Cooperative In-network Computation in Energy Harvesting Device Clouds

Chamil Kulatunga*, Kriti Bhargava, Dixon Vimalajeewa, Stepan Ivanov

Telecommunications Software and Systems Group
Arclabs Research and Innovation Centre
Waterford Institute of Technology, Carriganore
Waterford, Ireland

(ckulatunga, kbhargava, dvimalajeewa, sivanov) @tssg.org

Abstract

The Internet of Things paradigm is creating an environment where the big data originators will be located at the edge of the Internet. Accordingly, data analytic infrastructure is also being relocated to the network edges, to fulfill the philosophy of data gravity, under the umbrella of Fog Computing. The extreme edge of the hierarchical infrastructure consists of sensor devices that constitute the Wireless Sensor Networks. The role of these devices has evolved tremendously over the past few years owing to significant improvements in their design and computational capabilities. Sensor devices, today, are not only capable of performing sense and send tasks but also certain kinds of in-network processing. As such, triple optimization of sensing, computing and communication tasks is required to facilitate the implementation of data analytics on the sensor devices. A sensor node may optimally partition a computation task, for instance, and offload sub-tasks to cooperative neighbouring nodes for parallel execution to, in turn, optimize the network resources. This approach is crucial, especially, for energy harvesting sensor devices where the energy profile and, therefore, the computation capability of each device differs depending on the node location and time of day. Accordingly, future in-network computing must capture the energy harvesting information of sensor nodes to jointly optimize the computation and communication within the network. In this paper, we present a theoretical model for computation offloading in micro-solar powered energy harvesting sensor devices. Optimum data partitioning to minimize the total energy consumption has been discussed based on the energy harvesting status of sensor nodes for different scenarios. The simulation results show that our model reduced both energy losses and waste due to energy conversion and overflows respectively compared to a data partitioning algorithm that offloads computation tasks without taking the energy harvesting status of nodes into consideration. Our approach also improves energy balance of a WSN which is an important factor for its long-term autonomous operation.

Keywords: In-network analytics, cooperative computing, computation offloading, energy harvesting, low-latency applications, fog computing.

1. INTRODUCTION

With a growing number of devices in the Internet of Things (IoT) and high adopt-ability of cloud-based Big Data analytic platforms, the centralized cloud computing architecture has been recently challenged within the Internet community. Conventional cloud computing had been designed for monolithic applications assuming high availability of resources at large data centres. It saved CPEX for SMEs, particularly, the overall energy consumption of maintaining an Information and Communication Technologies (ICT) infrastructure. Furthermore, centralized clouds optimized resource utilization by statistically multiplexing peak-loads to avoid over-provisioning. This architecture functioned well until IoT devices generated some large datasets in remotely connected application domains such as smart agriculture [9] and Industry 4.0 [1]. Fog computing [26], is a new computing paradigm, that proposes the analysis of data (before aggregating it into big data sets) in a hierarchical and scalable way closer to the data sources. Although the term was coined by Cisco in 2012, the philosophy of data gravity where computation moves towards the data sources as far as they can, had been presented by Dave McCrory in 2010. Harnessing the computational power of the network devices for data processing has the potential to not only reduce the data in the backhaul network and, in turn, the latency experienced by the end users but also improve the overall energy consumption of the IoT platforms [10]. This is particularly useful for applications in rural agriculture and Industry 4.0 where backhaul connectivity is limited between the remote rural farms/factories and the cloud [7].

A number of interpretations of Fog nodes have been proposed, to date. Authors in [2], for instance, discusses Mobile Edge Computing where mobile operators leverage resources

*Corresponding author
Email address: `ckulatunga@tssg.org` (Chamil Kulatunga)

of the edge devices in 5G rather than the centralized servers used in cloud computing for data processing. Several forms of ad-hoc cloudlets (micro-clouds) have been proposed in [4] and [18]. Certain studies have also extended the concept of Fog Computing towards the extreme edge of the IoT in the private, enterprise, and community domains. This is primarily due to the design of pervasive low-power wireless technologies like ULP-PAN and LP-WAN as well as the tremendous improvement in computation capabilities of small devices (as mini-servers) such as CCTV cameras, mobile phones, and more recently, sensor devices that constitute Wireless Sensor Networks (WSN) [8]. In-network processing within WSN (referred here as in-network analytics) has been performed using different techniques such as data fusion, aggregation, compression and feature extraction [25], [21].

It is of particular importance in latency-sensitive applications such as object tracking, intrusion detection, monitoring structural and machine failures, where the result of the processing may not be useful at all times, the response time at event detection is of the order of fraction of a second. As a result, while numerous studies in the past have focused on optimizing sensing and networking tasks to improve the energy efficiency of WSN, attention is being drawn towards triple optimization that includes on-board computation given the increased capabilities of sensor nodes. Maximizing computation within WSN through resource optimization is more desirable as future sensor nodes will be powered via energy harvesting, for continuous use, from background sources such as solar, wind, vibration and radio frequency [15].

Cooperative computing via computation offloading has been suggested for maximizing the use of in-network computational resources. In computation offloading, a device can select (sometimes in an opportunistic way [5], [16]) a proximate infrastructure edge device (gateway) or another stationary or mobile device as an offloadee for parallel execution of tasks at different participating nodes [19]. Collaborative computing within WSN can enhance the capabilities of the resource constrained environment towards effective cyber-foraging approaches as shown in [20]. Multi-objective intelligent decisions can be made to optimize Fog computing resources and their application performance. The decision of how to optimally partition a task and where to offload given a completion time is an important research question which has not been much investigated in the literature. An analytical model for application partitioning in battery-powered WSN environment has been presented in [20]. An initiating node (IN) that is responsible for sensing data is designed that offloads partial computation to a neighbouring node known as the cooperating node (CN) such that the given task completion deadline is met while optimizing the energy resources of the network.

In this work, we consider in-network computation in WSN [14] and extend the cooperative computing approach discussed in [20] for different scenarios in an energy harvesting WSN. While in conventional WSN, the IN offloads less computation to CN owing to high communication energy, in case of energy-harvested nodes, the partitioning must be based on the level of stored energy as well as the current state of the device that determines the level of harvested energy. This is important to avoid over-flow of harvested energy (hence an energy waste) when battery is fully charged or energy conversion efficiency (50 – 70%) incurred by storing harvested energy into battery. Accordingly, we develop models for task partitioning to reduce the overall energy consumption of the network under different scenarios for latency-sensitive applications. Furthermore, we aim at improving the fairness within the network to ensure energy balancing. Our model and the simulation results show that our approach enables optimization of computation and communication for future energy harvested WSN and ensures sustainable operation.

2. COMPUTATIONAL POLICIES FOR CLEAN ENERGY

A node in a conventional sensor network forwards data without changing the payload. Instead, in-network processing allows a Fog node to not only function as a data source or merely relay a data chunk but also perform some computation on the data. In the early days of in-network processing, researchers were limited to a particular application within a sensor network such as calculation of average humidity or identifying a location of an event based on statistically correlated data aggregation. However, this is changing to embed more generic computational functionalities in WSN.

2.1. In-network Cooperative Computing in Wireless Sensor Networks

In-network processing has been applied for data aggregation, fusion, compression and feature abstraction in WSN to save energy by reducing the number of bits and, in turn, data packets transmitted to a centralized server. Computations are performed at specific aggregation nodes (cluster heads) along the path to the destination node (gateway or server). Offloading decisions are, therefore, simple and based on the forwarding algorithm used such as LEACH to answer the question of where rather than what. This has progressed recently to use a swarm of heterogeneous nodes (such as sensors, actuators, robots, smart phones, drones, cameras) that collectively form an in-network analytic platform and requires specification of where as well as what to send. Authors in [11] propose for instance a new in-network computation algorithm based on channel fading to improve the reliability of aggregation function compared to simultaneously sending all or only one sensor reading.

Computation offloading is a useful distributed computing paradigm at different levels of network resources from large data centres to implanted nano-sensors. Highly available cloud computing provides VM/container level computing resources to the users to perform computation tasks in geographically distributed data centres. Mobile edge computing brings cloud resources into the edge of the operator-managed network to reduce core network traffic of the operator and provide low-latency for the users. Enterprise and community-cloud allow the installation of micro data centres that execute micro-services at the proximity of a company office or a community.

The concept of cloudlets proposes the use of a set of mobile devices (different users) that collectively form an ad-hoc cloud [13]. Mobile computation offloading, for instance, can facilitate the execution of compute intensive tasks either on a nearby mobile (in terms of annotations) or on an infrastructure node (e.g. Androidx86).

Computation offloading in WSN is becoming increasingly important as the sensor devices exhibit improved capabilities in terms of computation power and reduced communication energy consumption. In conventional networks, sensor nodes transmit raw data to the sink node where some processing is performed and the results are communicated to the remote cloud. As a result, sensor nodes have prior knowledge of where and what to communicate. Moreover, the energy optimization is included in the algorithms. In modern-day WSN, sensor nodes can make on-the-fly decisions of where and what to compute under a subjected application completion deadline and, in turn, optimize energy usage. Therefore, the pre-designed computation offloading algorithms must be modified to make on-the-fly decisions. Accordingly, energy harvesting and in-network processing can be combined to develop a sustainable and autonomous network operation.

2.2. Heterogeneity in Energy Harvesting Sensor Nodes

Computational sensor nodes, in future, will be powered using diverse natural energy harvesting sources such as solar, wind, radio-frequency, thermal, vibration or piezoelectric [22]. Such energy sources demonstrate random spatial-temporal generation patterns leading to heterogeneity in stored energy between sensor nodes in both outdoor and indoor environments. Changes in the temporal patterns might be significant only on a macro time scale. For instance, while weather may differ from one city to another on a single day at a given time, a sensor network on a smart farm will experience the same effect at the same time. On the contrary, spatial variations among co-located mobile sensor nodes may be obtained due to different orientations and obstacles, for e.g., presence of IMU and GPS modules [3] for animal mobility and location tracking under direct sunlight vs shadows. This heterogeneity will be higher, particularly, in outdoor WSN such as those used in agricultural practices for pasture-based dairy farming (e.g. laying animals with solar-covered tags), site-specific irrigation in cultivation (e.g. leaves may grow into or fall onto the senor nodes) and soil monitoring (e.g. shadows of the plants may cover the soil monitoring sensors).

Optimal energy management in such environments has been proposed using adaptive duty cycling, adaptive communication strategies, routing decision making and application policy management. Authors in [27], for instance, propose optimization of the duty cycle to maximize the common active time based on unpredictable heterogeneity of energy harvesting nodes. The authors propose both online and offline algorithms based on the probability of the harvested energy obtained using a real deployment environment.

We consider cooperation between such sensor nodes to collectively perform computation tasks under a heterogeneous energy harvesting environment. For example, each sensor node in

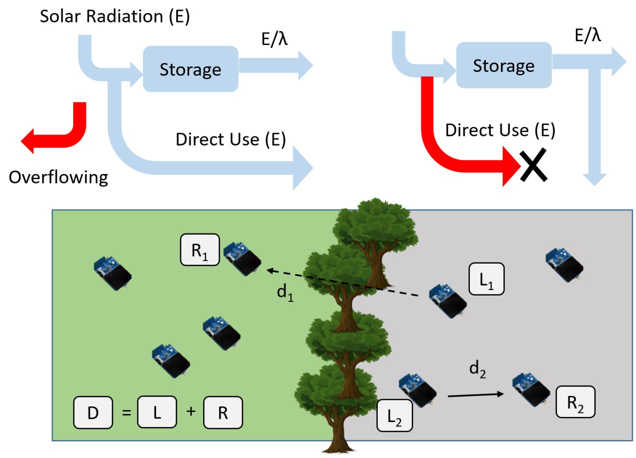

Figure 1: Heterogeneity of energy harvested will be captured by an appropriate data partitioning and in-network computation offloading.

such a scenario could partially perform some pre-processing or basic functional tasks such as averaging or compressing data. Balancing energy usage with computation offloading is important in such a Fog resource pooling environment due to three perspectives.

(a) Energy harvesting incurs a significant conversion loss while storing energy into a storage device like a battery or a capacitor. It accounts for about 25-35% of the total energy in battery storage and even higher for capacitors [27]. It is, therefore, preferable to use harvested energy directly whenever possible so as to minimize the conversion losses. Accordingly, any computation offloading to a node which is currently on solar power has a safe margin to use some energy to compensate for the communication overheads.

(b) If the amount of harvested energy is low, the system cannot perform both the charging and direct energy use operations together. That is, when the amount of harvested energy (E) is below a threshold (θ) a node must decide to either store the energy or use it directly but not both. Usually, in such situations, the most appropriate action is to store the harvested energy and consume the required energy from the battery. Therefore balancing stored energy within the nodes of a WSN is highly advantageous.

(c) Rechargeable batteries are a costly unit for energy harvesting sensor nodes. Therefore, they may have some limited capacity. Cooperative computing between the sensors is critical in such networks to optimize the energy usage via load balancing and avoid overflow of energy on nodes that are fully-charged with no computation task or energy deficit for others. Therefore, balancing energy consumption without using high capacity batteries is a positive trend in future WSN using energy harvesting.

2.3. Related Work

Mobile computation offloading has been widely researched in the recent years with varied objectives such as energy saving, transparent code migration and scalability. An optimal technique for application partitioning and fair node selection between two homogeneous nodes has been discussed in [24]. Computation offloading in WSN, however, did not gain much attention until *Z. Sheng* et al. [20] proposed optimal application partition and cooperation between two nodes to minimize overall energy consumption. Their work is based on cooperation between battery-powered homogeneous sensor nodes and assumes no selfish node behaviour. A cooperating node selection strategy that balances trade-off between fairness and energy consumption has been discussed.

Meanwhile, energy harvesting sensor nodes are becoming widely deployed and several studies discuss the heterogeneity in harvesting energy [15]. *N. Dang* et al. [6] presents predictive solar energy models for spatial-temporal weather conditions. Authors in [27] proposes a stochastic duty cycling approach to minimize energy consumption by taking into account the heterogeneous energy harvesting sensor networks. In [25], authors discuss the importance of triple optimization of sensing, networking and in-network data processing based on energy harvesting. The authors have implemented an optimization algorithm to recycle wasted energy due to battery overflow in an energy harvesting WSN. In this paper, we extend the work done by [20] and propose an approach to balance the energy in computational sensor network using cooperative computing in energy harvesting networks. We apply this approach for the scenario where certain solar powered sensor nodes are under sunlight while others are obstructed by shadows for a certain duration within a day.

3. MODELLING FOR COOPERATIVE COMPUTATION

In this section, we present our application model, computation and communication energy consumption models, and the micro-solar based energy harvesting model.

3.1. Application Model

In this work, we consider a lightweight analytic application that consists of a set of independent processing tasks to be computed cooperatively between two peer sensor nodes. We use the canonical model used in [28] to capture the essential characteristics of such a task-oriented application. Such tasks are normally arranged in a computational work-flow using a Dynamic Acyclic Graph (DAG) to be scheduled for execution in a distributed computing environment. A single processing task (A) is modelled with input data size (D) and a deadline for application completion (T). The Initiating Node (IN), which may be responsible for sensing the data, divides a single task into two sub-tasks for partial offloading to a target remote peer, referred to as the Cooperating Node (CN). The amount of processing data at the local node is denoted by L and the amount of data that is offloaded to the CN is denoted as R, where $D = L+R$. We assume there are no dependencies between the sub-tasks. For instance, in case of calculating average for a sensing variable, L and R may consist of n_L and n_R samples respectively. Note that, only R amount of input data is offloaded to the CN with no extra amount of code. We also assume that the response or the outcome of the processing sub-task at each node is negligible or locally consumed by another process. In the mentioned average calculation example, the local node will transmit only two values, which is the local average (A_L) and n_L, while CN will transmit its own local average (A_R) and n_R. An aggregation or the destination node will then calculate the overall average using the two responses from IN and CN.

3.2. Computation Energy Model

The energy consumption in embedded processors is dominated by dynamic power and can be regulated by the clock frequency using dynamic voltage and frequency scaling (DVFS) technique. Several attempts have been made to develop a simple and general computation energy estimation model for mobile and embedded processors. According to the literature, the computational energy consumption is proportional to the CPU load of a processor i.e. the number of CPU cycles required. Most of the work, therefore, considers the trade-off between energy (E) and task completion time (T) such that $E.T^{\alpha}$ is a constant for some values of α. In [24], the energy consumption for computing a task locally is calculated using eq. 1, where K (in the order of 10^{-11} starting from ARM to Intel) is called the computation coefficient. The value of K depends on the effective switched capacity (determined by the chip architecture and the clock-frequency), the processing capability of the node, and the application completion probability used in the model in [28]. As evident in eq. 1, a node consumes more energy for short completion deadlines T. A sensor node may, therefore, prefer more delay-tolerant tasks for local computation and offload tasks with large L and small T to a peer sensor node.

$$E_C = \frac{KL^3}{T^2} \tag{1}$$

3.3. Communication Energy Model

When a task is offloaded to another node, the energy used for communication depends on the number of bits transmitted [17]. This is energy consumed by the electronics in the physical layer and depends on the state of nodes - idle, transmit and receive. According to IEEE 802.15.4, energy consumption in the idle state can be neglected and, therefore, total energy consumption depends on the transmission of the number of bits at the sender and the reception of the same bits at the receiver which are equal in value but belong to two different nodes. A task can be scheduled for transmission to another node within one or more time-slots. This scheduling has been modelled using the Markov process based on whether the Additive White Gaussian Model (AWGN) channel state is good or bad. The energy used to communicate b bits within a time-slot t to another computational node depends on the path condition and the distance between the two nodes (represented as channel gain g) and is given by the following equation.

$$e = \frac{(2^b - 1)}{g}$$

According to one-shot channel allocation policy to transmit data task within a single time-slot, the scheduler must send L bits within one time-slot T. This is the simplest case in which all the data is sent within a single time-slot of communication window and the energy consumed is represented by a covex-monomial function as shown in eq. 2.

$$E_t = \rho \frac{L^n}{g} \tag{2}$$

Here, ρ is the communication coefficient of the link between the offloader and the offloadee and $g[0...1]$ is the channel gain of the link that is calculated proportional to $1/d^2$ according to AWGN in free-space propagation where d is the distance between the two nodes. According to [20], transmission in one-shot policy ($n = 1$) only depends on the channel state and it is the most optimal approach for latency-sensitive applications. It also minimizes the time shift between local and remote computation since it assumes a negligible delay in over-the-air transmission. Moreover, it saves energy that is otherwise incurred by overhead scheduling due to data split across multiple time-slots.

3.4. Total Energy Requirement Calculation per Task

The total energy consumption owing to computation and communication during processing a task between two nodes can be calculated as the summation of four components as shown in eq. 3. In [20], authors present the energy consumption for different input data sizes from 512 to 2048 bits. Here the job completion deadline is set to 20ms, $K = 5 \times 10^{-11}$ and $\rho = 0.05$. For large data sizes, the gain in energy consumption is much better in case of using cooperative computing and varies with the values of the computation and communication coefficients. After a distance of $5m$, however, cooperative computing is not effective and localized computation becomes the preferred mode for the entire task according to their analysis.

$$\begin{aligned} Total\ Energy(E) = {} & IN\{Computation\ L + Transmission\ R\} \\ & + CN\{Reception\ R + Computation\ R\} \end{aligned} \tag{3}$$

In this paper we extend this approach by taking into account the energy harvesting state of the IN and CN nodes and also the energy conversion efficiency. We estimate the required equivalent energy (E) (i.e. before the conversion) from the energy harvesting source within the optimization algorithms.

3.5. Micro-solar Energy Harvesting Model

We selected a latitude of 52^o and longitude of -8^o where the experimental smart farm for the project is located in Moorepark, Co. Cork, Ireland. We chose April. 1^{st} as the representative date of nigher a winter day nor a summer day for the solar energy harvesting model. We model the solar energy harvesting pattern as a Gaussian curve with 8 hours (T) clear sunlight from 8.00am to 4.00pm according to astronomical model developed by [23], [12]. We consider a discrete time model with a time-slot of 1 minute. A solar energy density of $15mWcm^{-3}$ is assumed for $5 \times 3cm$ area on a micro-solar panel associated with a sensor node. This implies $735\mu J$ energy can be generated by a sensor node on a day without any clouds and obstacles shadowing it. We also modeled a shadow

$$\cos\theta = \cos\alpha_p.\cos\alpha_s + \sin\alpha_p.\sin\alpha_s.\cos(\beta_p - \beta_s)$$

$$\cos\alpha_s = \sin\delta.\sin L + \cos\delta.\cos L.\cos h \qquad \sin\beta_s = -\cos\delta.\sin h/\sin\alpha_s$$

$$x = 2\pi n/365, \qquad h = 15(t - 12)$$

$$\delta = 0.302 - 22.93\cos x - o.229\cos 2x + 0.243\cos 3x + 3.851\sin x + 0.002\sin 2x - 0.055\sin 3x$$

Figure 2: The set of used energy harvesting astronomical modelling equations.

of 4 hours which will randomly cover sensor nodes within the field. Micro-solar panel inclination was set to 90^o and orientation to 45^o in our model.

4. ENERGY-AWARE TASK PARTITIONING

The aim of this work is to find the optimal data size for a task that is suitable for local computation (L) and remote computation (R) based on the state of harvested energy (under shadow, under sunlight with energy stored being under-flown and under sunlight with energy stored being over-flown) on both IN and CN. While we discuss the energy-aware application partitioning by IN and CN selection (in the following section), the energy state interchanges among the nodes using a distributed or a centralized approach is beyond the scope of this paper. The Lagrange Multiplier is used to solve the equal constrained optimization problem with an objective to minimize total required energy (E) from the solar panel at both nodes. When a task is to be processed at any given time, IN and CN may be in different states as shown in TABLE 1, resulting into different E_L and E_R values compared to non-energy harvesting-aware partitioning approach proposed by [20]. We calculate E accordingly as the summation of E_L and E_R values. We consider an energy gain factor λ as the reciprocal of the energy conversion efficiency in the equations for simplicity of deriving equations. For instance, λ=1.54 represents 65% efficiency and implies that if a task consumes $10\mu J$ stored energy from the battery when the node is under a shadow, the value of E will be $20\mu J$.

TABLE 1: Different energy harvesting states at IN and CN and the amount of total required energy in μJ using our energy-harvesting-aware task partitioning at $T = 20ms$. Local computing respectively consumes and computes $41.3\mu J$ (1024 bit)

CN \ IN		Shadow	Light	
			Underflow	Overflow
Shadow		12.1(526)	9.8(469)	0(1024)
Light	Underflow	9.8(472)	7.8(526)	0(1024)
	Overflow	1.6(122)	1.1(122)	0(1024)

In the following sub-sections, we discuss the optimal task partitioning in terms of number of bits and the total energy required at both IN and CN to execute the task in μJ under the different IN and CN states (TABLE 1). The data size (D) is set to 1,024 bits and the task completion deadline is changed from $5 - 100$ms. The channel gain between IN and CN is set to

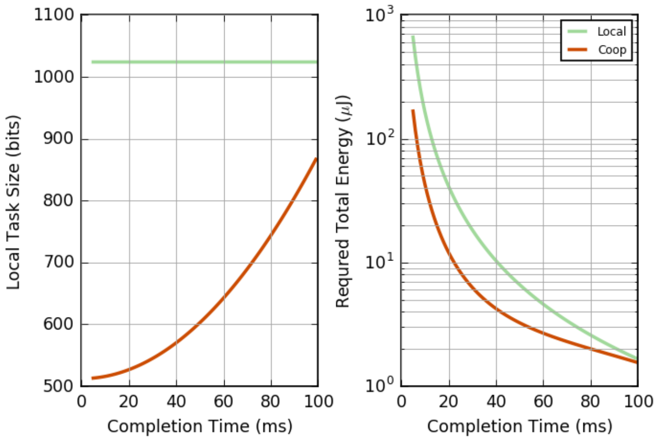

Figure 3: Cooperative computing gains with low energy when both nodes are under shadows. However, it does not gain any energy saving when completion deadline is larger than 100ms.

0.9 and the values of K and ρ are 10^{-11} and 10^{-3} respectively. Energy gain factor λ=1.54.

4.1. Shadow-Shadow

When IN and CN are under shadow, both nodes consume energy from the stored battery power for task processing. Such a scenario does not incur any waste from the harvested energy. In this case, E can be calculated as the sum of local energy requirement E_L at IN (for computation of local task L and transmission of data R to CN) and remote energy requirement E_R at CN (for reception of data R from IN and computation of data R).

$$E = E_L + E_R = \{\alpha L^3 + \beta R\}\lambda + \{\beta R + \alpha R^3\}\lambda \tag{4}$$

On solving eq. 4 using Lagrange constraint optimization in order to minimize E subjected to the constraint $L + R = D$, we obtain the values for L and R.

$$L = \frac{D}{2} + \frac{\beta}{3\alpha D} \quad \text{and} \quad R = \frac{D}{2} - \frac{\beta}{3\alpha D}$$

Even though the amount of task partition is the same as in the non-energy harvesting case, the energy requirement is multiplied by the energy gain factor λ when we calculate the amount of surplus energy to be stored at each node. Fig. 3 shows that cooperative computing gains with low energy and the amount of the locally computed data increase with the task completion deadline. After a certain time of completion deadline, however, IN processes all the data locally and does not achieve any advantage by cooperating with a CN.

4.2. Shadow-Light

In this case, the CN is under sunlight while energy is being harvested during the task processing. Therefore, remote computation R tends to be larger than in the previous case since energy required at the CN can be consumed directly from the energy harvesting source without incurring any conversion loss, if the battery is underflow (not charged up to the full capacity). Furthermore, it can use abundant energy if the battery overflows (battery fully charged and harvesting energy being wasted). Accordingly, we analyze this case separately for the two scenarios as the amount of L and R will be different.

Energy under-flowing: In this scenario, the energy is directly used from the solar panel at CN through the input regulator without incurring battery conversion loss. However, any surplus harvested energy can be stored in the CN battery without contributing towards energy waste as the battery is not charged to the full capacity. Therefore, E can be calculated as follows.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\}\lambda + \{\beta R + \alpha R^3\} \tag{5}$$

On solving eq. 5 to minimize E, we obtain the values for L and R as given below, where the value of A is obtained by solving the quadratic equation $aA^2 + bA + c = 0$ (see Appendix A) such that $L < D$.

$$L = \sqrt{\frac{A}{3\alpha\lambda)}} \quad \textit{and} \quad R = \sqrt{\frac{A - (1+\lambda)\beta}{3\alpha)}}$$

Furthermore, values of a, b and c are calculated as follows.

$$\begin{aligned} a &= (1-\lambda)^2 \\ b &= 2\lambda(1+\lambda)\{(1-\lambda)\beta - 3\alpha D^2\} \\ c &= \lambda^2[9\alpha^2 D^4 + \beta(1+\lambda)\{6\alpha D^2 + (1+\lambda)\beta\}] \end{aligned}$$

Energy over-flowing: If the battery at CN is fully charged, the energy required at CN is not considered for the total energy requirement calculation since CN in this case is wasting the harvested energy. However, transmission energy used for offloading data R to CN should be considered in the energy consumed at IN, which prevents offloading all the data D to CN.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\}\lambda + \{0\} \tag{6}$$

On solving eq. 6, we obtain $L = \sqrt{(\beta/3\alpha)}$ which is a trade-off between the required computation and communication energy at IN, and $R = D - L$. This shows that even though harvested energy at CN is wasted, IN cannot offload all the task to CN unless the completion deadline is very low.

As illustrated in Fig. 4, IN offloads more data to the CN when CN is under sunlight. We can see that if CN is overflowing, more computation can be offloaded than in the case of CN under-flowing. In case of the former, significant energy gain is observed for lower task completion deadlines when compared to the local computation only.

4.3. Light-Shadow

When IN is under sunlight, the size of local computation L tends to be larger than in the previous case. This is because energy consumed at the IN can be used directly from the energy harvesting source without incurring conversion loss or from the energy being wasted according to the level of charge of the battery (similar to the previous case). Therefore, this case is also investigated under two scenarios where the amount of L and R is different.

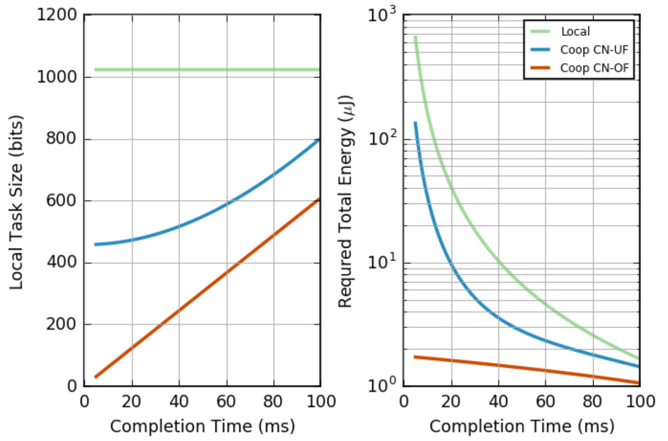

Figure 4: IN offloads more data to CN when it is under sunlight. CN over-flowing can achieve much lesser total energy consumption than underflowing scenario.

Energy under-flowing: In this scenario, energy is directly used without conversion loss but harvested energy can be stored in the IN battery rather than being wasted. Therefore, E can be calculated as follows.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\} + \{\beta R + \alpha R^3\}\lambda \tag{7}$$

On solving the optimization problem, we obtain the values for L and R as under.

$$L = \sqrt{\frac{A}{3\alpha}} \quad \text{and} \quad R = \sqrt{\frac{A - (1+\lambda)\beta}{3\alpha\lambda}}$$

The value of A can be obtained by solving the quadratic equation $aA^2 + bA + c = 0$ such that $L < D$ using the following values of a, b and c.

$$\begin{aligned} a &= (1-\lambda)^2 \\ b &= (1+\lambda)\{(\lambda-1)2\beta - 6\alpha\lambda D^2\} \\ c &= 9\alpha^2\lambda^2 D^4 + (1+\lambda)\beta\{6\alpha\lambda D^2 + (1+\lambda)\beta\} \end{aligned}$$

Energy over-flowing: In this scenario, the energy required at IN is not considered for the total required energy calculation since the node is wasting the harvested energy. Furthermore, all the computation is done locally at IN rather than offloading partial computation to CN. Accordingly, $E = E_L + E_R = 0 + 0$ and we obtain $L = D$ and $R = 0$. Fig. 5 shows that cooperative computing gains when IN is under sunlight.

4.4. Light-Light

This case results in three possibilities for deciding the values of L and R. The calculation of the total required energy for each scenario is explained below.

Both nodes energy under-flowing: When both IN and CN are under sunlight without energy over-flowing, nodes can consume energy directly from the energy source and store surplus energy in the battery without any waste. In this case, E can be calculated as shown in eq. 8, and the values of L and R can be calculated as in the shadow-shadow scenario in section 4.1

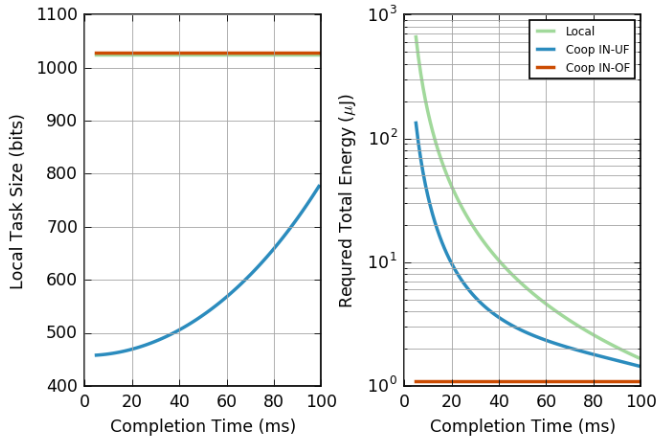

Figure 5: Overflowing IN does not offload any data to a CN. However, under-flowing IN offloads data in cooperative computing.

(however the energy required at each node will be differed by a factor of λ).

$$E = E_L + E_R = \{\alpha L^3 + \beta R\} + \{\beta R + \alpha R^3\} \tag{8}$$

On solving the eq. 8 to minimize E subject to the condition $L + R = D$, we can obtain the values for L and R as under.

$$L = \frac{D}{2} + \frac{\beta}{3\alpha D} \quad \textit{and} \quad R = \frac{D}{2} - \frac{\beta}{3\alpha D}$$

IN energy over-flowing: In this scenario, all the processing takes place locally at the IN irrespective of the CN state and the energy required at IN is not considered for the total energy calculation. Therefore, total energy is calculated as $E = E_L + E_R = 0 + 0$ and we obtain the $L = D$ and $R = 0$.

IN under-flowing and CN over-flowing: If the battery at CN is fully charged, the energy required at the CN is not considered for the total energy (E) calculation since CN, in this scenario, will waste the harvested energy. However, energy used for offloading data R to CN must be considered as the energy consumed at IN. Accordingly, total energy is calculated as given in eq. 9.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\} + \{\theta\} \tag{9}$$

We then obtain the value of $L = \sqrt{\beta/3\alpha}$ which is a trade-off between the required computation and communication energy at IN, and $R = D - L$. This shows that again even though harvested energy at CN is wasted, IN cannot offload all the data to CN. Also, the energy required by IN does not incur any conversion loss. Fig. 6 shows the gain in cooperative computing in these scenarios. As in the previous case, IN does not offload any data to CN in case of energy over-flowing whereas it offloads a considerable amount of data to CN when CN is over-flowing.

5. ENERGY-AWARE NODE SELECTION STRATEGY

The CN selection strategy must also be modified to make it suitable for our application model compared to non-energy harvesting scenario. In case of a non-energy harvesting environment, the minimum total energy strategy (MES), where the CN

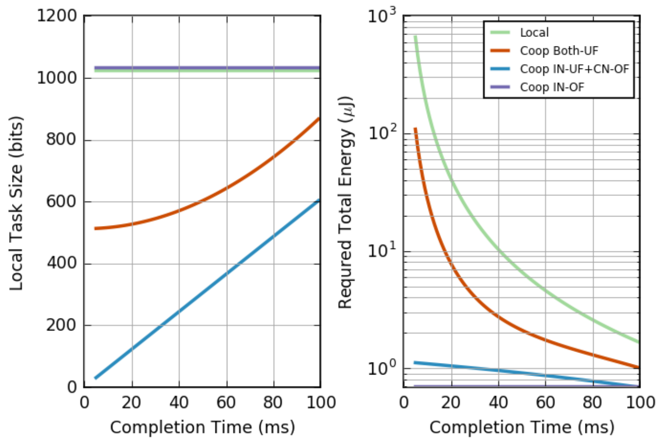

Figure 6: IN does not offload any data when it is overflowing. However, IN does not offload all the data when CN is overflowing due to communication energy used at the IN.

with minimum total cooperative energy cost is selected among the set of neighbouring nodes. This strategy does not consider past energy consumption (i.e. utilization). The drawback of it is that some nodes are overused due to cooperation and may lead to reduced battery lifetimes or many dead batteries, which affect the long-term autonomous functioning of the WSN. For example, a node in close proximity to a computationally-intensive node may cooperate heavily and may, therefore, be overused unfairly than what they save from cooperative computing.

In this work, CN selection is performed based on utility function as in [20], where authors define a utility function (U) based on the energy saved from cooperative computing compared to executing the complete data task locally at an IN. Our simplified utility function incorporating with the energy gain factor is given as follows.

$$U = \begin{cases} E_{LO} - E_L & \text{if IN} \\ -\gamma E_R & \text{if CN} \end{cases}$$

Here $E_{LO} = \frac{KD^3}{T^2}$ and $\gamma = 1$ if the CN is under sunlight and $\gamma = \lambda$ if CN is under shadow and under-flowing. The value of $\gamma = 0$ if the CN is over-flowing energy. Utility of IN will not change as the impact of the sunlight is already calculated in the required energy optimization. A Cooperation Index (CI) is then defined based on the cumulative utility as given below for $t = 0$ to $t - 1$ same as in [20]. A node can be used as a CN at time t if and only if the value of CI is positive.

$$CI = \begin{cases} 1 & \text{if } U(0:t-1) \geq 0 \\ 0 & \text{if } U(0:t-1) < 0 \end{cases}$$

This strategy is called positive utility strategy (PUS) [20]. Larger utility will have a higher chance to be selected as a CN. Designing an algorithm for this process based on the harvested energy (either in the past or predicted) is beyond the scope of this paper and remains as our future work.

6. PERFORMANCE EVALUATION

We simulated our energy harvesting-aware computation offloading algorithm (e-COFF) with 30 energy harvesting sensor nodes using the *SimGrid* simulator[1]. Nodes were randomly located within a $10m \times 10m$ geographical space. We selected latitude of 53^o, where the project site is located and day of the year as 91 (01^{st} April) in the micro-solar energy harvesting model, which harvested energy in a sinusoidal pattern within a day. We used randomly distributed obstacles for shadowing for a duration of 4 hours. The size of the solar panel at a node was selected as $5 \times 3cm$, which determined the multiplication factor of the sinusoidal harvesting pattern. We update the stored and wasted energy at each node per minute based on the harvested and the consumed energy during that period. We compared our results with non-energy harvesting-aware data offloading algorithm (COFF).

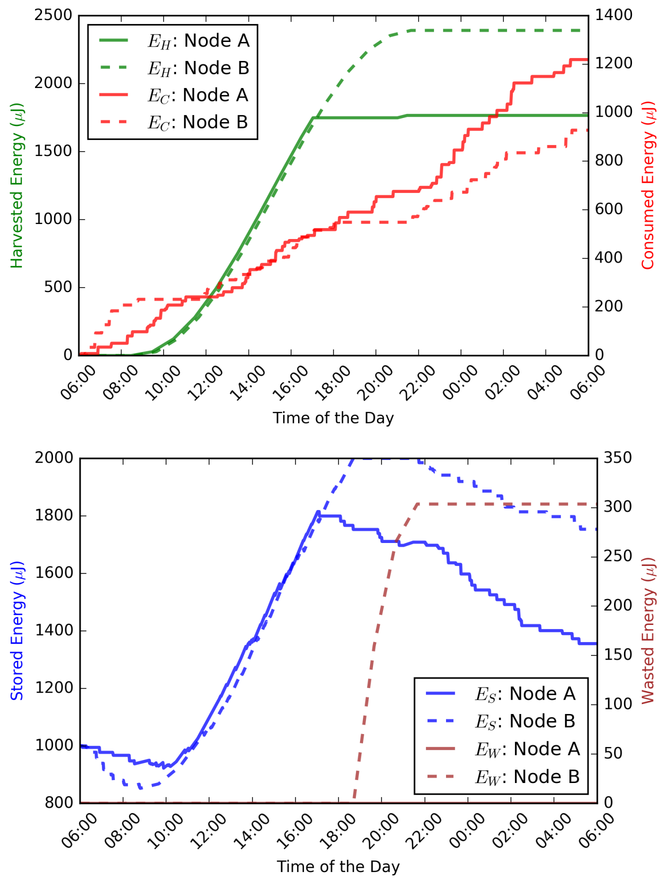

Figure 7: The amounts of measured energy performance parameters at two different energy harvesting sensor nodes for duration of 24 hours. Full battery capacity = 2000 mAh.

A computational task was created every $2s$ randomly by a selected senor node in the WSN with a size (D) of $1,024 bits$. We used a maximum capacity for a battery storage of a sensor node as 2000 mAh and set it to its half at the start of the day. Harvested (E_H), required to consume (E_C), stored (E_S) and wasted (E_W) energy at the end of 24 hour duration from 6.00am were measured. Task completion time (T), harvesting energy gain factor (λ), K and ρ were normally set respectively as 20ms, 1.54, 10^{-11} and 0.001 unless otherwise it was changed

[1] http://simgrid.gforge.inria.fr

in some sections. We calculated channel gain (g) according to the free-space wave propagation of AWGN as,

$$g = \frac{1}{\sigma\sqrt{2\pi}} exp^{\frac{-d^2}{2\sigma^2}}$$

where we selected σ as 8 in our simulations and d was calculated in the units of m.

As shown in Fig.7, the harvested energy (E_H) of Node B does not experience any shadow while Node A experiences shadow during the day. Moreover, Node A demands slightly more energy (i.e. required energy (E_C) for task executions either as an IN or CN before being converted) than Node B. As we can see in the bottom graph, Node B saturated with stored energy (E_S) from 6.00pm to 8.00pm resulting in a waste of energy (E_W). Node A's battery capacity does not overflow at any given time and therefore does not experience any waste of energy. This validates our chosen relative values of energy performance parameters in order to fulfill a requirement of self-sustainability of the wireless sensor network.

We then observe the probability distribution of the offloaded task sizes to a CN (R) and the end of the day stored energy (E_S) for the two algorithms; e-COFF and COFF. The top and the bottom graphs of Fig.8 shows the cumulative probability densities of R and (E_S) respectively with 30 different seed values set in the simulator. As we can see e-COFF offloads more data to a CN than the COFF algorithm does. The second figure shows COFF leaves with more sensor nodes towards lower energy levels at the end of the day while e-COFF leaves more stored energy towards higher energy levels.

Fig.9 shows the difference between the consumed energy of COFF and e-COFF (E_C of COFF - E_C of e-COFF). We have changed the computation coefficients (K) in the range of 10^{-11} to 10^{-10} and the communication coefficient (ρ) in the range of 0.01 and 0.001 both with a step size of 0.1. According to the figure, the performance improvements of the e-COFF is apparent for all the values of computation and communication coefficients since all the values in figure are positive. When both K and ρ are higher (top-left corner), performance improvement is significant.

Next, we change (top graph in 10) the energy gain factor (λ) from 2.5 to 1.0 (i.e. energy conversion efficiency from 0.4 to 1.0) with a step of 0.5 while keeping T at 20ms. In another experiment we also change task completion deadline (bottom graph in 10) from 5ms to 30ms with a step size of 5ms while keeping λ at 1.54. Figures show the consumed energy during the day and the stored energy at the end of the day. According to the figure at the top, e-COFF shows lesser (E_C) than the COFF. Our algorithm also shows that stored energy performance is also higher compared to COFF. Performance improvement of e-COFF is much better when energy gain factor λ is low. However, the performance improvement is not very apparent for the changing range of task completion deadline (T).

We then localize the task generations only to a subset of sensor nodes to investigate the adverse impact of the overuse of energy at a CN. In this case, we reduced the number of task originating nodes from 30 (all, which is the same as before)

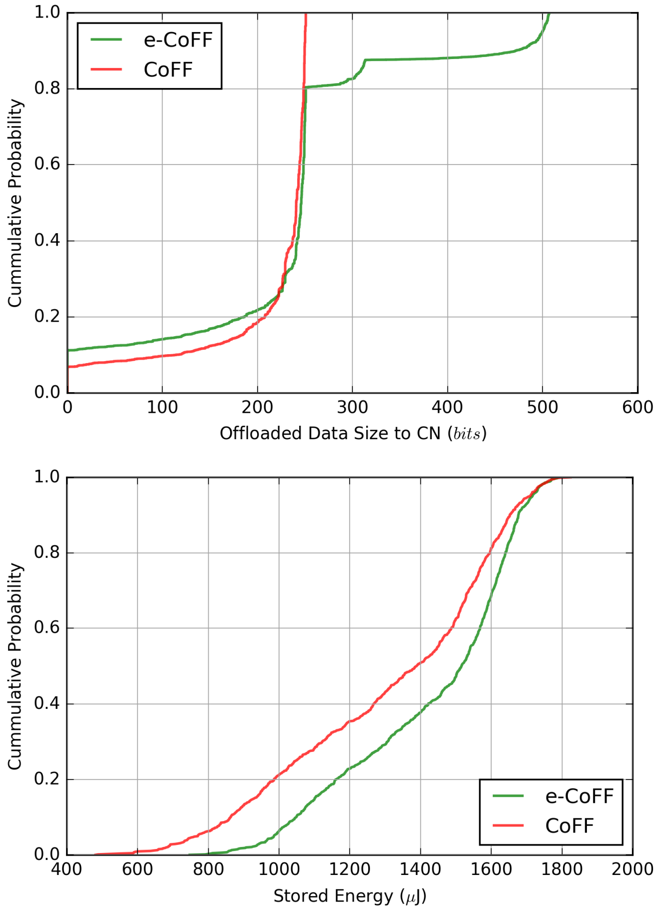

Figure 8: Top: CDF of the sizes of data chunks being offloaded to a remote CN (R). Bottom: CDF of the stored energy (E_S) at the end of the day. K=10^{-11}, ρ = 0.001, D = 1024 bits, T = 20ms.

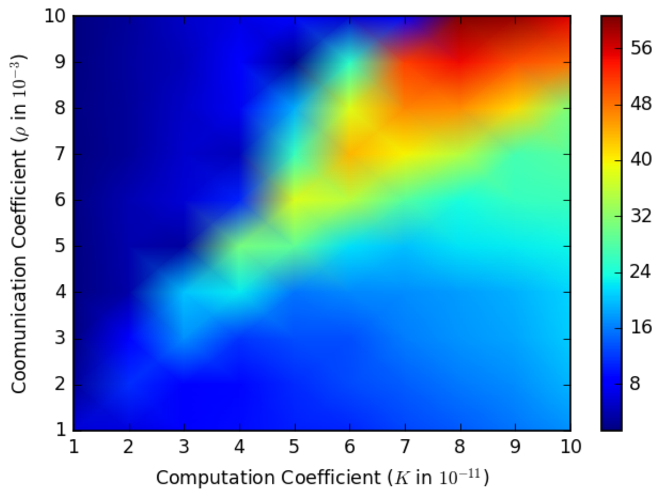

Figure 9: The difference of consumed energy (E_C) in mJ between the energy-unaware (COFF) and our energy-aware (e-COFF) data partitioning and computation offloading algorithms (task completion deadlines = 20ms). Both used Positive Utility Strategy (PUS) in selecting a CN.

to 5 with a step size of 5. We used two CN selection strategies; MES and PUS, with our e-COFF algorithm. Fig.11 shows the standard deviation of the end of the day stored energy E_S,

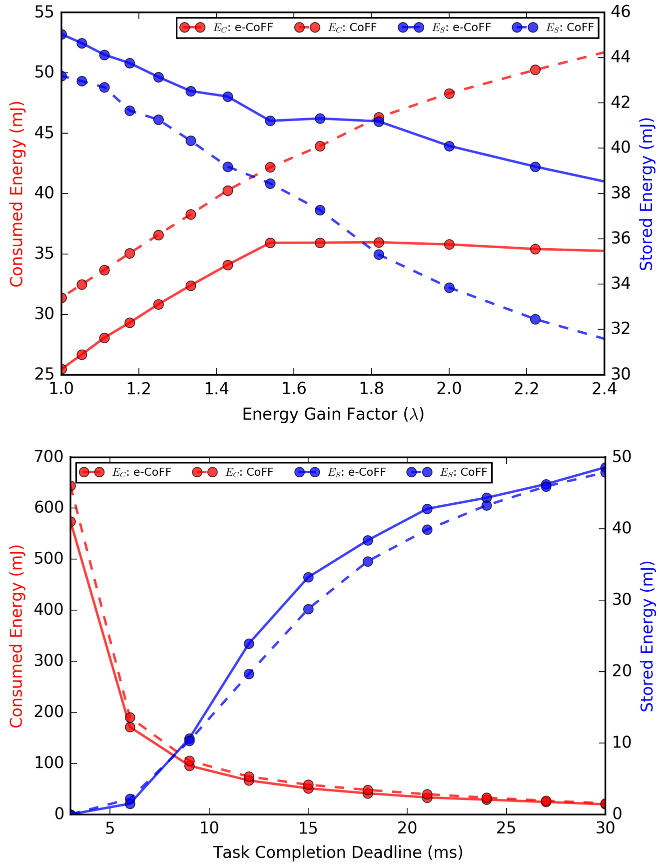

Figure 10: Consumed and stored energy of the two algorithms for different energy gain factors (λ) when t=20ms (top) and for different task completion deadlines when λ=1.54 (bottom).

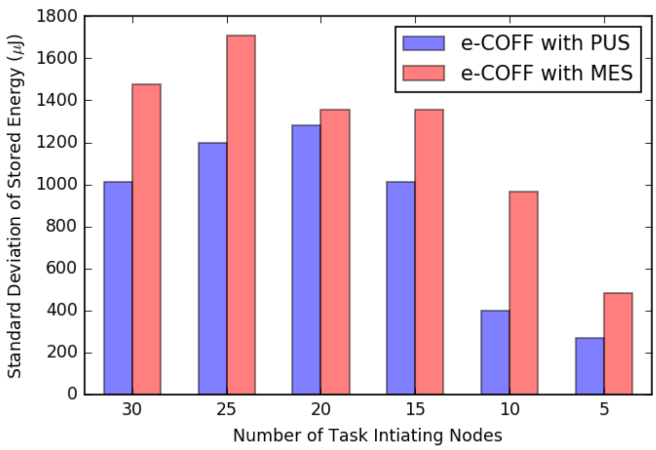

Figure 11: The standard deviation (STD) of the stored energy (E_S), where a smaller STD indicates a better energy balance, of 30 micro-solar energy harvesting sensor nodes at the end of the day (completion deadline=20ms and λ=1.54).

which is lower with the PUS strategy. It shows that the impact using the utilization factor in micro-energy harvesting where, if energy level of a node is low, becoming a CN persistently is critical. According to the figure use of CI solves the problem of overuse of CNs by INs in a computation intensive hotspots.

7. CONCLUSIONS

Energy-aware cooperative computing is a key technology that will benefit from energy harvesting in Fog computing applications. It is particularly important when the energy harvesting patterns and obstructions are dynamic, thereby, creating spatially heterogeneous energy sources. In this paper, we extend the optimal data partitioning algorithms developed for computation offloading by taking into account the state of the energy being harvested at the heterogeneous nodes. We evaluate our e-COFF algorithm under different scenarios and compare with COFF algorithm. Our results illustrate that overall energy consumption can be improved in a WSN by minimizing energy losses due to a poor energy conversion efficiency and waste due to energy overflows under constrained energy storage capacities. Our algorithm preformed the optimized data portioning with a positive utility cooperating node selection strategy, which balances the stored energy of the sensor nodes at the end of a day, which is useful concern for the sustainability of a WSN using micro-scale energy harvesting sources.

ACKNOWLEDGEMENT

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (TEAGASC) as part of the SFI TEAGASC Future Agri-Food Partnership, in a project [13/IA/1977] titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems".

Appendix A.

In this appendix, we discuss the optimal data partitioning for a scenario where the Initiating Node (IN) is under shadow while the Cooperating Node (CN) is under sunlight. The energy required by IN is obtained directly from the harvested energy whereas the energy required by CN is obtained from the battery. The total energy consumed is calculated as follows.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\} + \{\beta R + \alpha R^3\} + A\{D - L - R\}$$

Using gradient optimization with partial derivatives, we get

$$\frac{\partial E}{\partial L} = 3\alpha L^2 - A \rightarrow L^2 = \frac{A}{3\alpha}$$
$$\frac{\partial E}{\partial R} = \beta + \lambda\beta + 3\alpha R^2 - A \rightarrow R^2 = \frac{A - (1+\lambda)\beta}{3\alpha}$$

After solving the equation $(L + R)^2 = D^2$, we get a quadratic equation $aA^2 + bA + c = 0$ to find the roots for A.

References

[1] Baccarelli, E., Naranjo, P., Scarpiniti, M., Shojafar, M., Abawajy, J., 2017. Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study. IEEE Access 5, 9882–9910.

[2] Barbarossa, S., Sardellitti, S., Lorenzo, P. D., 2014. Communicating while computing: Distributed mobile cloud computing over 5g heterogeneous networks. IEEE Signal Processing Magazine 31 (6), 45–55.
[3] Bhargava, K., Ivanov, S., Kulatunga, C., Donnelly, W., 2017. Fog-enabled wsn system for animal behavior analysis in precision dairy. In: IEEE International Conference on Computing, Networking and Communications (ICNC).
[4] Chen, M., Hao, Y., Li, Y., Lai, C. F., Wu, D., 2015. On the computation offloading at ad hoc cloudlet: architecture and service modes. IEEE Communications Magazine 53 (6), 18–24.
[5] Conti, M., Kumar, M., 2010. Opportunities in opportunistic computing. IEEE Computer Magazine 43 (1).
[6] Dang, N., Bozorgzadeh, E., Venkatasubramanian, N., 2012. Quares: A quality-aware renewable energy-driven sensing framework. Sustainable Computing: Informatics and Systems 2 (4), 171–183.
[7] Eto, M., Katsuma, R., Tamai, M., Yasumoto, K., 2015. Efficient coverage of agricultural field with mobile sensors by predicting solar power generation. In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications. pp. 62–69.
[8] Giridhar, A., Kumar, P. R., 2006. Toward a theory of in-network computation in wireless sensor networks. IEEE Communications Magazine 44 (4), 98–107.
[9] Grogan, A., 2012. Smart farming. IET Engineering & Technology Magazine 7 (6).
[10] Hinton, F. J. K., Ayre, R., Alpcan, T., Tucker, R. S., 2016. Fog computing may help to save energy in cloud computing. IEEE Journal on Selected Areas in Communications 34 (5), 1728–1739.
[11] Jeon, S.-W., Jung, B. C., 2016. Opportunistic function computation for wireless sensor networks. IEEE Transactions on Wireless Communications 15 (6).
[12] Jeong, J., Culler, D., 2012. A practical theory of micro-solar power sensor networks. ACM Transactions on Sensor Networks (TOSN) 9 (1), 1–36.
[13] Jeong, S., Simeone, O., Kang, J., 2017. Mobile cloud computing with a uav-mounted cloudlet: optimal bit allocation for communication and computation. IET Communications 11 (7), 969–974.
[14] Khan, I., Belqasmi, F., Glitho, R., Crespi, N., Morrow, M., Polakos, P., 2015. Wireless sensor network virtualization: early architecture and research perspectives. IEEE Network 29 (3), 104–112.
[15] Ku, M. L., Chen, Y., Liu, K. J. R., 2015. Data-driven stochastic models and policies for energy harvesting sensor communications. IEEE Journal on Selected Areas in Communications 33 (8), 1505–1520.
[16] Kulatunga, C., Shalloo, L., Donnelly, W., Robson, E., Ivanov, S., 2017. Opportunistic wireless networking for smart dairy farming. IEEE IT Professional Magazine 19 (2), 16–23.
[17] Lee, J., Jindal, N., 2009. Energy-efficient scheduling of delay constrained traffic over fading channels. IEEE Transactions on Wireless Communications 8 (4), 1866–1875.
[18] Lewis, G. A., Echeverra, S., Simanta, S., Bradshaw, B., Root, J., 2014. Cloudlet-based cyber-foraging for mobile systems in resource-constrained edge environments. In: Companion Proceedings of the 36th International Conference on Software Engineering.
[19] Mtibaa, A., Harras, K. A., Habak, K., Ammar, M., Zegura, E., 2015. Towards mobile opportunistic computing. In: IEEE Conference of Cloud Computing.
[20] Sheng, Z., Mahapatra, C., Leung, V., Chen, M., Sahu, P., 2015. Energy efficient cooperative computing in mobile wireless sensor networks. IEEE Transactions on Cloud Computing PP (99), 1–1.
[21] Stojmenovic, I., 2014. Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems. IEEE Internet of Things Journal 1 (2), 122–128.
[22] Sudevalayam, S., Kulkarni, P., 2011. Energy harvesting sensor nodes: Survey and implications. IEEE Communications Surveys Tutorials 13 (3), 443–461.
[23] Taneja, J., Jeong, J., Culler, D., 2008. Design, modeling, and capacity planning for micro-solar power sensor network.
[24] Vaquero, L. M., Rodero-Merino, L., 2014. Finding your way in the fog: Towards a comprehensive definition of fog computing. ACM SIGCOMM Computer Communication Review 44 (25).
[25] Yang, S., Tahir, Y., y. Chen, P., Marshall, A., McCann, J., 2016. Distributed optimization in energy harvesting sensor networks with dynamic in-network data processing. In: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications. pp. 1–9.
[26] Zhang, B., Mor, N., Kolb, J., Chan, D. S., Lutz, K., Allman, E., Wawrzynek, J., Lee, E., Kubiatowicz, J., 2015. The cloud is not enough: Saving iot from the cloud. In: 7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15). pp. 12–21.
[27] Zhang, J., Wang, M., Li, Z., 2015. Stochastic duty cycling for heterogenous energy harvesting networks. In: 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC). pp. 1–9.
[28] Zhang, W., Wen, Y., Guan, K., Kilper, D., Luo, H., Wu, D. O., 2013. Energy-optimal mobile cloud computing under stochastic wireless channel. IEEE Transactions on Wireless Communications 12 (9), 4569–4581.

Chamil Kulatunga is a postdoctoral researcher in Telecommunications Software and Systems Group (TSSG), Waterford Institute of Technology (WIT), Ireland. His research interests include Distributed Analytics, Fog Computing and Smart Agriculture. Contact him at ckulatunga@tssg.org.

Kriti Bhargava is a PhD student in TSSG, WIT, Ireland. Her research interests include Internet of Things, Fog Computing and Data Mining. Contact her at kbhargava@tssg.org.

Dixon Vimalajeewa is a PhD student in TSSG, WIT, Ireland. His research interests include Distributed Learning Algorithms, Mathematical Modelling and Data Mining. Contact him at dvimalajeewa@tssg.org.

Stepan Ivanov is a postdoctoral researcher in TSSG, WIT, Ireland. His research interests include Wireless Communications, Internet of Things, Fog Computing and Edge Analytics. Contact him at sivanov@tssg.org.