, i_{e} and c_{e} to represent the average values and our experimental validation demonstrates that accurate energy estimates can be derived under this assumption.

Analogously, billing costs are incurred when servers are reserved from the cloud provider in order to process the queries uploaded by an IoT aggregator. Cloud providers offer a variety of usage-based pricing strategies for the consumption of computing resources that can be classified in three basic models: pay-as-you-go, subscription-based and auction-based. In this paper, we are primarily concerned with the first two models since auction-based models cannot guarantee reliable operation of compute instances.

In the *pay-as-you-go* model, users are billed with a static unit price per time interval without up-front costs or long-term commitments. Since the unit price remains constant, the total price increases linearly with the increase of the consumed units. Typically, a unit is a deployed virtual machine (e.g., an AWS EC2 instance) over a short time period (e.g., 1 hour) and the unit price follows a tiered model based on the compute capacity of the unit (in terms of CPU, memory and storage), the operating system and the region where the unit is deployed. For example the AWS EC2 "on-demand" instances use a linear per-unit pricing per hour. Variations of the linear pay-as-you-go model involve a step decrease in the unit price after certain utilization thresholds are reached, leading to a sublinear increase of the unit cost with the increase of usage time. An example of this model is the Sustained Usage pricing of the Google Cloud Engine.

In *subscription-based pricing* the user commits to long-term utilization for a pre-selected number of computing units by paying a fixed upfront price for the entire consumption period. Cloud providers may also offer a hybrid model that combines discounted pay-as-you-go pricing with an upfront payment for a fixed long-term time period (e.e 1 to 3 years). The AWS EC2 "reserved" instances and the Microsoft Azure Prepay implement a hybrid subscription/pay-as-you-go model. The unit price can be either linear or subject to a step decrease based on the usage volume.

Beyond their available pricing models, all cloud computing services today use some form of autoscaling mechanism in order to adjust the number of compute instances according to the demand. For example, in AWS Auto Scaling [39] one can set rules that scale the utilized compute instances for every monitoring interval according to the average query volume received during the previous monitoring interval. A typical AWS Auto Scaling setup would be[3]:

- 3 single-core AWS EC2 `m3.medium` on-demand instances when the average uploaded query volume was below a certain "quota" of c_b query bits ("idle" case) in the previous monitoring interval,
- 30 on-demand instances when the query volume exceeded c_b b ("active" case) in the previous monitoring interval.

Such configurations are prevalent in all cloud computing providers [AWS, Microsoft Azure, Google Compute Engine (GCE), etc.], where services are developed using a core number of instances, and additional compute instances are added when the demand exceeds a certain threshold [3], [4], [39], [40]. For example, based on current AWS EC2 pricing, each single-core `m3.medium` instance incurs (on average) billing cost of 0.067$ per hour under the on-demand configuration. Assuming that a search operation with a 256×32-bit query requires 10ms of cloud service time and under the AWS Auto Scaling rules stated above, this corresponds to billing cost of (approximately): 5.6×10^{-7} dollars-per-query under the "idle" case, or $i_\mathrm{b} \cong 6.8 \times 10^{-11}$ dollars-per-query-bit ($/b) and $p_\mathrm{b} \cong 6.8 \times 10^{-10}$ $/b for the "active" case. Similar billing rates can be calculated for other cloud providers under pay-as-you-go or subscription-based pricing. Notably, despite the fact that cloud infrastructure billing is levied on hourly or minute-by-minute increments (e.g., for AWS and GCE, resp.), because of the continued nature of the service, we do not have "termination gaps". Instead, some instances may idle for some time before they are reused or terminated, depending on the fluctuations of the query volume within each monitoring interval. The quota of c_b query bits that triggers the auto scaling can be set according to the application or the number of devices within each IoT aggregator and the billing rates are always linear to the number of queries since we always consider a fixed query and database size, which leads to the computation time increasing linearly to the number of queries.

Beyond the cost of the computing time, billing cost proportional to the expected query volume per monitoring interval, $E[\Psi_\mathrm{b}]$, must be constrained to V_max b, since: *(i)* all cloud providers charge for data transfers and storage; and *(ii)* excessive interference will occur if the average query volume rises above the capacity of the local network of each IoT aggregator. Assuming 0.15$ per gigabyte of query volume (based on current AWS pricing), this leads to (approximately) $g_\mathrm{b} = 1.9 \times 10^{-11}$ $/b. Then, in order to remain competitive against other solutions in the market, the service may wish to set an expectation that each user should be billed for B_mean $ on average for each device and each monitoring time interval of T seconds. Importantly, while the instantaneous rates i_b, p_b, g_b and the instantaneous query transmission rate from each IoT aggregator may fluctuate, given that we are interested in long monitoring intervals and infrastructure billing is typically applied in minute or even hourly increments, we utilize mean values for these rates, calculated by averaging over lengthy operational periods.

Evidently, the large number of system, data production, energy consumption, and cloud billing parameters of Table I makes the exhaustive exploration of the complete design space infeasible. Therefore, the creation of an analytic model that can establish closed-form relationships between the different parameters, as well as optimal settings under specified conditions for device energy consumption and billing cost is of paramount importance. This is the aim of the next section.

III. Characterization of Energy Consumption and Cloud Billing Cost

We derive analytic expressions for the expected energy consumption of a device (and its one-sided deviate), as well as the expected cloud billing for a group of n_tot devices on the same IoT aggregator. This allows us to derive closed-form conditions that ensure that the one-sided energy variation is minimized under a constraint on the expected energy consumption for each device, or, vice-versa. We also derive the

[3] The reported numbers of instances and instance types are only indicative and can be adjusted per IoT application.

Table I
NOMENCLATURE TABLE.

Symbol	Unit	Definition
n_{tot}, n_a	–	Total number of devices and devices of activity zone a (out of A total zones) within the same IoT aggregator
g_{e}	J/b	Energy for producing and transmitting a query bit
i_{e}	J/b	Energy during idle periods equal to the interval required to produce and transmit a query bit
c_{e}	–	Fraction of the average query volume below which the device application is in idle mode
$E_{\mathrm{max\_exp}}$	J	Upper bound of the expected energy consumption over T seconds
$E_{\mathrm{max\_var}}$	J^2	Upper bound of the one-sided variation from the expected energy consumption over T seconds
r_{tot}, r_a	b	Average query data production and transmission volume and average volume per device of activity zone a (over the monitoring interval)
$V_{\max}$	b	Maximum query data transmission volume of each IoT aggregator over the monitoring interval
$\Psi_{\mathrm{e}} \sim P_a(\psi_{\mathrm{e}})$, $\Psi_{\mathrm{b}} \sim P(\psi_{\mathrm{b}})$	b	RVs modeling the query data production and transmission volume per device (and activity zone a) and per IoT aggregator
$E[\Psi_{\mathrm{e}}]$, $E[\Psi_{\mathrm{b}}]$	b	Expected query data production and transmission per device and per aggregator over the monitoring time interval
g_{b}	\$/b	Billing cost (per query bit) incurred from uploading/storing a query
i_{b}	\$/b	Billing cost (per time interval corresponding to the processing time per query bit) incurred from "idle" periods
c_{b}	b	Number of query bits (quota) above which the cloud Auto Scaling mechanism switches from idle to active state
p_{b}	\$/b	Billing cost (per query bit) incurred from processing a query after exceeding the quota of c_{b} query bits per T seconds ("active" period)
B_{mean}	\$	Expected cloud billing cost over T seconds

conditions that minimize the incurred billing cost and ensure that the minimum value can be set to the expected billing of B_{mean} per monitoring period of T seconds, while satisfying the total query transmission volume constraint, $V_{\max}$, of the IoT aggregator.

The expected energy consumption of each mobile device of activity zone a over the monitoring period of T seconds is:

$$E_{\mathrm{exp}} = E[\Psi_{\mathrm{e}}]\, g_{\mathrm{e}} + i_{\mathrm{e}} \int_0^{c_{\mathrm{e}} E[\Psi_{\mathrm{e}}]} \left(c_{\mathrm{e}} E[\Psi_{\mathrm{e}}] - \psi_{\mathrm{e}}\right) P_a(\psi_{\mathrm{e}})\, d\psi_{\mathrm{e}}, \tag{2}$$

where the integral of the second term expresses the expected energy consumption for the time that the device will be in idle mode. This term expresses the energy consumed to produce no useful output, i.e., energy consumed that does not lead directly to the production of query volume (e.g., image acquisition and processing or buffering/standby).

We can also express the one-sided variability of the energy consumption when the application switches from idle to active state (i.e., when exceeding the $c_{\mathrm{e}} E[\Psi_{\mathrm{e}}]$-bit query volume):

$$E_{\mathrm{var}} = g_{\mathrm{e}}^2 \int_{c_{\mathrm{e}} E[\Psi_{\mathrm{e}}]}^{\infty} \left(\psi_{\mathrm{e}} - c_{\mathrm{e}} E[\Psi_{\mathrm{e}}]\right)^2 P_a(\psi_{\mathrm{e}})\, d\psi_{\mathrm{e}}. \tag{3}$$

For each monitoring interval of T seconds, higher values of E_{var} imply higher energy consumption fluctuation from the average energy consumption. Therefore, under a given energy budget of E_{exp} Joule for the monitoring time interval of T seconds, allowing for a large value for E_{var} will incur significant drop in the device battery level (and possibly other unintended consequences, such as device overheating, battery degradation over time, etc.). On the other hand, a small value of E_{var} will limit the query production volume handled by the device, or may require a very high value for c_{e} that may not be realistic for the application and hardware under consideration.

Let us now consider the expected cloud billing cost when receiving n_{tot} aggregated media query volumes from an IoT aggregator. We can express this cost via

$$\begin{aligned} B_{\mathrm{exp}} &= E[\Psi_{\mathrm{b}}]\, g_{\mathrm{b}} + i_{\mathrm{b}} \int_0^{c_{\mathrm{b}}} (c_{\mathrm{b}} - \psi_{\mathrm{b}})\, P(\psi_{\mathrm{b}})\, d\psi_{\mathrm{b}} \\ &+ p_{\mathrm{b}} \int_{c_{\mathrm{b}}}^{\infty} (\psi_{\mathrm{b}} - c_{\mathrm{b}})\, P(\psi_{\mathrm{b}})\, d\psi_{\mathrm{b}}, \end{aligned} \tag{4}$$

where: $E[\Psi_{\mathrm{b}}]\, g_{\mathrm{b}}$ corresponds to the data transfer/storage costs, the first integral corresponds to the partial moment expressing the "idle" billing cost, and the second integral corresponds to the "active" billing. Adding and subtracting $p_{\mathrm{b}} \int_0^{c_{\mathrm{b}}} (\psi_{\mathrm{b}} - c_{\mathrm{b}})\, P(\psi_{\mathrm{b}})\, d\psi_{\mathrm{b}}$ in B_{exp}, we get:

$$\begin{aligned} B_{\mathrm{exp}} &= E[\Psi_{\mathrm{b}}]\,(g_{\mathrm{b}} + p_{\mathrm{b}}) - p_{\mathrm{b}} c_{\mathrm{b}} \\ &+ (i_{\mathrm{b}} + p_{\mathrm{b}}) \int_0^{c_{\mathrm{b}}} (c_{\mathrm{b}} - \psi_{\mathrm{b}})\, P(\psi_{\mathrm{b}})\, d\psi_{\mathrm{b}}. \end{aligned} \tag{5}$$

Evidently, the expected billing cost depends on the coupling point, c_{b}, as well as on the PDF of the aggregate query data reaching the cloud service, $P(\psi_{\mathrm{b}})$, which is either a variant of the $P_a(\psi_{\mathrm{e}})$ distributions, or it is linked to them via (1). In the remainder of this section:

- We consider various cases for $P_a(\psi_{\mathrm{e}})$ and $P(\psi_{\mathrm{b}})$ and minimize the energy variance of (3) subject to an upper bound for the expression of (2), and vice-versa.
- We derive the number of query bits (quota), c_{b}, that minimizes the corresponding billing cost of (5) under various PDFs, $P(\psi_{\mathrm{b}})$.
- In order for the desired energy consumption and billing cost parameters to be met concurrently while obeying the total traffic volume constraint of each IoT aggregator, we associate the minimum billing cost with the desired value for the expected billing, B_{mean}, and the device query production volumes for activity zone. Therefore, we establish the corresponding number of devices, $n_1, \ldots n_A$, that can be admitted by each IoT aggregator under the optimal configuration.

A. Coupling of Device Energy Consumption and Cloud Infrastructure Billing

In order to control the overall energy consumption profile of the application, one may wish to minimize the expected one-sided energy variability, E_{var}, subject to the constraint that the expected energy consumption does not exceed $E_{\text{max\_exp}}$ Joule within T seconds. Both of these values are provided by the application or device developer in order to ensure the application does not degrade the user quality-of-experience, or disrupt other concurrently-running services on the device. We term this problem as the "primary" optimization problem, and its converse as the "dual" problem.

1) Primary energy minimization problem: We determine the value c_{e} that minimizes the one-sided variability of the energy consumption while satisfying a constraint on the average energy consumption, i.e., we consider the optimization problem

$$\begin{aligned} \underset{c_{\text{e}} \in \mathbb{R}^+}{\text{minimize}} \quad & E_{\text{var}} \\ \text{subject to} \quad & E_{\text{exp}} \le E_{\text{max\_exp}}. \end{aligned} \tag{6}$$

2) Dual energy minimization problem: Consider now a dual setting, in which one aims at minimizing the average energy consumption while satisfying a constraint on the maximum one-sided energy variation from idle to active mode. The activation threshold c_{e} that achieves this is found by solving the optimization problem

$$\begin{aligned} \underset{c_{\text{e}} \in \mathbb{R}^+}{\text{minimize}} \quad & E_{\text{exp}} \\ \text{subject to} \quad & E_{\text{var}} \le E_{\text{max\_var}}. \end{aligned} \tag{7}$$

3) Convexity of the energy minimization problems and closed-form solutions: We first show that both the primary and dual optimization problems of (6) and (7) are convex. Therefore, they can be solved using fast numerical methods, such as gradient descent or the Newton-Raphson method.

By taking the first and the second derivative of E_{exp} with respect to c_{e} we obtain

$$\frac{dE_{\text{exp}}}{dc_{\text{e}}} = i_{\text{e}} E[\Psi_{\text{e}}] F_a(c_{\text{e}} E[\Psi_{\text{e}}]) \tag{8}$$

$$\frac{d^2 E_{\text{exp}}}{dc_{\text{e}}^2} = i_{\text{e}} (E[\Psi_{\text{e}}])^2 P_a(c_{\text{e}} E[\Psi_{\text{e}}]), \tag{9}$$

where $F_a(\psi_{\text{e}})$ and $P_a(\psi_{\text{e}})$ are the cumulative distribution function (CDF) and the PDF of the query volume per device of activity zone a, Ψ_{e}, respectively. Since $\frac{d^2 E_{\text{exp}}}{dc_{\text{e}}^2} \ge 0$, E_{exp} is a convex function of c_{e}.

Analogously, by taking the first and the second derivative of E_{var} with respect to c_{e} we obtain

$$\begin{aligned} \frac{dE_{\text{var}}}{dc_{\text{e}}} &= 2g_{\text{e}}^2 (E[\Psi_{\text{e}}])^2 c_{\text{e}} \left[1 - F_a(c_{\text{e}} E[\Psi_{\text{e}}])\right] \\ &\quad -2g_{\text{e}}^2 E[\Psi_{\text{e}}] \int_{c_{\text{e}} E[\Psi_{\text{e}}]}^{+\infty} \psi_{\text{e}} P_a(\psi_{\text{e}}) d\psi_{\text{e}} \end{aligned} \tag{10}$$

$$\frac{d^2 E_{\text{var}}}{dc_{\text{e}}^2} = 2g_{\text{e}}^2 (E[\Psi_{\text{e}}])^2 \left[1 - F_a(c_{\text{e}} E[\Psi_{\text{e}}])\right]. \tag{11}$$

Thus, $\frac{d^2 E_{\text{var}}}{dc_{\text{e}}^2} \ge 0$ and E_{var} is also a convex function of c_{e}.

Given that E_{exp} and E_{var} are convex, the following proposition offers a way to derive the solutions of the problems (6) and (7) in closed form.

Proposition 1. *The solution to the optimization problem (6) is such that, at the optimal c_{e}, it holds*

$$E_{\text{exp}} = E_{\text{max\_exp}}. \tag{12}$$

The solution to the optimization problem (7) is such that, at the optimal c_{e}, it holds

$$E_{\text{var}} = E_{\text{max\_var}}. \tag{13}$$

Proof: See Appendix. ■

In other words, the solutions to both optimization problems are obtained when the constraints are met with equality. Therefore, whenever possible, by inverting the closed-form expressions of E_{exp} and E_{var} for different query volume PDFs, we can find the optimal c_{e} in closed form.

4) Billing parameter tuning to minimize the cloud infrastructure billing cost and meet the expected billing B_{mean}: We can now turn our attention to the billing cost B_{exp} in (5) for the n_{tot}-device aggregate query production volume over the monitoring time interval of T s. We note that the first and the second derivative of B_{exp} with respect to the coupling point c_{b} are given by

$$\frac{dB_{\text{exp}}}{dc_{\text{b}}} = -p_{\text{b}} + (i_{\text{b}} + p_{\text{b}}) F(c_{\text{b}}) \tag{14}$$

$$\frac{d^2 B_{\text{exp}}}{dc_{\text{b}}^2} = (i_{\text{b}} + p_{\text{b}}) P(c_{\text{b}}), \tag{15}$$

where $F(\psi_{\text{b}})$ and $P(\psi_{\text{b}})$ are CDF and the PDF of the aggregated query volume Ψ_{b}. Therefore, we can conclude that B_{exp} is a convex function of c_{b} when Ψ_{b} is modelled by a continuous distribution function. Moreover, the value of c_{b} that minimizes the billing cost is obtained by solving the equation $\frac{dB_{\text{exp}}}{dc_{\text{b}}} = 0$, i.e.,

$$c_{\text{b}} = F^{-1}\left(\frac{p_{\text{b}}}{i_{\text{b}} + p_{\text{b}}}\right), \tag{16}$$

where $F^{-1}(\cdot)$ is the inverse CDF of Ψ_{b}. Assuming any strictly-increasing CDF, c_{b} will be unique[4]. Therefore, in conjunction with the fact that $\forall c_{\text{b}} : \frac{d^2 B_{\text{exp}}}{dc_{\text{b}}^2} > 0$, B_{exp} attains a global minimum in function of c_{b}.

5) Number of devices in an IoT aggregator to concurrently satisfy cost and system constraints: In order to meet energy, billing and query volume constraints: $\{E_{\text{max\_exp}}$ or $E_{\text{max\_var}}\}$, B_{mean} and V_{max}, we first find c_{e} corresponding to (12) or (13). We can then match the device query volumes $r_1, \ldots, r_A$ with the minimum billing cost, $\min\{B_{\text{exp}}\}$, obtained by substituting c_{b} from (16) into (5). Finally, setting

$$\min\{B_{\text{exp}}\} = B_{\text{mean}}, \tag{17}$$

and constraining the average traffic volume of all devices,

$$r_{\text{tot}} = \sum_{a=1}^{A} n_a r_a, \tag{18}$$

by

$$r_{\text{tot}} \le V_{\text{max}}, \tag{19}$$

we obtain the number of devices, $n_1, \ldots, n_A$, that can be accommodated by an IoT aggregator when each device satisfies the energy settings of (6) or (7) and the IoT-uploaded volume

[4] Even if the CDF is monotonically increasing, all candidate extrema are equivalent with respect to the derived billing cost.

incurs the minimum billing cost of B_{mean} \$ per monitoring interval, while satisfying the query volume constraint $V_{\max}$.

Overall, via the energy-constrained analysis and the cloud-billing optimization, one can explore different energy and billing settings in order to accommodate particular types of mobile devices (with given energy consumption parameters), predetermined average query production volume, or given number of devices per IoT cluster of Fig. 1. We present detailed examples for this in the following subsections.

B. Illustrative Case: Ψ_{e} and Ψ_{b} Are Uniformly Distributed

For each activity zone a, when no knowledge of the underlying statistics of the query generation process exists, one can assume that both $P_a(\psi_{\mathrm{e}})$ and $P(\psi_{\mathrm{b}})$ are uniform over the intervals $[0, 2r_a]$ and $[0, 2r_{\mathrm{tot}}]$, respectively:

$$P_{\mathrm{U},a}(\psi_{\mathrm{e}}) = \begin{cases} \frac{1}{2r_a}, & 0 \le \psi_{\mathrm{e}} \le 2r_a \\ 0, & \text{otherwise} \end{cases}, \tag{20}$$

and

$$P_{\mathrm{U}}(\psi_{\mathrm{b}}) = \begin{cases} \frac{1}{2r_{\mathrm{tot}}}, & 0 \le \psi_{\mathrm{b}} \le 2r_{\mathrm{tot}} \\ 0, & \text{otherwise} \end{cases}. \tag{21}$$

This corresponds to the case where the IoT aggregator's upload query volume PDF matches the query generation PDF (20) and the aggregator merges and transmits query volumes of n_{tot} devices to the cloud service under the query volume PDF of (21).

For each activity zone a, $1 \le a \le A$, the expected value of Ψ_{e} is $E_{\mathrm{U}}[\Psi_{\mathrm{e}}] = r_a$ b. The expected value of Ψ_{b} is $E_{\mathrm{U}}[\Psi_{\mathrm{b}}] = r_{\mathrm{tot}}$ b. The cases where $c_{\mathrm{e}} > 2$ or $c_{\mathrm{b}} > 2r_{\mathrm{tot}}$ are of no practical relevance, because: *(i)* the first inequality means each device is always in idle mode, or *(ii)* the second inequality means the cloud infrastructure is constantly overprovisioned. Thus, we are only concerned with the case where: $0 < c_{\mathrm{e}} < 2$ and $0 < c_{\mathrm{b}} < 2r_{\mathrm{tot}}$.

1) Energy parameter tuning corresponding to the solution of the Primary and Dual minimization problems of (6) *and* (7)*:* Starting from the device energy consumption, by using (20) in (2), we obtain:

$$E_{\mathrm{exp,U}} = \left(g_{\mathrm{e}} + \frac{i_{\mathrm{e}} c_{\mathrm{e}}^2}{4}\right) r_a. \tag{22}$$

In addition, by using (20) in (3), we obtain:

$$E_{\mathrm{var,U}} = g_{\mathrm{e}}^2 \frac{(2 - c_{\mathrm{e}})^3}{6} r_a^2. \tag{23}$$

Then, given the average query volume r_a per time interval T, and the corresponding energy parameters i_{e} and g_{e}, it is possible to derive the activation threshold c_{e} that corresponds to the solution to (6) by solving $E_{\mathrm{exp}} = E_{\mathrm{max\_exp}}$ for c_{e}. Thus, we obtain

$$c_{\mathrm{e,U,primary}} = 2\sqrt{\frac{E_{\mathrm{max\_exp}} - g_{\mathrm{e}} r_a}{i_{\mathrm{e}} r_a}}, \tag{24}$$

provided that $E_{\mathrm{max\_exp}} > g_{\mathrm{e}} r_a$. The last inequality must hold or else the energy constraint does not suffice for the production of r_a b within T seconds. We also note that the constraint $c_{\mathrm{e}} < 2$ implies in this case that $E_{\mathrm{max\_exp}} < (g_{\mathrm{e}} + i_{\mathrm{e}}) r_a$. These two constraints provide the feasible range for the expected energy consumption under Uniformly-distributed query volumes as: $E_{\mathrm{max\_exp}} \in (g_{\mathrm{e}} r, (g_{\mathrm{e}} + i_{\mathrm{e}}) r_a)$.

Similarly, the solution to the constrained miminization of (7) is obtained by solving for c_{e} the equation $E_{\mathrm{var}} = E_{\mathrm{max\_var}}$, thus obtaining

$$c_{\mathrm{e,U,dual}} = 2 - \left(\frac{6E_{\mathrm{max\_var}}}{g_{\mathrm{e}}^2 r_a^2}\right)^{1/3}. \tag{25}$$

Note that if the constraint on the one-sided energy variation is such that $E_{\mathrm{max\_var}} \ge 4g_{\mathrm{e}}^2 r_a^2/3$, then such constraint is verified by all nonnegative values of c_{e}, and the minimum average energy consumption is achieved by setting $c_{\mathrm{e}} = 0$. Due to the finite support of the Uniform distribution, this effectively corresponds to the trivial case when the one-sided deviation is unlimited and the minimum energy consumption is obtained when no idle energy is consumed.

2) Billing parameter tuning to minimize the cloud infrastructure billing cost and meet the expected billing B_{mean}: For the case of uniform distribution, by replacing (21) in (5), we obtain the average billing cost as

$$B_{\mathrm{exp,U}} = (g_{\mathrm{b}} + p_{\mathrm{b}}) r_{\mathrm{tot}} - p_{\mathrm{b}} c_{\mathrm{b}} + (i_{\mathrm{b}} + p_{\mathrm{b}}) \frac{c_{\mathrm{b}}^2}{4r_{\mathrm{tot}}}, \tag{26}$$

and the optimal coupling point (16) is

$$c_{\mathrm{b,U}} = \frac{2p_{\mathrm{b}} r_{\mathrm{tot}}}{i_{\mathrm{b}} + p_{\mathrm{b}}}. \tag{27}$$

The corresponding minimum-possible billing cost is:

$$\min\{B_{\mathrm{exp,U}}\} = \left(g_{\mathrm{b}} + p_{\mathrm{b}} - \frac{p_{\mathrm{b}}^2}{i_{\mathrm{b}} + p_{\mathrm{b}}}\right) r_{\mathrm{tot}}. \tag{28}$$

The last equation shows that the minimum billing cost increases linearly to the average query data production volume of all n_{tot} devices.

3) Number of devices in an IoT aggregator to concurrently satisfy cost and system constraints: In order to meet *both* energy and billing costs: $\{E_{\mathrm{max\_exp}}, E_{\mathrm{max\_var}}\}$ and B_{mean}, we can first tune c_{e} according to (24) or (25). Then, by substituting $\min\{B_{\mathrm{exp,U}}\} = B_{\mathrm{mean}}$ in (28), the expected billing B_{mean} is achievable under the query transmission volume constraint $V_{\max}$ if

$$B_{\mathrm{mean}} \le V_{\max}\left(g_{\mathrm{b}} + p_{\mathrm{b}} - \frac{p_{\mathrm{b}}^2}{i_{\mathrm{b}} + p_{\mathrm{b}}}\right). \tag{29}$$

Otherwise, $V_{\max}$ cannot accommodate the query volume that guarantees billing equal to B_{mean}. When (29) is satisfied, we can use *proportional fairness* [41] to derive the number of devices from different activity zones, i.e.,

$$n_{\mathrm{U},a} = \frac{B_{\mathrm{mean}}}{\left(g_{\mathrm{b}} + p_{\mathrm{b}} - \frac{p_{\mathrm{b}}^2}{i_{\mathrm{b}} + p_{\mathrm{b}}}\right) A r_a}, \tag{30}$$

for $1 \le a \le A$. An interesting solution for $n_{\mathrm{U},a}$ occurs if B_{mean} is set so that the volume $V_{\max}$ is expected to be fully utilized, i.e., the constraint of (29) is met with equality. In such a case, the energy consumption parameters ($E_{\mathrm{exp,U}}$ and $E_{\mathrm{var,U}}$), the desired cloud billing cost (B_{mean}), and the aggregator's data transmission volume ($V_{\max}$) become *mutually coupled*. Then, the number of devices from different activity zones that can accommodated by the IoT aggregator simply becomes

$$n_{\mathrm{U},a} = \frac{V_{\max}}{A r_a}. \tag{31}$$

Overall, under the uniform distributions of (20) and (21), $n_{\mathrm{U},a}$ of (30) represents the number of devices that should be accommodated by an IoT aggregator [with each device having c_{e} according to (24) or (25)] in order to lead to the minimum billing cost being equal to B_{mean} and aggregated query volume below or equal to $V_{\max}$ b.

C. Energy-constrained Query Volume Production and Minimum Billing Cost under Pareto, Exponential and Half-Gaussian Distributions

We can now extend the previous calculation to other distributions expressing commonly observed data transmission rates in practical applications. We consider three additional PDFs for Ψ_{e} and Ψ_{b} that have been used to model the marginal statistics of many real-world data transmission applications and provide the obtained analytic results in this subsection. For each distribution and for each activity zone a, we couple its parameters to the average query volume of the uniform distribution, r_a. This facilitates comparisons of the energy consumption and billing cost achievable under different statistical characterizations for the query volume.

1) Pareto distribution and fixed query volume: This distribution has been used, amongst others, to model the marginal data size distribution of data production processes that result in substantial number of small data volumes and a few very large ones [42], [43]. For each activity zone a, $1 \le a \le A$, consider $P_{\mathrm{P},a}(\psi_{\mathrm{e}})$ as the Pareto distribution with scale v_{e} and shape $\alpha_{\mathrm{e}} > 2$,

$$P_{\mathrm{P},a}(\psi_{\mathrm{e}}) = \begin{cases} \alpha_{\mathrm{e}} \frac{v_{\mathrm{e}}^{\alpha_{\mathrm{e}}}}{\psi_{\mathrm{e}}^{\alpha_{\mathrm{e}}+1}}, & \psi_{\mathrm{e}} \ge v_{\mathrm{e}} \\ 0, & \text{otherwise} \end{cases}. \tag{32}$$

The expected value of Ψ_{e} is $E_{\mathrm{P}}[\Psi_{\mathrm{e}}] = \frac{\alpha_{\mathrm{e}} v_{\mathrm{e}}}{\alpha_{\mathrm{e}}-1}$ b. Thus, if we set $v_{\mathrm{e}} = \frac{\alpha_{\mathrm{e}}-1}{\alpha_{\mathrm{e}}} r_a$, we obtain $E_{\mathrm{P}}[\Psi_{\mathrm{e}}] = r_a$ b, i.e., we match the expected query volume per device to that of the Uniform distribution. The characterization of the energy consumption for queries with Pareto-distributed volumes is summarized in the following proposition.

Proposition 2. *The average energy consumption for Pareto-distributed media query volumes is given by*

$$E_{\mathrm{exp,P}} = \left[g_{\mathrm{e}} + i_{\mathrm{e}}\left[(\alpha_{\mathrm{e}}-1)^{\alpha_{\mathrm{e}}-1} c_{\mathrm{e}} (\alpha_{\mathrm{e}} c_{\mathrm{e}})^{-\alpha_{\mathrm{e}}} + c_{\mathrm{e}} - 1\right]\right] r_a, \tag{33}$$

and the one-sided variation of the energy consumption from idle mode to active mode is given by

$$E_{\mathrm{var,P}} = 2 g_{\mathrm{e}}^2 \frac{(\alpha_{\mathrm{e}}-1)^{\alpha_{\mathrm{e}}-1} c_{\mathrm{e}}^{2-\alpha_{\mathrm{e}}}}{\alpha_{\mathrm{e}}^{\alpha_{\mathrm{e}}} (\alpha_{\mathrm{e}}-2)} r_a^2. \tag{34}$$

Proof: The expressions (33) and (34) are obtained by substituting the Pareto PDF (32) in (2) and (3), respectively, and deriving the closed-form result of the integral expressions. ■

Note that Proposition 2 assumes that $c_{\mathrm{e}} \ge \frac{\alpha_{\mathrm{e}}-1}{\alpha_{\mathrm{e}}}$, since, otherwise, the device will never switch from active to idle state. In this case, the optimal solution, $c_{\mathrm{e,P,primary}}$, of (6) cannot be expressed in closed form, but it is obtained via efficient convex optimization algorithms such as gradient descent. On the other hand, from (34), we can derive the solution of (7) as

$$c_{\mathrm{e,P,dual}} = \left[\frac{\alpha_{\mathrm{e}}^{\alpha_{\mathrm{e}}} (\alpha_{\mathrm{e}}-2) E_{\mathrm{max\_var}}}{2 g_{\mathrm{e}}^2 (\alpha_{\mathrm{e}}-1)^{\alpha_{\mathrm{e}}-1} r_a^2}\right]^{1/(2-\alpha_{\mathrm{e}})}. \tag{35}$$

A particular case of interest for the Pareto distribution arises when $\alpha_{\mathrm{e}} \to +\infty$: in this limit case, the query volume per device converges to the expectation $E_{\mathrm{P}}[\Psi_{\mathrm{e}}] = r_a$, i.e., to *fixed-volume* query production per monitoring interval and activity zone. Then, since $c_{\mathrm{e}} \ge \frac{\alpha_{\mathrm{e}}-1}{\alpha_{\mathrm{e}}}$, as $\alpha_{\mathrm{e}} \to \infty$, the average energy consumption tends to

$$E_{\mathrm{exp,P}} = [g_{\mathrm{e}} + i_{\mathrm{e}}(c_{\mathrm{e}}-1)] r_a, \tag{36}$$

and the one-side energy variation from idle to active mode converges to zero (the device is in idle mode for a fixed part of every monitoring interval). Then, the activation threshold which meets the average energy consumption constraint $E_{\mathrm{max\_exp}}$ is given by

$$c_{\mathrm{e,P,primary}} = 1 + \frac{E_{\mathrm{max\_exp}} - g_{\mathrm{e}} r_a}{i_{\mathrm{e}} r_a}, \tag{37}$$

provided that $E_{\mathrm{max\_exp}} \ge g_{\mathrm{e}} r_a$ (which must hold or else the query production rate, r_a, is not achievable).

2) Exponential distribution: This distribution is relevant to our application context since the marginal statistics of compressed image and video traffic have often been modeled as exponentially decaying [44]. Consider $P_{\mathrm{E},a}(\psi_{\mathrm{e}})$ as the Exponential distribution with rate parameter $1/r_a$ for each activity zone a

$$P_{\mathrm{E},a}(\psi_{\mathrm{e}}) = \frac{1}{r_a} \exp\left(-\frac{\psi_{\mathrm{e}}}{r_a}\right), \tag{38}$$

for $\psi_{\mathrm{e}} \ge 0$. In this case, the expected value of Ψ_{e} is $E_{\mathrm{E}}[\Psi_{\mathrm{e}}] = r_a$ b. The characterization of the energy consumption for queries with exponentially distributed volumes is summarized in the following proposition.

Proposition 3. *The average energy consumption for Exponentially-distributed media query volumes is given by*

$$E_{\mathrm{exp,E}} = [g_{\mathrm{e}} + i_{\mathrm{e}}(c_{\mathrm{e}} + e^{-c_{\mathrm{e}}} - 1)] r_a, \tag{39}$$

and the one-sided variation of the energy consumption from idle mode to active mode is given by

$$E_{\mathrm{var,E}} = 2 g_{\mathrm{e}}^2 \exp(-c_{\mathrm{e}}) r_a^2. \tag{40}$$

Proof: The expressions (39) and (40) are obtained by substituting the Exponential PDF (38) in (2) and (3), respectively, and deriving the closed-form result of the integral expressions. ■

In this case, the closed form solution of the problem (6) can be derived from (39) as

$$\begin{aligned} c_{\mathrm{e,E,primary}} = {} & W_0\left(-\exp\left(-(E_{\mathrm{max\_exp}} + i_{\mathrm{e}} r_a - g_{\mathrm{e}} r_a)/(i_{\mathrm{e}} r_a)\right)\right) \\ & + (E_{\mathrm{max\_exp}} + i_{\mathrm{e}} r_a - g_{\mathrm{e}} r_a)/(i_{\mathrm{e}} r_a), \end{aligned} \tag{41}$$

where $W_0(\cdot)$ is the main branch of the standard Lambert W function [45]. Analogously, from (40) we can derive the closed form solution of (7) as

$$c_{\mathrm{e,E,dual}} = \ln \frac{2g_{\mathrm{e}}^2 r_a^2}{E_{\mathrm{max\_var}}}. \tag{42}$$

3) Half-Gaussian distribution: We consider now $P_{\mathrm{H},a}\left(\psi_{\mathrm{e}}\right)$ as the Half-Gaussian distribution with mean $E_{\mathrm{H}}\left[\Psi_{\mathrm{e}}\right] = r_a$ for each activity zone a

$$P_{\mathrm{H},a}\left(\psi_{\mathrm{e}}\right) = \begin{cases} \frac{2}{\pi r_a} \exp\left(-\frac{\psi_{\mathrm{e}}^2}{\pi r_a^2}\right), & \psi_{\mathrm{e}} \geq 0 \\ 0, & \text{otherwise} \end{cases}. \tag{43}$$

This distribution has been widely used in data gathering problems in science and engineering when the modeled data has non-negativity constraints. Some recent examples include the statistical characterization of motion vector data rates in Wyner-Ziv video coding algorithms suitable for WSNs [32], or the statistical characterization of sample amplitudes captured by an image sensor [37], [46]. The characterization of the energy consumption for queries with Half-Gaussian distributed volumes is summarized in the following proposition.

Proposition 4. *The average energy consumption for Half-Gaussian-distributed media query volumes is given by*

$$E_{\mathrm{exp,H}} = \left(g_{\mathrm{e}} + i_{\mathrm{e}} c_{\mathrm{e}} \mathrm{erf}\left(\frac{c_{\mathrm{e}}}{\sqrt{\pi}}\right) + i_{\mathrm{e}}\left(\exp\left(-\frac{c_{\mathrm{e}}^2}{\pi}\right) - 1\right)\right) r_a, \tag{44}$$

where $\mathrm{erf}(\cdot)$ *is the error function, and the one-side variation of the energy consumption from idle mode to active mode is given by*

$$E_{\mathrm{var,H}} = \frac{g_{\mathrm{e}}^2}{2}\left(\left(2c_{\mathrm{e}}^2 + \pi\right)\left(1 - \mathrm{erf}\left(\frac{c_{\mathrm{e}}}{\sqrt{\pi}}\right)\right) - 2c_{\mathrm{e}} \exp\left(-\frac{c_{\mathrm{e}}^2}{\pi}\right)\right) r_a^2. \tag{45}$$

Proof: The expressions (44) and (45) are obtained by substituting the Half-Gaussian PDF (43) in (2) and (3), respectively, and simplifying the integral expressions. ■

In this case, the solutions to (6) and (7) cannot be expressed in closed form. However, they can be efficiently computed using gradient descent given that the error function can be efficiently and accurately approximated with well known methods [47].

4) Billing cost under Pareto, Exponential and Half-Gaussian distribution: We now consider the billing cost for the processing of queries uploaded from n devices via an IoT aggregator. Let us first consider the aggregate query volume distribution modeled via a Pareto distribution with mean $E_{\mathrm{P}}[\Psi_{\mathrm{b}}] = r_{\mathrm{tot}}$ [with r_{tot} given by (18)], i.e.,

$$P_{\mathrm{P}}\left(\psi_{\mathrm{b}}\right) = \begin{cases} \alpha_{\mathrm{b}} \frac{v_{\mathrm{b}}^{\alpha_{\mathrm{b}}}}{\psi_{\mathrm{b}}^{\alpha_{\mathrm{b}}+1}}, & \psi_{\mathrm{b}} \geq v_{\mathrm{b}} \\ 0, & \text{otherwise} \end{cases}, \tag{46}$$

where $\alpha_{\mathrm{b}} > 2$ and $v_{\mathrm{b}} = \frac{\alpha_{\mathrm{b}}-1}{\alpha_{\mathrm{b}}} r_{\mathrm{tot}}$.

Proposition 5. *The average billing cost incurred from processing Pareto-distributed query volumes is given by*

$$B_{\mathrm{exp,P}} = (g_{\mathrm{b}} - i_{\mathrm{b}}) r_{\mathrm{tot}} + (i_{\mathrm{b}} + p_{\mathrm{b}}) \frac{(\alpha_{\mathrm{b}} - 1)^{\alpha_{\mathrm{b}}-1}}{\alpha_{\mathrm{b}}^{\alpha_{\mathrm{b}}} c_{\mathrm{b}}^{\alpha_{\mathrm{b}}-1}} r_{\mathrm{tot}}^{\alpha_{\mathrm{b}}} + i_{\mathrm{b}} c_{\mathrm{b}}. \tag{47}$$

The minimum billing cost is obtained when

$$c_{\mathrm{b,P}} = \left(\frac{i_{\mathrm{b}} + p_{\mathrm{b}}}{i_{\mathrm{b}}}\right)^{\frac{1}{\alpha_{\mathrm{b}}}} \frac{\alpha_{\mathrm{b}} - 1}{\alpha_{\mathrm{b}}} r_{\mathrm{tot}}, \tag{48}$$

and it is given by

$$\min\{B_{\mathrm{exp,P}}\} = \left[g_{\mathrm{b}} - i_{\mathrm{b}} + i_{\mathrm{b}}\left(\frac{i_{\mathrm{b}} + p_{\mathrm{b}}}{i_{\mathrm{b}}}\right)^{\frac{1}{\alpha_{\mathrm{b}}}}\right] r_{\mathrm{tot}}. \tag{49}$$

Proof: The proof stems from the evaluation of the general solution expressed in (16) under the usage of the Pareto PDF. ■

In order to ensure that the average billing cost is B_{mean} when the maximum query volume constraint, V_{max}, is satisfied, we first need to guarantee that

$$B_{\mathrm{mean}} \leq V_{\mathrm{max}}\left[g_{\mathrm{b}} - i_{\mathrm{b}} + i_{\mathrm{b}}\left(\frac{i_{\mathrm{b}} + p_{\mathrm{b}}}{i_{\mathrm{b}}}\right)^{\frac{1}{\alpha_{\mathrm{b}}}}\right]. \tag{50}$$

Then, by determining the number of devices, $n_1, \ldots, n_A$, for activity zone based on proportional fairness, and by setting $\min\{B_{\mathrm{exp,P}}\} = B_{\mathrm{mean}}$ in (49), we obtain

$$n_{\mathrm{P},a} = \frac{B_{\mathrm{mean}}}{\left[g_{\mathrm{b}} - i_{\mathrm{b}} + i_{\mathrm{b}}\left(\frac{i_{\mathrm{b}}+p_{\mathrm{b}}}{i_{\mathrm{b}}}\right)^{\frac{1}{\alpha_{\mathrm{b}}}}\right] A r_a}, \tag{51}$$

where B_{mean} is bounded by the constraint of (50). If B_{mean} is set such that V_{max} is expected to be fully utilized, i.e., (50) becomes an equality, then $n_{\mathrm{P},a}$ is given by (31). We also note that, when assuming that the aggregate query volume is Pareto distributed, by letting $\alpha_{\mathrm{b}} \to +\infty$, we can analyze the case when the aggregate query volume at the IoT is fixed and equal to r_{tot}. In this case, if $c_{\mathrm{b}} \geq r_{\mathrm{tot}}$, the average billing cost is simply given by

$$B_{\mathrm{exp,P}} = (g_{\mathrm{b}} - i_{\mathrm{b}}) r_{\mathrm{tot}} + i_{\mathrm{b}} c_{\mathrm{b}}, \tag{52}$$

which is minimized by setting c_{b} equal to the mean, i.e., $c_{\mathrm{b,P}} = r_{\mathrm{tot}}$.

Let us consider the aggregate query volume distribution modeled via an Exponential distribution with mean $E_{\mathrm{E}}[\Psi_{\mathrm{b}}] = r_{\mathrm{tot}}$, i.e.,

$$P_{\mathrm{E}}(\psi_{\mathrm{b}}) = \frac{1}{r_{\mathrm{tot}}} \exp\left(-\frac{1}{r_{\mathrm{tot}}} \psi_{\mathrm{b}}\right), \tag{53}$$

for $\psi_{\mathrm{b}} \geq 0$.

Proposition 6. *The average billing cost incurred from processing Exponentially-distributed query volumes is given by*

$$B_{\mathrm{exp,E}} = (g_{\mathrm{b}} - i_{\mathrm{b}}) r_{\mathrm{tot}} + i_{\mathrm{b}} c_{\mathrm{b}} + (i_{\mathrm{b}} + p_{\mathrm{b}}) r_{\mathrm{tot}} e^{-\frac{c_{\mathrm{b}}}{r_{\mathrm{tot}}}}. \tag{54}$$

The minimum billing cost is obtained when

$$c_{\mathrm{b,E}} = r_{\mathrm{tot}} \ln \frac{i_{\mathrm{b}} + p_{\mathrm{b}}}{i_{\mathrm{b}}}, \tag{55}$$

and it is given by

$$\min\{B_{\mathrm{exp,E}}\} = \left(g_{\mathrm{b}} + i_{\mathrm{b}} \ln \frac{i_{\mathrm{b}} + p_{\mathrm{b}}}{i_{\mathrm{b}}}\right) r_{\mathrm{tot}}. \tag{56}$$

Proof: The proof stems from the evaluation of the general solution expressed in (16) under the usage of the Exponential PDF. ■

In order to ensure that the average billing cost is B_{mean} when the maximum query volume constraint $V_{\max}$ is satisfied, we first need to guarantee that

$$B_{\text{mean}} \le V_{\max}\left(g_{\text{b}} + i_{\text{b}} \ln \frac{i_{\text{b}} + p_{\text{b}}}{i_{\text{b}}}\right). \tag{57}$$

Then, by adopting proportional fairness to allocate the number of devices for each activity zone $n_1, \ldots, n_A$, and by setting $\min\{B_{\text{exp,P}}\} = B_{\text{mean}}$ in (56), we obtain

$$n_{\text{E},a} = \frac{B_{\text{mean}}}{\left(g_{\text{b}} + i_{\text{b}} \ln \frac{i_{\text{b}} + p_{\text{b}}}{i_{\text{b}}}\right) A r_a}, \tag{58}$$

where the value of B_{mean} is upper-bounded by (57). Similarly as before, if (57) is met with equality, then $n_{\text{E},a}$ is given by the simple solution of (31).

Finally, consider the case when the aggregate query volume is Half-Gaussian distributed with mean $E_{\text{H}}[\Psi_{\text{b}}] = r_{\text{tot}}$, i.e.,

$$P_{\text{H}}(\psi_{\text{b}}) = \begin{cases} \frac{2}{\pi r_{\text{tot}}} \exp\left(-\frac{\psi_{\text{b}}^2}{\pi r_{\text{tot}}^2}\right), & \psi_{\text{b}} \ge 0 \\ 0, & \text{otherwise} \end{cases}. \tag{59}$$

Proposition 7. *The average billing cost incurred from processing Half-Gaussian-distributed query volumes is given by*

$$\begin{aligned} B_{\text{exp,H}} = {} & (g_{\text{b}} + p_{\text{b}}) r_{\text{tot}} - p_{\text{b}} c_{\text{b}} + (i_{\text{b}} + p_{\text{b}}) \\ & \times \left(c_{\text{b}} \text{erf}\left(\frac{c_{\text{b}}}{\sqrt{\pi} r_{\text{tot}}}\right) + rn \left(\exp\left(-\frac{c_{\text{b}}^2}{\pi r_{\text{tot}}^2}\right) - 1 \right) \right). \end{aligned} \tag{60}$$

The minimum billing cost is obtained when

$$c_{\text{b,H}} = r_{\text{tot}} \sqrt{\pi} \text{erf}^{-1}\left(\frac{p_{\text{b}}}{p_{\text{b}} + i_{\text{b}}}\right), \tag{61}$$

and it is given by

$$\begin{aligned} \min\{B_{\text{exp,H}}\} = r_{\text{tot}} \Bigg[& g_{\text{b}} - i_{\text{b}} + (i_{\text{b}} + p_{\text{b}}) \\ & \times \exp\left(-\left(\text{erf}^{-1}\left(\frac{p_{\text{b}}}{p_{\text{b}} + i_{\text{b}}}\right)\right)^2\right)\Bigg]. \end{aligned} \tag{62}$$

Proof: The proof stems from the evaluation of the general solution expressed in (16) under the usage of the Half-Gaussian PDF. ∎

In order to ensure that the average billing cost is B_{mean} when the maximum query volume constraint $V_{\max}$ is satisfied, we first need to guarantee that

$$\begin{aligned} B_{\text{mean}} \le V_{\max} \Bigg[& g_{\text{b}} - i_{\text{b}} + (i_{\text{b}} + p_{\text{b}}) \\ & \times \exp\left(-\left(\text{erf}^{-1}\left(\frac{p_{\text{b}}}{p_{\text{b}} + i_{\text{b}}}\right)\right)^2\right)\Bigg]. \end{aligned} \tag{63}$$

Via a proportionally-fair allocation of the number of devices for each activity zone [and by setting $\min\{B_{\text{exp,P}}\} = B_{\text{mean}}$ in (62)], we obtain

$$n_{\text{H},a} = \frac{B_{\text{mean}}}{\left[g_{\text{b}} - i_{\text{b}} + (i_{\text{b}} + p_{\text{b}}) \exp\left(-\left(\text{erf}^{-1}\left(\frac{p_{\text{b}}}{p_{\text{b}} + i_{\text{b}}}\right)\right)^2\right)\right] A r_a}, \tag{64}$$

where the value of B_{mean} is upper-bounded by (63). If the billing is set such that $V_{\max}$ is expected to be fully utilized, i.e., (63) is met with equality, then $n_{\text{H},a}$ is given by (31), i.e., mutual coupling is achieved between the energy consumption parameters ($E_{\text{exp,U}}$ and $E_{\text{var,U}}$), the desired cloud billing cost (B_{mean}), and the aggregator's data transmission volume ($V_{\max}$).

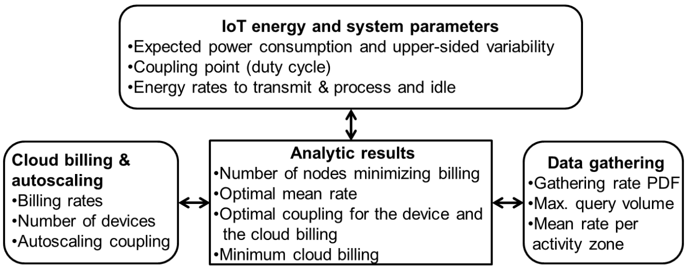
IoT energy and system parameters
•Expected power consumption and upper-sided variability
•Coupling point (duty cycle)
•Energy rates to transmit & process and idle
Cloud billing & autoscaling
•Billing rates
•Number of devices
•Autoscaling coupling
Analytic results
•Number of nodes minimizing billing
•Optimal mean rate
•Optimal coupling for the device and the cloud billing
•Minimum cloud billing
Data gathering
•Gathering rate PDF
•Max. query volume
•Mean rate per activity zone

Figure 2. Conceptual illustration of the linkage between: IoT system parameters, cloud billing & autoscaling, and data gathering. When parameters from two out of three domains are provided, our analytic framework can be used to tune the parameters of the third.

D. Discussion

The results of this section can be used in practical applications to assess the impact in the required energy rates when the statistics of the query generation and transmission follow a certain PDF and the cloud billing costs are fixed. Conversely, if a particular IoT device technology is chosen, under the knowledge of the system and data gathering parameters, one can establish the appropriate cloud billing rates and the number of devices to include in each IoT aggregator in order to minimize the cloud infrastructure cost. Finally, for given IoT and cloud infrastructure parameters, one can assess the achievable query generation and transmission rates such that the IoT cluster leads to the optimal coupling between energy consumption and cloud billing cost. Thus, as shown in Fig. 2, our analytic results allow for the linkage of energy, data gathering and cloud billing parameters within IoT clusters of devices. Hence, our analysis can be used for early-stage exploration of the capabilities of a particular IoT infrastructure, in conjunction with the data gathering requirements of a particular application, prior to embarking in cumbersome development and testing in the field.

IV. Evaluation of the Analytic Results

To validate the proposed analytic modeling framework of Propositions 2–7, we performed a series of experiments based on a visual sensor network connected to an IoT aggregator, and eventually to an AWS S3 repository plus EC2 cluster of on-demand instances. The following subsections present the hardware and application specifications, as well as the achieved results.

A. System Specification

We utilized a visual sensor network composed of multiple BeagleBone Linux embedded platforms [17], [48]. Each BeagleBone is equipped with a RadiumBoard CameraCape board to provide for the video frame acquisition. For energy-efficient processing, we downsampled all input images to QVGA (320 × 240) resolution. Further, our deployment involved:

1) a portable computer acting as the IoT aggregator, i.e., collecting all bitstreams via a star topology with $n_{\mathrm{tot}} = 12$ nodes and the recently-proposed (and available as open source) TFDMA protocol [49] for contention-free MAC-layer coordination;
2) an AWS S3 bucket where the IoT aggregator continuously uploads all queries via a TCP/IP connection using a Cron Job and the AWS Command Line Interface;
3) one reserved AWS instance running as the control server and assigning query volumes from S3 to AWS EC2 on-demand instances that serve as compute units; via AWS Auto Scaling, within each monitoring instance of T seconds, the number of on-demand instances is set to:
    - 3 when the query volume is below c_{b} b ("idle" case).
    - 30 when the volume exceeds c_{b} b ("active" case).

    Under our deployment and the utilized application, the uploaded query vectors are matched with the feature vectors extracted from $80,000$ images of similar content. The corresponding billing rates per query bit for this matching operation were found to be $i_{\mathrm{b}} = 6.27 \times 10^{-11}$ \$/b and $p_{\mathrm{b}} = 6.27 \times 10^{-10}$ \$/b. Regarding query traffic upload and storage costs, the corresponding billing rate per query bit was found to be $g_{\mathrm{b}} = 2.09 \times 10^{-10}$ \$/b.

We note that no WiFi or other IEEE802.15.4 networks were concurrently operating in the utilized channels of the 2.4 GHz band. However, even if IEEE 802.11 or other IEEE 802.15.4 networks coexist with the proposed deployment, well-known channel hopping schemes like TSCH [50] can be used at the MAC layer to mitigate such external interference. Moreover, experiments have shown that such protocols can scale to hundreds or even thousands of nodes [50]. Therefore, our evaluation is pertinent to such scenarios that may be deployed in the next few years within an IoT paradigm [51].

B. Visual Similarity Identification Based on the Vector of Locally Aggregated Descriptors (VLAD)

Each BeagleBone runs a basic motion detection algorithm (based on successive frame differencing) that generates a visual query only when sufficient motion is detected between the captured video frames. The query vectors were generated using the state-of-the-art VLAD algorithm of Jegou *et. al.* [16], which is based on SIFT feature extraction and compaction using local feature centers and a PCA projection matrix, both of which are derived offline via training with representative video data [16]. The VLAD descriptor (i.e., query) size was set to 8192 b (256 coefficients of 32 b each).

With respect to the visual feature extraction, dedicated energy-measurement tests were performed with the Beaglebone following the energy measurement setup of our previous work [17] (repeated tests with a resistor in series to the Beaglebone board and a high-frequency oscilloscope to capture the power consumption profile across repeated monitoring intervals). Under the utilized setup, we measured the average energy cost to produce and transmit a query bit, as well as the average initialization cost per frame for both application scenarios. The resulting energy rates were: $g_{\mathrm{e}} = 1.78 \times 10^{-6}$ J/b and $i_{\mathrm{e}} = 6.10 \times 10^{-7}$ J/b. Moreover, under the utilized application, the Beaglebone can generate up to 1 query per second while being constantly active, i.e., $8192T$ b per monitoring interval of T seconds. By setting mean query rates such that $E[\Psi_{\mathrm{e}}] \leq 2048T$ b (i.e., up to 0.25 queries per second), this allows for $c_{\mathrm{e}} \in (0, 4)$. In practice, we restricted the utilized values for c_{e} to $(0, 2]$ since higher values lead to the frame acquisition frequently exceeding 1 frame per second, which can lead to system instability.

C. Results with Controlled Query Generation that Matches the Marginal PDFs Considered in the Theoretical Analysis

Under the settings described previously, our first goal is to validate the analytic expressions of Section III that form the mathematical foundation for Propositions 2–4. To this end, we create a controlled query data production process on each node by: *(i)* artificially setting several sets of query volumes according to the marginal PDFs of Section III via rejection sampling [52], a.k.a., Monte Carlo sampling; *(ii)* setting the mean query volume size per monitoring interval, r, to predetermined values. The sets containing query volume sizes are preloaded onto the memory of each sensor node during the setup phase. At run time, each BeagleBone node runs a special routine, which, per monitoring interval t: *(i)* reads the corresponding query volume size, $v(t)$, from the preloaded set; *(ii)* captures and processes $\frac{v(t)}{8192}$ frames, *(iii)* transmits the produced $v(t)$ query bits to the IoT aggregator; *(iv)* if $v(t) < c_{\mathrm{e}} E[\Psi_{\mathrm{e}}]$, captures and processes $\frac{c_{\mathrm{e}} E[\Psi_{\mathrm{e}}] - v(t)}{8192}$ additional frames without transmitting queries. In this way, we emulate the actual operation of the node under various query volumes that match the statistical models considered by our analysis, and various thresholds c_{e} for switching between "idle" and "active" states. This controlled experiment is designed to confirm the validity of our analytic derivations when the operating conditions match the model assumptions precisely.

Indicative experimental results for monitoring time interval of $T = 60$ s are reported in Fig. 3 and Fig. 4 for $r = 81,920$ b. It is evident that the theoretical results match the Monte Carlo experiments regarding energy consumption for all the tested distributions, with all the R^2 values (coefficients of determination) between the experimental and the model points being above 0.9964. We have observed the same level of accuracy for the proposed model under a variety of data sizes (r) and active time interval durations (T), but omit these repetitive experiments for brevity of exposition.

Similar experiments have been carried out in order to validate the analytic expressions of Propositions 5–7 regarding the average billing cost. Specifically, we have submitted indicative queries to the cloud-computing service with volumes that have been generated according to the marginal PDFs of Section III via rejection sampling under various numbers of devices per IoT cluster (n_{tot}) and various average query volumes. The aggregated queries are uploaded to the dedicated S3 bucket for the service and are processed by a number of instances that is controlled by the AWS Auto Scaling rules stated in the previous subsection. In this case, we used $T = 600$ s and

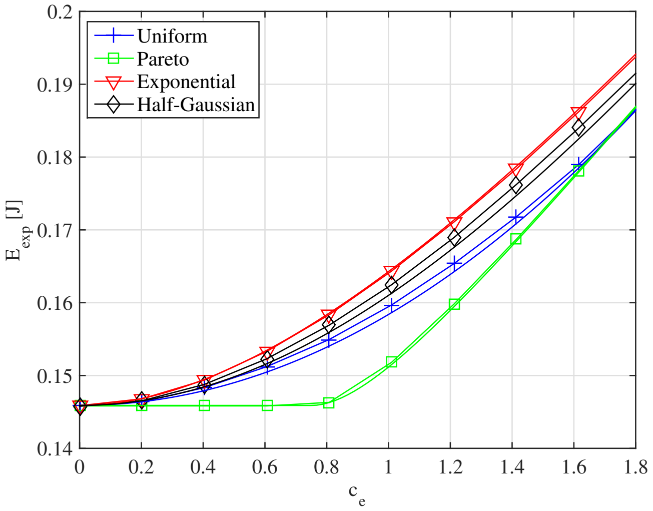

Figure 3. Average energy consumption E_{exp} vs. c_{e}. The average query volume was set to $r = 81,920$ b. For the case of Pareto distribution, we used $\alpha_{\mathrm{e}} = 4$. Lines with markers: Monte Carlo experiments; Lines without markers: theoretical predictions.

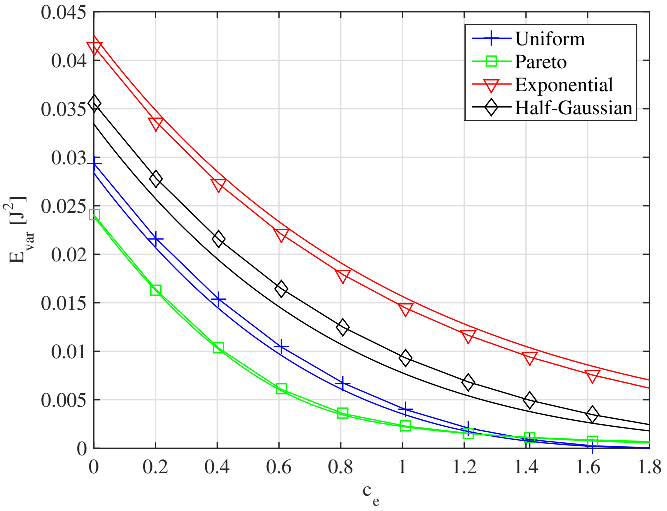

Figure 4. One-sided energy consumption E_{var} vs. c_{e}. The average query volume was set to $r = 81,920$ b. For the case of Pareto distribution, we used $\alpha_{\mathrm{e}} = 4$. Lines with markers: Monte Carlo experiments; Lines without markers: theoretical predictions.

varied the value of c_{b} in order to see the incurred infrastructure billing costs under a variety of Auto Scaling thresholds.

Fig. 5 presents indicative results under this setup. Evidently, the theoretical results follow the trends of the experimental data, with R^2 coefficients being above 0.9947 for all the distributions under consideration. However, the theoretical predictions tend to always underestimate the experimental values. This underestimation is due to the fact that our analysis does not take into account some practical latency and cost aspects of the service, for example that switching between “idle”, “active” states is not instantaneous and other cost overheads (such as the cost of the control server) are not taken into account by our analysis. Similar results to Fig. 5 have been obtained for a variety of average query volumes and monitoring intervals, but are omitted for brevity of exposition.

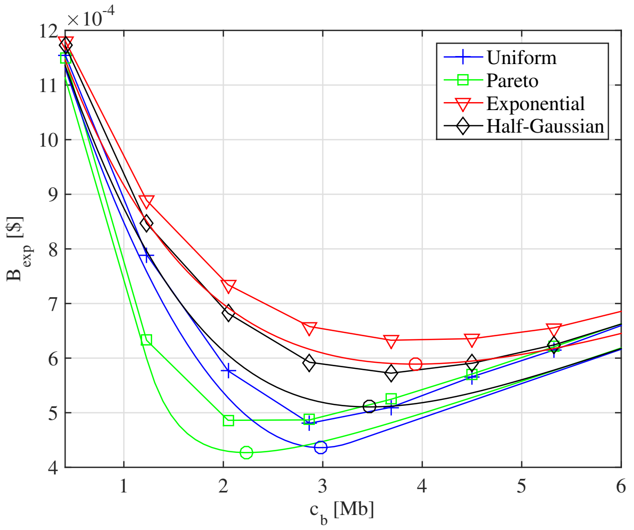

Figure 5. Average billing cost B_{exp} vs. c_{b}. The average query volume per device was set to $r = 163,840$ b and the experiment corresponds to $n = 10$ devices. For the case of Pareto distribution, we used $\alpha_{\mathrm{e}} = 4$. Lines with markers: Monte Carlo experiments; Lines without markers: theoretical predictions. The circles indicate minimum billing values as predicted by the analysis in Section III.

D. Results with Real Data

We now present results when repeating the visual query generation, transmission and cloud-based processing for 25 monitoring intervals under a practical deployment within several research staff offices of the Electronic and Electrical Engineering Department of University College London. The deployment environment comprises a large shared office space, which is composed of areas with low query generation activity (seated desk areas with low movement of people) and areas with high query generation activity (corridor areas with high movement of people).

1) Accuracy of energy estimation under a fixed setup and one activity zone: In the first batch of tests, each device's camera is set to fixed capture rate of 5 frames per second. Via successive frame differencing for motion detection, VLAD queries where generated when the contents of frames varied beyond a preset threshold, e.g., when people passed (or moved) in front of the device camera. Back-end query similarity identification was done using prestored VLAD signatures of $80,000$ images of similar content based on the AWS setup described in the previous subsection.

Once data has been collected, we fitted[5] the query production volumes to one of the distributions used in Section III, i.e., assuming only one activity zone. In the performed experiment, and under monitoring interval of $T = 60$ s for the devices, we found that the real data query volume histogram agreed best with the Exponential distribution with $r = 82,616$

[5] Fitting is performed by matching the average data size r of each distribution to the average data size of the JPEG compressed frames or the set of visual features.

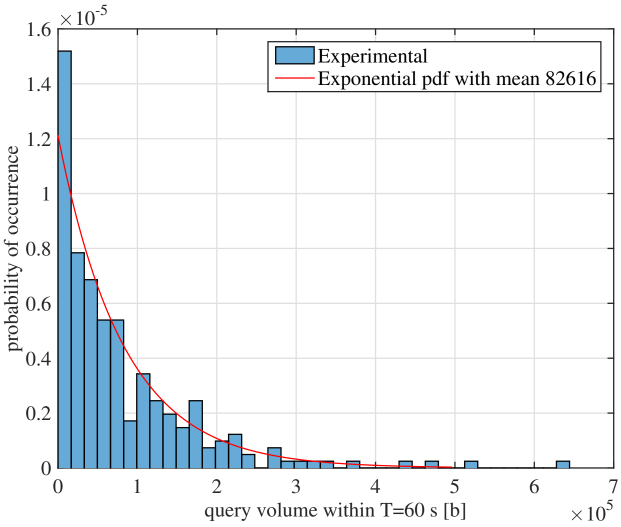

Figure 6. Probability histogram of query volume for $T = 60$ s generated from our deployment experiments and the best fit obtained via the Exponential distribution.

b. For $T = \{600, 1200\}$ s, the best fit was found to be the Pareto distribution with: $r = 816,250$ b and $\alpha = 3.89$, and $r = 1,569,700$ b and $\alpha = 3.95$, respectively. An example for the fit obtained with the Exponential distribution is given in Fig. 6. Moreover, with respect to c_{e}, we found that, under the acquisition of 5 frames per second, the system switched between "idle" and "active" states at $c_{\mathrm{e}} \cong 0.75$.

Under this setup and with the fitted values for Exponential and Pareto PDFs, Table II presents the obtained experimental and theoretical values (via Propositions 2 and 3) for the expected energy and the one-sided energy variance for two monitoring intervals. It is observed that, despite the modeling mismatch due to the fitting shown in Fig. 6, the theoretical predictions on the expected energy consumption are always within 2% of the experimentally-derived values, whereas theoretical predictions on the one-sided energy variation are within 20% of experimental data. As such, the proposed energy-consumption model can be used for early-stage testing of feasible application deployments with respect to their energy consumption in order to determine the impact of various options, prior to time-consuming experimentation in the field.

2) Energy and billing cost reduction under parameter tuning and two activity zones: We consider a further example to showcase how the theoretical modeling presented in Section III can be leveraged in order to allow parameter tuning within $T = \{600, 1200\}$. When $T = 600$ s, we set the maximum admissible one-sided energy variation to $E_{\mathrm{max\_var}} = 0.35$ J^2, whereas, when $T = 1200$ s, we set $E_{\mathrm{max\_var}} = 1$ J^2. We report experimental results generated from our deployment for a scenario where $A = 2$ different zones are present, corresponding to "low" and "high" activity. This was achieved by positioning some devices in areas with low movement of people (seated desk area of a large shared office) and some others in areas with high movement (corridor area of a large shared office). The devices contained in the first (i.e., "low activity") zone produce query volumes that are best approximated by the Pareto distribution with $r_1 = 160,000$ b and $\alpha_1 = 2.42$, if $T = 600$ s, and $r_1 = 320,000$ b and $\alpha_1 = 2.58$, if $T = 1200$ s. On the other hand, the devices in the second (i.e., "high activity") zone are best approximated via the Pareto distribution, albeit with $r_2 = 4,915,600$ b and $\alpha_2 = 3.27$, if $T = 600$ s, and $r_2 = 9,572,600$ b and $\alpha_2 = 4.10$, if $T = 1200$ s. The IoT aggregator admits $n_1 = 10$ devices from the low-activity zone and $n_2 = 2$ devices from the high-activity zone, thus resulting in a total query volume occupation of 11.43 Mb, when $T = 600$ s, and 22.35 Mb, when $T = 1200$ s. Instead of presetting the frame acquisition to fixed value (5 frames per second, which led to $c_{\mathrm{e}} \cong 0.75$), we now change the acquisition rate, thereby controlling the activation threshold c_{e}. Our aim is to set c_{e} to the value that minimizes the expected energy consumption while verifying the constraint on the one-sided energy variation, which is given by (35). We then compare the expected energy consumption obtained with such setting, with the one obtained via two baseline solutions that impose $c_{\mathrm{e}} = 1.5$ or $c_{\mathrm{e}} = 2$ (corresponding to acquiring 10 and 13.3 frames per second). The obtained values from (35) were found to be $c_{\mathrm{e,P,dual}} = 0.82$, corresponding to capturing 5.5 frames per second, for the case $T = 600$ s, and $c_{\mathrm{e,P,dual}} = 0.92$, corresponding to capturing 6 frames per second, for the case $T = 1200$ s. The obtained energy consumption results, reported in Table III, show that by selecting c_{e} via the proposed analytic framework, we can achieve gains of up to $23,55\%$ with respect to baseline settings. It is important to note that, beyond the presented comparisons of Table III, the optimal tuning of c_{e} always led to decreased energy consumption in comparison to all other baseline settings attempted, thereby experimentally confirming the validity of Propositions 1 and 2.

Let us now consider the billing cost associated to the cloud infrastructure. Under the utilized setup, we determined the autoscaling threshold, c_{b}, that is expected to lead to the minimum cloud infrastructure billing cost based on Proposition 5. We then benchmarked the obtained cost of the system under this threshold against the intuitive (albeit *ad-hoc*) baseline setting of $c_{\mathrm{b}} = r_{\mathrm{tot}} = r_1 n_1 + r_2 n_2$, which corresponds to the autoscaling threshold being set to match the average query volume of all n_{tot} devices. The results, given in Table IV, show that the obtained billing cost is 14% (for $T = 600$ s) and 12% (for $T = 1200$ s) lower than the case of the same query volume processing under the baseline autoscaling threshold. This demonstrates that establishing the system parameters based on the theoretical analysis can lead to important cost savings within cloud-based media query processing systems. Importantly, the optimal values derived by (48) have consistently outperformed all other baseline settings attempted, thereby experimentally confirming the validity of Proposition 5.

V. Conclusions

We propose a novel theoretical framework for establishing trade-offs in the energy consumption and infrastructure billing cost of Internet-of-Things oriented deployments comprising mobile devices generating media queries that are processed by a back-end cloud computing service. Our analysis incorporates energy consumption and cloud infrastructure billing

Table II
EXPECTED ENERGY CONSUMPTION AND ONE-SIDED VARIATION. EXPERIMENTAL RESULTS FROM OUR DEPLOYMENT AND THEORETICAL PREDICTION, $c_e = 0.75$.

	Theoretical	Experimental
$T = 60$ s	$E_{exp} = 0.1588$ J $E_{var} = 0.0201$ J^2	$E_{exp} = 0.1603$ J $E_{var} = 0.0254$ J^2
$T = 1200$ s	$E_{exp} = 2.7965$ J $E_{var} = 1.4349$ J^2	$E_{exp} = 2.8440$ J $E_{var} = 1.2234$ J^2

Table III
EXPECTED ENERGY CONSUMPTION WITH ONE-SIDED VARIATION CONSTRAINT GENERATED FROM OUR DEPLOYMENT EXPERIMENTS. THE BASELINE SOLUTIONS CORRESPOND TO SETTING c_e EQUAL TO 1.5 OR 2. THE PROPOSED SOLUTION IS OBTAINED WITH $c_e = 0.82$ AND $c_e = 0.92$, DERIVED VIA (35).

	$c_e = 1.5$	$c_e = 2$	c_e as in (35)
$T = 600$ s $E_{var} \leq 0.35$ J^2	$E_{exp} = 1.72$ J (13.26%)	$E_{exp} = 1.95$ J (23, 55%)	$E_{exp} = 1.49$ J ($c_e = 0.82$)
$T = 1200$ s $E_{var} \leq 1.00$ J^2	$E_{exp} = 3.30$ J (12, 40%)	$E_{exp} = 3.75$ J (22, 95%)	$E_{exp} = 2.89$ J ($c_e = 0.92$)

rates when the devices and the cloud computing system adapt their resource consumption according to the volume of generated queries by switching between "idle" and "active" states. Experiments with an embedded platform and Amazon Web Services based back-end processing for visual query generation, transmission and similarity detection demonstrate that the proposed model forms a framework that accurately incorporates the effect of various system parameters with respect to energy consumption and cloud billing costs. Therefore, variations of the proposed analytic modeling can be used for early-stage analysis of possible deployments, or limit studies of the expected performance under a wide range of parameter settings, prior to costly deployments in the field.

VI. APPENDIX

A. Proof of Proposition 1

We observe that E_{exp} is strictly-increasing in c_e, since $\frac{dE_{exp}}{dc_e} > 0$ for all values of c_e larger than the left extremum of the support of Ψ_e. Moreover, E_{var} is strictly-decreasing in c_e. In order to prove this, we express the dependence of E_{var} from c_e by using the notation $E_{var}(c_e)$, and we consider two values $c'_e \geq 0$ and $c''_e \geq 0$ such that $c'_e > c''_e$. Then,

$$E_{var}(c'_e) = g_e^2 \int_{c'_e E[\Psi_e]}^{+\infty} (\psi_e - c'_e E[\Psi_e])^2 P_a(\psi_e) d\psi_e \quad (65)$$

$$\leq g_e^2 \int_{c''_e E[\Psi_e]}^{+\infty} (\psi_e - c'_e E[\Psi_e])^2 P_a(\psi_e) d\psi_e \quad (66)$$

$$< g_e^2 \int_{c''_e E[\Psi_e]}^{+\infty} (\psi_e - c''_e E[\Psi_e])^2 P_a(\psi_e) d\psi_e \quad (67)$$

$$= E_{var}(c''_e), \quad (68)$$

where the first inequality follows from integrating a positive function over a subset, and the second inequality follows from $(\psi_e - c'_e E[\Psi_e])^2 > (\psi_e - c''_e E[\Psi_e])^2$, when $\psi_e \geq c''_e E[\Psi_e]$.

The monotonicity properties of E_{exp} and E_{var} imply that the constraints in (6) and (7) are active at the optimum point. Therefore, on recalling that such optimization problems are convex, complementary slackness [53] implies that the solution of the problem (6) is such that $E_{exp} = E_{max\_exp}$ and the solution of (7) is such that $E_{var} = E_{max\_var}$.

Table IV
EXPECTED BILLING COST GENERATED FROM OUR DEPLOYMENT EXPERIMENTS. THE BASELINE SOLUTION CORRESPONDS TO SETTING $c_b = r_1 n_1 + r_2 n_2$. THE PROPOSED SOLUTION IS OBTAINED WITH c_b AS IN PROPOSITION 5.

	Baseline	Proposition 5	Saving
$T = 600$ s $n_1 = 10$ $n_2 = 2$	$B_{exp} = 3.38 \cdot 10^{-3}$ \$ $c_b = 11.43$ Mb	$B_{exp} = 2.89 \cdot 10^{-3}$ \$ $c_b = 14.90$ Mb	14 %
$T = 1200$ s $n_1 = 10$ $n_2 = 2$	$B_{exp} = 5.86 \cdot 10^{-3}$ \$ $c_b = 22.35$ Mb	$B_{exp} = 5.15 \cdot 10^{-3}$ \$ $c_b = 27.75$ Mb	12 %

REFERENCES

[1] D. Siewiorek, "Generation smartphone," *IEEE Spectrum*, vol. 49, no. 9, pp. 54–58, 2012.

[2] Y. Wen, X. Zhu, J. Rodrigues, and C. W. Chen, "Cloud mobile media: Reflections and outlook," *IEEE Trans. on Multimedia*, vol. 16, no. 4, pp. 885–902, June 2014.

[3] X. Ma, Y. Zhao, L. Zhang, H. Wang, and L. Peng, "When mobile terminals meet the cloud: computation offloading as the bridge," *IEEE Network*, vol. 27, no. 5, pp. 28–33, 2013.

[4] W. Zhang, Y. Wen, J. Wu, and H. Li, "Toward a unified elastic computing platform for smartphones with cloud support," *IEEE Network*, vol. 27, no. 5, pp. 34–40, 2013.

[5] V. C. M. Leung, M. Chen, M. Guizani, and B. Vucetic, "Cloud-assisted mobile computing and pervasive services," *IEEE Network*, vol. 27, no. 5, pp. 4–5, 2013.

[6] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *2012 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2012, pp. 59–66.

[7] B. Girod, V. Chandrasekhar, D. M. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 61–76, 2011.

[8] J. Serra *et al.*, "Audio cover song identification and similarity: background, approaches, evaluation, and beyond," in *Advances in Music Information Retrieval*. Springer, 2010, pp. 307–332.

[9] B. C. Becker and E. G. Ortiz, "Evaluation of face recognition techniques for application to facebook," in *8th IEEE Internat. Conf. on Automatic Face & Gesture Recognition, 2008. FG'08*. IEEE, 2008, pp. 1–6.

[10] H. Sellahewa and S. A. Jassim, "Wavelet-based face verification for constrained platforms," in *SPIE Proc. Defense and Secur. Conf.* International Society for Optics and Photonics, 2005, pp. 173–183.

[11] H. Bredin, A. Miguel, I. H. Witten, and G. Chollet, "Detecting replay attacks in audiovisual identity verification," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., 2006, ICASSP 2006*, vol. 1. IEEE, 2006, pp. I–I.

[12] Y.-G. Jiang, Q. Dai, T. Mei, Y. Rui, and S.-F. Chang, "Super fast event recognition in internet videos," *IEEE Trans. on Multimedia*, vol. 17, no. 8, pp. 1174–1186, Aug 2015.

[13] S. Marcel *et al.*, "MOBIO: Mobile biometric face and speaker authentication," in *Proc. IEEE Conf. Comput. Vision and Pat. Rec.*, San Francisco, CA, USA, 2010.

[14] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *IEEE Int. Conf. on Comput. Vis. and Patt. Rec. (CVPR)*, 2012, pp. 2911–2918.

[15] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2010, pp. 3384–3391.

[16] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Patt. Anal. and Machine Intel.*, vol. 34, no. 9, pp. 1704–1716, 2012.

[17] A. Redondi *et al.*, “Energy consumption of visual sensor networks: Impact of spatio-temporal coverage,” *IEEE Trans. on Circ. and Syst. for Video Technol.*, vol. 24, no. 12, pp. 2117–2131, 2014.

[18] S. Ren *et al.*, “Dynamic scheduling for energy minimization in delay-sensitive stream mining,” *IEEE Trans. on Signal Processing*, vol. 62, no. 20, pp. 5439–5448, 2014.

[19] H. Besbes *et al.*, “Analytic conditions for energy neutrality in uniformly-formed wireless sensor networks,” *IEEE Trans. on Wireless Commun.*, vol. 12, no. 10, pp. 4916–4931, 2013.

[20] L. Benini, A. Bogliolo, and G. De Micheli, “A survey of design techniques for system-level dynamic power management,” *IEEE Trans. on Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 3, pp. 299–316, 2000.

[21] L. Tse-Hua and A. Tewfik, “A resource management strategy in wireless multimedia communications-total power saving in mobile terminals with a guaranteed QoS,” *IEEE Trans. on Multimedia*, vol. 5, no. 2, pp. 267–281, June 2003.

[22] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, “Energy management in wireless sensor networks with energy-hungry sensors,” *IEEE Instr. & Meas. Mag.*, vol. 12, no. 2, pp. 16–23, 2009.

[23] Q. Li *et al.*, “Streaming-viability analysis and packet scheduling for video over in-vehicle wireless networks,” *IEEE Trans. on Vehic. Technol.*, vol. 56, no. 6, pp. 3533–3549, 2007.

[24] D. Anastasia and Y. Andreopoulos, “Throughput-distortion computation of generic matrix multiplication: Toward a computation channel for digital signal processing systems,” *IEEE Trans. on Signal Process.*, vol. 60, no. 4, pp. 2024–2037, 2012.

[25] V. Spiliotopoulos *et al.*, “Quantization effect on vlsi implementations for the 9/7 dwt filters,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, 2001, ICASSP'01*, vol. 2. IEEE, 2001, pp. 1197–1200.

[26] Y. Andreopoulos and M. van der Schaar, “Incremental refinement of computation for the discrete wavelet transform,” *IEEE Trans. on Signal Process.*, vol. 56, no. 1, pp. 140–157, 2008.

[27] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, Sep. 2007.

[28] J. Yang and S. Ulukus, “Optimal packet scheduling in an energy harvesting communication system,” *IEEE Trans. on Communications*, vol. 60, no. 1, pp. 220–230, 2012.

[29] K. Tutuncuoglu and A. Yener, “Optimum transmission policies for battery limited energy harvesting nodes,” *IEEE Trans. on Wireless Communications*, vol. 11, no. 3, pp. 1180–1189, 2012.

[30] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, “Senseye: a multi-tier camera sensor network,” in *13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 229–238.

[31] A. Rowe, D. Goel, and R. Rajkumar, “Firefly mosaic: A vision-enabled wireless sensor networking system,” in *Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International*. IEEE, 2007, pp. 459–468.

[32] M. Tagliasacchi, S. Tubaro, and A. Sarti, “On the modeling of motion in Wyner-Ziv video coding,” in *IEEE Int. Conf. on Image Process.* IEEE, 2006, pp. 593–596.

[33] A. Redondi, M. Cesana, and M. Tagliasacchi, “Low bitrate coding schemes for local image descriptors,” in *MMSP*, 2012, pp. 124–129.

[34] G. WernerAllen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, “Fidelity and yield in a volcano monitoring sensor network,” in *7th symposium on Operating systems design and implementation*. ACM, 2006, pp. 381–396.

[35] D. Palma, L. Bencini, G. Collodi, G. Manes, F. Chiti, R. Fantacci, and A. Manes, “Distributed monitoring systems for agriculture based on wireless sensor network technology,” *International Journal on Advances in Networks and Services,*, vol. 3, 2010.

[36] A. Redondi, M. Chirico, L. Borsani, M. Cesana, and M. Tagliasacchi, “An integrated system based on wireless sensor networks for patient monitoring, localization and tracking,” *Ad Hoc Networks*, vol. 11, no. 1, pp. 39–53, 2013.

[37] E. Lam and J. Goodman, “A mathematical analysis of the DCT coefficient distributions for images,” *IEEE Trans. on Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[38] B. Foo, Y. Andreopoulos, and M. van der Schaar, “Analytical rate-distortion-complexity modeling of wavelet-based video coders,” *IEEE Trans. on Signal Process.*, vol. 56, no. 2, pp. 797–815, Feb. 2008.

[39] M. Ryan, *AWS System Administration: Best Practices for Sysadmins in the Amazon Cloud*. O'Reilly Media, Inc., 2015.

[40] A. Li, X. Yang, S. Kandula, and M. Zhang, “Cloudcmp: comparing public cloud providers,” in *Proc. 10th ACM SIGCOMM Conf. on Internet Meas.* ACM, 2010, pp. 1–14.

[41] F. P. Kelly, A. K. Maulloo, and D. K. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, pp. 237–252, 1998.

[42] V. Paxson and S. Floyd, “Wide area traffic: the failure of Poisson modeling,” *IEEE/ACM Trans. on Networking*, vol. 3, no. 3, pp. 226–244, Mar. 1995.

[43] K. Park, G. Kim, and M. Crovella, “On the relationship between file sizes, transport protocols, and self-similar network traffic,” in *Proc. 1996 Int. Conf. on Network Prot.* IEEE, 1996, pp. 171–180.

[44] M. Dai, Y. Zhang, and D. Loguinov, “A unified traffic model for MPEG-4 and H.264 video traces,” *IEEE Trans. on Multimedia*, vol. 11, no. 5, pp. 1010–1023, May 2009.

[45] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey, and D. E. Knuth, “On the Lambert W function,” *Advances in Computational mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[46] Y. Andreopoulos and I. Patras, “Incremental refinement of image salient-point detection,” *IEEE Trans. on Image Process.*, vol. 17, no. 9, pp. 1685–1699, Sept. 2008.

[47] S. Winitzki, “Uniform approximations for transcendental functions,” in *Computational Science and Its Applications - ICCSA 2003*. Springer, 2003, pp. 780–789.

[48] A. Canclini, L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, “Comparison of two paradigms for image analysis in visual sensor networks,” in *Proc. 11th ACM Conf. Embedded Netw. Sens. Syst. (SENSYS)*. ACM, 2013, p. 62.

[49] D. Buranapanichkit and Y. Andreopoulos, “Distributed time-frequency division multiple access protocol for wireless sensor networks,” *IEEE Wireless Comm. Let.*, vol. 1, no. 5, pp. 440–443, 2012.

[50] C.-F. Shih, A. E. Xhafa, and J. Zhou, “Practical frequency hopping sequence design for interference avoidance in 802.15.4e TSCH networks,” in *Proc. IEEE Int. Conf. on Comm. (ICC)*. IEEE, 2015, pp. 6494–6499.

[51] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Gen. Comp. Syst. J.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[52] W. Gilks and P. Wild, “Adaptive rejection sampling for Gibbs sampling,” *Applied Statistics*, pp. 337–348, 1992.

[53] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.