End-to-end energy models for Edge Cloud-based IoT platforms: Application to data stream analysis in IoT

Yunbo Li [a,b], Anne-Cécile Orgerie [a,*], Ivan Rodero [c], Betsegaw Lemma Amersho [a,d], Manish Parashar [c], Jean-Marc Menaud [b]

[a] *Univ Rennes, Inria, CNRS, IRISA, France*
[b] *IMT Atlantique, LS2N, Nantes, France*
[c] *Rutgers Discovery Informatics Institute (RDI²), Rutgers University, Piscataway, NJ, USA*
[d] *Aalto University, Finland*

HIGHLIGHTS

- Estimates the energy consumption of IoT applications.
- End-to-end energy cost model for data stream analysis in IoT.
- Comparison of edge and core cloud solutions for IoT.

ARTICLE INFO

ABSTRACT

Internet of Things (IoT) is bringing an increasing number of connected devices that have a direct impact on the growth of data and energy-hungry services. These services are relying on Cloud infrastructures for storage and computing capabilities, transforming their architecture into more a distributed one based on edge facilities provided by Internet Service Providers (ISP). Yet, between the IoT device, communication network and Cloud infrastructure, it is unclear which part is the largest in terms of energy consumption. In this paper, we provide end-to-end energy models for Edge Cloud-based IoT platforms. These models are applied to a concrete scenario: data stream analysis produced by cameras embedded on vehicles. The validation combines measurements on real test-beds running the targeted application and simulations on well-known simulators for studying the scaling-up with an increasing number of IoT devices. Our results show that, for our scenario, the edge Cloud part embedding the computing resources consumes 3 times more than the IoT part comprising the IoT devices and the wireless access point.

1. Introduction

In 2011, Ericsson and Cisco started to announce that we will reach 50 billion devices connected to the Internet by 2020 [1,2]. Indeed, connected devices progressively invade our everyday lives with ever-widening application fields: personal health equipment, intelligent buildings, smart grids, connected vehicles, etc. The count in 2016 was under 20 billion of devices, including Internet-of-Things (IoT) devices, smartphones, tablets and computers [3]. Current forecasts estimate approximately 30 billion devices by 2020 [3].

All these objects, linked to telecommunication networks (most commonly the Internet), can interact with other connected devices or with distributed computing infrastructures, such as Clouds, for instance, to store information or perform computations. The growth in the number of connected objects and supporting infrastructures poses scientific challenges, notably in terms of managing the scaling, the heterogeneity of the communications networks used (Ethernet, WiFi, 3G, etc.), the migration of computations between objects and supporting infrastructures, and their energy consumption.

The development of IoT (Internet of Things) equipment, the popularization of mobile devices, and emerging wearable devices bring new opportunities for context-aware applications in Cloud computing environments [4]. Since 2008, the U.S. National Intelligence Council lists the IoT among the six technologies that are most likely to impact U.S. national power by 2025 [5]. The disruptive potential impact of IoT relies on its pervasiveness: it should constitute an integrated heterogeneous system connecting an unprecedented number of physical objects to the Internet [4]. A

* Corresponding author.
E-mail address: anne-cecile.orgerie@irisa.fr (A.-C. Orgerie).

basic example of such objects includes vehicles and their numerous sensors.

Among the many challenges raised by IoT, one is currently getting particular attention: making computing resources easily accessible from the connected objects to process the huge amount of data streaming out of them. Cloud computing has been historically used to enable a wide number of applications. It can naturally offer distributed sensory data collection, global resource and data sharing, remote and real-time data access, elastic resource provisioning and scaling, and pay-as-you-go pricing models [6]. However, it requires the extension of the classical centralized Cloud computing architecture towards a more distributed architecture that includes computing and storage nodes installed close to users and physical systems [7]. Such an edge Cloud architecture needs to deal with flexibility, scalability and data privacy issues to allow for efficient computational offloading services [8].

While computation offloading to the edge can be beneficial from a Quality of Service (QoS) point of view, from an energy perspective, it is relying on less energy-efficient resources than centralized Cloud data centers [9]. On the other hand, with the increasing number of applications moving on to the Cloud, it may become untenable to meet the increasing energy demand which is already reaching worrying levels [10]. Edge nodes could help to alleviate slightly this energy consumption as they could offload data centers from their overwhelming power load [9] and reduce data movement and network traffic. In particular, as edge Cloud infrastructures are smaller in size than centralized data center, they can make a better use of renewable energy [11].

On the other side, as IoT involves billions of connected devices mainly communicating through wireless networks, their power consumption is a major concern and limitation for the widespread of IoT [12]. An IoT device does not consume a lot of power by itself, typically from few milliWatts to few Watts [13,14]. Yet, the increasing number of devices produces a scale effect and causes also a non negligible impact on Cloud infrastructures that provide the computing power required by IoT devices to offer services [15]. To cope with the traffic increase caused by IoT devices, Cloud computing infrastructures start to explore the newly proposed distributed architectures, and in particular edge Cloud architectures where small data centers are located at the edge of the Cloud, typically in Internet Service Providers' (ISP) edge infrastructures [16,17].

While the current state of the art offers numerous studies on energy models for IoT devices [18,19] and Cloud infrastructures [20,21], to the best of our knowledge, none of them provides the overall picture. It is thus hard to estimate the energy consumption induced by the increase of IoT devices on Cloud infrastructures for instance. The issue resides in having an end-to-end energy estimation of all the involved devices and infrastructures, including network devices from ISP and Cloud servers. Such results could also serve to identify which part consumes the most, and should then focus the energy-efficient efforts.

In this paper, we propose to investigate the end-to-end energy consumption of IoT platforms. Our aim is to evaluate, on a concrete use-case, the benefits of edge computing platforms for IoT regarding energy consumption. We propose end-to-end energy models for estimating the consumption when offloading computation from the objects to the edge or to the core Cloud, depending on the number of devices and the desired application QoS, in particular trading-off between performance (response time) and reliability (service accuracy).

Our validation use-case targets the Internet of Vehicles (IoV) which can be seen as a convergence of the mobile internet and the IoT [22]. In particular, we focus on video streams from cameras that need to be analyzed usually for object detection and tracking. In this particular case, as it is often the case with IoT applications,

a high QoS level is required. Indeed, data lose their value when they cannot be analyzed fast enough. Through real measurements and simulations of this precise scenario, we evaluate the energy consumed by each part of the IoT platform: the connected devices, wired communication networks and Cloud computing infrastructures.

The paper is organized as follows. Section 2 presents the related work. Our validation scenario is described in Section 3. The end-to-end energy model is specified in Sections 4 and 5 presents the validation results. The validity and limits of our model are discussed in Section 6. Section 7 concludes this work.

2. Related work

2.1. Offloading data to the edge

Processing data streams analysis consumes enormous computational resources and the response time is usually crucial for many applications. Moving the data to the Cloud for analysis can be a solution [23] in a variety of application scenarios that require enormous computational resources as well as QoS guarantees. However, it might pose a risk of network bottleneck if thousands data streams are produced from IoT devices at the same time and then transmitted to a central Cloud (*core*) for quick analysis. Although lowering the analysis time profits large computational resources from Cloud, it cannot avoid the time for data transferring through the network from user to the physical location of Cloud, which might be thousands miles away [24]. Furthermore, the increasing number of data streams over the network consume a large amount of energy [24–27].

To meet the demand of low latency response times, computation offloading to edge can be an answer [28]. The edge represents small-scale data centers that are close to the data source. The concept of processing data at the edge is based on the advantage of lower latency than core, therefore been able to quickly return result to the device. Nevertheless, considering the large amount of data streams that needs to process, the core which has more computational resources may be a more energy-efficient choice.

Besides, a new model emerges: decentralized Cloud infrastructures [29]. Cloud providers expect to improve the performance of their Cloud and to leverage their available infrastructure. Indeed, telecommunication operators, like Orange, try to deploy micro data centers (20 to 50 servers by micro-DC) at the network border, closer to customers. In this new model, by deploying data centers closer to the user, the response time would greatly improve. This work focuses on a small-/medium-sized data centers as they continue to keep increasing their share of the market. Placing computing and storage nodes at the Internet's edge has grown more and more popular in the recent years. These nodes are often placed in a small data center which is near mobile devices. In particular, edge computing [30] enables to provide response time-critical Cloud services for users.

Existing literature has addressed video analysis algorithms and tools. Haar feature-based cascade classifiers [31] is a typical method for object detection which is effective and capable of achieving high detection rates. It is based on machine learning approach AdaBoost [32] and trains a cascade function from a large set of positive and negative images. The classifiers used in this paper are included in the OpenCV distribution[1] 2.4.13. We trained our own Haar classifier which is used to analyze video streams for objects detection.

---

[1] OpenCV is designed for computational efficiency and with a strong focus on real-time applications http://opencv.org.

2.2. Energy consumption of network and cloud devices

Since the appearance of Cloud computing, the demand for computing and storage resources in data centers has rapidly grown, leading to a consequent increase of their energy consumption. As an example, for 2010, Google used 900 000 servers and consumed 260 million watts of electricity [33]. Electricity becomes a key issue for deploying data center equipment.

Since the servers are among the primary energy consumers of data centers [34], many green proposals have addressed the problem of the server's energy-efficiency. Dynamic voltage and frequency scaling that exploits server performance knobs is an example of such proposals. Meanwhile, virtualization technology brings new opportunity for saving energy. It enables processing multiple requests on the same server, thus making it possible to run the workload on fewer servers by consolidation.

A study from 2014 presents the estimated energy consumption of data center market based on the number of installed servers and infrastructure electricity consumption [35]. Small- and medium-sized data centers account for nearly half the energy consumption of the market; they are typically composed of less than 100 servers with a light cooling system. The energy consumption of hyper-scale Cloud providers such as Google, Amazon and Facebook, only occupies 4 percent of the global data center energy consumption [35] due to their aggressive deployment of energy efficiency mechanisms that lower their power bill. With the adoption of edge Cloud computing, the part in the global Cloud's energy bill for small- and medium-sized data centers is expected to increase since edge Cloud infrastructures rely on highly distributed small data centers.

Nowadays, most of Cloud providers implement their commercial Clouds in large-scale data centers and operate them in a centralized fashion. Although they enable to achieve high performance computing ability and manageability, a powerful cooling system is needed to lower the temperature of this large infrastructure equipment. Yet, the cooling system is expensive and consumes huge amounts of energy. Instead, previous work [36] point out that small-size data centers have numerous advantages compared to large-scale data centers. First, small size data centers limit the amount of heat-dissipation and it can thus be more easier to manage. Then, smaller power consumption usually uses smaller power supplies and lower heat-dissipation overhead, which also reduces the cost and area of infrastructure equipment. Further, a small-scale data center is more suitable to build highly geographically distributed infrastructures.

Globally, the growth rate of data center energy consumption has slowed down in recent years [37]. Server shipments experienced a five-year rapid growth period with 15% annual growth rate from 2000 to 2005. From 2005–2010, the annual growth rate fell to 5% probably due to economic recession and also because the energy efficiency mechanisms started to be implemented in server, storage, network and infrastructure along with virtualization techniques. After 2010, the growth rate drops to 3% and is expected to stay stable by 2020. Currently, the most widely used and industry-acknowledged metric for assessing the energy efficiency of data centers is the PUE [38,39]: Power Usage Effectiveness. Usually, an ideal PUE value is equal to 1.0: this indicates that the energy consumed by IT equipment is same as the total facility energy.

Computer networks are the crucial elements that interconnect the data centers providing the computing and storage capacities of these distributed Clouds. As the size of Clouds increases and their traffic demands diversify, computer network resources, inside and in-between the data centers, are often stretched to their limits and, in many cases, become a performance bottleneck [40]. Besides, they represent a non negligible part of the energy consumption of these highly distributed systems [41,42].

Table 1

360p, 480p and 720p represent 3 different resolutions of the same video.

	Resolution	Bit rate
360p	640×360	514 kb/s
480p	720×480	706 kb/s
720p	1280×720	1176 kb/s

3. Driving use case

Quite a few IoT applications are being developed and deployed in various industries including environmental monitoring, healthcare service, inventory and production management, food supply chain, transportation, workplace and home support, security, and surveillance [12]. In this paper, we study one particular use-case. We first describe the application scenario and then explore the possible Cloud architecture for supporting these applications.

3.1. Application characteristics

The scenario that is explored in this paper presents heavy traffic needed to be processed. Multiple devices are sending continuous flows of data to a collection point that sends them to the Cloud. For instance, it represents a camera-based monitoring service like a road traffic analyzer [17]. Continuous data flows are sent from the camera to the Cloud hosting the IoT service. The generated data can represent huge volumes that have to be processed in real time [43]. Augmented reality applications exhibits similar traffic characteristics [44].

The motivation of this work is to provide a framework that can balance performance and energy cost tradeoffs for real-time data analysis of high-rate data from many devices. A typical use-case scenario consists in cameras that can be embedded in small devices such as Google Glass, GigaSight [45] or any other devices. The camera captures frames continuously that can be seen as a high-rate data stream. Since such a video analysis, that detects interesting objects (i.e., areas of interest) from it, consumes computing power, it thus requires energy. To increase the computation performance and to reduce energy consumption on the end-device, data is often offloaded to the Cloud to be analyzed. Although data offloading to high performance servers at the Cloud can accelerate the analysis processing, the efficiency of the whole procedure is highly dependent to the network condition and to the costs associated to the network service.

We considered that the videos can be encoded through H.264 codec in 3 resolutions (360p, 480p and 720p). More details are shown in Table 1 and we use the FFmpeg tool [46] for decoding.

3.2. Cloud infrastructure for data stream analysis

The edge typically has less computing capacity (e.g., compute servers) than the resources available in the Cloud core. However, these edge servers are closer to the edge-users and therefore, for users, the latency to edge servers is lower than the latency to the core. The core Cloud represents the federation of large data centers where each data center is composed of thousands of servers. Such a model of data centers [47] with federation of resources and autonomic management mechanisms offers a large pool of computing resources. While the core has more powerful servers the energy costs associated to data movement present different tradeoffs that need to be investigated.

In this paper, we make the assumption that all the considered vehicles are equipped with an on-board camera and are capable of uploading the video captured by their cameras continuously to edge and core Clouds. The edge/core analyzes each data stream in real time and returns the road condition to the user. The application goal is preventing traffic jam and possible traffic accidents by
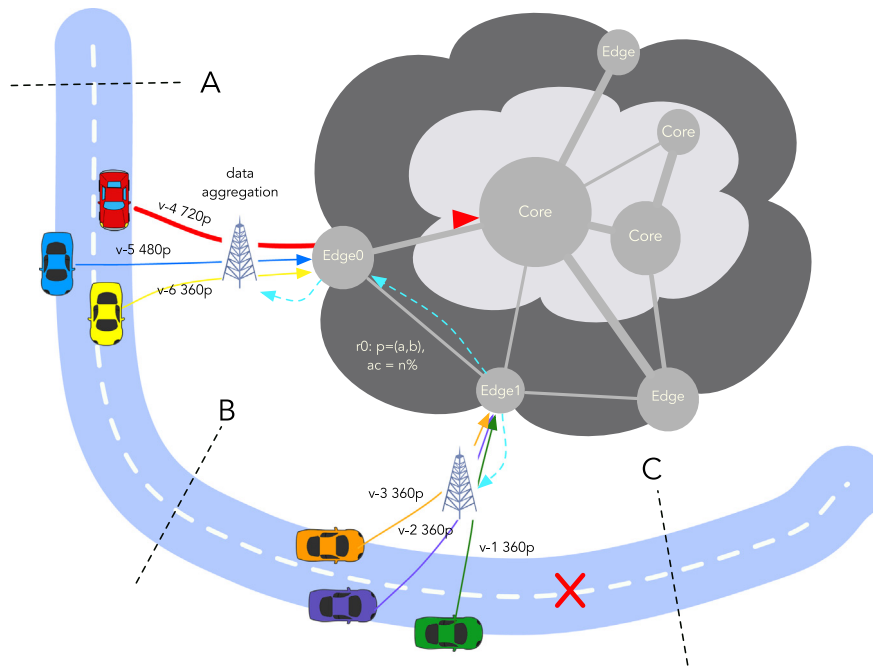
Fig. 1. Use case for IoV with edge and core Clouds.

sharing the produced information to users in an online manner. Integrating this into next generation of vehicles with autopilot technology can help improving the road safety for the drivers (i.e., the users).

Fig. 1 depicts our use-case: an object is detected by analyzing the data stream from car cameras, the resulting analysis identifies an object in the middle of the road which may be dangerous for the other vehicles behind on this road. The edge-1 immediately informs all the vehicles that are in section BC of the road. At the same time, a message is sent from edge-1 to the edge-0 in order to inform the vehicles in section AB of the road.

4. System model and assumptions

In this section, we describe our end-to-end models for the IoT scenarios and Cloud architectures described above. The architecture of an IoT service is composed of several elements: the IoT devices themselves, the collecting point gathering the data from the IoT devices, the Cloud infrastructure used to process and to store the data and the network that link the collecting point and the Cloud. For the sake of clarity, we divide these components into three parts as depicted on Fig. 2:

- the IoT part comprising the IoT devices and the collecting point;
- the networking part comprising several switches and routers, their number depends on the Cloud architecture (centralized or edge);
- the Cloud part including the data center resources employed by the IoT service.

When the considered Cloud model is an edge Cloud, the networking part is minimal. This model can be adapted to support fog computing infrastructures where IoT devices can be part of the Cloud resources [7,20]. In the following, we characterize the energy model of each part.

4.1. IoT part

The IoT part comprises the IoT devices and the collecting point. We consider n IoT devices using WiFi (802.11) to communicate.
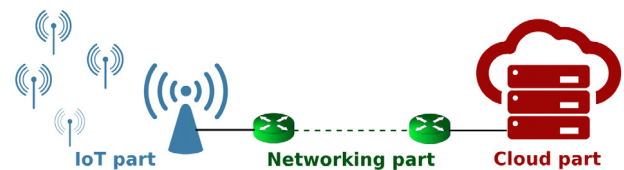

Fig. 2. Three main infrastructure parts of an IoT service deployment.

Table 2
IoT device values from [49].

Parameter	Value
Voltage	3.3 V
Idle current	0.273 A
CCA busy state current	0.273 A
Tx current	0.38 A
Rx current	0.313 A
Channel switching current	0.273 A
Sleep current	0.033 A

Following previous work, we use a state model for the energy consumption of their WiFi NIC [48]. The NIC is either in idle state (IDLE), performing carrier sense (CCA BUSY), receiving (RX), transmitting (TX) and or switching between states (SWITCHING) [14]. The corresponding energy-related values are exposed in Table 2. These values were measured for a 802.11n NIC [49]. The model can be adapted for other wireless communication types.

For the devices themselves, we consider WiFi cameras with a power consumption of 5 Watts [50] and monitoring sensors with a power consumption of 0.06 Watts [51]. In the case of an access point, we use the energy values from an LTE base station: 333 Watts when idle and 528 Watts at max [52,20]. A WiFi base station presents similar power consumption values [53].

4.2. Networking part

Network devices, such as routers and switches, typically presents a linear relation between power consumption and load

Table 3
Network device values from [20].

Parameter	Edge router	Core router
Idle consumption	4095 Watts	11 070 Watts
Max consumption	4550 Watts	12 300 Watts
Traffic	560 Gbps	4480 Gbps
Energy	37 nJ/bit	12.6 nJ/bit

Table 4
Overall power consumption in Watts of the IoT devices (simulated with ns3).

# devices	360p	480p	720p
1	6.907	6.908	6.909
2	12.869	12.87	12.873
3	18.831	18.832	18.837
4	24.792	24.795	24.801
5	30.754	30.757	30.765
6	36.716	36.719	36.728
7	42.677	42.682	42.692
8	48.639	48.644	48.656
9	54.601	54.606	54.62
10	60.562	60.568	60.583

and a high static consumption (power when the device is on but idle) [41,20]. Table 3 presents typical values for edge and core routers. Here, we assume that the networking part – between the access point and the Cloud – comprises only these two types of devices. We assume that the edge Cloud is linked to the access point through edge routers while the core Cloud is reached through core routers. As explained in [20], we will use an energy-per-bit model, thus assuming a proportional sharing of the router static energy consumption among the network flows.

The energy consumption of the network flows depends also on the number of routers they have to cross. We consider a core Cloud located 10 hops away from the access point for a 100 ms Round-Trip-Time according to values measured in [54] between clients and Amazon Cloud. Finally, we consider a PUE of 1.7 in the access point and the network points-of-presence, which is a typical value for small data centers [55].

4.3. Cloud part: Edge and core model

Inspired by the previous works on video stream analysis [45,56] and edge-computing [57], our model involves two types of computing resources.

Because a user is physically close to the edge, the servers placed at the edge enables low latency for users. The data transfers from users to edge can have a lower latency than direct transferring to the core Cloud. Conversely, the computation capacities at the edge Cloud is limited and can be seen as a small-scale data center, the considered edge comprises between 20 to 50 servers. Each server has limited physical resources in terms of CPU, RAM and ingress bandwidth. We assume that there is no centralized storage system at the edge Cloud: each server has its own hard disk [58]. Once the edge cannot satisfy the computational task QoS requirement, it transfers the task to core where sufficient computing resources are available.

The core represents a federation of inter-connected data centers which are usually far from users. Although the servers placed at the core Cloud have higher latency than edge servers, either their number or their performance (of core servers) are higher than at the edge. From the energy cost perspective, the data processing at the core is faster than data processing at the edge. However, a large volume of data needs to be transferred to core to process such that the communication cost between user-core through the Internet cannot be ignored.

A job is a request from a vehicle that requires computing resources for processing. It can be submitted to the edge and the core at anytime. Once the request is accepted, a Virtual Machine (VM) is created on a server at the edge or core to process the analysis. A VM is considered as the basic unit of resource allocation. Each VM is created with its specific CPU and RAM requirements. When the vehicle leaves this section of road, the VM is destroyed and it releases its reserved resources back to the server.

In addition to the computing power, we will use the PUE in order to account for the full energy cost of using computing resources in Cloud environments. The PUE for the Edge Cloud is assumed to be around 1.7 and for the centralized Cloud around 1.2, according to a recent U.S. study [55].

5. Evaluation

This section provides the validation results on the considered use-case. First, we instantiate our energy model of the IoT devices using the ns3 simulator [59] and we present simulation results about the energy consumption of the IoT devices (Section 5.1). Then, we describe the setup of our experimental testbed (Section 5.2) used to provide real measurements on Cloud servers for the execution of the stream analysis application (object detection). The experiments explore the various possible Cloud configurations in order to execute the given application (Section 5.3). These small-scale measurements are integrated within our simulator [60] in order to extrapolate these values for the large-scale scenario and to include telecommunication networks (Section 5.4). We also explore the trade-off between application's accuracy (i.e. object detection accuracy depending on the number of sources providing streams and the image resolution of each stream) and energy consumption (Section 5.5). Finally, we gather all the results to draw the complete picture of the energy consumption induced by the studied application on the considered infrastructure, and consequently compare the consumption of the different parts (Section 5.6).

5.1. IoT devices consumption

In order to evaluate the energy consumption of the IoT devices, we have used the ns3 simulator [59] and its implementation of the energy consumption model described in Section 4.1 for the IoT devices [48]. The simulations mimic the behavior of 802.11n (5 GHz band) devices. The IoT device is moving at a constant speed (using the constant velocity mobility model of ns3) within the range of the access point. Simulations are following the application's bit rates according to the values provided in Table 1. As there are no random variables in this simulation, only one run is conducted.

Fig. 3 shows the dynamic power consumption for an increasing number of IoT devices: this consumption only takes into account the dynamic energy needed to send data streams. As expected, 720P streams are consuming more power. Table 4 presents the overall power consumption (including the static power consumption of the cameras) for an increasing number of IoT devices. One can see that communications are almost negligible in comparison with the static power consumption of the devices. Consequently the video format has an almost negligible influence on the overall power consumption (on the order of a mW for one device). However, the traffic varies from one format to another.

If we consider an ideal access point providing 54 Mbps of bandwidth (classical value for 802.11n [61]), it means that ideally it can serve up to 105 streams at 360p, 76 streams at 480p and 45 streams at 720p. Fig. 4 presents an extrapolation of the total power consumption (IoT devices and access points) for up to 300 vehicles with 54Mbps of bandwidth per access point. This ideal extrapolation is based on values obtained through the simulations illustrated by Fig. 3 and Table 4. The steps correspond to the
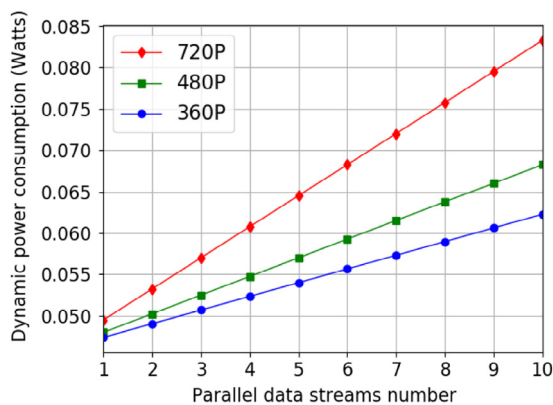
Fig. 3. Dynamic power consumption of the wireless communications for the IoT devices (simulated with ns 3).

Fig. 4. Power consumption for the entire IoT part including devices and access points (extrapolation from ns3 simulations presented in Table 4.

successive addition of new access points to support the traffic and to compute this extrapolation, we assume ideal conditions without interference.

5.2. Setup for the cloud and networking parts

The first half of our experiments consists in measuring the power consumption and performance degradation with different application resolutions on our experimental test-bed Grid'5000, a French platform for experimenting distributed system [62]. We use the same servers as mentioned in [60]: the servers are Dell PowerEdge R720 from the *Taurus* cluster at Grid'5000 Lyon site. Each server is composed of two Intel Xeon E5-2630 processors (2.3 GHz) each with 6 cores, 32 GB of RAM and 600 GB of disk space. The processors support hyper-threading technology thus the total of 12 physical cores servers can provide 24 virtual CPUs. We employed KVM as the virtualization solution along with Linux on x86-based servers. The experiment results are used for building power and performance models. The network energy consumption model is defined in a similar way as it is in [20] and based on per-bit cost. These models were integrated into our simulator described in [60]. In order to extrapolate to large-scale, the second half of our experiments are held using this simulator.

The servers are placed at both edge and core. Most of previous studies [41] agree on the fact that the dynamic server power consumption mainly depends on the working CPU frequency. The server power consumption is taken for different CPU load profiles as described in [60]. Furthermore, our experimental results show in particular that a server on idle state consumes roughly half of its maximal power consumption. From the latency point-of-view, we assume a 100 ms Round-Trip-Time (RTT) between the vehicles and the core Cloud. This value is similar to what can be observed for accessing an Amazon Cloud for instance [54].

5.3. VM size and time analysis

Due to the server limited computational capacity, allocating resources to VMs needs to be carefully done. The goal of our first experiment is to evaluate the video analysis performance and energy consumption on different VM sizes. In this experiment, we create two individual VMs on two servers from the Taurus cluster. The VM-1 is given 2 virtual CPU and 2 GB RAM, and the VM-2 is given 4 virtual CPU and 4 GB RAM.

The analysis time per frame of VM-1 and VM-2 are shown in Fig. 5. VM-2 is 26%, 33% and 35% faster than VM-1 for resolutions of 360p, 480p and 720p respectively. Clearly, the VM-2 benefits from more computational resources (i.e. the application makes
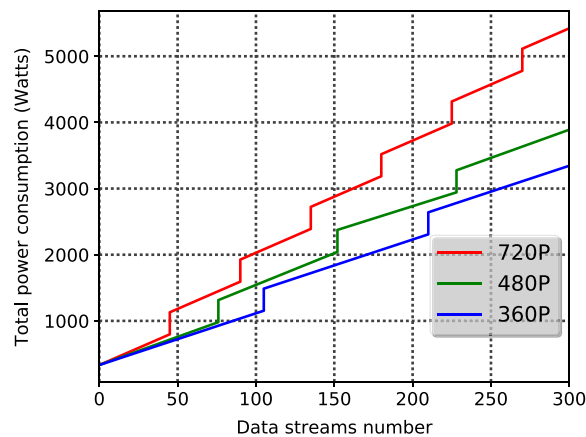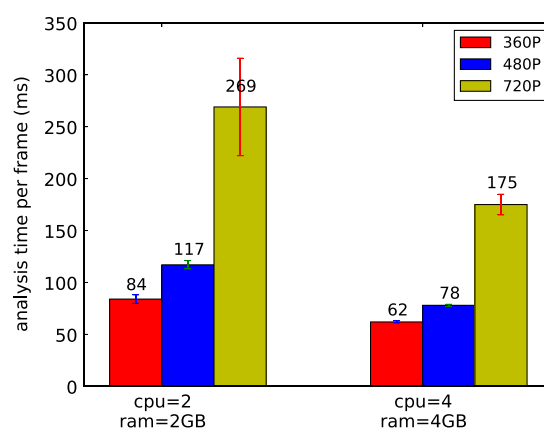

Fig. 5. Time analysis on different VM sizes (measured on our servers).

advantage of parallel computations) and it results in a reduced analysis time.

We then move on to another experiment where we vary the number of VMs. We first create VM-1 $\sim$ 4 on server Taurus-12, four identical VMs, and each VM has the same hardware configuration: 2 vCPU and 2 GB RAM. These VMs process only 1 data stream at a time. VM-5 is created on server Taurus-13 with 8 vCPU and 8 GB RAM. Unlike VM-1 $\sim$ 4, it processes 4 data streams in parallel. We conducted the experiments on analyzing the same video. Each experiment, corresponding to one video format in one scenario, has been run 10 times in order to get statistically sounding results.

The results are shown in Fig. 6. For each point presented in Fig. 6, 10 iterations of the same experiment have been run to get significant results. Error bars illustrate the standard deviation of the obtained measurements. Fig. 6(a) is when 4 individual small VMs are used on Taurus-12 and each VM only processes 1 data stream. Analysis time is similar for the four VMs for each resolution format. In Fig. 6(b), it shows the processing of 4 data streams in parallel within a large size VM on Taurus-13. We observe that processing 4 streams in 1 large VM is faster than processing in 4 small size VMs. We attribute this to the fact that the KVM virtualization layer adds a penalty. In case of 4 VMs, the computational resources given to each VM from KVM is not always from the same physical cores. In other words, there is a scheduling cost if a VM is not always using at least one physical core. Moreover, as we are executing the same application four times in parallel, there might be a positive memory mutualization effect for the large VM case that does not appear in the case of separate VMs.
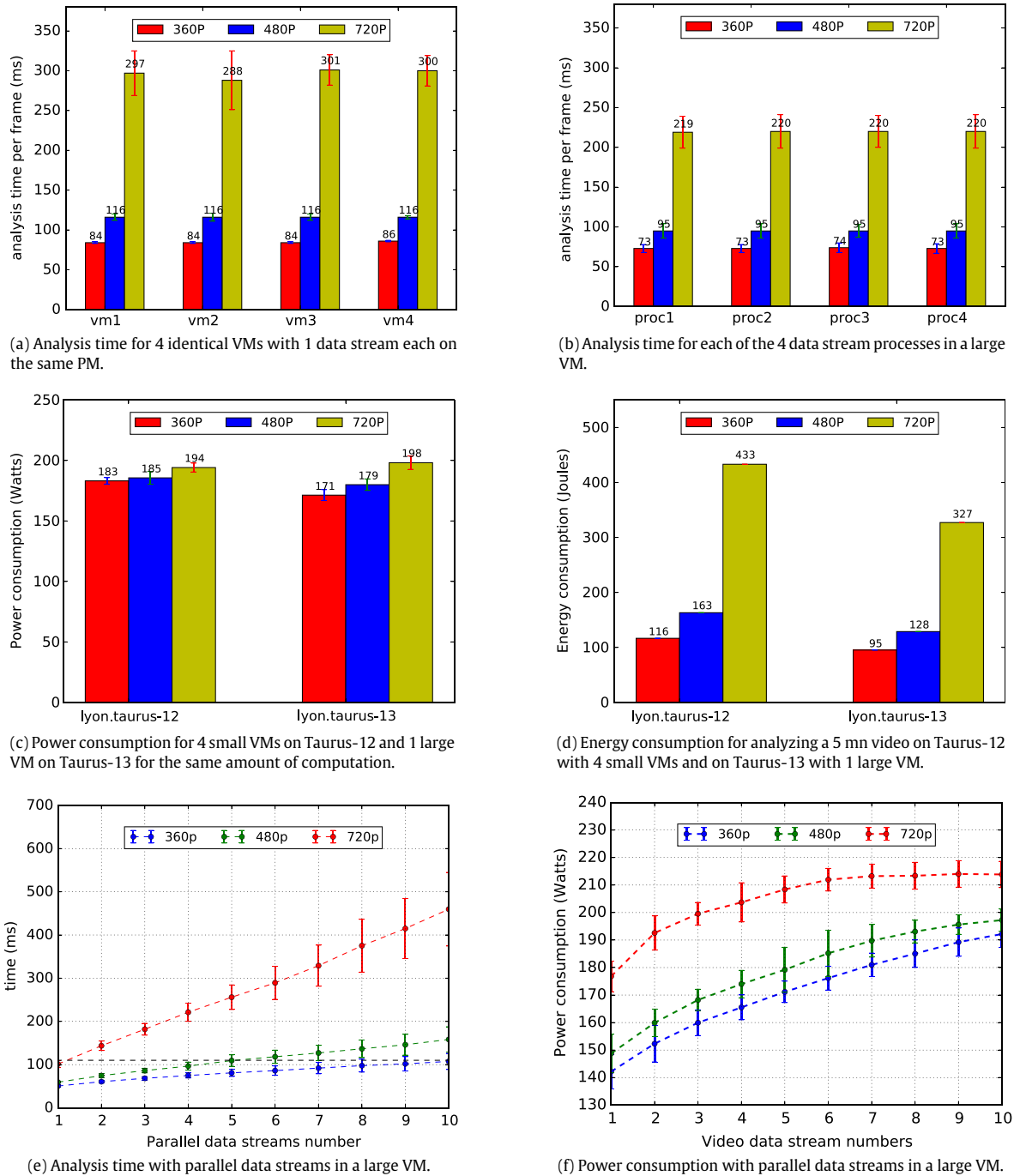
(a) Analysis time for 4 identical VMs with 1 data stream each on the same PM.

(b) Analysis time for each of the 4 data stream processes in a large VM.

(c) Power consumption for 4 small VMs on Taurus-12 and 1 large VM on Taurus-13 for the same amount of computation.

(d) Energy consumption for analyzing a 5 mn video on Taurus-12 with 4 small VMs and on Taurus-13 with 1 large VM.

(e) Analysis time with parallel data streams in a large VM.

(f) Power consumption with parallel data streams in a large VM.

Fig. 6. Energy consumption and frame analysis resolution time in 360p, 480p and 720p (measured on our servers).

Fig. 6(c) shows the power consumption of Taurus-12 with 4 small VMs processing 1 data stream each, and Taurus-13 with 1 large VM hosting the same 4 data streams. As shown in Fig. 6(c), the average power consumption (in Watts) for processing 4 data streams in 1 larger VM is lower compared with 4 small VMs. For analyzing a 5 min video, as shown in Fig. 6(d), the large VM (VM-5 on Taurus-13) with faster speed of frame analysis and lower instantaneous power consumption, consumes less energy in total.

We also observe that the processing time increases significantly when the resolution format increases. For instance, if we expect to analyze 8 frames per second (e.g., Simoens et al. [45] select 1 out of 24 frames for analyzing) for a relevant application precision: in

order to ensure the accuracy of analysis process, this throughput should be as maximum as possible. It means that we have to analyze 1 frame in every 3 frames with a video at 25 fps (frame per second). It means that the average analysis time per frame must be smaller than 125 ms. To compute the maximum number of videos that can be analyzed in parallel, we assume that 1 VM is used for analyzing 1 format of video. We measure the analysis time on VM-5 for a video in the 3 resolution formats.

As shown in Fig. 6(e), the large VM (VM-5 on Taurus-13) supports in parallel up to 11 videos streams in resolution 360p, 4 video streams for 480p video and only 1 for 720p video. Fig. 6(f) shows the respective energy consumption in the 3 resolution formats for the large VM.

5.4. Edge and core Clouds' energy consumption

In this subsection, we evaluate the effects of offloading computation tasks at the edge for system performance of our framework and energy consumption at edge and core. We study the scalability of our framework by increasing the number of vehicles (video sources). We assume that there is no bottlenecks in the network between user-edge and edge-core. The experiments in this subsection are performed using simulations based on the real measurements done in the previous subsection. These simulations consider the average values displayed on Fig. 6.

Edge usually has less computational resources in comparison with core. In initial configuration, edge has 5 servers and core has 100 servers. Each edge server has 24 virtual CPU and 24 GB of RAM and the core servers are twice as powerful as edge servers. To avoid the energy consumption associated with VM placement, we assume all the VMs are same size that consists of 8 virtual CPU and 8 GB of RAM at edge. The VMs have 24 virtual CPU and 24 GB of RAM at core implying the time analysis is reduced. We only consider 360p and 720p video formats in this scenario in order to illustrate that different resolutions impact energy consumption and performance. As mentioned before (Section 5.3), a VM processes one format of video in the experiment thus a VM at maximum processes 1 video stream for 720p, and 10 video streams for 360p in parallel as shown in Fig. 6(e). All the requests of data analysis are processed at the edge by default. If edge does not have sufficient resources for processing, the request is transferred to core.

The goal of this experiment is to measure the total energy consumption at both the edge and core. We first assume that all the data streams are 360p. At beginning, there are few vehicles in the system. These vehicles first offload their data to edge to be processed. When increasing the number of data streams, the edge energy consumption increases by processing these data streams. As shown in Fig. 7(a), we observe that the core does not consume any energy before the edge computing resources are exhausted. Core starts to process data when the number of data streams exceeds 112 in the system. In Fig. 7(b), all the 360p videos are replaced by 720p, the edge quickly drained its resources when processing 720p videos as it consumes more computation resources than 360p videos. The core receives the first request of data analysis from the 16th vehicle. From that moment, all the new data arrivals are directed to core to be processed.

In Fig. 7(c), we can observe that the average delay of 360p videos is significantly lower than for 720p videos. Indeed most analysis tasks are performed at edge instead of at core. Once the edge has exhausted all its resources, the new arrivals are migrated at the core. On the scale of 300 vehicles, edge is capable of processing 37.3% of 360p videos streams in the system. In contrast to 360p, processing 720p video streams consumes much more computing resources than processing 360p videos. The edge can only process 5% of data streams and all the other data streams have to move to core for processing. Despite the fact that the core possesses more powerful computational resources which might even reduce the time analysis, the latency from the network between edge and core cannot be ignored. Fig. 7(c) also demonstrates that the average delay of all videos are mainly depending on the number of data streams offloading to the core. When increasing the number of data streams moving to core, the network energy consumption for the edge Cloud is also increased as shown in Fig. 7(d). Yet, it should be noticed that this energy consumption is almost negligible in comparison to the Cloud energy consumption.

5.5. Application's accuracy

Processing analysis in higher resolution video format often outputs a result with higher detection accuracy. However, it consumes

Table 5
The detection accuracy of different objects from [45].

Classes	720p	480p	360p
Car	96.7%	91%	88.5%
Body	97.7%	94.9%	90.7%
Dog	96.1%	94.9%	90.7%
Total	96.7%	92.3%	87.9%

Table 6
The number of cameras needed for achieving the indicated number of nines.

# nines	720p	480p	360p
99.9%	3	4	6
99.99%	4	6	8
99.999%	5	7	11

consequently larger computational resources including CPU/RAM and bandwidth for transmission. Reducing the resolution is a clear way to save computing resources and network utilization, and thus energy. Edge servers can process more videos streams in parallel without significant performance degradation. It potentially decreases network usage, thus more video streams can be processed at edge. However, scaling down the video affects the detection accuracy. As mentioned in [45], lowering the resolution of video significantly reduces the detection accuracy. We show the initial accuracy settings used in this subsection for object detection in Table 5.

Assuming that there is only one car in the section *AB* of road, the detection accuracy for car is equal to 96.7%, 91%, 88.5% with 720p, 480p and 360p video format respectively. Now, we assume that there are two cars in the same section, their cameras both capture with resolution 360p. When one of the two cameras detects an object on the road and another did not, one can wonder in this case, which camera should be used for the definitive result? Furthermore, we replace one camera by using 720p resolution. Suppose the two results are still different, should we always believe the result with higher resolution (720p) because of its higher detection accuracy by default?

Unfortunately, we cannot directly conclude which result of the two is more believable. Even though the 720p videos often offers a higher detection accuracy than 360p videos, this only shows that 720p is more likely to be correct, but it is not conclusive. However, when increasing the number of cameras, we show that the correct probability of result is not only depending on the initial detection accuracy, but also related to the number of cameras in the system. Suppose there are $2n + 1$ cars in the same section of road. All the cars upload video streams with the same resolution and then they output $2n + 1$ results. Intuitively, if there is a result appearing at least for half of the total number of cars, we prefer to select this result as the final result. We define the *reliability* as the probability of a result that appears to exceed $n + 1$ times among $2n + 1$ results. We have proven in a previous work [17] that this final result becomes more believable when the number of cameras increase.

We introduce the **nines** conception which is typically expressed as a percentage with a number of nines (e.g., 99% $\rightarrow$ two nines, 99.9% $\rightarrow$ three nines, etc.). This conception is similar with the *high availability* conception in system design which aims to ensure an agreed level of operational performance. It could thus be used as a negotiated metric between the client and the Cloud provider within an SLA (Service Level Agreement). From the Cloud provider's perspective, the video resolution format can be seen as a green lever allowing for a controlled application performance degradation in return for lower resource allocation and thus, energy savings.

As shown in Fig. 8 and Table 6, the resolution 360p requires 6 cameras working simultaneously in order to achieve *three nines*,
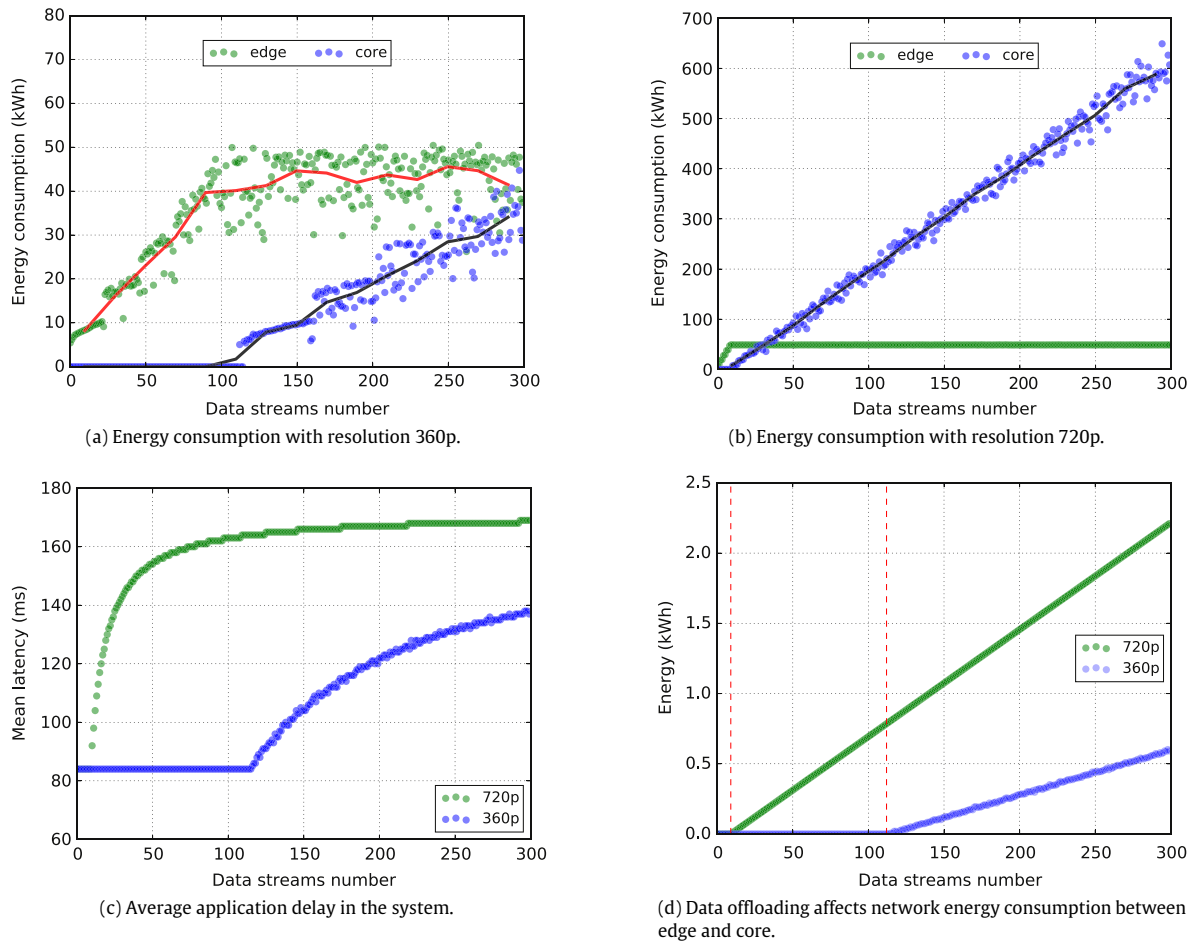
(a) Energy consumption with resolution 360p.

(b) Energy consumption with resolution 720p.

(c) Average application delay in the system.

(d) Data offloading affects network energy consumption between edge and core.

Fig. 7. Energy consumption and performance at edge and core Clouds (simulated with our simulator using measured data presented in Fig. 6).
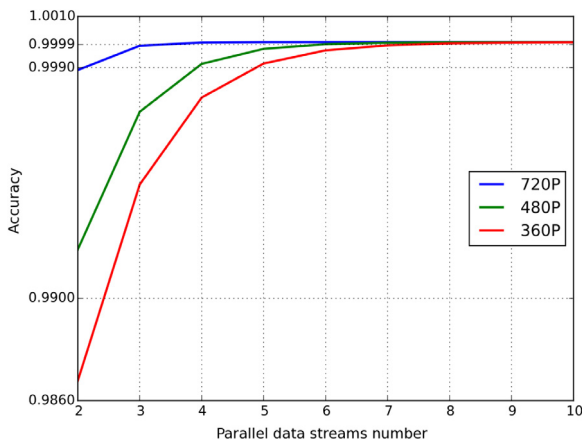

Fig. 8. Application's accuracy (from the statistical formula detailed in [17]).

the 480p requires 4 cameras and the 720p requires only 3 cameras to achieve the same level of accuracy. The higher the resolution is, the lower is the number of cameras required for reaching a given level of reliability.

When correlating these results with the figures presented in Section 5.3, one can observe that to reach the highest level of accuracy (99.999%), 5 720p streams are required, thus needing 5 large VMs; and 11 360p streams, equating to only 1 large VM. Moreover, both options require approximately the same amount

Table 7
Estimation of the power cost per 360p stream for each part (using simulations presented in Figs. 4 and 7(d), and real measurements shown in Fig. 6(f)).

Scenario	IoT	Network	Cloud
Edge cloud	10.96 Watts	0.07 Watts	32.3 Watts
Core cloud	10.96 Watts	0.11 Watts	22.8 Watts

of network traffic. Hence, the energy-performance trade-off clearly lies in favor of lower-quality videos.

5.6. End-to-end energy consumption

In order to compare the energy consumption of each part, we have computed the cost per stream for a 360p video. For the Cloud part, from Fig. 6(f), we can estimate the cost of a single stream processed in a VM on a given server assuming a proportional sharing of the server static energy consumption [21]. Then we multiply this cost by the PUE of the considered Cloud (edge or core). For the IoT part, we use Fig. 4 and the staircase curve is approximated by a linear function in order to obtain a single cost per stream. This method is less precise than using the computed values directly, but it allows for simpler comparisons with the other parts. Finally, for the networking part, as in Fig. 7(d), the cost per bit is employed with different numbers of devices depending on the case (edge Cloud or core Cloud).

Table 7 reports the computed power cost per stream for a 360p video for each part as defined in Section 4 depending on the use-case: edge Cloud or core Cloud. As we use linear models in order to

obtain a cost per frame, these values assume that all the employed devices are either power proportional or fully loaded (i.e. the infrastructure is well sized in relation to the number of users). While this can be a reasonable assumption for the computing resources (routers typically use over-commit techniques in data centers in order to increase their utilization ratio), it is less exact for networking devices that are usually highly redundant equipment in order not to be too much loaded. But, such a rough model is sufficient to have an idea about the main consuming part and the overall trend as we can see on Table 7.

From this estimation, we can see that predominant consumption is the Cloud consumption (computing resources) in both cases: edge and core Clouds. But, in the case of the edge Cloud, it represents three quarters of the overall cost, while it represents two thirds for the core Cloud case. In both cases, the networking part is negligible, although routers are the most consuming devices per unit. Yet, if they are suitably loaded, their energy efficiency is high due to their large capacities. Finally, the IoT part, that includes the IoT device and the access point, accounts for one quarter of the overall cost for the edge Cloud case and one third for the core Cloud. These estimations advocate for a better energy efficiency of Cloud infrastructures.

6. Discussion

As mentioned in Section 5.4, the edge could be capable of generating its own energy and storing the surplus energy into batteries [17]. Due to its limited computing resources, the edge Cloud cannot support huge amounts of processing that needs to occur at the same time. Thus, new incoming data streams have to move to core Cloud for quick analysis. As we conclude previously, in order to reduce the energy consumption, it is then better to decrease the resolution format for all videos with a penalty on detection accuracy. From an environmental point of view, if the user expects high accuracy of detection and to consume clean energy instead of brown one, he first needs to ensure that the data is processed at the edge. As the number of users grows, we then have to increase not only the number of edge servers, but also the solar photovoltaic panels that are able to provide as much energy as the servers need or to switch to the core Cloud with lower performances in terms of latency.

To reduce the total energy consumption, another alternative solution, displayed in Fig. 9, consists in changing the division of labor between edge and core. The finite computing resources at edge are no longer used for data analysis but for video decoding, sampling and encoding. As such 720p videos in particular consumes a lot of computing resources. Even when taking all the edge servers, it is still far from enough for processing all the 720p videos in the system. Thus, carefully using edge resources is important for the overall framework optimization. As described in Section 5, it needs to analyze 8 frames every second for a video at 25 frames per second. It means that we select 1 frame out every 3 frames for processing.

In particular, we expect that the sampling work can be done at the edge. When a new video stream arrives, the edge performs decoding, sampling and encoding successively on this video and then transfers it to core. Although the data has to move to core for processing, their size is reduced and the energy consumption over the network is also reduced. Unfortunately, the result of this scenario is unsatisfactory. Decoding a video at 720p is extremely fast but encoding will take 15 times more than decoding in our experiment. It leads to an additional delay (roughly 100 ms in our experiments) while the latency is crucial in this scenario. This opportunity for data movement could be explored through the development of a framework that couples the tasks and computes partially on the transferring path, thus reducing the network cost
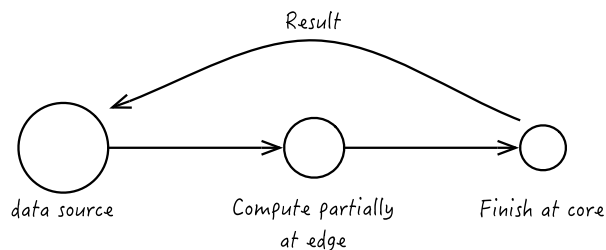

Fig. 9. Computing partially at edge.

[63]. This option could also save energy because, as shown in Section 5.6, the networking part is almost negligible in terms of energy cost.

Finally, as the Cloud, whether edge or core, is the most consuming part for IoT platforms, the increasing number of devices will raise major challenges for the Cloud infrastructures in terms of energy consumption. From Table 7, one can indeed roughly estimates that for a 6 Watts device like a camera, sending data with a 514 kbps data rate, the data processing requires a VM consuming 5 times more power in an edge Cloud.

7. Conclusion

Data loses its value when it cannot be analyzed quick enough. Offloading the data to process video streams at edge effectively reduces the response time and avoids unnecessary data transmission between edge and core, thus reducing the network energy overhead. Moreover, it can extend for instance the battery lifetime of end-user equipment (e.g., wearable equipment). Meanwhile, the traditional energy consumption and carbon footprint can be reduced by building self-producing electricity edge.

We proposed an analytic model to estimate the energy consumption of Edge Cloud-based IoT Platforms for IoT. This model can be used to decide whether to offload computation from the objects to the edge or to the core Cloud, depending on the number of devices and the desired application QoS. This model was validated on our application use-case that deals with video stream analysis from vehicle cameras. Our results show the relationship between the number of cameras, the application accuracy and the processing energy cost, opening new research directions on finding relevant trade-offs between application performance degradation and energy consumption of underlying Cloud systems.

The scenario in this paper can be seen as a concrete example to demonstrate the advantages of offloading the data to process at the edge or core in an energy saving context. In addition, although this work is camera-based, it can be applied to any other scenario where the data streams need to be processed in real-time as it provides the analytical framework for such applications.

Our results also show the predominance of the Cloud consumption on the overall energy cost of IoT platforms. This predominance is even greater in the case of an edge Cloud where the energy cost is three times bigger for the Cloud infrastructure than for IoT and networking parts together. Consequently, with the connected devices explosion, it becomes urgent to improve the energy efficiency of Cloud infrastructures, and especially for small-sized data centers, in order to limit the impact of IoT on global energy consumption.

References

[1] Ericsson, More than 50 billion connected devices, White Paper, 2011.

[2] D. Evans, The Internet of Things - How the Next Evolution of the Internet Is Changing Everything, Cisco White Paper, 2011.

[3] A. Nordrum, Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated, IEEE Spectrum, 2016.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376.

[5] S.C.B. Intelligence, Six Technologies with Potential Impacts on US Interests out to 2025, Technical Report, National Intelligent Concil, 2008.

[6] S. Abdelwahab, B. Hamdaoui, M. Guizani, A. Rayes, Enabling smart cloud services through remote sensing: An internet of everything enabler, IEEE Internet Things J. 1 (3) (2014) 276–288.

[7] L.M. Vaquero, L. Rodero-Merino, Finding your way in the fog: Towards a comprehensive definition of fog computing, ACM SIGCOMM Comput. Commun. Rev. 44 (5) (2014) 27–32.

[8] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos, Z. Chen, P. Pillai, M. Satyanarayanan, Quantifying the impact of edge computing on mobile applications, in: ACM SIGOPS Asia-Pacific Workshop on Systems, 2016, pp. 5:1–5:8.

[9] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, D.S. Nikolopoulos, Challenges and Opportunities in Edge Computing, CoRR abs/1609.01967.

[10] T. Bawden, Global warming: Data centres to consume three times as much energy in next decade, experts warn, Independent, 2016. http://www.independent.co.uk/environment/global-warming-data-centres-to-consume-three-times-as-much-energy-in-next-decade-experts-warn-a6830086.html .

[11] Í. Goiri, W. Katsak, K. Le, T.D. Nguyen, R. Bianchini, Parasol and greenswitch: Managing datacenters powered by renewable energy, ACM SIGARCH Comput. Archit. News 41 (1) (2013) 51–64.

[12] L.D. Xu, W. He, S. Li, Internet of things in industries: A survey, IEEE Trans. Ind. Inform. 10 (4) (2014) 2233–2243.

[13] P. Serrano, A. Garcia-Saavedra, G. Bianchi, A. Banchs, A. Azcorra, Per-frame energy consumption in 802. 11 devices and its implication on modeling and design, IEEE/ACM Trans. Netwing 23 (4) (2015) 1243–1256.

[14] H. Wu, S. Nabar, R. Poovendran, An energy framework for the network simulator 3 (NS-3), in: International Conference on Simulation Tools and Techniques, SIMUTools, 2011, pp. 222–230.

[15] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376.

[16] S. Islam, J.-C. Grégoire, Network edge intelligence for the emerging next-generation internet, Future Internet 2 (4) (2010) 603–623.

[17] Y. Li, A.-C. Orgerie, I. Rodero, M. Parashar, J.-M. Menaud, Leveraging renewable energy in edge Clouds for data stream analysis in IoT, in: IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid, 2017, pp. 186–195.

[18] L. Rosyidi, R.F. Sari, Energy harvesting aware protocol for 802.11-based internet of things network, in: IEEE Region 10 Conference, TENCON, 2016, pp. 1325–1328.

[19] Y.W. Kuo, C.L. Li, Design of long range low power sensor node for the last mile of IoT, in: IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW, 2016, pp. 1–2.

[20] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, R.S. Tucker, Fog computing may help to save energy in cloud computing, IEEE J. Sel. Areas Commun. 34 (5) (2016) 1728–1739.

[21] M. Kurpicz, A.-C. Orgerie, A. Sobe, How much does a VM cost? Energy-proportional accounting in VM-based environments, in: Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP, 2016, pp. 651–658.

[22] F. Yang, S. Wang, J. Li, Z. Liu, Q. Sun, An overview of internet of vehicles, China Commun. 11 (10) (2014) 1–15.

[23] A. Ishii, T. Suzumura, Elastic stream computing with clouds, in: 2011 IEEE International Conference on Cloud Computing (CLOUD), IEEE, 2011, pp. 195–202.

[24] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, J. Stefa, Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: Review, challenges, and a case study, IEEE Netw. 30 (2) (2016) 54–61.

[25] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall, Reducing network energy consumption via sleeping and rate-adaptation, in: USENIX Symposium on Networked Systems Design and Implementation, NSDI, vol. 8, 2008, pp. 323–336.

[26] W. Fisher, M. Suchara, J. Rexford, Greening backbone networks: Reducing energy consumption by shutting off cables in bundled links, in: ACM SIGCOMM workshop on Green networking, 2010, pp. 29–34.

[27] S. Figuerola, M. Lemay, V. Reijs, M. Savoie, B.St. Arnaud, Converged optical network infrastructures in support of future internet and grid services using IaaS to reduce GHG emissions, J. Lightwave Technol. 27 (12) (2009) 1941–1946.

[28] W. Zhu, C. Luo, J. Wang, S. Li, Multimedia cloud computing, IEEE Signal Process. Mag. 28 (3) (2011) 59–69.

[29] M. Bertier, F. Desprez, G. Fedak, A. Lebre, A.-C. Orgerie, J. Pastor, F. Quesnel, J. Rouzaud-Cornabas, C. Tedeschi, Beyond the clouds: How should next generation utility computing infrastructures be designed? in: Cloud Computing, Computer Communications and Networks, Springer, 2014, pp. 325–345.

[30] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30–39.

[31] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, vol. 1, 2001, pp. 511–518.

[32] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Mach. Learn. 37 (3) (1999) 297–336.

[33] J. Koomey, Growth in data center electricity use 2005 to 2010, White paper, 2011.

[34] R. Schmidt, D. Beaty, J. Dietrich, Increasing energy efficiency in data centers, ASHRAE J. (2007) 18–24.

[35] J. Whitney, P. Delforge, Data center efficiency assessment scaling up energy efficiency across the data center industry: Evaluating key drivers and Barriers, NRDC White Paper, 2014.

[36] K. Church, A.G. Greenberg, J.R. Hamilton, On delivering embarrassingly distributed cloud services., in: HotNets, 2008, pp. 55–60.

[37] A. Shehabi, S. Smith, N. Horner, I. Azevedo, R. Brown, J. Koomey, E. Masanet, D. Sartor, M. Herrlin, W. Lintner, United states data center energy usage report, Lawrence Berkeley National Laboratory, Berkeley, California. LBNL-1005775 Page 4.

[38] C. Belady, A. Rawson, J. Pfleuger, T. Cader, Green grid data center power efficiency metrics: PUE and DCIE, Technical Report, Green Grid, 2008.

[39] Green Grid, Harmonizing global metrics for data center energy efficiency, global taskforce reaches agreement regarding data center productivity, Technical Report, The Green Grid Consortium, 2014.

[40] M. Chowdhury, M. Zaharia, J. Ma, M. Jordan, I. Stoica, Managing data transfers in computer clusters with orchestra, in: SIGCOMM, 2011, pp. 98–109.

[41] A.-C. Orgerie, M. Dias de Assunção, L. Lefèvre, A survey on techniques for improving the energy efficiency of large-scale distributed systems, ACM Comput. Surv. (CSUR) 46 (4) (2014) 47.

[42] R. Bolla, R. Bruschi, F. Davoli, F. Cucchietti, Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures, IEEE Commun. Surv. Tutor. 13 (2) (2011) 223–244.

[43] C.W. Tsai, C.F. Lai, M.C. Chiang, L.T. Yang, Data mining for internet of things: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2014) 77–97.

[44] B.B.P. Rao, P. Saluia, N. Sharma, A. Mittal, S.V. Sharma, Cloud computing for Internet of Things amp; sensing based applications, in: International Conference on Sensing Technology, ICST, 2012, pp. 374–380.

[45] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, M. Satyanarayanan, Scalable crowdsourcing of video from mobile devices, in: ACM International Conference on Mobile Systems, Applications, and Services, 2013, pp. 139–152.

[46] https://www.ffmpeg.org.

[47] J. Diaz-Montes, M. Zou, I. Rodero, M. Parashar, Enabling autonomic computing on federated advanced cyberinfrastructures, in: ACM Cloud and Autonomic Computing Conference, ACM, 2013, p. 20.

[48] M. Ergen, P. Varaiya, Decomposition of energy consumption in IEEE 802.11, in: IEEE International Conference on Communications, ICC, 2007, pp. 403–408.

[49] D. Halperin, B. Greenstein, A. Sheth, D. Wetherall, Demystifying 802.11n power consumption, in: International Conference on Power Aware Computing and Systems, HotPower, 2010, pp. 1–5.

[50] D-Link, DCS-2330L HD Wireless N Day/Night Outdoor Cloud Camera, Datasheet, 2014. http://www.dlink.com/uk/en/-/media/consumer_products/dcs/dcs-2330l/datasheet/dcs_2330l_datasheet_deu.pdf.

[51] Libelium, Waspmote technical guide, 2017. http://www.libelium.com/downloads/documentation/waspmote_technical_guide.pdf.

[52] G. Auer, et al., Energy efficiency analysis of the reference systems, areas of improvements and target breakdown, Tech. rep., European Project NFSO-ICT-247733 EARTH - Deliverable D2.3 2012.

[53] M. Deruyck, W. Vereecken, W. Joseph, B. Lannoo, M. Pickavet, L. Martens, Reducing the power consumption in wireless access networks: Overview and recommendations, Prog. Electromagn. Res. 132 (2012) 255–274.

[54] I. Cuadrado-Cordero, F. Cuadrado, C. Phillips, A.-C. Orgerie, C. Morin, Microcities: A platform based on microclouds for neighborhood services, in: International Conference Algorithms and Architectures for Parallel Processing, ICA3PP, 2016, pp. 192–202.

[55] A. Shehabi, et al., United States Data Center Energy Usage Report, Technical Report, Lawrence Berkeley National Laboratory LBNL-1005775, 2016.

[56] A. Anjum, T. Abdullah, M. Tariq, Y. Baltaci, N. Antonopoulos, Video stream analysis in clouds: An object detection and classification framework for high performance video analytics, IEEE Transactions on Cloud Computing.

[57] Y. Mao, J. Zhang, K.B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, arXiv preprint arXiv:1605.05488.

[58] N. Beldiceanu, B. Dumas Feris, P. Gravey, S. Hasan, C. Jard, T. Ledoux, Y. Li, D. Lime, G. Madi-Wamba, J.-M. Menaud, P. Morel, M. Morvan, M.-L. Moulinard, A.-C. Orgerie, J.-L. Pazat, O.H. Roux, A. Sharaiha, Towards energy-proportional Clouds partially powered by renewable energy, Computing (2016) 20.

[59] ns3 network simulator, http://www.nsnam.org/ns-3-11/.

[60] Y. Li, A.-C. Orgerie, J.-M. Menaud, Opportunistic scheduling in clouds partially powered by green energy, in: GreenCom: IEEE International Conference on Green Computing and Communications, 2015, pp. 448–455.

[61] V. Shrivastava, S. Rayanchu, J. Yoonj, S. Banerjee, 802.11n under the microscope, in: ACM SIGCOMM Conference on Internet Measurement, IMC, 2008, pp. 105–110.

[62] D. Balouek, et al., Adding virtualization capabilities to the grid'5000 testbed, in: Cloud Computing and Services Science, vol. 367, Springer, 2013, pp. 3–20.

[63] I. Petri, J. Diaz-Montes, M. Zou, O.F. Rana, T. Beach, H. Li, Y. Rezgui, In-transit data analysis and distribution in a multi-cloud environment using Comet-Cloud, in: International Conference on Future Internet of Things and Cloud, FiCloud, 2014, pp. 471–476.

Ivan Rodero is an assistant research professor at Rutgers University and a member of the RDI2 and the NSF Cloud and Autonomic Computing Center. His research interests include parallel and distributed computing, high-performance computing, energy efficiency, autonomic computing, grid computing, cloud computing, and data analytics at extreme scales. He has a Ph.D. in computer science and engineering from the Technical University of Catalonia. He is a member of IEEE, the ACM, and the American Association for the Advancement of Science (AAAS).

Betsegaw Lemma Amersho is currently M.Sc. student at Université Rennes 1 in France and Aalto University in Finland. He received his B.Sc. in computer science from University of Gondar, Ethiopia in 2009. From 2010 to 2015, he has been working as a Lecturer at University of Gondar. His research interests include Cloud computing, network simulation and energy efficiency in distributed systems.
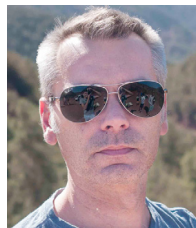
Yunbo Li received an M.Sc. degree in Telecom/Networking from Université Pierre & Marie Curie (UPMC-Sorbonne University), Paris, France in 2013, and a Ph.D. degree in Computer Science from IMT Atlantique, Nantes, France in 2017. His Ph.D. subject was on Resource allocation in a Cloud partially powered by renewable energy sources. His interests are in networks and cloud computing. His works are mainly on resource allocation in context of cloud-/edge-computing and integrating renewable energy.

Anne-Cécile Orgerie received her Ph.D. degree in Computer Science from Ecole Normale Supérieure de Lyon (France) in September 2011. From October 2011 to September 2012, she was working as a postdoc at the Department of Electrical and Electronic Engineering at the University of Melbourne (Australia) on the PetaFlow project (French–Japanese project). She has been a full time researcher at CNRS in the IRISA laboratory (Rennes, France) since October 2012. Her research interests focus on energy efficiency, cloud computing, and distributed systems.

Manish Parashar is a professor in the Department of Computer Science at Rutgers University, he is the director of the Rutgers Discovery Informatics Institute (RDI2) and the US National Science Foundation (NSF) Cloud and Autonomic Computing Center (CAC) at Rutgers, and he is the associate director of the Rutgers Center for Information Assurance. His research interests focus on applied parallel and distributed computing and computational and data-intensive science and engineering. He has a Ph.D. in computer engineering from Syracuse University. He received the IBM Faculty award twice, as well as a US NSF Career award, and he is an American Association for the Advancement of Science (AAAS) and IEEE Fellow.

Jean-Marc Menaud is a professor at IMT Atlantique, Nantes, France. He received his Ph.D. in computer science from Université of Rennes 1, France. He works on software solutions to manage virtual machine in cluster environment, and on dynamic consolidation systems for virtualized environment using the constraint programming approach. Around this topic, he is also involved in the modeling, scaling and validation of cloud IaaS management system. A technology transfer has enhanced his research, by the EasyVirt start'up creation in 2011, a young innovative enterprise specialized in the data center management and optimization. He is the author or co-author of more than 40 international publications or communications.