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

Francesco Renna, Joseph Doyle, Yiannis Andreopoulos
Dept. of Electronic and Electrical Engineering
University College London (UCL)
London, UK
f.renna,j.doyle,i.andreopoulos@ucl.ac.uk

Vasileios Giotsas
Dithen
http://www.dithen.com
London, UK
v.giotsas@dithen.com

***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, 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) spot instances, with the AWS Auto Scaling being used to control the number of instances according to the demand.**

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]–[7]. Examples of early commercial services in this domain include Google Goggles, Google Glass object recognition, Facebook automatic face tagging [8], Microsoft's Photo Gallery face recognition, as well as technology described in recent publications from Google, Siemens and others[1].

[1]See "A Google Glass app knows what you're looking at" MIT Tech. Review (Sept. 30, 2013) and EU projects SecurePhone [9], [10] and MoBio [11], [12].

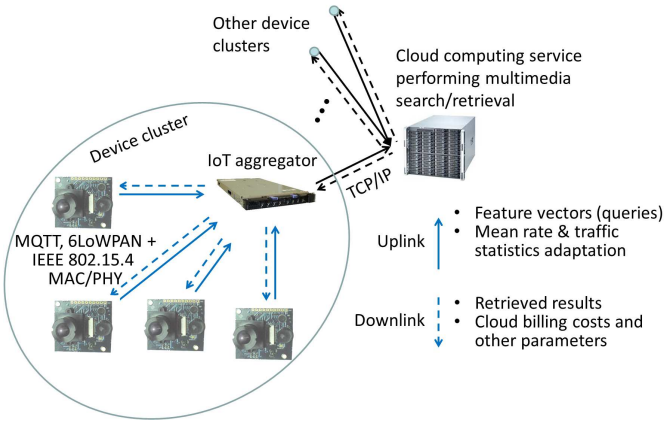

Figure 1. System hierarchy for a media search application within an IoT context. Low-power devices send query data to an IoT aggregator using low-power protocols for the application, network, medium access control and physical layers, such as MQTT, 6LowPAN, and IEEE 802.15.4 MAC/PHY. The IoT aggregator 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 [7], [13]–[15]. Such feature vectors can be seen as *queries* in a multimedia search application [6], [13]. For example, Serra *et. al.* [7] 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 compact vectors of 64~8192 elements [14], [15]. All such compacted feature vectors can be matched to equivalent vectors of very large content libraries within a cloud-computing service within the context of classification, retrieval and similarity identification for so-called "big data" applications. Because devices of the same type run the same application software for the query extraction and transmission, they incur, on average, the same

energy consumption per bit of each type of query. Therefore, they can be partitioned into "device clusters" that represent a multitude of identical devices (Fig. 1). 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 that provides for recognition and retrieval purposes [1]–[3], [16].

In this paper, we consider the energy consumption and billing costs incurred by such 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 settings, under the assumption of identical devices producing data traffic with the same statistical characterization. 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 on each 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.

A key advantage of our work in comparison to previous work on optimal energy management policies [16]–[19] and resource prediction and analysis [20]–[23] (see also [24] and references therein), is that we provide closed-form expressions for the minimum-required billing cost in order for each mobile device to remain within the predetermined energy consumption constraints. 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) spot instances. Our results illustrate how the proposed model can be applied to real-world IoT-oriented 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 three 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 mobile 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) or IBM IoT Foundation. This is where all device queries are uploaded (e.g., using an application-layer protocol like MQTT) in order 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 reshape the IoT query traffic volume before uploading it to the cloud-computing service and also 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 event-driven activation within an audio/visual surveillance application. We therefore assume that the query data production volume during T seconds is modeled as a random variable. 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

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

[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 the present paper.

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) $P(\psi_{\mathrm{e}})$. A model for the marginal statistics of this data volume can be derived 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. Examples of systems with variable query data production and transmission rates include visual sensor networks transmitting image features [25]–[28], as well as activity recognition networks where the data acquisition is irregular and depends on the events occurring in the monitored area [29]–[31].

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 $P(\psi_{\mathrm{e}})$ 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, for an aggregator of n devices:

$$P(\psi_{\mathrm{b}}) = \underbrace{P(\psi_{\mathrm{e}}) \star \ldots \star P(\psi_{\mathrm{e}})}_{n \text{ times}}, \tag{1}$$

where $\star$ denotes the convolution operator, 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 devices. Note that, as the number of devices n increases, the corresponding PDF $P(\psi_{\mathrm{b}})$ can be approximated with increasing precision via the Half-Gaussian distribution.[3] Since the query data production volume may be non-stationary, we assume its marginal statistics for $P(\psi_{\mathrm{e}})$ and $P(\psi_{\mathrm{b}})$, which are derived starting from a doubly stochastic model for these processes as explained in related work [32], [33].

2) Energy and Cloud Infrastructure Billing Parameters: We assume that 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. Since we are considering prolonged periods of operation in our analysis and the utilized sensors, transceivers and embedded processors consume energy in a stable manner when handing data, g_{e} can be calculated by averaging several "on" periods for sensing, processing and transmission for each device under consideration and normalizing to the amount of bits delivered to the IoT aggregator. For example, under a visual search application, g_{e} 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 [15], and the application and transceiver-incurred energy consumption to transmit this 8192-bit stream to the aggregator (e.g., using MQTT and the IEEE 802.15.4e MAC/PHY). 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.

On the other hand, 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 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 $c_{\mathrm{e}} E[\Psi_{\mathrm{e}}]$ bits, 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 [20], [23], as well as on the specifics of the application [34]–[36], 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.

In order to control the overall energy consumption profile of the application, the expected energy consumption within T seconds should be equal to E_{mean} Joule and the expected upper-sided deviation should not exceed E_{updev} Joule. 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 (e.g., sudden drop of battery life or device/battery overheating), or disrupt other concurrently-running services on the device.

Analogously, when servers are reserved from the cloud provider in order to process the queries uploaded by an IoT aggregator, this incurs billing costs. 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 [37] 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[4]:

[3]The analysis associated to Half-Gaussian distributed query volumes will be carried out in future works.

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

- 3 single-core AWS EC2 `m3.medium` spot instances when the average uploaded query volume is below a certain "quota" of c_{b} query bits ("idle" case),
- 30 spot instances when the query volume exceeds c_{b} bits ("active" case).

Based on current AWS EC2 pricing, each single-core `m3.medium` spot instance incurs infrastructure billing cost of 0.01$ per hour. 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): 8.3×10^{-8} dollars-per-query under the "idle" case, or $i_{\mathrm{b}} \cong 1.0 \times 10^{-11}$ dollars-per-query-bit ($/b); $p_{\mathrm{b}} \cong 1.0 \times 10^{-10}$ $/b for the "active" case. The quota of c_{b} query bits can be set according to the application or the number of devices, n, within each IoT aggregator.

Beyond the cost of the computing time, billing cost proportional to the expected query volume per monitoring interval, $E[\Psi_{\mathrm{b}}]$, must be accounted for, since all cloud providers charge for data transfers and storage. 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. Finally, 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.

Evidently, the large number of system, data production, energy consumption, and cloud billing parameters makes the exhaustive exploration of the complete design space infeasible. Therefore, although not all parameters describing the overall system are controlled by the same entity, 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 the utmost 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 upper-side deviate), as well as the expected cloud billing for a group of n devices on the same IoT aggregator. This allows us to derive closed-form conditions that ensure the value of E_{mean} Joule is met for each device, while also meeting the corresponding energy upper-side deviation of E_{updev} Joule. We also derive the 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.

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

$$E_{\mathrm{exp}} = E\left[\Psi_{\mathrm{e}}\right] g_{\mathrm{e}} + i_{\mathrm{e}} \int_{0}^{c_{\mathrm{e}} E[\Psi_{\mathrm{e}}]} \left(c_{\mathrm{e}} E\left[\Psi_{\mathrm{e}}\right] - \psi_{\mathrm{e}}\right) P_{\mathrm{e}}\left(\psi_{\mathrm{e}}\right) d\psi_{\mathrm{e}}, \tag{2}$$

where the integral of the second term expresses the expected energy consumption for the time that the device will be in idle mode. 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 $(c_{\mathrm{e}} E\left[\Psi_{\mathrm{e}}\right])$-bit query volume):

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

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, 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 utilized hardware. Therefore, in this paper we explore this tradeoff by imposing operational values for the mean energy

$$E_{\mathrm{exp}} = E_{\mathrm{mean}} \tag{4}$$

and the corresponding upper-side energy deviation

$$E_{\mathrm{var}} = E_{\mathrm{updev}}^2, \tag{5}$$

and we explore their impact on the system parameters and the query data production volume.

In a similar fashion, let us now consider the expected cloud billing cost when receiving n aggregated query volumes from n devices. We can express this cost via

$$\begin{aligned} B_{\mathrm{exp}} &= E\left[\Psi_{\mathrm{b}}\right] g_{\mathrm{b}} + i_{\mathrm{b}} \int_{0}^{c_{\mathrm{b}}} \left(c_{\mathrm{b}} - \psi_{\mathrm{b}}\right) P\left(\psi_{\mathrm{b}}\right) d\psi_{\mathrm{b}} \\ &+ p_{\mathrm{b}} \int_{c_{\mathrm{b}}}^{\infty} \left(\psi_{\mathrm{b}} - c_{\mathrm{b}}\right) P\left(\psi_{\mathrm{b}}\right) d\psi_{\mathrm{b}}, \end{aligned} \tag{6}$$

where: $E\left[\Psi_{\mathrm{b}}\right] g_{\mathrm{b}}$ 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 $p_{\mathrm{b}} \int_{0}^{c_{\mathrm{b}}} \left(\psi_{\mathrm{b}} - c_{\mathrm{b}}\right) P\left(\psi_{\mathrm{b}}\right) d\psi_{\mathrm{b}}$ in B_{exp}, we get:

$$\begin{aligned} B_{\mathrm{exp}} &= E\left[\Psi_{\mathrm{b}}\right] \left(g_{\mathrm{b}} + p_{\mathrm{b}}\right) - p_{\mathrm{b}} c_{\mathrm{b}} \\ &+ \left(i_{\mathrm{b}} + p_{\mathrm{b}}\right) \int_{0}^{c_{\mathrm{b}}} \left(c_{\mathrm{b}} - \psi_{\mathrm{b}}\right) P\left(\psi_{\mathrm{b}}\right) d\psi_{\mathrm{b}}. \end{aligned} \tag{7}$$

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, $P\left(\psi_{\mathrm{b}}\right)$, which is either of the same form as $P\left(\psi_{\mathrm{e}}\right)$, or it is linked to it via (1). In the remainder of this section:

- We consider different cases for $P(\psi_{\mathrm{e}})$ and $P(\psi_{\mathrm{b}})$ to derive the conditions to match the energy consumption expression of (2) to E_{mean} in (4) and allow parameter tuning that guarantees that (3) does not exceed the threshold E_{updev} in (5).
- We derive the number of query bits (quota), c_{b}, that minimizes the corresponding billing cost of (7) under various PDFs $P(\psi_{\mathrm{b}})$.
- In order for the desired energy consumption and billing cost parameters to be met concurrently, we associate the minimum billing cost with the desired value for the expected billing, B_{mean}, and the device query production volume, r, thereby establishing the number of devices, n, that should be admitted by each IoT aggregator.

A. Illustrative Case: $P(\psi_{\mathrm{e}})$ and $P(\psi_{\mathrm{b}})$ are Uniformly Distributed

When no knowledge of the underlying statistics of the query generation process exists, one can assume that both $P(\psi_{\mathrm{e}})$ and $P(\psi_{\mathrm{b}})$ are uniform over the intervals $[0, 2r]$ and $[0, 2rn]$, respectively:

$$P_{\mathrm{U}}(\psi_{\mathrm{e}}) = \begin{cases} \frac{1}{2r}, & 0 \le \psi_{\mathrm{e}} \le 2r \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

and

$$P_{\mathrm{U}}(\psi_{\mathrm{b}}) = \begin{cases} \frac{1}{2rn}, & 0 \le \psi_{\mathrm{b}} \le 2rn \\ 0, & \text{otherwise} \end{cases}. \tag{9}$$

This corresponds to the case where the IoT aggregator's upload query volume PDF matches the query generation PDF of (8) and the aggregator merges query volumes of n devices.

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

1) Energy Parameter Tuning to Meet the Settings of (4) *and* (5)*:* Starting from the device energy consumption, by using (8) in (2), we obtain:

$$E_{\mathrm{exp,U}} = \left(g_{\mathrm{e}} + \frac{i_{\mathrm{e}} c_{\mathrm{e}}^2}{4}\right) r \Leftrightarrow \; r = \frac{4E_{\mathrm{exp,U}}}{4g_{\mathrm{e}} + i_{\mathrm{e}} c_{\mathrm{e}}^2}. \tag{10}$$

In addition, by using (8) in (3), we obtain:

$$E_{\mathrm{var,U}} = g_{\mathrm{e}}^2 \frac{(2 - c_{\mathrm{e}})^3}{6} r^2, \tag{11}$$

and by substituting (10) in (11), we can express the one-side variability of the energy consumption between idle and active state as a function of the idle threshold c_{e} as

$$E_{\mathrm{var,U}} = \frac{8 g_{\mathrm{e}}^2 E_{\mathrm{exp,U}}^2 (2 - c_{\mathrm{e}})^3}{3(4g_{\mathrm{e}} + i_{\mathrm{e}} c_{\mathrm{e}}^2)^2}. \tag{12}$$

Therefore, by imposing the constraint (4) for $E_{\mathrm{exp,U}}$, we can derive the value of r that matches the expected energy consumption. Moreover, (12) offers a tool to efficiently tune c_{e} so that the setting of (5) for $E_{\mathrm{var,U}}$ is met. In this way, one can carry out a detailed exploration of the mean query production volumes and coupling data volumes per device that satisfy any *a-priori* energy settings for E_{mean} and E_{updev}, as well as any energy parameters g_{e} and i_{e}, within the monitoring time interval T.

Alternatively, from (10) we can derive the activation threshold c_{e} that guarantees the average energy consumption, for a given average query volume of r bits, as

$$c_{\mathrm{e}} = 2\sqrt{\frac{E_{\mathrm{exp,U}} - g_{\mathrm{e}} r}{i_{\mathrm{e}} r}}, \tag{13}$$

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

Based on (13), the one-side variability of energy consumption can be expressed as a function of the average query volume r:

$$E_{\mathrm{var,U}} = \frac{4}{3} g_{\mathrm{e}}^2 r^2 \left(1 - \sqrt{\frac{E_{\mathrm{exp,U}} - g_{\mathrm{e}} r}{i_{\mathrm{e}} r}}\right)^3. \tag{14}$$

Via (14), we can numerically determine the value of r for which the corresponding one-sided variability of the energy consumption agrees with the setting of (5).

2) 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 (7) for the n-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

$$\frac{dB_{\mathrm{exp}}}{dc_{\mathrm{b}}} = -p_{\mathrm{b}} + (i_{\mathrm{b}} + p_{\mathrm{b}}) F_{\mathrm{b}}(c_{\mathrm{b}}) \tag{15}$$

$$\frac{d^2 B_{\mathrm{exp}}}{dc_{\mathrm{b}}^2} = (i_{\mathrm{b}} + p_{\mathrm{b}}) P_{\mathrm{b}}(c_{\mathrm{b}}), \tag{16}$$

where $F_{\mathrm{b}}(\psi_{\mathrm{b}})$ and $P_{\mathrm{b}}(\psi_{\mathrm{b}})$ are the cumulative distribution function (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} obeys to a continuous distribution with given PDF and CDF. Moreover, the value of c_{b} that

minimizes the billing cost is obtained by solving the equation $\frac{dB_{\rm exp}}{dc_{\rm b}} = 0$, i.e.,

$$c_{\rm b} = F^{-1}\left(\frac{p_{\rm b}}{i_{\rm b}+p_{\rm b}}\right), \tag{17}$$

where $F^{-1}(\cdot)$ represents the inverse function of the CDF of $\Psi_{\rm b}$. Assuming any strictly-increasing CDF, $c_{\rm b}$ will be unique[5]. Therefore, in conjunction with the fact that $\forall c_{\rm b}: \frac{d^2 B_{\rm exp}}{dc_{\rm b}^2} > 0$, $B_{\rm exp}$ has a unique minimum in function of $c_{\rm b}$.

For the case of uniform distribution, by replacing (9) in (7), we obtain the average billing cost as

$$B_{\rm exp,U} = (g_{\rm b}+p_{\rm b})\, rn - p_{\rm b}c_{\rm b} + (i_{\rm b}+p_{\rm b})\,\frac{c_{\rm b}^2}{4rn}, \tag{18}$$

and the optimal coupling point is

$$c_{\rm b,U} = \frac{2p_{\rm b}rn}{i_{\rm b}+p_{\rm b}}. \tag{19}$$

The corresponding minimum-possible billing cost for $c_{\rm b} \in (0,\infty)$ is achieved under $c_{\rm b} = c_{\rm b,U}$, and it is:

$$\min\{B_{\rm exp,U}\} = \left(g_{\rm b}+p_{\rm b}-\frac{p_{\rm b}^2}{i_{\rm b}+p_{\rm b}}\right) rn. \tag{20}$$

The last equation shows that the minimum billing cost increases linearly to the average query data production volume of all n devices. If the minimum value is set to any *a-priori* determined expected billing, i.e., $\min\{B_{\rm exp,U}\} = B_{\rm mean}$, the corresponding device query volume becomes:

$$r_{\rm b,U} = \frac{B_{\rm mean}}{\left(g_{\rm b}+p_{\rm b}-\frac{p_{\rm b}^2}{i_{\rm b}+p_{\rm b}}\right) n}. \tag{21}$$

3) Number of Devices in an IoT Aggregator to Concurrently Satisfy Energy Consumption and Billing Costs: In order to meet *both* energy and billing costs: $\{E_{\rm mean}, E_{\rm updev}\}$ and $B_{\rm mean}$, we can match the derived query volume of (21) with $r_{\rm e,U}$ derived from (10) and, by tuning $c_{\rm e}$ via (13) and setting $c_{\rm b,U}$ to the value given by (19), derive:

$$r_{\rm b,U} = r_{\rm e,U} \Leftrightarrow n_{\rm U} = \frac{B_{\rm mean}}{\left(g_{\rm b}+p_{\rm b}-\frac{p_{\rm b}^2}{i_{\rm b}+p_{\rm b}}\right) r_{\rm e,U}}. \tag{22}$$

The value of $n_{\rm U}$ of (22) comprises the numbers of devices that should be accommodated by an IoT aggregator that receives and transmits queries under the uniform distributions of (8) and (9) when each device meets the energy settings of (4) and (5) and the IoT-uploaded volume leads to minimum billing cost of $B_{\rm mean}$.

Overall, via the energy-constrained analysis that derived (10) and (12) and the cloud-billing optimization that derived (19)–(22), one can explore different energy and billing settings in order to accommodate: particular types of mobile devices (with given energy consumption parameters), given average query production volume, or given number of devices per IoT cluster of Fig. 1.

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

B. Energy-constrained Query Volume Production and Minimum Billing Cost under Pareto and Exponential Distributions

We can now extend the previous calculation to other distributions expressing commonly observed data transmission rates in practical applications. We consider two additional PDFs for Ψ 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. The proofs follow the same process as for the uniform distribution. For each distribution, we couple its parameters to the average query volume of the uniform distribution, r. 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 [38], [39]. Consider $P_{\rm P}(\psi_{\rm e})$ as the Pareto distribution with scale $v_{\rm e}$ and shape $\alpha_{\rm e} > 2$ ($\alpha_{\rm e} \in \mathbb{N}$),

$$P_{\rm P}(\psi_{\rm e}) = \begin{cases} \alpha_{\rm e}\frac{v_{\rm e}^{\alpha_{\rm e}}}{\psi_{\rm e}^{\alpha_{\rm e}+1}}, & \psi_{\rm e} \geq v_{\rm e} \\ 0, & \text{otherwise} \end{cases}. \tag{23}$$

The expected value of $\Psi_{\rm e}$ is $E_{\rm P}[\Psi_{\rm e}] = \frac{\alpha_{\rm e}v_{\rm e}}{\alpha_{\rm e}-1}$ bits. Thus, if we set

$$v_{\rm e} = \frac{\alpha_{\rm e}-1}{\alpha_{\rm e}} r, \tag{24}$$

we obtain $E_{\rm P}[\Psi_{\rm e}] = r$ bits, 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 1. *The average energy consumption for Pareto-distributed device query volumes is given by*

$$E_{\rm exp,P} = \left[g_{\rm e} + i_{\rm e}\left[(\alpha_{\rm e}-1)^{\alpha_{\rm e}-1} c_{\rm e}(\alpha_{\rm e}c_{\rm e})^{-\alpha_{\rm e}} + c_{\rm e} - 1\right]\right] r, \tag{25}$$

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

$$E_{\rm var,P} = 2g_{\rm e}^2 \frac{(\alpha_{\rm e}-1)^{\alpha_{\rm e}-1} c_{\rm e}^{2-\alpha_{\rm e}}}{\alpha_{\rm e}^{\alpha_{\rm e}}(\alpha_{\rm e}-2)} r^2. \tag{26}$$

Proof: See Appendix. ∎

Note that Proposition 1 assumes that $c_{\rm e} \geq \frac{\alpha_{\rm e}-1}{\alpha_{\rm e}}$, since, otherwise, the device will never switch from active to idle state. Moreover, from (25), we can derive the average query

volume corresponding to any given values for $E_{\mathrm{exp,P}}$ and c_{e} as

$$r = \frac{E_{\mathrm{exp,P}}}{g_{\mathrm{e}} + i_{\mathrm{e}}\left[(\alpha_{\mathrm{e}}-1)^{\alpha_{\mathrm{e}}-1}c_{\mathrm{e}}(\alpha_{\mathrm{e}}c_{\mathrm{e}})^{-\alpha_{\mathrm{e}}} + c_{\mathrm{e}} - 1\right]}, \quad (27)$$

and the one-sided energy variance associated to r as

$$E_{\mathrm{var,P}} = g_{\mathrm{e}}^2 \frac{(\alpha_{\mathrm{e}}-1)^{\alpha_{\mathrm{e}}-1}c_{\mathrm{e}}^{2-\alpha_{\mathrm{e}}}}{\alpha_{\mathrm{e}}^{\alpha_{\mathrm{e}}}(\alpha_{\mathrm{e}}-2)} \times \frac{E_{\mathrm{exp,P}}^2}{\left[g_{\mathrm{e}} + i_{\mathrm{e}}\left[(\alpha_{\mathrm{e}}-1)^{\alpha_{\mathrm{e}}-1}c_{\mathrm{e}}(\alpha_{\mathrm{e}}c_{\mathrm{e}})^{-\alpha_{\mathrm{e}}} + c_{\mathrm{e}} - 1\right]\right]^2}. \quad (28)$$

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 $E_{\mathrm{P}}[\Psi_{\mathrm{e}}] = r$, i.e., to *fixed-volume* query production per monitoring interval. Then, since $c_{\mathrm{e}} \geq \frac{\alpha_{\mathrm{e}}-1}{\alpha_{\mathrm{e}}}$, the average energy consumption converges to

$$E_{\mathrm{exp,P}} = \left[g_{\mathrm{e}} + i_{\mathrm{e}}(c_{\mathrm{e}} - 1)\right] r, \quad (29)$$

as $\alpha_{\mathrm{e}} \to \infty$, and the one-side energy variation from idle to active mode converges to zero (the device is in idle mode for a portion of the time of every monitoring interval). Then, the average query volume that meets the expected energy consumption constraint $E_{\mathrm{exp,P}}$ is simply given by

$$r = \frac{E_{\mathrm{exp,P}}}{g_{\mathrm{e}} + i_{\mathrm{e}}(c_{\mathrm{e}} - 1)}. \quad (30)$$

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 [40]. Consider $P_{\mathrm{E}}(\psi_{\mathrm{e}})$ as the Exponential distribution with rate parameter $\frac{1}{r}$

$$P_{\mathrm{E}}(\psi_{\mathrm{e}}) = \frac{1}{r}\exp\left(-\frac{1}{r}\psi_{\mathrm{e}}\right), \quad (31)$$

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

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

$$E_{\mathrm{exp,E}} = \left[g_{\mathrm{e}} + i_{\mathrm{e}}\left(c_{\mathrm{e}} + e^{-c_{\mathrm{e}}} - 1\right)\right] r, \quad (32)$$

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

$$E_{\mathrm{var,E}} = 2g_{\mathrm{e}}^2 \exp(-c_{\mathrm{e}}) r^2. \quad (33)$$

Proof: See Appendix. ∎

From (32), it is straightforward to derive the average query volume corresponding to any given values of $E_{\mathrm{exp,E}}$ and c_{e} as

$$r = \frac{E_{\mathrm{exp,E}}}{g_{\mathrm{e}} + i_{\mathrm{e}}\left[c_{\mathrm{e}} + \exp(-c_{\mathrm{e}}) - 1\right]}, \quad (34)$$

and the one-sided energy variation associated to r as

$$E_{\mathrm{var,E}} = \frac{2g_{\mathrm{e}}^2 \exp(-c_{\mathrm{e}}) E_{\mathrm{exp,E}}^2}{\left[g_{\mathrm{e}} + i_{\mathrm{e}}\left(c_{\mathrm{e}} + \exp(-c_{\mathrm{e}}) - 1\right)\right]^2}. \quad (35)$$

In addition, for any given values of $E_{\mathrm{exp,E}}$ and r, we can also derive the threshold c_{e} as

$$c_{\mathrm{e}} = W_0\left(-\exp\left(-\frac{E_{\mathrm{exp,E}} + i_{\mathrm{e}}r - g_{\mathrm{e}}r}{i_{\mathrm{e}}r}\right)\right) + \frac{E_{\mathrm{exp,E}} + i_{\mathrm{e}}r - g_{\mathrm{e}}r}{i_{\mathrm{e}}r}, \quad (36)$$

where $W_0(\cdot)$ is the main branch of the standard Lambert W function. The corresponding one-sided energy variability associated to c_{e} is given by

$$E_{\mathrm{var,E}} = -2g_{\mathrm{e}}^2 r^2 W_0\left(-\exp\left(-\frac{E_{\mathrm{exp,E}} + i_{\mathrm{e}}r - g_{\mathrm{e}}r}{i_{\mathrm{e}}r}\right)\right). \quad (37)$$

3) Billing Cost under Pareto and Exponential 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 $E_{\mathrm{P}}[\Psi_{\mathrm{b}}] = rn$, i.e.,

$$P_{\mathrm{P}}(\psi_{\mathrm{b}}) = \begin{cases} \alpha_{\mathrm{b}}\frac{v_{\mathrm{b}}^{\alpha_{\mathrm{b}}}}{\psi_{\mathrm{b}}^{\alpha_{\mathrm{b}}+1}}, & \psi_{\mathrm{b}} \geq v_{\mathrm{b}} \\ 0, & \text{otherwise} \end{cases}, \quad (38)$$

where $\alpha_{\mathrm{b}} > 2$ $(\alpha_{\mathrm{b}} \in \mathbb{N})$ and $v_{\mathrm{b}} = \frac{\alpha_{\mathrm{b}}-1}{\alpha_{\mathrm{b}}}rn$.

Proposition 3. *The average billing cost incurred from processing Pareto-distributed query volumes is given by*

$$B_{\mathrm{exp,P}} = (g_{\mathrm{b}} - i_{\mathrm{b}})rn + (i_{\mathrm{b}} + p_{\mathrm{b}})\frac{(\alpha_{\mathrm{b}}-1)^{\alpha_{\mathrm{b}}-1}}{\alpha_{\mathrm{b}}^{\alpha_{\mathrm{b}}}}(rn)^{\alpha_{\mathrm{b}}}c_{\mathrm{b}}^{1-\alpha_{\mathrm{b}}} + i_{\mathrm{b}}c_{\mathrm{b}}. \quad (39)$$

The minimum billing cost is obtained when

$$c_{\mathrm{b,P}} = \left(\frac{i_{\mathrm{b}} + p_{\mathrm{b}}}{i_{\mathrm{b}}}\right)^{\frac{1}{\alpha_{\mathrm{b}}}} \frac{\alpha_{\mathrm{b}} - 1}{\alpha_{\mathrm{b}}} rn, \quad (40)$$

and it is given by

$$\min\{B_{\mathrm{exp,P}}\} = \left[g_{\mathrm{b}} - i_{\mathrm{b}} + i_{\mathrm{b}}\left(\frac{i_{\mathrm{b}} + p_{\mathrm{b}}}{i_{\mathrm{b}}}\right)^{\frac{1}{\alpha_{\mathrm{b}}}}\right] rn. \quad (41)$$

Proof: The proof stems from the evaluation of the general solution expressed in (17) under the usage of the Pareto PDF. ∎

In order to ensure that the average billing cost is B_{mean} and average query volume per device is $r_{\mathrm{e,P}}$, the IoT aggregator must handle

$$n_{\mathrm{P}} = \frac{B_{\mathrm{mean}}}{r_{\mathrm{e,P}}\left[g_{\mathrm{b}} - i_{\mathrm{b}} + i_{\mathrm{b}}\left(\frac{i_{\mathrm{b}}+p_{\mathrm{b}}}{i_{\mathrm{b}}}\right)^{\frac{1}{\alpha_{\mathrm{b}}}}\right]} \quad (42)$$

devices. This is derived by setting $\min\{B_{\mathrm{exp,P}}\} = B_{\mathrm{mean}}$ in (41) and solving for n. 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 rn.

In this case, if $c_{\rm b} \geq rn$, the average billing cost is simply given by

$$B_{\rm exp,P} = (g_{\rm b} - i_{\rm b})rn + i_{\rm b}c_{\rm b}, \tag{43}$$

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

Finally, let us consider the aggregate query volume distribution modeled via an Exponential distribution with mean $E_{\rm E}[\Psi_{\rm b}] = rn$, i.e.,

$$P_{\rm E}(\psi_{\rm b}) = \frac{1}{rn}\exp\left(-\frac{1}{rn}\psi_{\rm b}\right), \tag{44}$$

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

Proposition 4. *The average billing cost incurred from processing Exponentially-distributed query volumes is given by*

$$B_{\rm exp,E} = (g_{\rm b} - i_{\rm b})rn + i_{\rm b}c_{\rm b} + (i_{\rm b} + p_{\rm b})nre^{-\frac{c_{\rm b}}{nr}}. \tag{45}$$

The minimum billing cost is obtained when

$$c_{\rm b,E} = rn\ln\frac{i_{\rm b} + p_{\rm b}}{i_{\rm b}}, \tag{46}$$

and it is given by

$$\min\{B_{\rm exp,E}\} = \left(g_{\rm b} + i_{\rm b}\ln\frac{i_{\rm b} + p_{\rm b}}{i_{\rm b}}\right)rn. \tag{47}$$

Proof: The proof stems from the evaluation of the general solution expressed in (17) under the usage of the Exponential PDF. ■

In order to ensure that the average billing cost is $B_{\rm mean}$ and average query volume per device is $r_{\rm e,P}$, the IoT aggregator must handle

$$n_{\rm E} = \frac{B_{\rm mean}}{r_{\rm e,E}\left(g_{\rm b} + i_{\rm b}\ln\frac{i_{\rm b}+p_{\rm b}}{i_{\rm b}}\right)} \tag{48}$$

devices. This is derived by setting $\min\{B_{\rm exp,E}\} = B_{\rm mean}$ in (47) and solving for n.

IV. Evaluation of the Analytic Results

To validate the proposed analytic modeling framework of Propositions 1–4, we performed a series of experiments based on a visual sensor network connected to an IoT aggregator, and eventually to an AWS S3 plus EC2 cluster of spot instances. The following subsections present the hardware and application specifications, as well as the achieved results. Beyond our specific experimental results, we ensure to retain our description as broad as possible in order to indicate ways to carry out similar experiments within other IoT-oriented platforms, such as IBM IoT Foundation and Bluemix, AWS IoT, Cisco OpenStack, etc.

A. System Specification

We utilized a visual sensor network composed of multiple BeagleBone Linux embedded platforms [41], [42]. 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 (320x240) resolution. Further, our deployment involved:

1) a portable computer acting as the IoT aggregator, i.e., collecting all bitstreams via a star topology with $n = 10$ nodes and the recently-proposed (and available as open source) TFDMA protocol [43] for contention-free MAC-layer coordination;
2) an AWS S3 bucket where the IoT aggregator uploads all queries via a TCP/IP connection using script code running on the AWS Command Line Interface;
3) One reserved AWS instance running as the control server and assigning query volumes from S3 to AWS EC2 spot instances that serve as compute units; via AWS Auto Scaling, within each monitoring instance of T seconds, the number of spot instances are set to:
   - 3 when the query volume is below $c_{\rm b}$ bits ("idle" case).
   - 30 when the query volume exceeds $c_{\rm b}$ bits ("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 $i_{\rm b} = 6.27 \times 10^{-11}$ \$/b and $p_{\rm b} = 6.27 \times 10^{-10}$ \$/b. Regarding query traffic upload and storage costs, the corresponding billing rate per query bit was found to be $g_{\rm b} = 2.09 \times 10^{-10}$ \$/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 [44] 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 [44]. Therefore, our evaluation is pertinent to such scenarios that may be deployed in the next few years within an IoT paradigm [45].

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. [15], 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 [15]. The VLAD descriptor (i.e., query) size was set to 256 coefficients of 32 bits 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 [41] (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: $g_{\mathrm{e}} = 1.78\times10^{-6}$ J/b and $i_{\mathrm{e}} = 6.10\times10^{-7}$ J/b. Moreover, under the utilized application, the Beaglebone can process up to 1 frame per second while being constantly active. Therefore, the maximum query rate is 1 query per second, i.e., 8192 b/s. By setting mean query rates such that $E[\Psi_{\mathrm{e}}] \leq 2048$ bits per second, this theoretically allows for "idle" energy consumption with $c_{\mathrm{e}} < 3$. In practice, we only utilized $c_{\mathrm{e}} \in (0, 2)$ for "idle" energy consumption (i.e., up to twice the number of frames captured and processed with no query generation), as higher values caused 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 1–2. 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 [46], 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 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. 2 and Fig. 3 for

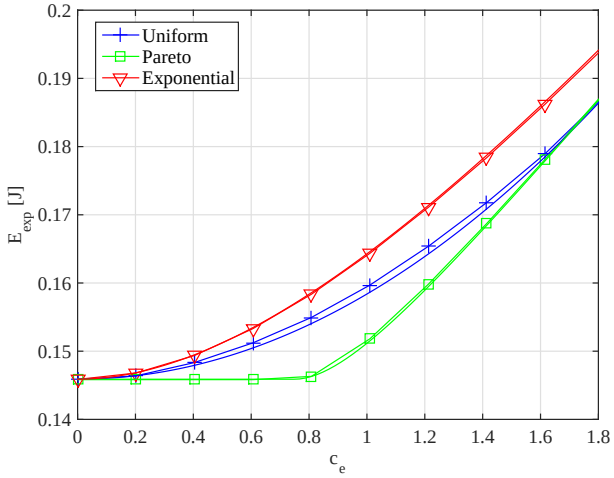

Figure 2. Average energy consumption E_{exp} vs. c_{e}. The average query volume was set to $r = 81,920$ b. For the case of Pareto distribution, we used $\alpha_{\mathrm{e}} = 4$. Lines with markers: Monte Carlo experiments; Lines without markers: theoretical predictions.

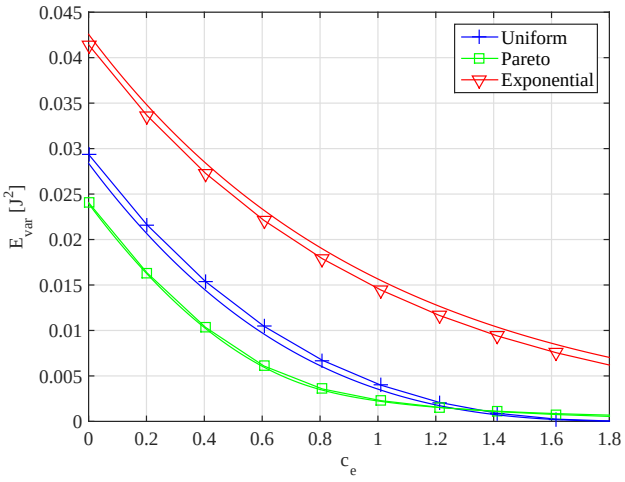

Figure 3. One-sided energy consumption E_{var} vs. c_{e}. The average query volume was set to $r = 81,920$ b. For the case of Pareto distribution, we used $\alpha_{\mathrm{e}} = 4$. Lines with markers: Monte Carlo experiments; Lines without markers: theoretical predictions.

$r = 81,920$ 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 0.998. 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 3 and 4 regarding the average billing cost. Specifically, we have

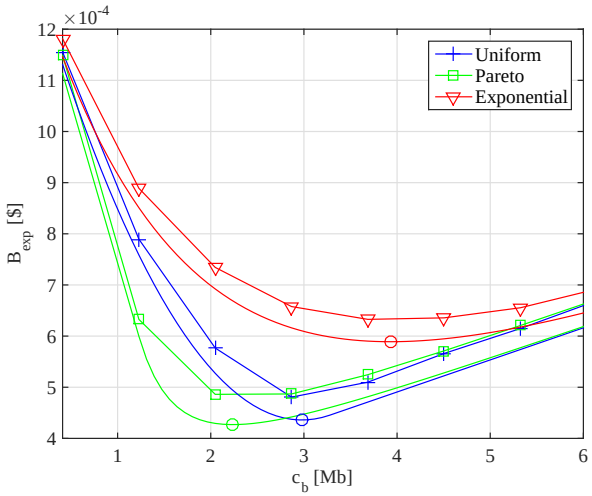

Figure 4. Average billing cost B_{exp} vs. c_{b}. The average query volume per device was set to $r = 163,840$ b and the experiment corresponds to $n = 10$ devices. For the case of Pareto distribution, we used $\alpha_{\text{e}} = 4$. Lines with markers: Monte Carlo experiments; Lines without markers: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

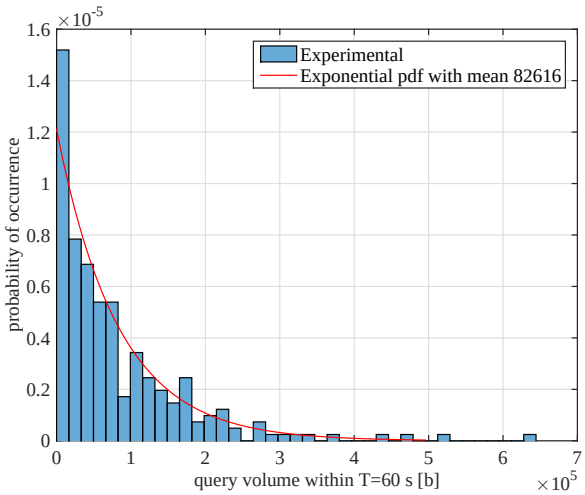

Figure 5. Probability histogram of query volume for $T = 60$ s and the best fit obtained via the Exponential distribution.

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

Fig. 4 presents indicative results under this setup. Evidently, the theoretical results follow the trends of the experimental data, with R^2 coefficients being above 0.9983 for all the distributions under consideration. However, the theoretical predictions always tend to 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. 4 have been obtained for a variety of average query volumes and monitoring intervals, but are omitted for brevity of exposition.

D. Results with User Generated Data

We now present system tuning results when repeating the visual query generation, transmission and cloud-based processing for 25 monitoring intervals based on real video captures and VLAD query generation using real data. The experiment was carried out within several offices of the Electronic and Electrical Engineering Department of University College London, and activation of query generation, transmission and processing was triggered when people passed (or moved) in front of the device cameras. Backend 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[6] the query production volumes to one of the distributions used in Section III. In the performed experiment and under monitoring interval of $T = 60$ s for the devices, we found that the query volume histogram agreed best with the Exponential distribution with $r = 82,616$ 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. 5. Moreover, with respect to c_{e}, we found that, for all cases of monitoring intervals under consideration, the system switched between "idle" and "active" states at $c_{\text{e}} \cong 0.5$. Therefore, our analytic results utilized this value for all results of this subsection.

Under this setup and with the fitted values for Exponential and Pareto PDFs, Table I presents the obtained experimental and theoretical values (via Proposions 1 and 2) for the expected energy and the upper-sided energy variance for two monitoring intervals. It is observed that the theoretical predictions are always within 10% of the experimentally-derived values. As such, the proposed energy-consumption model can be used for early-stage testing of plausible application deployments with respect to their energy efficiency

[6]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.

Table I
EXPECTED ENERGY CONSUMPTION AND UPPER-SIDED VARIATION. EXPERIMENTAL RESULTS AND THEORETICAL PREDICTION. FOR ALL CASES, WE SET $c_{\mathrm{e}} = 0.5$.

	Theoretical	Experimental
$T = 60$ s	$E_{\mathrm{exp}} = 0.1679$ J $E_{\mathrm{var}} = 0.0316$ J^2	$E_{\mathrm{exp}} = 0.1538$ J $E_{\mathrm{var}} = 0.0317$ J^2
$T = 1200$ s	$E_{\mathrm{exp}} = 2.7955$ J $E_{\mathrm{var}} = 2.9276$ J^2	$E_{\mathrm{exp}} = 2.8053$ J $E_{\mathrm{var}} = 2.8411$ J^2

Table II
EXPECTED BILLING COST. THE AD-HOC SOLUTION CORRESPONDS TO SETTING $c_{\mathrm{b}} = rn$. THE PROPOSED SOLUTION IS OBTAINED WITH c_{b} SET ACCORDING TO PROPOSITION 3.

	Ad-hoc	Proposition 3	Saving
$T = 600$ s $n = 10$	$B_{\mathrm{exp}} = 5.82 \cdot 10^{-4}$ \$ $c_{\mathrm{b}} = 1.54$ Mb	$B_{\mathrm{exp}} = 4.70 \cdot 10^{-4}$ \$ $c_{\mathrm{b}} = 2.51$ Mb	19 %
$T = 1200$ s $n = 10$	$B_{\mathrm{exp}} = 7.70 \cdot 10^{-4}$ \$ $c_{\mathrm{b}} = 1.88$ Mb	$B_{\mathrm{exp}} = 6.25 \cdot 10^{-4}$ \$ $c_{\mathrm{b}} = 3.09$ Mb	19 %

in order to determine the impact of various options, prior to more detailed experimentation in the field.

To present a further example of this capability, with the Pareto-distributed query volume statistics for $T = \{600, 1200\}$ s and under the use of $n = 10$ devices, we determined the Auto Scaling threshold, c_{b}, that is expected to lead to the minimum cloud infrastructure billing cost based on Proposition 3. Then, we benchmarked the obtained cost of the system under this threshold against the intuitive (albeit *ad-hoc*) setting of $c_{\mathrm{b}} = nr$, which corresponds to the Auto Scaling threshold being set to match the average query volume of all n devices. The results, given in Table II, show that the obtained billing cost is 19% lower than the case of the same query volume processing under the ad-hoc Auto Scaling threshold. In terms of practical deployments, it is important to emphasize again that not all the system parameters can be tuned by the same entity. For example, c_{b} is controlled by the cloud provider, whereas c_{e} depends on the specific device and the processing task performed. However, the proposed framework provides an analytic link between such parameters and the energy and billing costs, that can be used by the different stakeholders in a variety of ways. Moreover, the experimental example reported demonstrates that tuning the system based on the theoretical analysis can lead to important cost savings under real-world conditions for cloud-based processing of IoT-generated queries.

V. CONCLUSIONS

We propose a novel theoretical framework for establishing trade-offs in the energy consumption and infrastructure billing cost of Internet-of-Things (IoT) oriented deployments comprising mobile devices generating queries that are processed by a back-end cloud computing service. Our analysis incorporates energy consumption and cloud infrastructure billing 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 Beaglebone Linux embedded platforms and Amazon Web Services (AWS) 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. Our framework could be expanded in future work by: *(i)* expanding our analytic results beyond the specific cases of distributions used to characterize the query data volumes; *(ii)* considering the case of simple aggregation of the IoT devices' traffic by the IoT aggregator (Fig. 1); *(iii)* extending the experimental validation to different testbeds and applications, e.g., within IBM IoT Foundation and Bluemix, AWS IoT, Cisco OpenStack, etc.

ACKNOWLEDGEMENTS

FR, JD, and YA were supported by EPSRC, grants EP/K033166/1, EP/M00113X/1. VG was supported by Innovate UK, project ACAME, grant no. 131983.

APPENDIX

A. Energy Consumption: $P(\psi_{\mathrm{e}})$ Is Pareto Distributed

In this case, Ψ_{e} is drawn from a Pareto distribution with scale v_{e} and shape α_{e}, with $\alpha_{\mathrm{e}} > 2$, as in (23). Note that the expected value of Ψ_{e} is given by $E_{\mathrm{P}}[\Psi_{\mathrm{e}}] = \frac{\alpha_{\mathrm{e}} v_{\mathrm{e}}}{\alpha_{\mathrm{e}-1}}$. Therefore, we set $v_{\mathrm{e}} = \frac{\alpha_{\mathrm{e}}-1}{\alpha_{\mathrm{e}}} r$ in order to be consistent with the analysis carried out for the case of uniformly distributed Ψ_{e}.

If $c_{\mathrm{e}} r \leq v_{\mathrm{e}}$, the device is never in idle state, and the corresponding expected energy consumption is simply $E_{\mathrm{exp,P}} = g_{\mathrm{e}} r$. Under $c_{\mathrm{e}} r > v_{\mathrm{e}}$, and by using (23) in (2), we obtain

$$E_{\mathrm{exp,P}} = \left(g_{\mathrm{e}} + i_{\mathrm{e}} \left((\alpha_{\mathrm{e}} - 1)^{\alpha_{\mathrm{e}}-1} c_{\mathrm{e}} (\alpha_{\mathrm{e}} c_{\mathrm{e}})^{-\alpha_{\mathrm{e}}} + c_{\mathrm{e}} - 1\right)\right) r. \quad (49)$$

Thus, the value of the average query volume that meets the expected energy consumption constraint is

$$r = \frac{E_{\mathrm{exp,P}}}{g_{\mathrm{e}} + i_{\mathrm{e}} \left((\alpha_{\mathrm{e}} - 1)^{\alpha_{\mathrm{e}}-1} c_{\mathrm{e}} (\alpha_{\mathrm{e}} c_{\mathrm{e}})^{-\alpha_{\mathrm{e}}} + c_{\mathrm{e}} - 1\right)}. \quad (50)$$

Similarly, by using (23) in (3), and under $c_{\mathrm{e}} r > v_{\mathrm{e}}$, we can write the upper-sided variability of the energy consumption when the application switches from "idle" to "active" state as

$$E_{\mathrm{var,P}} = 2 g_{\mathrm{e}}^2 \frac{(\alpha_{\mathrm{e}} - 1)^{\alpha_{\mathrm{e}}-1} c_{\mathrm{e}}^{2-\alpha_{\mathrm{e}}}}{\alpha_{\mathrm{e}}^{\alpha_{\mathrm{e}}} (\alpha_{\mathrm{e}} - 2)} r^2. \quad (51)$$

Then, by substituting (50) into (51), we can express the one-side variability of the energy consumption as

$$E_{\mathrm{var,P}} = g_{\mathrm{e}}^2 \frac{(\alpha_{\mathrm{e}}-1)^{\alpha_{\mathrm{e}}-1} c_{\mathrm{e}}^{2-\alpha_{\mathrm{e}}}}{\alpha_{\mathrm{e}}^{\alpha_{\mathrm{e}}}(\alpha_{\mathrm{e}}-2)} \cdot \frac{E_{\mathrm{exp,P}}^2}{\left(g_{\mathrm{e}} + i_{\mathrm{e}}\left((\alpha_{\mathrm{e}}-1)^{\alpha_{\mathrm{e}}-1} c_{\mathrm{e}} (\alpha_{\mathrm{e}} c_{\mathrm{e}})^{-\alpha_{\mathrm{e}}} + c_{\mathrm{e}} - 1\right)\right)^2}. \quad (52)$$

We note that $\alpha_{\mathrm{e}} > 2$ is a necessary and sufficient condition for the upper-sided energy variability to be finite.

B. Energy Consumption: $P(\psi_{\mathrm{e}})$ Is Exponentially Distributed

Consider now the case where Ψ_{e} is exponentially distributed with rate parameter $\frac{1}{r}$ as in (31), with the expected value of Ψ_{e} set to $E_{\mathrm{E}}[\Psi_{\mathrm{e}}] = r$.

The expected energy consumption at each device can be computed by substituting (31) in (2), thus obtaining

$$E_{\mathrm{exp,E}} = \left(g_{\mathrm{e}} + i_{\mathrm{e}}\left(c_{\mathrm{e}} + e^{-c_{\mathrm{e}}} - 1\right)\right) r. \quad (53)$$

Moreover, the one-side variability for the energy consumption when the application switches from idle to active state is obtained by substituting (31) in (3), leading to

$$E_{\mathrm{var,E}} = 2 g_{\mathrm{e}}^2 e^{-c_{\mathrm{e}}} r^2. \quad (54)$$

From (53), we can derive the average query volume that meets the average energy consumption constraint in function of the activation rate c_{e}:

$$r = \frac{E_{\mathrm{exp,E}}}{g_{\mathrm{e}} + i_{\mathrm{e}}\left(c_{\mathrm{e}} + e^{-c_{\mathrm{e}}} - 1\right)}. \quad (55)$$

Then, by substituting (55) in (54), we obtain the expression of the upper-sided energy variability associated to a single device as a function of the activation rate c_{e}

$$E_{\mathrm{var,E}} = \frac{2 g_{\mathrm{e}}^2 e^{-c_{\mathrm{e}}} E_{\mathrm{exp,E}}^2}{\left(g_{\mathrm{e}} + i_{\mathrm{e}}\left(c_{\mathrm{e}} + e^{-c_{\mathrm{e}}} - 1\right)\right)^2}. \quad (56)$$

Provided that $E_{\mathrm{exp,E}} > g_{\mathrm{e}} r$, we can also determine the activation rate c_{e} that guarantees a given average energy consumption constraint for any average query volume. This is achieved by solving equation (53) for c_{e}, thus obtaining

$$c_{\mathrm{e}} = W_0\left(-\exp\left(-\frac{E_{\mathrm{exp,E}} + i_{\mathrm{e}} r - g_{\mathrm{e}} r}{i_{\mathrm{e}} r}\right)\right) + \frac{E_{\mathrm{exp,E}} + i_{\mathrm{e}} r - g_{\mathrm{e}} r}{i_{\mathrm{e}} r}, \quad (57)$$

where $W_0(\cdot)$ is the main branch of the standard Lambert W function [47]. Finally, the corresponding one-side energy variability is given by

$$E_{\mathrm{var,E}} = -2 g_{\mathrm{e}}^2 r^2 W_0\left(-\exp\left(-\frac{E_{\mathrm{exp,E}} + i_{\mathrm{e}} r - g_{\mathrm{e}} r}{i_{\mathrm{e}} r}\right)\right). \quad (58)$$

REFERENCES

[1] W. Zhu *et al.*, "Multimedia cloud computing," *IEEE Signal Proces. Mag.*, vol. 28, no. 3, pp. 59–69, 2011.

[2] X. Ma *et al.*, "When mobile terminals meet the cloud: computation offloading as the bridge," *IEEE Network*, vol. 27, no. 5, pp. 28–33, 2013.

[3] 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.

[4] 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.

[5] T. Soyata *et al.*, "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.

[6] B. Girod *et al.*, "Mobile visual search," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 61–76, 2011.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] N. Poh *et al.*, "An evaluation of video-to-video face verification," *IEEE Trans. Inf. Forens. and Sec.*, vol. 5, no. 4, pp. 781–801, 2010.

[12] 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.

[13] 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.

[14] F. Perronnin *et al.*, "Large-scale image retrieval with compressed Fisher vectors," in *IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2010, pp. 3384–3391.

[15] H. Jégou *et al.*, "Aggregating local image descriptors into compact codes," *IEEE Trans. Patt. Anal. and Machine Intel.*, vol. 34, no. 9, pp. 1704–1716, 2012.

[16] 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.

[17] 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.

[18] B. Zhang, R. Simon, and H. Aydin, "Energy management for time-critical energy harvesting wireless sensor networks," in *Lecture Notes in Comp. Sc.: Stabil., Safety, and Secur. of Distr. Syst.*, vol. 6366, 2010, pp. 236–251.

[19] C. Alippi *et al.*, "Energy management in wireless sensor networks with energy-hungry sensors," *IEEE Instr. & Meas. Mag.*, vol. 12, no. 2, pp. 16–23, 2009.

[20] N. Kontorinis *et al.*, "Statistical framework for video decoding complexity modeling and prediction," *IEEE Trans. on Circ. and Syst. for Video Technol.*, vol. 19, no. 7, 2009.

[21] Y. Andreopoulos *et al.*, "A local wavelet transform implementation versus an optimal row-column algorithm for the 2d multilevel decomposition," in *Proc. IEEE Int. Conf. Image Processing, 2001, ICIP*, vol. 3, 2001, pp. 330–333.

[22] ——, "High-level cache modeling for 2-d discrete wavelet transform implementations," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 34, no. 3, pp. 209–226, 2003.

[23] Y. Andreopoulos and M. Van der Schaar, "Adaptive linear prediction for resource estimation of video decoding," *IEEE Trans. on Circ. and Syst. for Video Technol.*, vol. 17, no. 6, pp. 751–764, 2007.

[24] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surv. Tut.*, vol. 16, no. 1, pp. 337–368, 2014.

[25] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "SensEye: a multi-tier camera sensor network," in *ACM international conference on Multimedia*. ACM, 2005, pp. 229–238.

[26] A. Rowe *et al.*, "FireFly Mosaic: A vision-enabled wireless sensor networking system," in *IEEE International Real-Time Systems Symposium (RTSS)*. IEEE, 2007, pp. 459–468.

[27] 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.

[28] A. Redondi *et al.*, "Low bitrate coding schemes for local image descriptors," in *MMSP*, 2012, pp. 124–129.

[29] G. Werner-Allen, 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.

[30] D. Palma *et al.*, "Distributed monitoring systems for agriculture based on wireless sensor network technology," *Int. Journal on Advances in Networks and Services,*, vol. 3, 2010.

[31] A. Redondi *et al.*, "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.

[32] 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.

[33] B. Foo *et al.*, "Analytical rate-distortion-complexity modeling of wavelet-based video coders," *IEEE Trans. on Signal Process.*, vol. 56, no. 2, pp. 797–815, Feb. 2008.

[34] I. Andreopoulos *et al.*, "A hybrid image compression algorithm based on fractal coding and wavelet transform," in *Proc. IEEE Int. Symp. Circuits and Systems, 2000 (ISCAS 2000).*, vol. 3. IEEE, 2000, pp. 37–40.

[35] Y. Andreopoulos *et al.*, "A new method for complete-to-overcomplete discrete wavelet transforms," in *Int. Conf. Dig. Signal Process., 2002, (DSP 2002).*, vol. 2. IEEE, 2002, pp. 501–504.

[36] A. Munteanu *et al.*, "Control of the distortion variation in video coding systems based on motion compensated temporal filtering," in *Proc. IEEE Int. Conf. Image Process., ICIP 2003*, vol. 2. IEEE, 2003, pp. II–61.

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

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] A. Canclini *et al.*, "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.

[43] 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.

[44] 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.

[45] J. Gubbi *et al.*, "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.

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

[47] 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.