


## Article

# Zero-Energy Computation Offloading with Simultaneous Wireless Information and Power Transfer for Two-Hop 6G Fog Networks

Daniele Tarchi <sup>1,\*</sup> , Arash Bozorgchenani <sup>2</sup>  and Mulubrhan Desta Gebremeskel <sup>1</sup>

<sup>1</sup> Department of Electrical, Electronic and Information Engineering “Guglielmo Marconi”, University of Bologna, 40136 Bologna, Italy; mulubrha.gebremeskel@studio.unibo.it

<sup>2</sup> School of Computing and Communications, Lancaster University, Lancaster LA14WA, UK; a.bozorgchenani@lancaster.ac.uk

\* Correspondence: daniele.tarchi@unibo.it

**Abstract:** Currently, we are faced with an ever-increasing number of devices and *objects* connected to the Internet aimed at creating the so-called *Internet of Things* framework, fostering the creation of a connected world of objects. One of the main challenges we are actually facing is constituted by the constrained sizes of such objects: reduced memory, reduced computational capacity, and reduced battery sizes. Particular attention should be devoted to energy efficiency, since a potential energy shortage would negatively impact not only its operation but also network-wide operation, considering the tight connections among any object. According to the 6G system’s use-case related to self-sustainability and zero-energy networks, this paper focuses on an energy-efficient fog network architecture for IoT scenarios, jointly implementing computation offloading operations and simultaneous wireless information and power Transfer (SWIPT), hence, enabling the possibility of jointly transferring energy and computational tasks among the nodes. The system under consideration is composed of three nodes, where an access point (AP) is considered to be always connected to the power network, while a relay node and an end node can harvest energy from the AP. The proposed solution allows to jointly optimize the computation offloading and the energy harvesting phases while maximizing the network lifetime, so as to maximize the operational time of the network. Numerical results obtained on MATLAB demonstrate that the proposed algorithm performs better than the other benchmarks considered for comparison.

**Keywords:** zero energy devices; energy harvesting; wireless power transfer; fog computing; computation offloading



**Citation:** Tarchi, D.; Bozorgchenani, A.; Gebremeskel, M.D. Zero-Energy Computation Offloading with Simultaneous Wireless Information and Power Transfer for Two-Hop 6G Fog Networks. *Energies* **2022**, *15*, 1632. <https://doi.org/10.3390/en15051632>

Academic Editor: Sangheon Park

Received: 27 January 2022

Accepted: 19 February 2022

Published: 22 February 2022

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the modern-day digital era, the number of wireless devices becoming part of the connected world is ever increasing. Most of the wireless devices are small in size, with limited computational capacity, and with a limited energy source [1]. When these devices have some intensive jobs to process, due to their limited computing capacity they require a long processing time, which will indirectly consume a huge amount of energy. The problem can be addressed by means of computation offloading, i.e., by transferring computation-intensive jobs to a more powerful remote device [2]. Currently, cloud computing is widely used for these aims [3]. Despite the superior services that cloud computing provides to end-users, it has a very serious limitation: the connectivity between the end devices and the cloud is over the Internet, and the distance between them can be thousands of kilometers. This cripples the communication for delay-sensitive cloud-based applications, such as connected vehicles, fire detection and firefighting, smart grid, and content delivery applications.

Fog computing is a term coined by Cisco [4], which refers to bringing the resources and services originally provided by cloud computing closer to the user-end devices, including computation, networking services, and storage [5]. The ever-increasing number of smaller devices, with reduced computational speeds, smaller memory size, and power constraints demand a networking scheme that provides these devices all the services they need. These devices can be positioned anywhere, e.g., in factories, in water-bodies, alongside a railway track, in a vehicle, or in an oil refinery plant, with the only requirement being a network connection. As its name implies, *fog* is a cloud that is closer to the ground. Likewise, fog computing brings all the functionalities of cloud computing to the edge of the network. As noted in the literature [3,6], cloud computing is not a good choice for IoT applications, and hence fog could be used as an alternative.

IoT can be seen as a framework enabling the interconnection of *any* object, provided that each object can be uniquely identified and that no human intervention is required, so as to enable an effective object-to-object interaction. The IoT is the interconnection of various devices that are communicating with each other on the web, all collecting and sharing data. The advent of powerful integrated circuits (ICs), and the ubiquity of wireless networks, enabled the creation of *things* varying from the smallest ant-sized objects up to objects as big as an airplane. Adding sensors to these devices makes them intelligent and able to share real-time data without the intervention of humans. The word "thing" is used because virtually anything in the physical world can be connected to the Internet, exploiting connected sensors to collect data, and communication protocols to transceive data so that the *thing* becomes an IoT device.

IoT has diverse applications, from the manufacturing industry to automotive, from transportation and logistics to retail, and from healthcare to safety across all industries. Thanks to the availability of low-cost and low-power sensors, data collection becomes an easy task. Moreover, concerning devices that are either deployed massively or in difficult and dangerous places, whenever these devices run out of energy, battery replacement becomes complex. Therefore, in addition to computation offloading, which reduces the energy consumption and the processing delay, energy harvesting (EH) allows for their lifetime to be increased [7].

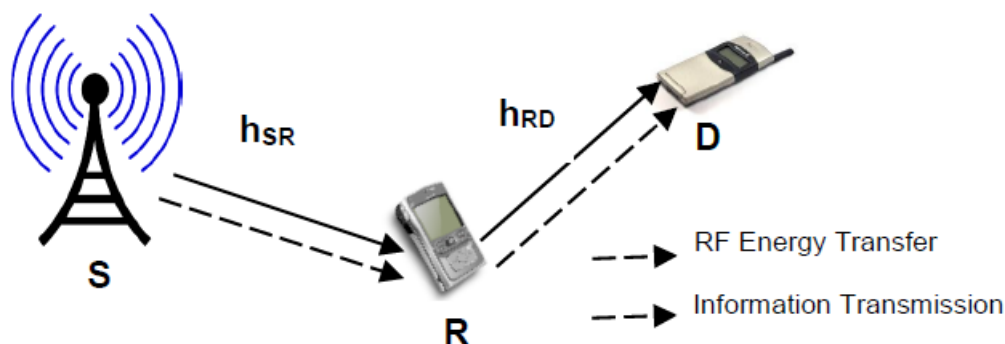
The upcoming 6G communication systems foresee the introduction of several new innovations aimed at fostering the 5G standard [8]. Among others, there is an increasing interest towards the concept of sustainability [9]. Indeed, it was clear from recent studies that telecommunication equipment has an ever-increasing impact on global energy consumption. The ultimate idea is that of going toward a net-zero-emission approach with the possibility of introducing zero-energy devices, i.e., devices able to operate with no impact in terms of energy consumption. This is possible thanks to an efficient exploitation of EH solutions, making it possible for devices to recharge their batteries. Indeed, powering mobile devices using wires is not viable. The rechargeable battery is the conventional source of power for most portable devices. The bigger the batteries, the better capacity; however, this is achieved at the cost of an increased physical size, which will increase the overall weight of the device significantly. Since these batteries have limited capacity, they have to be frequently recharged. Nevertheless, it is very inconvenient to connect those devices to cables, and hence a mechanism must be sought to recharge them through wireless connections. As a result, two novel wireless power supply methods are developed: EH [10] and wireless power transfer (WPT) [11].

EH, named also as power harvesting or energy scavenging, is the process of harvesting energy from external power sources, such as solar power, heat, wind, electromagnetic wave radiation, and kinetic energy, etc. The harvested energy is stored and mostly used for small mobile devices, such as wearable electronics and wireless sensors. Even though the harvested energy is significantly reduced, and there were concerns whether this energy is sufficient for computation-intensive applications, thanks to the rapid development of silicon technology, even a tiny amount of energy can perform plenty of operations. With EH, the need for replacing batteries on devices is reduced, especially those devices that are used in

hazardous environments. Some papers indeed considered the possibility of integrating EH solutions, exploiting solar panels with fog computing nodes. As an example, in study [12], the authors proposed a forecasting procedure able to increase the node lifetime. Despite the proposed solution, which allows an increase to the lifetime, its value still remains limited.

However, EH is unreliable because of fluctuations, high dependency on climate conditions, and human motions. Besides this, since the harvested energy is insignificant, it cannot be used for power-hungry devices, such as smartphones. Unlike the passive EH mechanism, WPT provides a stable and manageable wireless power supply by using a permanent power source. In WPT, the wireless links are used as a source of energy, hence the wireless links themselves can be exploited for recharging batteries and activating devices at distance. WPT is at the basis of a self-sustainable implementation of the upcoming 6G networks [13]. Simultaneous wireless and information power transfer (SWIPT) is a particular type of WPT, where the transmission of information (data), as well as power, is performed simultaneously based on either time switching or power splitting [14]. To achieve this simultaneity, a receiver is designed that is able to separately process information and harvest RF energy [15]. This is particularly useful for cooperative networks that depend on energy and information relaying. Those nodes which are more energy-constrained harvest energy from nearby powerful nodes that have ample amounts of energy.

Figure 1 shows a typical cooperative communication network employing SWIPT. Here, the source  $S$  is connected to a direct source of power and hence does not have any energy constraints. The intermediate relay node ( $R$ ) can be either the same device as the end node  $D$  or different. It can be assumed that the relay is relatively close to the source  $S$  compared to the end node  $D$ . Therefore, the relay can harvest energy from the source, and at the same time, the end node can harvest energy from the relay. Information can also be transmitted from the source to the relay and then to the end node, or vice versa.



**Figure 1.** System model for a SWIPT for cooperative communication network.

It is worth noting that several different SWIPT implementations exist, where the energy and the information are managed in different ways. For instance, in a time-switching SWIPT, the time frame is divided into slots, where one is assigned to the information sharing, while the other is assigned to the power transfer. In separate receivers, it is instead assumed that each device is equipped with two receivers, one for EH and one for information decoding. There are also two other schemes: in power splitting, the idea is to use a basic amount for information decoding and the surplus energy for EH, while in antenna switching, it is assumed that each device has one antenna for EH and one for information decoding. In the following study, we will implicitly refer to the time-switching architecture.

### 1.1. Related Works

In this sub-section, we list the most relevant activities in the areas of mobile edge computing (MEC)/fog computing and WPT. Recently, there have been some review papers in these areas. The authors in reference [16] present a comprehensive review on fog computing technology, and provide a comparison with other technologies, such as cloud

computing, MEC, and cloudlet computing. A comprehensive survey in reference [17] studies the methodologies in WPT-enabled MEC offloading.

There have also been some technological designs to improve the performance of the WPT technology. The authors in reference [18] present the non-maximally coefficient symmetry multi-rate filter bank in order to reduce the delay and hardware complexity for a wideband channelizer. The authors in reference [19] design a quadratic sandwich rectenna circuit that increases the output power, making them suitable for wireless devices in wireless telecommunication systems.

A consideration of WPT-enabled MEC systems has also been observed in several works. The authors in reference [20] propose an energy-efficient approach for the NOMA grouping selection. In particular, based on the exploitation of WPT, they demonstrate a user grouping method in a NOMA-based system for implementing a cooperative scheme, while also maximizing their communication efficiency. There have also been some works focusing on optimizing the transmitted power for energy minimization. The authors in reference [21] jointly optimize the time assignments for EH and offloading, and the transmit power at the device for offloading. Similar works are observed in references [22,23]. A game-theoretical approach is proposed in reference [24] for the resource allocation problem in a wireless-powered MEC system for IoT applications. They try to obtain the optimal power transfer for both the access point (AP) and the harvesting devices. The authors in reference [25] propose an optimization framework, aiming to maximize a utility function based on data transfer while minimizing energy consumption; the optimization is performed by jointly considering wireless power transfer at the access-point side and offloading power consumption at the end-device side, while taking into account delay constraints. The authors in reference [26] optimize the computing mode selection for mobile devices along with the system transmission time allocation in an MEC system allowing a WPT. Similar work is studied in reference [27], whereby the authors use the perfect and imperfect channel state information for the transmission mode selection in a wireless-powered sensor network. In reference [28], the authors focus on maximizing the harvested energy and minimizing the consumed energy of mobile devices in an MEC-WPT system by formulating a problem to optimize the EH time, task offloading time, task offloading size, and the devices' CPU frequencies.

The authors in reference [29] analyzed the performance of an amplify-and-forward system with RF, EH, and information processing. The authors proposed a time-switching-based relaying protocol, where the receiver allocates a portion of its time for EH and the rest of it for information processing, and a power-splitting-based relaying protocol, where a portion of the received power is used for EH and the rest of it for information processing. The work in reference [15] was extended to the systems in study [29], with decode-and-forward relaying. The authors analyzed the throughput of each cooperative network that employs the time-switching protocol and the power-splitting protocol. Assuming a relay node with an EH constraint, the exact analytical expressions of the achievable throughput and ergodic capacity are derived for the decode-and-forward relaying networks for both the time-switching and power-splitting schemes. Chen et al. [30] studied a time-switching cooperative network with a downlink information transfer and an uplink information transmission. The researchers suggested a harvest-then-cooperate protocol, where the intermediate relay and the source harvest energy from the same hybrid AP. Unlike the conventional cooperative networks, the relay and the source have no batteries, and hence the only source of energy is the harvested power from the AP. A delay-limited transmission mode is considered, and the result is supported both by theoretical results as well as numerical simulations.

### 1.2. Paper Contribution

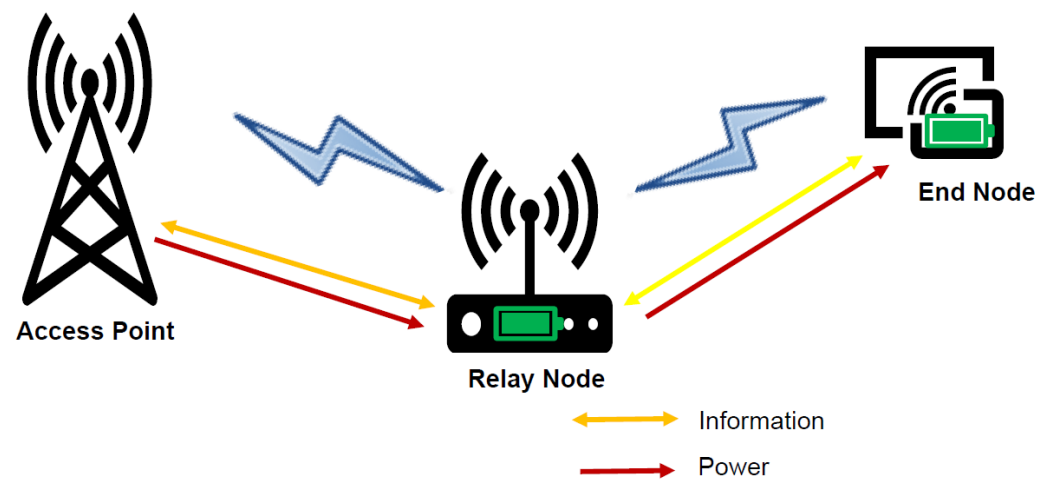
The possibility of integrating energy and communication results in the introduction of a novel concept, where the integration among communication, energy, and computing could lead to lots of advantages, is promising for the implementation of the upcoming 6G

networks [31]. The developed idea behind our work is indeed within this realm. The aim of this paper is to formulate the computation offloading for a SWIPT-based fog network. By doing so, computation-intensive processes are offloaded in case a more energy-efficient situation appears. This has a two-fold advantage. One is that energy consumption is reduced, thereby increasing the lifetime of the nodes. Reduction in the packet processing delay is the second advantage. The objective of the work is to develop an algorithm that outlines the thresholds for computation offloading, while in reference [32], we defined the main elements of the fog architecture for joint offloading and SWIPT limited to a two-nodes scenario; in this work, a more general scenario is considered, where an intermediate node is also used for both energy transfer and computation offloading. In addition to this, two policies have been proposed, aiming at optimizing the performance of the system by increasing the node lifetime under different conditions. Theoretical analysis has been also extended to a more general case including three nodes, as well the numerical results.

The rest of the paper is structured as follows: In Section 2, the system model is presented, and both energy- and offloading-related parameters are introduced. In Section 3, the feasibility analysis is carried out, allowing for an understanding as to what extent the proposed solution is viable regarding the energy constraints. In Section 4, the proposed algorithm for optimizing the computation offloading is described, while in Section 5, the numerical results obtained through computer simulation are shown. In Section 6, a final discussion of the proposed idea is given together with some proposals for further developments, while in Section 7, the conclusions are drawn.

## 2. System Model

The system is designed for a two-hop relay network. We assume the presence of an AP plugged in to the electrical network; hence, there are no energy constraints from the AP side. A battery-operated intermediate node is also considered with limited energy and computational power. Finally, there is a battery-operated end node with both energy and computational power constraints (see Figure 2).



**Figure 2.** The two-hop relay network architecture.

Compared to the other two nodes, the AP has a high transmission and reception power (i.e.,  $P_{tx}$  and  $P_{rx}$ ). It has also a higher computational speed with respect to the relay node and the end node. The relay node is a battery-operated terminal. It can harvest energy from the AP; since we assume a time-slotted SWIPT, the relay node is able to harvest energy from the AP only when it is not active in any transmission/reception phase. Moreover, we assume that the relay node has two separate antennas: while one antenna interacts with the AP, the other one interacts with the end node. This allows to enable the possibility to transfer the energy to the end node whenever the end node battery is depleted. The transmission, reception, and computation powers of the relay node are assumed to be the



same as the end node, while its computational speed is higher than that of the end node. Similarly to the relay node, the end node is also a battery-operated device. Whenever there is a need, it can harvest energy from the relay node. Concerning the processing phase, it is assumed that the end node is the only one responsible for the packet generation to be processed. Exploiting the computation-offloading principle, we assume that the generated packets can be either locally processed or offloaded to the relay node depending on the availability of resources in terms of energy and bandwidth.

### 2.1. SWIPT and System Design Equations

The goal of this paper is to design a novel algorithm that exploits SWIPT to improve the lifetime of the whole network and reduce the computational delay. A time-switching approach is used so that the antenna is used both for information transmission/reception and EH.

#### 2.1.1. System Design Equations

In Table 1, the parameters later used for modeling the system are listed. The end node/relay node, depending on various parameters and also the availability of resources, can perform the packet processing either locally or via offloading. Energy consumption and computational delay are different for local and offloaded processing.

**Table 1.** System parameters.

| Parameter  | End Node  | Relay Node  | AP                                       |
|--|---|---|--|
| Power transmission, reception                    | $p_{tx}^{FN}, p_{rx}^{FN}$                            | $p_{tx}^{FR}, p_{rx}^{FR}$                            | $p_{tx}^{AP}, p_{rx}^{AP}$               |
| Idle power, harvested power                      | $p_{id}^{FN}, p_{hp}^{FN}$                            | $p_{id}^{FR}, p_{hp}^{FR}$                            |  |
| Computation speed, computation power             | $\eta_{com}^{FN}, p_{com}^{FN}$                       | $\eta_{com}^{FR}, p_{com}^{FR}$                       | $\eta_{com}^{AP}, p_{com}^{AP}$          |
| Transmission, reception, idle, computation times | $t_{tx}^{FN}, t_{rx}^{FN}, t_{com}^{FN}, t_{id}^{FN}$ | $t_{tx}^{FR}, t_{rx}^{FR}, t_{com}^{FR}, t_{id}^{FR}$ | $t_{tx}^{AP}, t_{rx}^{AP}, t_{com}^{AP}$ |

#### 2.1.2. End Node—Consumed and Harvested Energy

Given any instant of time  $t$ , the total amount of consumed energy at the end node up to time  $t$  is given by:

$$E_c^{FN}(t) = \alpha p_{tx}^{FN} t_{tx}^{FN} + \alpha p_{rx}^{FN} t_{rx}^{FN} + (1 - \alpha) p_{com}^{FN} t_{com}^{FN} + p_{id}^{FN} t_{id}^{FN} \quad (1)$$

such that  $t \leq t_{tx}^{FN} + t_{rx}^{FN} + t_{com}^{FN} + t_{id}^{FN}$ . The consumed energy is the sum of the transmission, reception, computation, and idle energy. The total time needs to be lower or equal to the sum of the other terms. If the operations are non-overlapping, the total time becomes equal to the sum of the individual time intervals. Otherwise, if there is concurrent processing, the total time becomes less than the sum of the individual terms. The term  $\alpha$  is introduced to make a distinction between local processing and offloaded processing, and is a binary value (i.e., 0 or 1). If the operations are performed locally,  $\alpha$  is set to 0. Hence, energy consumption is due to local computation and idle power. On the other hand, if the packets are offloaded,  $\alpha$  value is set to 1. In this case, energy consumption is due to the transmission, reception, and idle time.

If SWIPT is enabled, the end node harvests energy from the relay node. Assuming free space propagation with propagation constant  $\beta$ , a fraction of the transmitted power from the relay node reaches the end node. Using the Friis transmission equation, the received power can be modeled as:

$$P_{rx} = \frac{P_{tx} G_{tx} G_{rx}}{\left(\frac{\lambda}{4\pi d}\right)^\beta} \quad (2)$$

where  $P_{rx}$  is the received power,  $P_{tx}$  is the transmitted power,  $G_{tx}$  is the gain of the transmitter,  $G_{rx}$  is the gain of the receiver,  $\lambda$  is the wavelength,  $d$  is the distance between the

transmitter and the receiver antennas, and  $\beta$  is the propagation constant. The energy harvested is then given by:

$$P_h^{FN} = \eta^h P_{tx}^{FR} |h_1|^2 \quad (3)$$

where  $\eta^h$  is the EH efficiency and  $|h_1|^2$  is the channel gain between the transmitter and receiver antennas of the relay node and the end node. The overall harvested energy at a certain time instant  $t$  is:

$$E_h^{FN}(t) = P_h^{FN} (t - t_{tx}^{FN} - t_{rx}^{FN}) \quad (4)$$

where  $t_{tx}^{FN}$ ,  $t_{rx}^{FN}$ , and  $t_{com}^{FN}$  are calculated as follows. The transmission time  $t_{tx}^{FN}$  depends on the data size and bandwidth available, hence:

$$t_{tx}^{FN} = \frac{L}{r} \quad (5)$$

where  $L$  is the amount of data transmitted over the link and  $r$  is data rate, calculated through the Shannon–Hartley theorem:

$$r = B \log_2 \left( 1 + \frac{P_{rx}}{P_n} \right) \quad (6)$$

where  $P_{rx}$  is the received power at the relay node using (2), and  $P_n$  is the noise power. The noise power is the total power that resides in the bandwidth  $B$ , which is the product of the noise power spectral density and the total bandwidth, i.e.,  $P_n = N_0 \cdot B$ . Assuming the noise to be white (i.e., thermal noise), the noise spectral density is given by  $N_0 = kT$ , where  $k$  is the Boltzmann's constant in Joule per Kelvin (i.e.,  $1.380649 \times 10^{-23} \text{ J K}^{-1}$ ),  $T$  is the receiver system noise temperature in Kelvin (i.e.,  $T = 290 \text{ K}$ ).

The reception time  $t_{rx}^{FN}$  is similar to  $t_{tx}^{FN}$  except that the received power at the end node is used in the Shannon–Hartley theorem. After the offloading operation is finished, the size of the processed task/data is usually smaller than the original task, due to the processing [2]. In most cases, the processed data is negligible, however, in this paper, the processed data is considered to be one-fifth of the original data.

The computation time is, on the other hand, dependent only on the packet size and the computational speed of the end node, hence:

$$t_{com}^{FN} = \frac{O}{\eta_{com}^{FN}} \quad (7)$$

where  $O$  is the data size (in bytes) multiplied by the task operations per byte. Having derived the harvested and the consumed energy, assuming that there is initial energy at the node, the remaining energy at any time  $t$  is given as:

$$E_r^{FN}(t) = E_r^{FN}(0) - E_c^{FN}(t) + E_h^{FN}(t) \quad (8)$$

where  $E_r^{FN}(0)$  is the initial energy of the node.

### 2.1.3. Relay Node—Consumed and Harvested Energy

The relay node energy consumption is different from the end node, since the relay node interacts with both the end node and the AP. At the same time, a fraction of its energy is consumed in transmitting the energy to the end node through the wireless link. Given any instant of time  $t$ , the total amount of consumed energy at the relay node up to time  $t$  is given by:

$$E_c^{FR}(t) = P_{tx}^{FR} (t_{tx}^{FR-N} + t_{tx}^{FR-AP}) + P_{rx}^{FR} (t_{rx}^{FR-N} + t_{rx}^{FR-AP}) + P_{com}^{FR} t_{com}^{FR} + P_{id}^{FR} t_{id}^{FR} + P_h^{FN} t_h^{FN} \quad (9)$$

such that:

$$t \leq (t_{tx}^{FR-N} + t_{tx}^{FR-AP}) + (t_{rx}^{FR-N} + t_{rx}^{FR-AP}) + t_{com}^{FR} + t_{id}^{FR} + t_h^{FN} \quad (10)$$

where  $t_{tx}^{FR-N}$  and  $t_{rx}^{FR-N}$  are the transmission time and reception time between the relay node and end node, respectively,  $t_{tx}^{FR-AP}$  and  $t_{rx}^{FR-AP}$  are the transmission time and reception time between the relay node and AP, respectively, and  $t_h^{FN}$  is the time to harvest for the end node. The energy harvested by the relay node is given by:

$$P_h^{FR} = \eta^h P_{tx}^{AP} |h_1|^2 \quad (11)$$

where  $|h_1|^2$  is the channel gain between transmitter and receiver antennas of the relay node and the AP. The expression is the same as that of the end node in (3) except that the energy transmitter is the AP. The remaining energy at the relay node at any instant of time  $t$  is:

$$E_r^{FR}(t) = E_r^{FR}(0) - E_c^{FR}(t) + E_h^{FR}(t) \quad (12)$$

where  $E_r^{FR}(0)$  is the initial energy of the relay node.

### 3. Feasibility Analysis

In an energy-constrained network, the primary goal is to improve the lifetime of the whole network. For fog networks with no SWIPT capability, the network is able to work as the battery at the nodes that are not depleted. As soon as the battery depletes, the network starts to fail. This becomes even worse when the intermediate relay node battery gets depleted faster. A failed end node does not affect system behavior as much. However, a failed relay node affects all the end nodes which are connected to it.

In order to understand the feasibility of the SWIPT-enabled fog network, we assume that given any network, with a given bandwidth, packet generation rate, and packet size, if the harvested energy up to the time instant  $t$  is greater than or equal to the consumed energy, the network is alive at least up to time instant  $t$ . Hence, we define the stability states for the end node and the relay node as:  $E_h^{FN}(t) \geq E_c^{FN}(t)$  and  $E_h^{FR}(t) \geq E_c^{FR}(t)$ .

#### 3.1. Local and Offloading Thresholds

We assume the packet generation rate follows a Poisson distribution with a mean value  $\lambda$ . This means the higher the value of  $\lambda$ , the more packets are generated in a given interval of time, which makes the packet inter-arrival very short. However, the smaller the lambda value, the fewer packets that are generated, and this makes the packet inter-arrival longer. Therefore, whether a given packet should be processed locally or offloaded depends on the packet inter-arrival after the local processing and offloading thresholds are calculated from the other parameters.

#### 3.2. Threshold Values for the end node

Assuming an arbitrary packet inter-arrival time  $\Delta t$ , the stable state for the end node is:

$$E_h^{FN}(t) \geq E_c^{FN}(t) \quad (13)$$

Exploiting (1) and (4), we can rewrite (13) as:

$$P_h^{FN}(\Delta t - \alpha t_{tx}^{FN} - \alpha t_{rx}^{FN}) \geq P_{id}^{FN}(\Delta t - \alpha t_{tx}^{FN} - \alpha t_{rx}^{FN} - (1 - \alpha)t_{com}^{FN}) + \alpha P_{tx}^{FN} t_{tx}^{FN} + \alpha P_{rx}^{FN} t_{rx}^{FN} + (1 - \alpha)P_{com}^{FN} t_{com}^{FN} \quad (14)$$

The above equation is the generalized formula for a stable state for both the local processing and the offloading. In the case of local processing (i.e.,  $\alpha = 0$ ), (14) becomes:

$$P_h^{FN}(\Delta t) \geq P_{id}^{FN}(\Delta t - t_{com}^{FN}) + P_{com}^{FN} t_{com}^{FN} \quad (15)$$



Rearranging (15) and solving for  $\Delta t$ , we obtain:

$$\Delta t \geq \frac{t_{com}^{FN}(P_{com}^{FN} - P_{id}^{FN})}{P_h^{FN} - P_{id}^{FN}}$$

In order to respect the stability state under the condition of local processing, the inter-arrival time ( $T_{loc}$ ) should be:

$$T_{loc} \geq \frac{t_{com}^{FN}(P_{com}^{FN} - P_{id}^{FN})}{P_h^{FN} - P_{id}^{FN}} \quad (16)$$

For offloaded processing ( $\alpha = 1$ ), (14) becomes:

$$P_h^{FN} \geq P_{id}^{FN}(\Delta t - \alpha t_{tx}^{FN} - t_{rx}^{FN}) + P_{tx}^{FN} t_{tx}^{FN} + P_{rx}^{FN} t_{rx}^{FN} \quad (17)$$

Rearranging (17) and solving for  $\Delta t$ , we obtain:

$$\Delta t \geq \frac{t_{tx}^{FN}(P_{tx}^{FN} + P_h^{FN} - P_{id}^{FN}) + t_{rx}^{FN}(P_{rx}^{FN} + P_h^{FN} - P_{id}^{FN})}{P_h^{FN} - P_{id}^{FN}}$$

For offloaded processing, the minimum inter-arrival time that fulfills the stability condition is:

$$T_{off} \geq \frac{t_{tx}^{FN}(P_{tx}^{FN} + P_h^{FN} - P_{id}^{FN}) + t_{rx}^{FN}(P_{rx}^{FN} + P_h^{FN} - P_{id}^{FN})}{P_h^{FN} - P_{id}^{FN}} \quad (18)$$

The associated processing delays for both the local processing and the offloaded operations are given by:

$$D_{loc}^l = t_{com}^{FN} \quad (19a)$$

$$D_{off}^l = t_{tx}^{FN} + t_{rx}^{FN} + t_{com}^{FR} \quad (19b)$$

### 3.3. Threshold Values for the Relay Node

As already discussed above, the stability condition is respected only if:

$$E_h^{FR}(t) \geq E_c^{FR}(t) \quad (20)$$

It is possible to write the energy stability equation, similar to (14), for the relay node as:

$$P_h^{FR}(\Delta t - \alpha t_{tx}^{FR} - \alpha t_{rx}^{FR}) \geq P_{id}^{FR}(\Delta t - \alpha t_{tx}^{FR} - \alpha t_{rx}^{FR} - (1 - \alpha)t_{com}^{FR}) + \alpha P_{tx}^{FR} t_{tx}^{FR} + \alpha P_{rx}^{FR} t_{rx}^{FR} + (1 - \alpha)P_{com}^{FR} t_{com}^{FR} + P_h^{FN} \Delta t \quad (21)$$

For local processing ( $\alpha = 0$ ), after algebraic simplification and solving for  $\Delta t$ , we get:

$$\Delta t \geq \frac{t_{com}^{FR}(P_{com}^{FR} - P_{id}^{FR})}{P_h^{FR} - P_{id}^{FR}}$$

For a stable state condition, the minimum inter-arrival time that must be respected for local processing is:

$$T_{loc} \geq \frac{t_{com}^{FR}(P_{com}^{FR} - P_{id}^{FR})}{P_h^{FR} - P_{id}^{FR}} \quad (22)$$

while for offloaded processing ( $\alpha = 1$ ), simplifying the expression and solving for  $\Delta t$ , we obtain:

$$\Delta t \geq \frac{t_{tx}^{FR}(p_{tx}^{FR} + p_h^{FR} - p_{id}^{FR}) + t_{rx}^{FR}(p_{rx}^{FR} + p_h^{FR} - p_{id}^{FR})}{p_h^{FR} - p_h^{FN} - p_{id}^{FR}}$$

For offloaded processing, the minimum packet inter-arrival that must be fulfilled for the stable state condition is:

$$T_{off} \geq \frac{t_{tx}^{FR}(p_{tx}^{FR} + p_h^{FR} - p_{id}^{FR}) + t_{rx}^{FR}(p_{rx}^{FR} + p_h^{FR} - p_{id}^{FR})}{p_h^{FR} - p_h^{FN} - p_{id}^{FR}} \quad (23)$$

The processing delays associated for both the local and offloaded processing results are:

$$D_{loc}^l = t_{com}^{FR} \quad (24a)$$

$$D_{off}^l = t_{tx}^{FR} + t_{rx}^{FR} + t_{com}^{AP} \quad (24b)$$

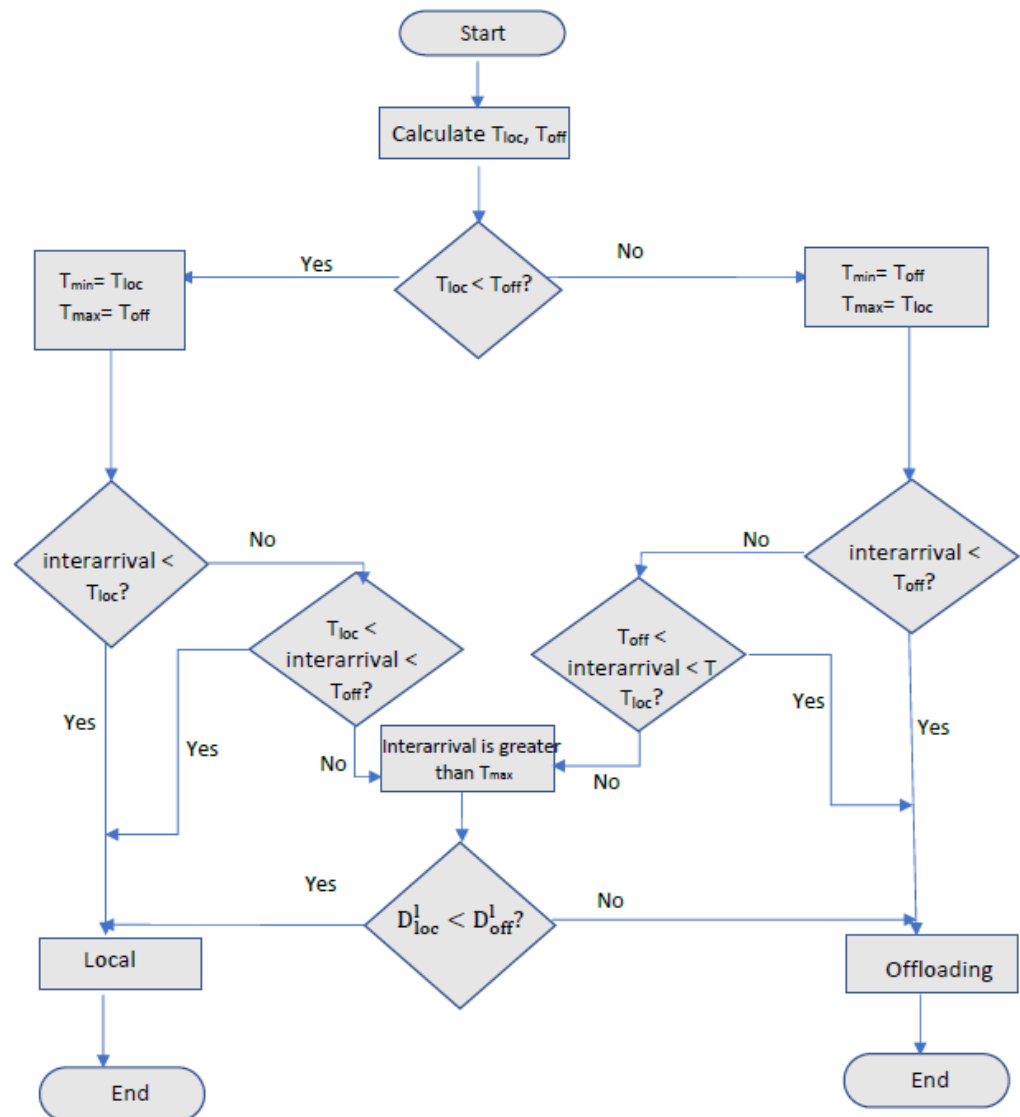
#### 4. SWIPT-Enabled Computation Offloading Decision Algorithm

For properly managing the decisions to be taken by the nodes under different conditions, we develop one algorithm that in turn considers two optimization policies, where the offloading decision is taken depending on the node's state. In addition, we consider four benchmarks, two of which are considered to have always-local processing operations (i.e., performing the packet computation at the nodes); one of them is considered to have SWIPT enabled and the other does not. Instead, the other two benchmarks are considered to always offload the packets to the remote AP for processing; one of them is SWIPT-enabled and the other one is not.

##### 4.1. Optimization with SWIPT—Policy 1

In the optimization with the SWIPT algorithm, we take all system parameters, in terms of packet inter-arrival, bandwidth, and packet size, into consideration. The decision for offloading depends on the values  $T_{loc}$ ,  $T_{off}$ , packet inter-arrival time, and the processing delay. The algorithm is shown in Figure 3 for the end node.

By focusing on the flowchart in Figure 3 it is possible to see that at first, the threshold values for  $T_{loc}$  and  $T_{off}$  are calculated. Then, they are compared, and the minimum of the two values is taken. After having calculated the inter-arrival interval, three possibilities stand: (i) the inter-arrival interval can be less than both thresholds, (ii) or in between the two thresholds, (iii) or greater than the two thresholds. If the inter-arrival is below both thresholds or in between the two thresholds, the decision is taken based on the minimum of the two, either the local or the offloading threshold. However, if the packet inter-arrival interval becomes greater than both values, the one which introduces lower processing delay is selected; this can be seen in the flowchart selection block before deciding between local or offloading actions. The goal of the algorithm is to find an optimum value of packet size, bandwidth, and packet generation rate ( $\lambda$ ), such that the energy balance is optimized.



**Figure 3.** Flowchart of the algorithm.

#### 4.2. Optimization with SWIPT—Policy 2

This modified version of the previous optimization technique (Policy 1) follows the same procedures to calculate the threshold values for the local computational and offloaded computation. The only difference is that, while the node in Policy 1 follows a greedy approach to harvest energy from the relay node, in Policy 2 we take into consideration the energy status of the end and relay nodes. However, since the relay node is also battery operated, its energy depletes quickly if it continuously transfers energy to the end node. In Policy 2, the end node is allowed to harvest energy either if its remaining energy falls below a certain threshold, or if the packet inter-arrival at the relay node is large, such that it can harvest enough energy during that time interval.

#### 4.3. Benchmark Procedures

Four benchmark procedures are considered as a comparison for evaluating the system performance.

##### 4.3.1. Local Processing with and without SWIPT

In these two procedures, regardless of the parameters, whenever there is a packet arrival, the packets are processed locally. Both the end node and the relay node process their

packets locally. For the local processing without SWIPT, the two nodes are independent of each other and there is no interaction. For the scenario where the SWIPT is present, the end node can harvest energy from the relay node throughout the time. The relay node can also harvest from the AP throughout the time. There is a partial interaction between the end node and the relay node, and between the relay node and the AP. The interaction is limited to the transmission of power.

#### 4.3.2. Offloaded Processing with and without SWIPT

In the offloaded processing, both nodes generate packets independently. All the generated packets are, however, offloaded. The packets generated at the relay node are offloaded to the AP for processing. The packets generated at the end node are offloaded twice: firstly, to the relay node, and secondly, to the AP. For the offloading scenario without SWIPT, the interaction between the nodes is still partial. The interaction is only due to the packets that are offloaded from one node to the other node. On the other hand, for the offloading scenario with SWIPT enabled, there is a full interaction among the nodes. Both nodes share packets and energy. The end node is offloading packets to the relay node and the relay node is offloading packets to the AP. Likewise, the end node harvests energy from the relay node and the relay node harvests energy from the AP.

### 5. Numerical Results

The numerical results are obtained through computer simulations in MATLAB. The values for the different parameters are given in Table 2. The simulation parameters are fixed throughout the simulations. The packet size, packet arrival rate, and bandwidth are variable parameters. By varying these three parameters, various simulation results are obtained.

**Table 2.** Simulation parameters.

| Parameter                 | End Node                         | Relay Node                       | AP                               |
|---------------------------|----------------------------------|----------------------------------|----------------------------------|
| Power trans/recep. [W]    | $p_{tx}^{FN}, p_{rx}^{FN} = 1.3$ | $p_{tx}^{FR}, p_{rx}^{FR} = 1.3$ | $p_{tx}^{AP}, p_{rx}^{AP} = 1.5$ |
| Idle power [W]            | $p_{id}^{FN} = 0.01$             | $p_{id}^{FR} = 0.01$             | -                                |
| Computation power [W]     | $p_{com}^{FN} = 0.9$             | $p_{id}^{FR} = 0.9$              | -                                |
| Harvest efficiency        | $\eta^h = 0.01$                  | $\eta^h = 0.02$                  | -                                |
| Computation speed [FLOPs] | $\eta_{com}^{FN} = 15$ G         | $\eta_{com}^{FR} = 15$ G         | $\eta_{com}^{AP} = 100$ G        |
| Task operations [FLOPs/B] | 10,000                           | 10,000                           | 10,000                           |

Each simulation runs for a maximum of 500,000 iterations. Each time slot has a 6 ms duration. In total, each simulation is considered to run for 3000 s. The environmental propagation parameter  $\beta$  in (2) has been set to 2.7. The optimization algorithms, with both policies, as well as the four benchmark procedures, have been considered in each of the following scenarios. In each scenario, different value ranges for the parameters are considered, and three comparisons are plotted: the lifetime of the network, the harvested energy, and the delay. For the algorithms optimizing the behavior, the percentage of the locally processed and offloaded packets is also plotted.

- Lifetime of the network:** Both the end node and the relay node are battery operated. Each is assumed to have an initial energy equal to 20 J. The lifetime of the network is calculated based on the remaining energy on the nodes. Whichever of the two, either the relay node or the end node, whose remaining energy falls below the threshold value will let the network down, and that determines the lifetime of the network. A node is assumed to be off if the remaining energy falls below 10% of its battery capacity.
- Harvested energy:** The system architecture is based on the time-switching SWIPT. The antenna is used for EH if there is no packet transmission or reception. The relay node can harvest energy from the AP. The end node, on the other hand, harvests

energy from the relay node. The harvested efficiency of the relay node is assumed to be twice that of the end node.

- c. **Locally processed/offloaded percentage:** Generated packets can either be processed locally or offloaded for remote processing. Depending on the threshold values, packets can be offloaded or not. Based on comparing the packet inter-arrival with the local and offloaded thresholds, the decision is made. The end node decides to process the packets locally or offloads them to the relay node. The relay node, in turn, makes its own decision either to process the packets locally or offload them to the AP.
- d. **Packet generation:** The packets are generated randomly following Poisson distribution. The average packet arrival rate is provided to the simulation for each time slot.

As already described in the previous section, our proposed algorithm with the two policies and four benchmark procedures are considered. The local with-SWIPT and without-SWIPT algorithms process the packets locally. The local with-SWIPT algorithm can harvest energy from a remote node. For example, the end node harvests energy from the relay node and the relay node, on the other hand, harvests energy from the AP.

The two other benchmarks are the offloading with SWIPT and without SWIPT. Both of them offload their packets, i.e., the end node offloads the packets to the relay node and the relay node offloads its own packets and the packets coming from the end node to the AP. The offloading with SWIPT can harvest energy whenever its antennas are not transmitting or receiving packets.

The last two algorithms are the SWIPT with Optimization P1 (Policy 1) and SWIPT with Optimization P2 (Policy 2). Both of them make the offloading decisions based on the local and offloading threshold values already derived in the previous section. By comparing these threshold values with the packet inter-arrival time, they decide whether to offload a packet or process it locally. The difference is that Policy 1 uses this decision just to decide whether to offload or not. However, in this scenario, the end node can harvest at any time but this consumes a lot of energy from the relay node. As the end node does not have any information about the energy status of the relay node, the relay node can run out of energy and hence the lifetime of the network is impacted. Policy 2 is introduced to alleviate this problem. The end node does not harvest energy from the relay node all the time. It harvests energy from the relay only if its remaining energy drops below 50% of its initial energy, and also if the packet inter-arrival time at the relay node lets the relay node harvest more energy than consume energy.

### 5.1. Threshold Differences between the Local and Offloading Computations

As described previously, in the optimization algorithms, the offloading decision is based on the comparison between the packet inter-arrival time, and the local and offloading thresholds. Based on the decision, we can have three regions, i.e., (a) inter-arrival is less than both thresholds, (b) inter-arrival is in between the two thresholds, and (c) the inter-arrival is above both thresholds. The first two regions depend on the threshold with the lower value, and hence the offloading decision is based on the minimum of the two. However, the third region depends on the operation that has the minimum packet processing time. The regions are shown in Figure 4.

The goal of the optimization is to properly manage the offloading decision based on the threshold values. As shown in the tables below, the difference between the two thresholds ( $T_{off} - T_{loc}$ ) is evaluated for different bandwidth and packet size values; the evaluation of the differences allows a better understanding of the decision to be taken among option A and option B in Figure 4. Table 3 shows the different values for the end node, while Table 4 shows these values for the relay node.

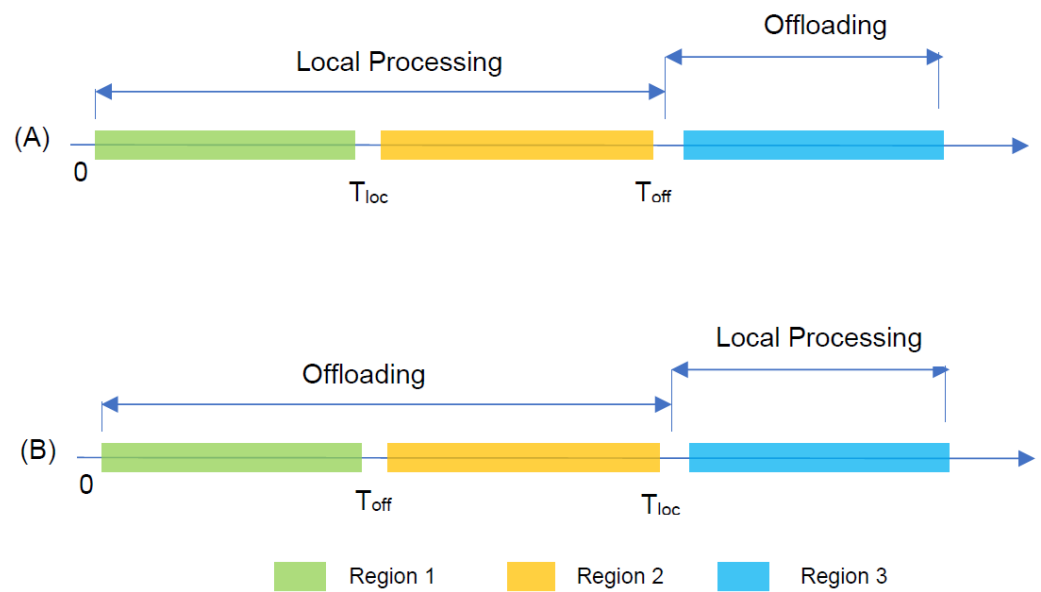
**Table 3.** Threshold difference values between offloading and local for the end node ( $T_{off} - T_{loc}$ ).

|         | 500 B    | 1 kB     | 2 kB     | 5 kB     | 10 kB    | 20 KB    | 50 KB     | 100 kB    | 200 kB    | 500 kB    | 1 MB      | 2 MB      | 5 MB      |
|---------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 200 Hz  | 202.1801 | 404.3602 | 808.7205 | 2021.801 | 4043.602 | 8087.205 | 20,218.01 | 40,436.02 | 80,872.05 | 202,180.1 | 404,360.2 | 808,720.5 | 2,021,801 |
| 500 Hz  | 82.94292 | 165.8858 | 331.7717 | 829.4292 | 1658.858 | 3317.717 | 8294.292  | 16,588.58 | 33,177.17 | 82,942.92 | 165,885.8 | 331,771.7 | 829,429.2 |
| 1 kHz   | 42.2657  | 84.5315  | 169.063  | 422.6575 | 845.315  | 1690.63  | 4226.575  | 8453.15   | 16,906.3  | 42,265.75 | 84,531.5  | 169,063   | 422,657.5 |
| 500 kHz | 0.0047   | 0.0094   | 0.0188   | 0.0471   | 0.0943   | 0.1886   | 0.4716    | 0.9432    | 1.8865    | 4.7163    | 9.4326    | 18.8653   | 47.1634   |
| 1 MHz   | −0.0457  | −0.0915  | −0.1830  | −0.4576  | −0.9153  | −1.8306  | −4.5766   | −9.1532   | −18.3065  | −45.7663  | −91.5326  | −183.065  | −457.663  |
| 10 MHz  | −0.0930  | −0.18617 | −0.37234 | −0.93085 | −1.86171 | −3.72342 | −9.3085   | −18.6171  | −37.2342  | −93.0854  | −186.171  | −372.342  | −930.854  |

**Table 4.** Threshold difference values between offloading and local for the relay node ( $T_{off} - T_{loc}$ ).

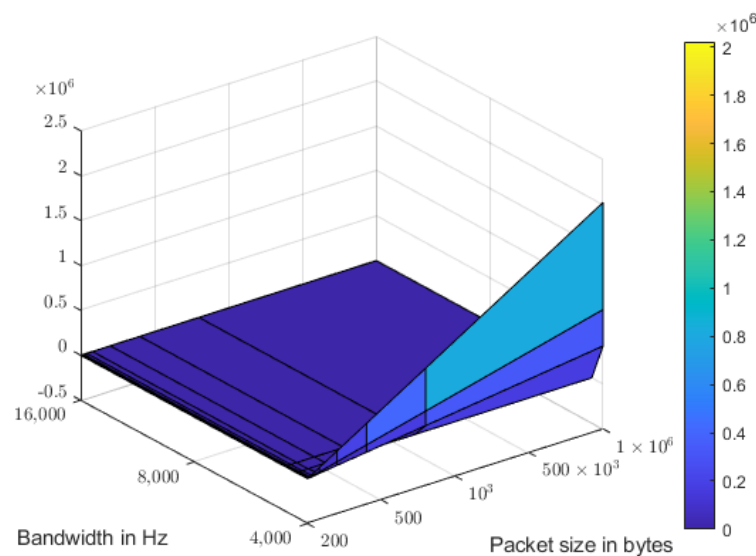
|         | 500 B   | 1 kB     | 2 kB     | 5 kB     | 10 kB     | 20 KB    | 50 KB     | 100 kB     | 200 kB    | 500 kB     | 1 MB       | 2 MB       | 5 MB     |
|---------|---------|----------|----------|----------|-----------|----------|-----------|------------|-----------|------------|------------|------------|----------|
| 200 Hz  | 69.0115 | 138.0231 | 276.0463 | 690.1157 | 1380.2315 | 2760.463 | 6901.1574 | 13,802.315 | 27,604.63 | 69,011.574 | 138,023.15 | 276,046.3  | 690,115  |
| 500 Hz  | 28.3163 | 56.6326  | 113.2653 | 283.1632 | 566.3264  | 1132.653 | 2831.6324 | 5663.2648  | 11,326.53 | 28,316.324 | 56,632.648 | 113,265.3  | 283,163  |
| 1 kHz   | 14.4335 | 28.8671  | 57.7342  | 144.3355 | 288.6711  | 577.3422 | 1443.3556 | 2886.7113  | 5773.4225 | 14,433.556 | 28,867.113 | 57,734.225 | 144,335  |
| 500 kHz | 0.0016  | 0.0212   | 0.0425   | 0.1063   | 0.2126    | 0.4252   | 1.0630    | 2.1261     | 4.2523    | 10.6308    | 21.2617    | 42.5235    | 106.056  |
| 1 MHz   | −0.0065 | −0.0131  | −0.0263  | −0.0659  | −0.1319   | −0.2638  | −0.6595   | −1.3191    | −2.6382   | −6.5956    | −13.1912   | −26.3825   | −65.9563 |
| 10 MHz  | −0.0227 | −0.0454  | −0.0909  | −0.2274  | −0.4548   | −0.9096  | −2.2742   | −4.5484    | −9.0968   | −22.7421   | −45.4842   | −90.9684   | −227.42  |





**Figure 4.** Local and offloading computation thresholds. (A)  $T_{off} > T_{loc}$  (B)  $T_{loc} > T_{off}$

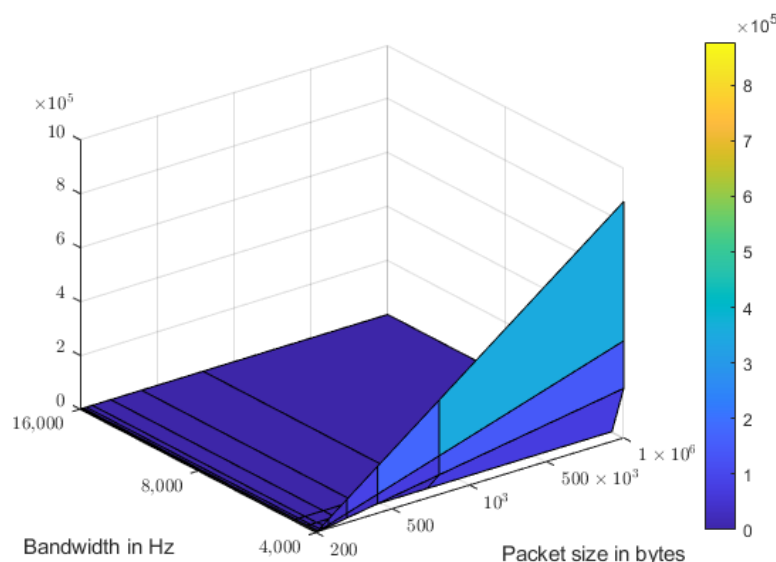
The same results reported in Tables 3 and 4 are also reported in Figures 5 and 6, allowing a better understanding of the impact of the parameters over the offloading thresholds.



**Figure 5.** Threshold difference between offloading and local for end node.

As can be observed from Tables 3 and 4, and from Figures 5 and 6, the threshold difference reduces for both cases, with small values of bandwidth and small values of packet size. As the packet size increases, keeping the bandwidth constant, the offloading threshold increases. It is also possible to notice that for a fixed packet size, when the bandwidth is increased, the threshold become negative; this corresponds with the local and offloading thresholds being swapped, which is what is represented in the scheme labeled with (B) and with respect to that labeled with (A) in Figure 4. Moreover, there is a trade-off between bandwidth and packet size that is around 500 kHz, while the impact of packet size is linear. It can be seen that the magnitude of the values for the end node is much

higher than that for the relay node. This is expected, since the impact of the optimization is much higher for the end node that is able to harvest only a limited portion of the energy transmitted by the relay. It is worth being reminded that in case the packet inter-arrival becomes big enough, which means its value is greater than both thresholds, the decision is made based on the computation that introduces the lowest packet processing delay. This is due to the fact that the inter-arrival time is sufficient for EH.



**Figure 6.** Threshold difference between offloading and local for Relay node.

### 5.2. Variable Bandwidth, Fixed Packet Arrival Rate, and Fixed Packet Size

An average packet arrival equal to  $\lambda = 0.03$  is assumed for both the end node and the relay node for each time slot. This corresponds to generating five packets per second. The packet is fixed to 2 kB. The simulation is performed for the following values of bandwidth: 200 kHz, 500 kHz, 1 MHz, 2 MHz, and 5 MHz. The bandwidth has a positive correlation with the lifetime. As shown in Figure 7, except for the local computation, which is independent of the variability of the bandwidth, all the algorithms show an increase in the lifetime as bandwidth increases. This is because as bandwidth increases, the interaction time of the antenna during the transmission and reception of the packets is lower, i.e., the data rate is higher. The lower the interaction for the antenna means the lower the energy consumption. All the algorithms incorporating the SWIPT are indeed showing better performance than those without SWIPT.

The harvested energy is directly related to the bandwidth. Since there is one antenna employing a time-switching technique to be used for both packet transmission/reception and EH, the amount of time it becomes idle from the transmission/reception of packets is crucial for the amount of harvested energy. Therefore, if the bandwidth is higher, the antenna spends less time in transmitting and receiving packets, and hence more energy can be harvested. As shown in Figure 8, SWIPT & Local is the asymptotic behaviour. In this procedure, since the antenna is idle all the time, it can be used to harvest energy. This is the theoretical maximum energy that can be achieved. Whichever algorithm we employ, the harvested energy always approaches the SWIPT & Local but never crosses it. The average harvested energy is increasing for all scenarios as bandwidth increases. Since the Local and Offloading without SWIPT do not harvest energy, they are shown with a zero value at the bottom of the plot.

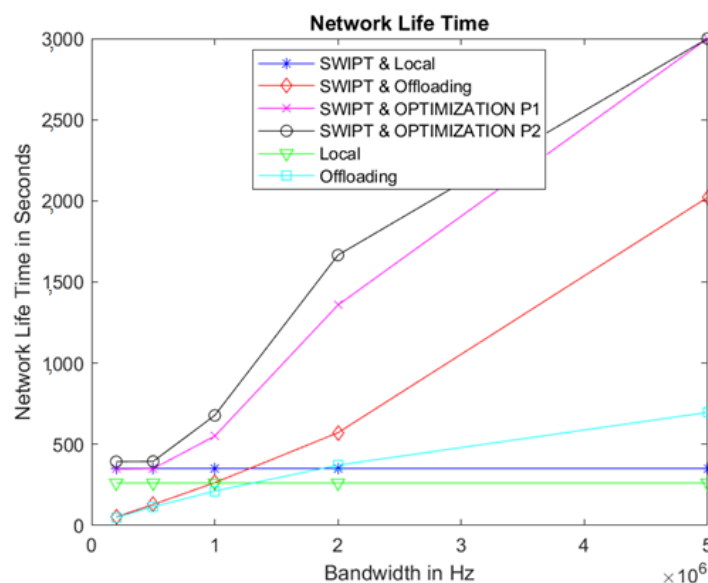


Figure 7. Network lifetime for variable bandwidth.

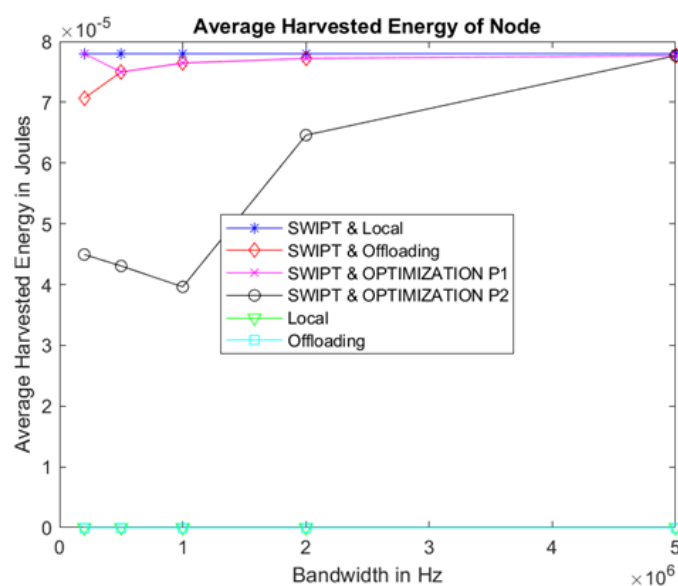


Figure 8. Average harvested energy of the end node.

Task processing delay is the amount of time it takes a packet to be processed whether performed locally or offloaded. If the packet is processed locally, the task delay is the same as the computation time of the node. However, if the packet is offloaded, the task processing delay is equal to the sum of the retransmission time plus the packet computation time at the remote device. Therefore, the variability of the bandwidth does not have any effect if the packet is processed locally. However, if the packet is offloaded, the higher the bandwidth, the lower will be the packet processing time. As depicted in Figure 9, both the offloading procedures (i.e., with and without SWIPT) show a decrease in average task delay as bandwidth increases.

Figure 10 shows the percentage of packets that are locally processed at the end node and offloaded to the relay node. For small values of bandwidth, the end node realizes it is energy-consuming to offload the packet. Hence, it processes the packets locally. For a bandwidth equal to 500 kHz, around 15% of all the generated packets are processed locally, while the rest are offloaded to the relay node. However, for larger values of bandwidth, it

is economical in terms of energy consumption to offload the packets, and therefore the end node offloads all of its packets to the relay node.

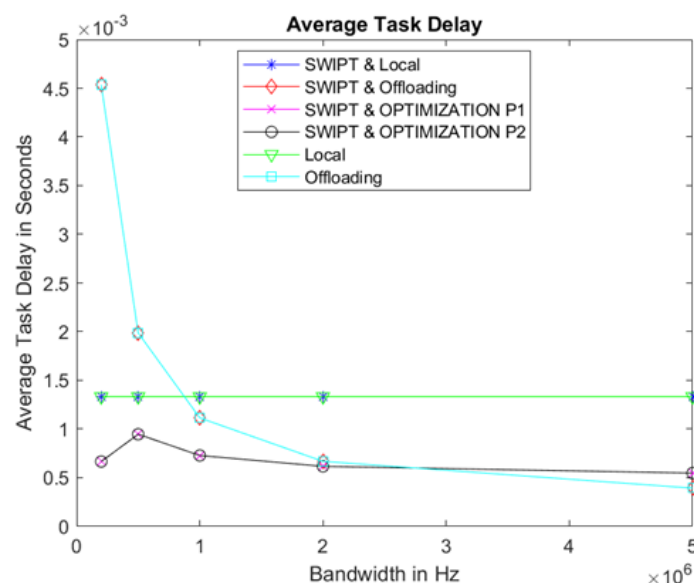


Figure 9. Average task delay of the end node.

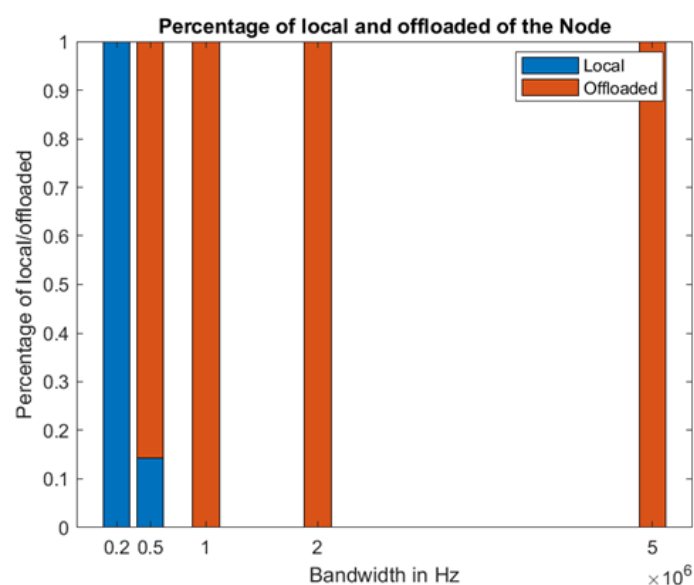


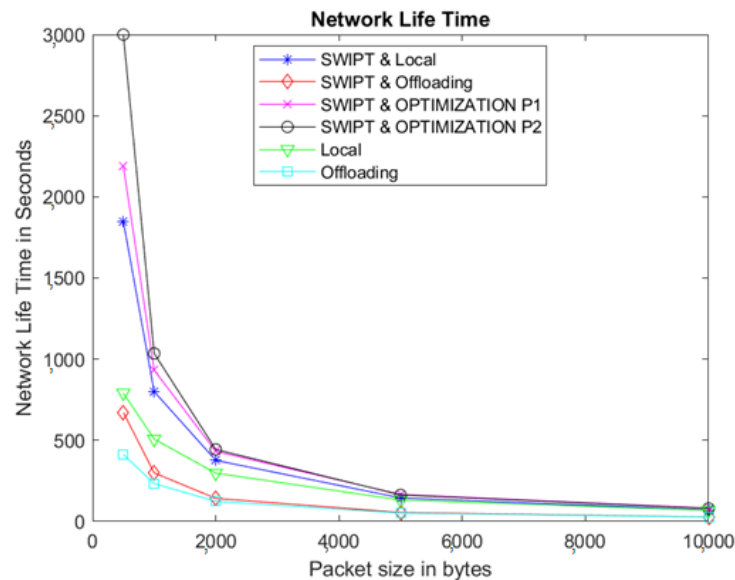
Figure 10. Percentage of locally computed and offloaded packets.

### 5.3. Variable Packet Size, Fixed Bandwidth, and Fixed Packet Arrival Rate

An average packet arrival  $\lambda = 0.3$  is assumed for both the end node and the relay node for each time slot. This corresponds to generating 50 packets per second. The bandwidth is assumed to be 600 kHz. The simulations are made for the following packet sizes: 500 B, 1 kB, 2 kB, 5 kB, and 10 kB.

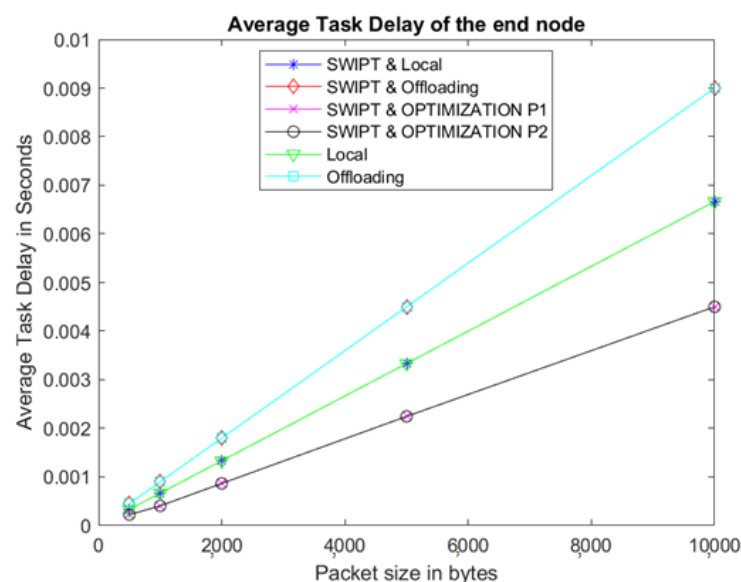
The packet size and lifetime of the network have a negative correlation. Larger packet size means the nodes need a lot of power to process it. For local computation, the processor takes a lot of time to do the operation and hence consumes a lot of energy. For the offloaded operation, the antennas spend a long time during the transmission and reception of packets. An increased antenna activity means more consumed energy. As shown in Figure 11, all the scenarios are decreasing for increasing packet sizes. The local ones are decreasing their energy due to the local activity, while the procedures employing offloading are consuming

energy due to increased antenna activity. As expected, the two optimization algorithms are showing better performance than the others. Particularly, Policy 2 is showing a better network lifetime than Policy 1 because of the interactive nature of the energy transfer between the end node and the relay node.



**Figure 11.** Network lifetime with variable packet size.

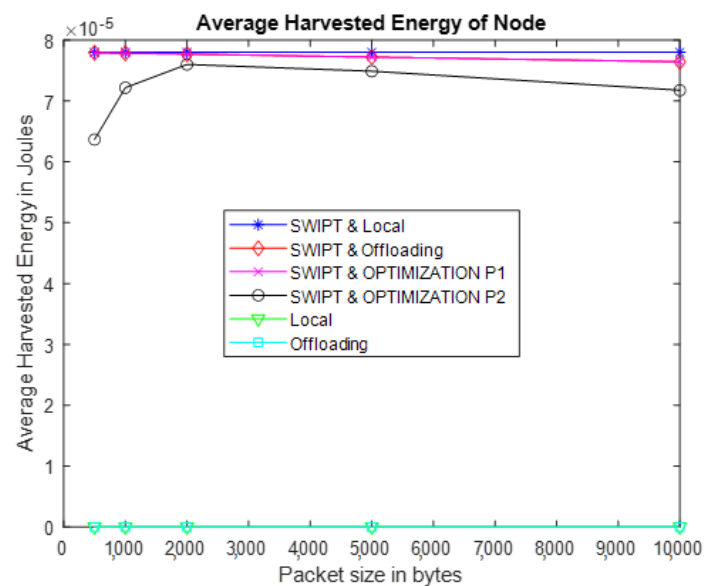
The average task delay also has a positive correlation with the packet size. The larger the packet size, the longer it takes for the packet to be processed. For local computation, the delay is only due to the computation time. For the offloading, the larger the packet size, the longer it takes to transmit it through a limited-bandwidth channel. As shown in Figure 12, the average task delay increases linearly with the packet size.



**Figure 12.** Average task delay of the end node with variable packet size.

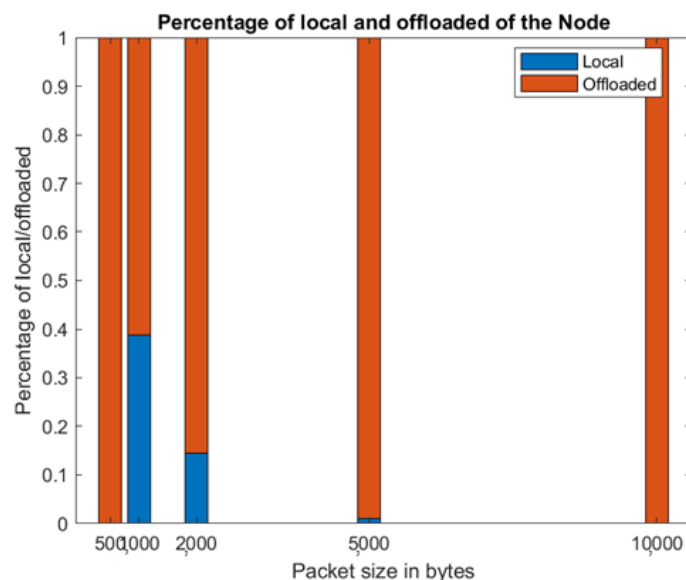
The energy harvested in the local computation is still unaffected because antennas are not employed. However, for all the scenarios that employ offloading, the increase in packet size impacts the harvested energy negatively. As the packet size increases, whenever the antenna wants to offload the packets to the remote device, it spends a longer time

during transmission and reception. Since a single antenna is being used in a time-switching manner, this affects the harvested energy. Therefore, as can be seen in Figure 13, the plot shows a decrease in the average harvested energy as packet sizes increase.



**Figure 13.** Average harvested energy of the end node with variable packet size.

Based on the packet inter-arrival time, and comparing this value with the local and offloading threshold values derived in the previous section, packets are either offloaded or processed locally. The end node offloads its packets to the relay node (if it has to). However, the relay node has two roles: it has to process its own generated packets and also packets coming from the end node. As depicted in Figure 14, for small-value packets, the end node can process a fraction of the total-generated packets locally.

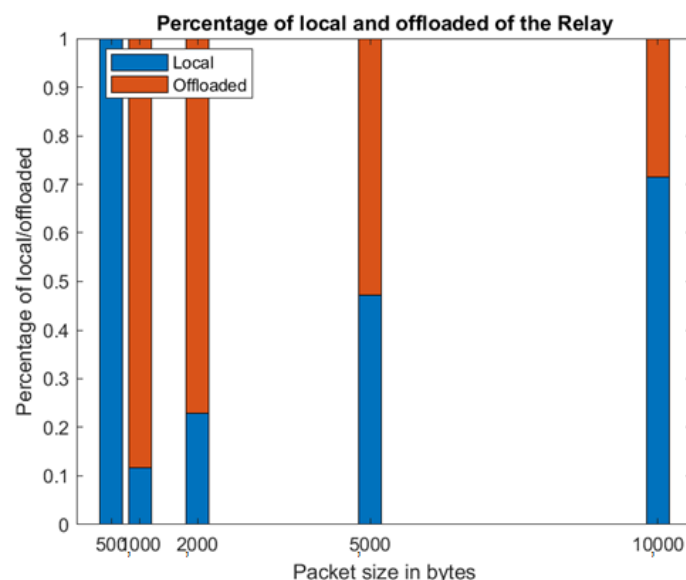


**Figure 14.** Percentage of locally computed and offloaded packets of the end node.

However, as the sizes of the packet become larger, it offloads the packets to the relay node for remote processing. However, offloading packets to the relay node is an extra burden to the relay node. As shown in Figure 15, for a small-size packet, the relay node processes a fraction of the total packets (both generated and received from the end node) locally. However, as packet size increases, energy consumption increases due to the increase



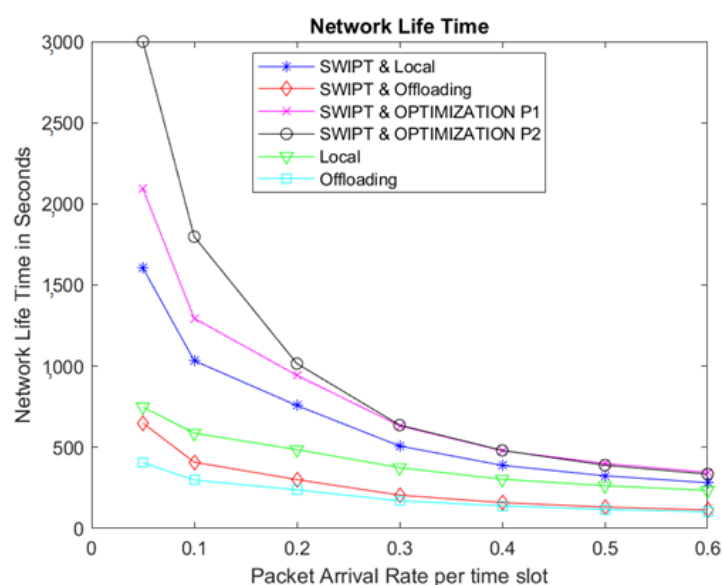
in transmission and reception time. Therefore, the relay node processes a majority of the packets locally, while a fraction of them are offloaded.



**Figure 15.** Percentage of locally computed and offloaded packets of the relay node.

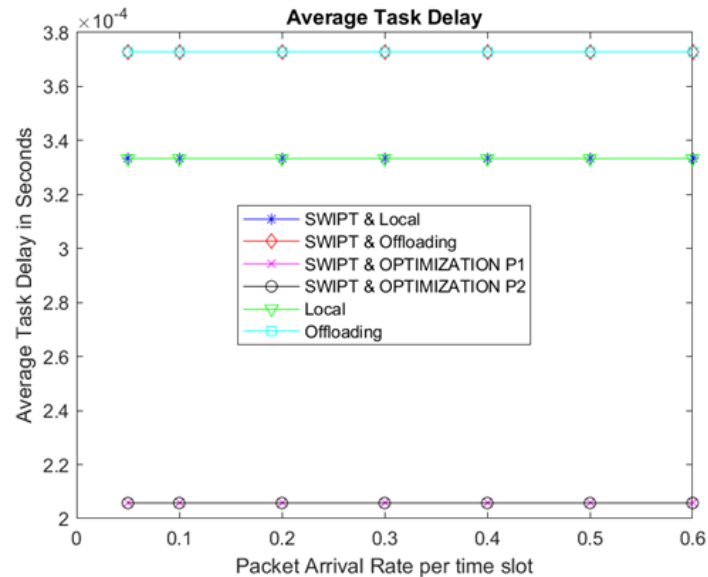
#### 5.4. Variable Packet Arrival Rate, Fixed Bandwidth, and Fixed Packet Size

In this simulation, different values of packet arrival rates are used. The bandwidth is 600 kHz and the packet size is 2 kB. The following  $\lambda$  packet arrival rates are assumed for all simulations: 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6. The average packet arrival rate shows the number of packets that are generated in a single time slot. Therefore, the bigger the number, the more packets are generated in a given time slot. The higher packet arrival rate shortens the packet inter-arrival time. More packets mean more energy is consumed. For the local processing, more packets mean the local processor is always busy; however, since the antenna is idle, it can harvest energy throughout this time. For those scenarios employing offloading, the generation of more packets makes the antenna always busy transmitting and receiving packets. This consumes energy on the one hand, and decreases the energy to be harvested on the other hand. As shown in Figure 16, in all the algorithms, the life time decreases as the average packet arrival rate increases.



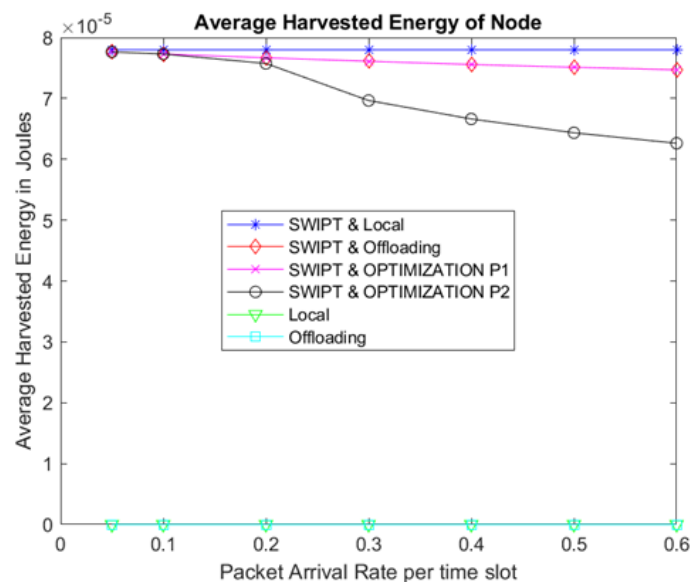
**Figure 16.** Network lifetime for the variable packet arrival rate.

The average task delay is unaffected by the change in the average packet arrival rate. Even though more packets are generated and the total task delay changes accordingly, the average does not change. This is shown in Figure 17.



**Figure 17.** Average task delay of the end node with variable packet arrival rate.

The more packets generated, the busier the antennas will be transmitting and receiving packets. This compromises the time that the antennas could have been used to harvest energy from the remote node. Hence, the harvested energy decreases as the average packet arrival rate increases, as shown in Figure 18.



**Figure 18.** Average harvested energy of the end node with variable packet arrival rate.

## 6. Discussion

Due to the increased number of fog and IoT devices, energy consumption is an issue. Since most of these wireless devices are battery operated, their lifetime is very limited. Whether standalone or cooperating with other devices, the failure of one device might affect the network. Different approaches are proposed to power these devices, of which this paper has focused on simultaneous wireless EH and information transfer. By doing

so, the energy requirements of the devices are fulfilled, and computation offloading is performed based on some thresholds. Three parameters, the bandwidth, the packet size, and the packet arrival rate, are considered, to see how the different scenarios behave as these parameters change. At each simulation, one parameter is made variable while fixing the other two. By doing so, different values are plotted. The network lifetime, the average task delay, and the harvested energy are plotted as a function of the changing parameter to compare and contrast among the different scenarios. The optimization algorithm calculates the local threshold and the offloading threshold. It compares these values with the packet inter-arrival interval. Based on the decision, the packets are either processed locally or offloaded to the remote device for offloaded operation. Both the end node and the relay node can harvest energy. However, the end node harvests energy from the relay node only if its remaining energy is below a threshold or the relay node can harvest enough energy from the AP in any packet inter-arrival interval. This work was done for a two-hop fog network. This work can be expanded by using multiple end nodes, multiple relay nodes, and multiple APs. For example, the end node will wisely select the one with the best computational capacity or/and ample energy. Moreover, it will be applicable for mesh-type networks where there are multiple relay points.

## 7. Conclusions

In this paper, we have studied the effect of the SWIPT technology in a three-node fog network employing computation offloading. The time-switching operation is considered for managing the simultaneous transmission of energy and tasks to be processed. An optimization algorithm has been proposed by considering two different policies in terms of the relay node energy management. The performance of the proposed policies are compared with four benchmark procedures, considering static approaches. We have demonstrated that for implementing such a situation there are some feasibility intervals where the offloading is advantageous, while for some parameter combinations there is no advantage. We have also demonstrated that by opportunely deriving the offloading procedure it is possible to optimize the energy consumption and transfer, hence maximizing the network lifetime.

**Author Contributions:** Conceptualization, D.T.; methodology, D.T. and A.B.; software, M.D.G.; validation, M.D.G., A.B., and D.T.; formal analysis, M.D.G., A.B., and D.T.; investigation, M.D.G.; resources, M.D.G. and D.T.; data curation, M.D.G.; writing—original draft preparation, D.T.; writing—review and editing, D.T. and A.B.; visualization, M.D.G.; supervision, D.T.; project administration, D.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Rayes, A.; Salam, S. *Internet of Things From Hype to Reality—The Road to Digitization*, 2nd ed.; Springer: Cham, Switzerland, 2019. doi:10.1007/978-3-319-99516-8.
2. Bozorgchenani, A.; Mashhadi, F.; Tarchi, D.; Salinas Monroy, S.A. Multi-Objective Computation Sharing in Energy and Delay Constrained Mobile Edge Computing Environments. *IEEE Trans. Mob. Comput.* **2021**, *20*, 2992–3005. doi:10.1109/TMC.2020.2994232.
3. Mouradian, C.; Naboulsi, D.; Yangui, S.; Glitho, R.H.; Morrow, M.J.; Polakos, P.A. A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 416–464. doi:10.1109/COMST.2017.2771153.
4. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*; Association for Computing Machinery: New York, NY, USA, 2012; pp. 13–16. doi:10.1145/2342509.2342513.

5. Atlam, H.F.; Walters, R.J.; Wills, G.B. Fog Computing and the Internet of Things: A Review. *Big Data Cogn. Comput.* **2018**, *2*, 10:1–10:18. doi:10.3390/bdcc2020010.
6. Yannuzzi, M.; Milito, R.; Serral-Gracià, R.; Montero, D.; Nemirovsky, M. Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. In Proceedings of the 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, 1–3 December 2014; pp. 325–329. doi:10.1109/CAMAD.2014.7033259.
7. Ma, D.; Lan, G.; Hassan, M.; Hu, W.; Das, S.K. Sensing, Computing, and Communications for Energy Harvesting IoTs: A Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1222–1250. doi:10.1109/COMST.2019.2962526.
8. Viswanathan, H.; Mogensen, P.E. Communications in the 6G Era. *IEEE Access* **2020**, *8*, 57063–57074. doi:10.1109/ACCESS.2020.2981745.
9. Hu, J.; Wang, Q.; Yang, K. Energy Self-Sustainability in Full-Spectrum 6G. *IEEE Wirel. Commun.* **2021**, *28*, 104–111. doi:10.1109/MWC.001.2000156.
10. Sanislav, T.; Mois, G.D.; Zeadally, S.; Folea, S.C. Energy Harvesting Techniques for Internet of Things (IoT). *IEEE Access* **2021**, *9*, 39530–39549. doi:10.1109/ACCESS.2021.3064066.
11. Zhang, Z.; Pang, H.; Georgiadis, A.; Cecati, C. Wireless Power Transfer—An Overview. *IEEE Trans. Ind. Electron.* **2019**, *66*, 1044–1058. doi:10.1109/TIE.2018.2835378.
12. Bozorgchenani, A.; Disabato, S.; Tarchi, D.; Roveri, M. An energy harvesting solution for computation offloading in Fog Computing networks. *Comput. Commun.* **2020**, *160*, 577–587. doi:10.1016/j.comcom.2020.06.032.
13. López, O.L.A.; Alves, H.; Souza, R.D.; Montejo-Sánchez, S.; Fernández, E.M.G.; Latva-Aho, M. Massive Wireless Energy Transfer: Enabling Sustainable IoT Toward 6G Era. *IEEE Internet Things J.* **2021**, *8*, 8816–8835. doi:10.1109/JIOT.2021.3050612.
14. Ponnimbaduge Perera, T.D.; Jayakody, D.N.K.; Sharma, S.K.; Chatzinotas, S.; Li, J. Simultaneous Wireless Information and Power Transfer (SWIPT): Recent Advances and Future Challenges. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 264–302. doi:10.1109/COMST.2017.2783901.
15. Nasir, A.A.; Zhou, X.; Durrani, S.; Kennedy, R.A. Relaying Protocols for Wireless Energy Harvesting and Information Processing. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 3622–3636. doi:10.1109/TWC.2013.062413.122042.
16. Bani Yassein, M.; Alzoubi, O.; Rawasheh, S.; Shatnawi, F.; Hmeidi, I. Features, Challenges and Issues of Fog Computing: A Comprehensive Review. *WSEAS Trans. Comput.* **2020**, *19*, 86–97. doi:10.37394/23205.2020.19.12.
17. Mustafa, E.; Junaid, S.; uz Zaman, S.K.; Jehangiri, A.I.; Din, S.; Rehman, F.; Mustafa, S.; Maqsood, T.; Khan, A.N. Joint wireless power transfer and task offloading in mobile edge computing: A survey. *Clust. Comput.* **2021**. doi:10.1007/s10586-021-03376-3.
18. Vaidya, K.S.; Dethle, C.G.; Akojwar, S. Low Complexity Non Maximally Coefficient Symmetry Multi Rate Filter Bank for Wideband Channelization. *WSEAS Trans. Circuits Syst.* **2021**, *20*, 57–65. doi:10.37394/23201.2021.20.7.
19. Agwil, R.; Tatu, S. Quadrature Sandwich Rectenna for Wireless Power Transfer. *WSEAS Trans. Commun.* **2020**, *19*, 173–178. doi:10.37394/23204.2020.19.19.
20. Khazali, A.; Tarchi, D.; Shayesteh, M.G.; Kalbkhani, H.; Bozorgchenani, A. Energy Efficient Uplink Transmission in Cooperative mmWave NOMA Networks with Wireless Power Transfer. *IEEE Trans. Veh. Technol.* **2022**, *71*, 391–405. doi:10.1109/TVT.2021.3124076.
21. Liu, Y.; Xiong, K.; Zhang, Y.; Zhou, L.; Lin, F.; Liu, T. Multi-Objective Optimization of Fog Computing Assisted Wireless Powered Networks: Joint Energy and Time Minimization. *Electronics* **2019**, *8*, 137. doi:10.3390/electronics8020137.
22. Zheng, H.; Xiong, K.; Fan, P.; Zhou, L.; Zhong, Z. SWIPT-Aware Fog Information Processing: Local Computing vs. Fog Offloading. *Sensors* **2018**, *18*, 3291. doi:10.3390/s18103291.
23. Zheng, H.; Xiong, K.; Fan, P.; Zhong, Z.; Letaief, K.B. Fog-Assisted Multiuser SWIPT Networks: Local Computing or Offloading. *IEEE Internet Things J.* **2019**, *6*, 5246–5264. doi:10.1109/JIOT.2019.2899458.
24. Liu, B.; Xu, H.; Zhou, X. Resource Allocation in Wireless-Powered Mobile Edge Computing Systems for Internet of Things Applications. *Electronics* **2019**, *8*, 206. doi:10.3390/electronics8020206.
25. Feng, J.; Pei, Q.; Yu, F.R.; Chu, X.; Shang, B. Computation Offloading and Resource Allocation for Wireless Powered Mobile Edge Computing with Latency Constraint. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 1320–1323. doi:10.1109/LWC.2019.2915618.
26. Bi, S.; Zhang, Y.J. Computation Rate Maximization for Wireless Powered Mobile-Edge Computing with Binary Computation Offloading. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 4177–4190. doi:10.1109/TWC.2018.2821664.
27. Wang, L.; Shao, H.; Li, J.; Wen, X.; Lu, Z. Optimal Multi-User Computation Offloading Strategy for Wireless Powered Sensor Networks. *IEEE Access* **2020**, *8*, 35150–35160. doi:10.1109/ACCESS.2020.2967559.
28. Li, L.; Xu, G.; Liu, P.; Li, Y.; Ge, J. Jointly Optimize the Residual Energy of Multiple Mobile Devices in the MEC–WPT System. *Future Internet* **2020**, *12*, 233. doi:10.3390/fi12120233.
29. Nasir, A.A.; Zhou, X.; Durrani, S.; Kennedy, R.A. Throughput and ergodic capacity of wireless energy harvesting based DF relaying network. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 4066–4071. doi:10.1109/ICC.2014.6883957.
30. Chen, H.; Li, Y.; Luiz Rebelatto, J.; Uchôa-Filho, B.F.; Vucetic, B. Harvest-Then-Cooperate: Wireless-Powered Cooperative Communications. *IEEE Trans. Signal Process.* **2015**, *63*, 1700–1711. doi:10.1109/TSP.2015.2396009.

31. Qi, Q.; Chen, X.; Zhong, C.; Zhang, Z. Integration of Energy, Computation and Communication in 6G Cellular Internet of Things. *IEEE Commun. Lett.* **2020**, *24*, 1333–1337. doi:10.1109/LCOMM.2020.2982151.
32. Bozorgchenani, A.; Tarchi, D.; Corazza, G.E. Computation Offloading Decision Bounds in SWIPT-Based Fog Networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019. doi:10.1109/GLOBECOM38437.2019.9014108.