IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 16, 2019, accepted April 1, 2019, date of publication April 15, 2019, date of current version April 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2911117

Energy Efficient IoT Virtualization Framework With Peer to Peer Networking and Processing

ZAINEB T. AL-AZEZ, AHMED Q. LAWEY, TAISIR E. H. EL-GORASHI, AND JAAFAR M. H. ELMIRGHANI, (Senior Member, IEEE)
School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, U.K.

Corresponding author: Zaineb T. Al-Azez (elztaa@leeds.ac.uk)

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) for the INTERNET (EP/H040536/1) and STAR (EP/K016873/1) Projects.

ABSTRACT In this paper, an energy efficient IoT virtualization framework with peer-to-peer (P2P) networking and edge processing is proposed. In this network, the IoT task processing requests are served by peers. IoT objects and relays that host virtual machines (VMs) represents the peers in the proposed P2P network. We have considered three scenarios to investigate the saving in power consumption and the system capabilities in terms of task processing. The first scenario is a 'relays only' scenario, where the task requests are processed using relays only. The second scenario is an 'objects only' scenario, where the task requests are processed using the IoT objects only. The last scenario is a hybrid scenario, where the task requests are processed using both IoT objects and VMs. We have developed a mixed integer linear programming (MILP) model to maximize the number of processing tasks served by the system, and minimize the total power consumed by the IoT network. Based on the MILP model principles, we developed an energy efficient virtualized IoT P2P networks heuristic (EEVIPN). Our results show that the hybrid scenario serves up to 77% (57% on average) processing task requests, but with higher energy consumption compared to the other scenarios. The relays only scenario serves 74% (57% on average) of the processing task requests with 8% saving in power consumption compared to the hybrid scenario. In contrast, 28% (22% on average) of task requests can be handled by the objects only scenario with up to 62% power saving compared to the hybrid scenario.

INDEX TERMS IoT, P2P, VMs, energy efficiency.

I. INTRODUCTION

The dramatic recent developments in IoT are mainly driven by the tremendous need and benefits that can be gained from connecting our physical world to the Internet. It is expected that there will be 50 billion (and by some estimates, more) IoT interconnected devices in the coming years [1]. This growth in the number of connected devices opens the doors to new applications, for example in agriculture, transportation, manufacturing, smart homes, smart healthcare, and M2M communications [2], [3]. Many challenges such as energy efficiency, reliability, security, interoperability and scalability have to be overcome before the planned growth in the number and functionalities of IoT can be realized [4]. Given the expected number of devices, one of the most important challenges is energy efficiency and hence greening the associated networks, which grabbed attention in both the academic and industrial domains. Cloud computing is investigated as one of the solutions to the energy efficiency challenge in networks and data centers [5]–[8]. However, with the large data generated by the connected IoT objects (expected to generate 2.3 trillion gigabytes of data every day by year 2020) [2], emerging cloud computing with IoT poses new challenges which have to be addressed. Among these challenges is the hunger for more processing capabilities, high communication bandwidth, security, and latency requirements [9].

A number of solutions were suggested to address these issues. The work started with distributed content placement, thus bringing content closer to users [10], distributed data centers, thus bringing the processing capabilities closer to users and IoT devices [11] and distributed processing of big data, where processing the huge data generated by IoT

The associate editor coordinating the review of this manuscript and approving it for publication was Monjur Mourshed.

devices near the source can extract knowledge from the data and hence transmit the small volume 'extracted knowledge' messages, thus saving network and processing resources and hence energy [12]. However, a different and potentially more efficient solution, advocated here, is to process the IoT data by the IoT objects themselves or by the devices in the nearest layer to these objects. According to Allied Business Intelligence (ABI), it is expected that 90% of the data created by the endpoints will be processed and stored locally rather than being handled by the conventional clouds [10]. Since some complicated data processing tasks cannot be done by most of the IoT devices and sensors because of their limited capabilities, edge computing is proposed to provide more resources to serve such tasks in efficient and fast ways. One of the suggested ways to do this is the dynamic installation of virtual machines (VMs) in the edge cloud to process the raw data generated by the tasks requested by the IoT objects. The processed results are then sent back to the objects [2].

In [13], we considered a single IoT network consisting of IoT network elements (relays, coordinators and gateways). In [13], data processing and traffic aggregation were done by VMs hosted in cloudlets, where these mini clouds are distributed over the IoT network elements. The work was extended in [14] where two separated IoT networks were considered with the deployment of a Passive Optical Access Network (PON). The main goal of our previous work is to investigate the potential energy efficiency gains that can be made if a use is made of distributed cloudlets at the edge of the network compared to centralized cloudlets at highest layer of the implemented model.

There is a recent trend in research toward proposing IoT platforms based on local computing close to the objects such as fog and edge computing. Such platforms have many common characteristics with our proposed architecture. In [15], a combination of fog computing and microgrid is proposed in order to reduce the energy consumed by IoT applications. A set of measurements and experiments were implemented considering different processing and traffic requirements. In [15], dynamic decisions can be made by the proposed IoT gateway to minimize the consumed energy by choosing the most efficient location for processing a task in the fog or in the cloud. This decision is affected by the type of deployed IoT application, weather forecasting and the availability of renewable sources. An edge computation platform is presented in [16] where the design of an IoT gateway virtualized environment for IoT applications is proposed using lightweight virtualization technologies. In this work [16], IoT data processing can be achieved by making use of container-based virtualization technologies such as Docker containers.

Although IoT and wireless sensor networks (WSN) share many features, IoT is a more encompassing concept and term compared to WSN. A WSN can be part of an IoT.in WSN sensors are assumed to communicate wirelessly, therefore a wired set of sensors is strictly not a WSN, but such wired set of sensors can be part of an IoT. In IoT, the devices can be contacted via Internet, which is not a prerequisite in WSN. WSNs focus more on optimizing the use of limited resources, whereas IoT focuses in many cases on quality of service, resilience and data processing as in the current paper. In addition, IoT devices can make use of P2P communication capabilities and architectures [17]. A number of advantages could be realized by using P2P communication systems compared to conventional communication systems such as energy efficiency, traffic reduction [17] and reliability. Based on the potential energy efficiency advantage, we introduce our energy efficient IoT network considering a combination of P2P communication between the IoT objects and edge computing while installing VMs in the relays. Computing tasks and the communication between the peers in our network is achieved through two stages. In the first stage, objects send the requests for tasks to be served by other peers (represented by IoT objects and relays hosting VMs) through the directly connected relays in the network. In the second stage the results of the processed tasks are received. We assume the traffic generated by task requests is reduced after processing by different percentages depending on the complexity of the requested tasks.

The remainder of this paper is organized as follows: In Section II, we describe the MILP model developed to optimize the network and hence construct an energy efficient P2P IoT network. Section III discusses the MILP model results. In Section IV, we introduce our network heuristic and discuss its results. Conclusions are given in Section V.

II. ENERGY EFFICIENT MILP FOR P2P IoT NETWORKS

The MILP model developed considers the architecture shown in Fig. 1. The proposed architecture is constructed of two layers. The first layer represents the IoT objects. The upper layer consists of the relay devices that realize traffic transportation between peers. In our framework, each object is capable of processing three types of tasks that are required by other objects. The task processing capabilities and task requirements for the IoT objects are specified by the MILP model parameters. Each relay node has the ability to host VMs in order to process the tasks requested by IoT objects. The number of relays that can handle all task types is limited to a subset of total number of relays. For example, in the results section we consider a scenario in which 10 out of 25 relays host VMs that can handle all tasks types.

Fig. 1 illustrates all the processing cases we have considered in our P2P platform. Internal processing is shown in case (a), where the object has the ability to process its own request. Consequently, the network power consumption associated with sending the task request to another object or relay or receiving a task result from them will be eliminated. One application of this case might be in smart lights. In case (b), the object sends its task request to the object's neighbor (the directly connected relay device) to be processed by the hosted VM, for example a healthcare device. Some of the objects in our model have the ability to process task requests generated by other objects but considering fairness constraint limitations. The fairness constraint states that each object

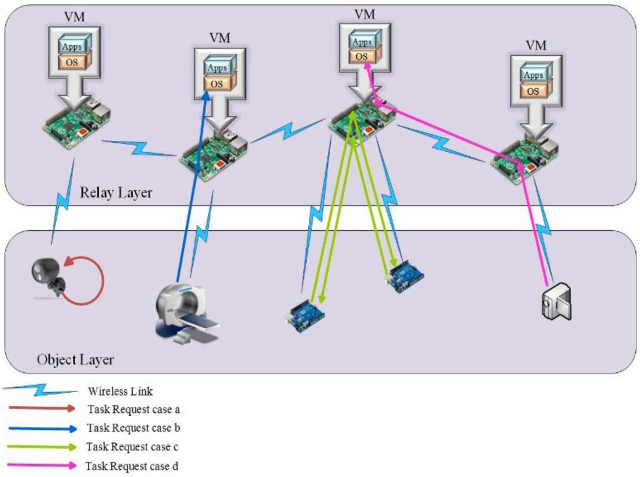

FIGURE 1. The proposed architecture with P2P communication and processing.

should reciprocate equally to other objects choosing it to process its requested task. Object to object communication such as two Arduino devices with different capabilities is illustrated in case (c). The last task processing case is case (d). In this case, none of the objects themselves or the other objects or even the VM hosted by its directly connected relay have the ability to process the requested tasks. In spite of that, relays can process all types of tasks, but the capacity of each relay-processor is limited to a specific maximum workload. So, in order to process this task, the relay sends the task request to other relays to be processed by the nearest possible relay hosting VM (keep in mind that not all the relays host VMs) such as a smart camera sending small size images to be processed.

The MILP model objective consists of two main parts. The first part maximizes the number of logical end-to-end connections between objects and between both relays and other objects. Maximizing this number means maximizing the number of served tasks. The second part of the objective considers minimizing the total power consumption of all elements in our network. The total power consumption in our model is made up of two parts. The first part is the traffic induced power consumption in objects and relays caused by uplink and downlink traffic flow through the network. The uplink traffic is generated by the task requests (the row data) while the downlink traffic is the reduced traffic generated after task processing (the information). The second part of the power consumption equation represents the processing induced power consumption in objects and relays produced by the tasks processing in objects and hosted VMs.

The MILP model objective is subject to many constraints. These constraints are related to VMs placements, fairness constraint between the objects, tasks number limitations, uplink and downlink capacities, and processing capability limitations.

For more clarity in the MILP expressions and notations, we have used superscripts to index the type of variables and parameters, while we have used subscripts as indices of these variables and parameters.

TABLE 1. List of parameters and their definitions.

Notation	Description
O	Set of objects
VM	Set of virtual machines
R	Set of relays
P	Set of peers $(O \cup R)$
TN	Set of all IoT network nodes ($TN = P \cup R$)
P_r^R	Set of peers of relay r
N_a	Set of neighbors of node a
K	Set of tasks
K^p	Subset of tasks that can be served by each peer p. $K^p \subset K$
R_p^P	Neighbor relay of peer p if the peer is an object or R_p^P is peer p if the peer is the hosting relay
Q_{ik}	Task k required by object i
W_k	The workload required by each task k (GHz)
B^V	The number of possible locations occupied by VMs
ψ_j^r	The processor capacity of each relay j (GHz)
Ω_j^r	The maximum power consumed by the processor used in each relay j (W)
ψ_j^O	The processor capacity of each object j ((GHz)
Ω_j^O	The maximum power consumed by each object j ((W)
M_k	The traffic demand of each task k (bit/s) (row data)
C_k	The traffic resulting after processing each task k (bit/s) (information)
L^{DO}	Maximum traffic that can be downloaded by each object (bits/s)
L^{Dr}	Maximum traffic that can be downloaded by each relay (bits/s)
L^{UO}	Maximum traffic that can be uploaded by each object (bits/s)
L^{Ur}	Maximum traffic that can be uploaded by each relay (uploading tasks results) (bits/s)
X_i	Maximum number of upload slots for each object i
E^{elec}	Energy consumed per bit by the electronics of the transceiver (Joules/bit)
D_{mn}	Distance between any node pair in the IoT network (m,n) (meter) $m, n \in TN$
ϵ	Transmission amplifier power coefficient (Joule/bit.m^2)

First, the sets, parameters, and variables of our P2P IoT MILP model are defined in Tables 1 and 2:

The total IoT network power consumption is composed of:

1. The processing induced power consumption of each peer, which can be calculated by summing the workloads of all processed tasks by the peer and multiplying the summation by the energy per processed bit. The processing power in our work is composed of two parts:
   a) Processing induced power consumption of each object:

$$P_j^{op} = \left(\sum_{i \in O, k \in K_j^P} U_{ijk}.W_k \right) . \frac{\Omega_j^O}{\psi_j^O} \quad \forall j \in O \qquad (1)$$

   b) Processing induced power consumption of each relay:

$$P_j^{rp} = \left(\sum_{i \in O, k \in K_j^P} U_{ijk}.W_k \right) . \frac{\Omega_j^r}{\psi_j^r} \quad \forall j \in R \qquad (2)$$

TABLE 2. List of variables and their definitions.

Notation	Description
U_{ijk}	Binary variable which is set to 1 if peer i processes task k requested by object j, otherwise it is set to 0
V_j	Binary variable which is set to 1 if there is a virtual machine in that location otherwise it is set to 0
I_j^{DM}	Download rate (downloading task request) for each peer j (kbps)
I_i^{DC}	Download rate (downloading task result) for each object i (kbps)
I_i^{UM}	Upload rate (uploading task request) for each object i (kbps)
I_j^{UC}	Upload rate (uploading task result) for each peer j (kbps)
λ_{xy}^{Q}	Total traffic passing from relay x (neighbor of source object) to relay y (neighbor of destined object or hosting the destined VM)
λ_{xy}^{S}	Total traffic (tasks result traffic) passing from the relay x (neighbor of source object or source relay) to relay y (neighbor of destined object)
λ_{xyab}^{Q}	Relay to relay traffic (x, y) passing through the link between the intermediate relays pair (a, b)
λ_{xyab}^{S}	Relay to relay traffic (x, y) (tasks results traffic) passing through the link between the intermediate relays pair (a, b)
λ_{ab}^{Q}	Intermediate traffic between any two relays pair (a, b)
λ_{ab}^{S}	Intermediate traffic (tasks results traffic) between any two relays pair (a, b)

2. The traffic induced power consumption, which consists of two basic parts, the sending part and the receiving part. Both parts are based on radio energy dissipation (Friis free-space equation) used in [18].
The energy needed for data transmission and reception, considering path loss is expressed as:
Transmission energy consumption:

$$E_{Tx} = l.E_{elec} + l.\epsilon.d^{\alpha} \tag{3}$$

Receive energy consumption:

$$E_{Rx} = l.E_{Rx-elec} \tag{4}$$

where l is the number of bits, E_{elec} and $E_{Rx-elec}$ are the energy consumed per bit by the electronics in the transmitter and the receiver respectively, ϵ is the transmission factor, d is the distance between the transmitter and the receiver and α is the propagation loss exponent. In our work, $\alpha = 2$ is used as we consider free-space propagation. The transmission power is equal to the bit rate times the propagation energy per bit [18], hence the power consumption is calculated by multiplying the energy per bit by the traffic (bit/s).

a) The traffic induced power consumption of each object:

$$\begin{aligned} P_i^{otr} &= \sum_{j\in P} \sum_{k\in K_j^P : i\neq j} U_{ijk}.M_k.\left(E^{elec} + \epsilon.D_{ig}^2\right) \\ &+ \left(\sum_{j\in O: i\neq j} \sum_{k\in KP_j} U_{jik}.C_k.\left(E^{elec} + \epsilon.D_{ig}^2\right)\right) \\ &+ \sum_{j\in O: i\neq j} \sum_{k\in K_j^P} U_{jik}.M_k.E^{elec} \\ &+ \left(\sum_{j\in P: i\neq j} \sum_{k\in KP_i} U_{ijk}.C_k.E^{elec}\right) \end{aligned}$$

$$g = R_i^P \quad \forall i \in O \tag{5}$$

The first two terms represent the sending power while the third and fourth parts represent the receiving power. The first term calculates the power consumed by each object in sending its requests to other peers in order to process them. The second part represents the power consumed by each object in sending back the results of the tasks processed by itself to the original request generator. The third part represents the power consumed by each object in receiving the task requests from other objects. The last part shows the power consumed by each object in receiving the results of its task requests.

b) Traffic induced power consumption of each relay:

$$\begin{aligned} P_a^{rtr} &= \sum_{b\in N_a\cap R: a\neq b} \left(\lambda_{ab}^{Q}.\left(E^{elec} + \epsilon.D_{ab}^2\right)\right) \\ &+ \sum_{b\in N_a\cap R: a\neq b} \left(\lambda_{ab}^{S}.\left(E^{elec} + \epsilon.D_{ab}^2\right)\right) \\ &+ \sum_{j\in P_a^R\cap O} \left(I_j^{DM}.\left(E^{elec} + \epsilon.D_{aj}^2\right)\right) \\ &+ \sum_{i\in P_a^R\cap O} \left(I_i^{DC}.\left(E^{elec} + \epsilon.D_{ai}^2\right)\right) \\ &+ \sum_{b\in N_a\cap R: a\neq b} \left(\lambda_{ba}^{Q}.E^{elec}\right) \\ &+ \sum_{b\in N_a\cap R: a\neq b} \left(\lambda_{ba}^{S}.E^{elec}\right) \\ &+ \sum_{i\in P_a^R\cap O} I_i^{UM}.E^{elec} + \sum_{j\in P_a^R\cap O} I_j^{UC}.E^{elec} \\ &\forall a \in R \end{aligned} \tag{6}$$

Traffic induced power consumption of the relays P_a^{rtr} consists of 8 terms. The first four terms represent the sending power and the last four terms represent the receiving power. The first and second terms represent the power consumed in sending the task requests and task results respectively from a relay to another relay. The third and fourth terms calculate the power consumed in sending the task requests and task results respectively to the objects directly connected to that relay. The fifth and sixth terms describe the power consumed in receiving task requests and task results respectively by each relay from another neighbor relay. The seventh term calculates the power consumed by each relay in receiving the task requests from the directly connected object while the last term represents the power consumed by each relay in receiving the task results from other peers (directly connected object to the relay or relay hosting VM).

Objective: **Maximize**

$$\left(\sum_{i\in O, j\in P, k\in K_j^P} F.U_{ijk}\right) - \left(\sum_{j\in O} P_j^{op} + \sum_{j\in R} P_j^{rp}\right) - \left(\sum_{i\in O}\left(P_i^{otr}\right) + \sum_{a\in R}\left(P_a^{rtr}\right)\right) \quad (7)$$

Equation (7) gives the model objective where the number of logical end-to-end connections between objects and other peers is maximized while the network power consumption and the processing power consumption are minimized. The parameter F takes care of the units and is also used to scale the number of connections so that they become comparable in magnitude to the consumed power.

Constraints

Subject to:

1. U indicator setting constraints

$$\sum_{j\in P} U_{ijk} \leq Q_{ik} \quad \forall i \in O,\ \forall k \in K \quad (8)$$

Constraint (8) ensures that only one peer (one object or one relay) can serve each request of each object.

2. Fairness constraints

$$\sum_{k\in K_j^P} U_{ijk} = \sum_{k\in K_i^P} U_{jik} \quad \forall i \in O,\ \forall j \in O \quad (9)$$

Constraint (9) is the fairness constraint which ensures that each object reciprocates equally to other objects that serve a request of this object.

3. Virtual Machine Calculations constraints

$$\sum_{i\in O, k\in K_j^P} U_{ijk} \geq V_j \quad \forall j \in VM \quad (10)$$

$$\sum_{i\in O, k\in K_j^P} U_{ijk} \leq A.V_j \quad \forall j \in VM \quad (11)$$

$$\sum_{j\in VM} V_j = \mathrm{B}^{\mathrm{v}} \quad (12)$$

Constraints (10) and (11) locate a virtual machine in an appropriate relay in order to process the requested tasks. Constraint (12) limits the number of selected locations occupied by the virtual machines to 10 only out of 25 possible locations.

4. Processing power consumption calculations

$$\sum_{i\in O, k\in K_j^P} U_{ijk}.W_k \leq \psi_j^O \quad \forall j \in O \quad (13)$$

$$\sum_{i\in O, k\in K_j^P} U_{ijk}.W_k \leq \psi_j^r \quad \forall j \in R \quad (14)$$

Constraints (13) and (14) ensure that the summation of the whole workloads of processed tasks by each object and each relay respectively do not exceed its maximum processing workload capability

5. Traffic calculations and capacity constraints

$$\lambda_{xy}^Q = \left(\left(\sum_{i\in P_x^R\cap O}\left(\sum_{j\in P_y^R : i\neq j}\left(\sum_{k\in K_j^P} U_{ijk}.M_k\right)\right)\right)\right) \quad \forall x \in R,\quad \forall y \in R : x \neq y \quad (15)$$

$$\lambda_{xy}^S = \left(\left(\sum_{i\in P_x^R}\left(\sum_{j\in P_y^R\cap O : i\neq j}\left(\sum_{k\in K_j^P} U_{ijk}.C_k\right)\right)\right)\right) \quad \forall x \in R,\quad \forall y \in R : x \neq y \quad (16)$$

$$\sum_{b\in N_a\cap R : a\neq b} \lambda_{xyab}^Q - \sum_{b\in N_a\cap R : a\neq b} \lambda_{xyba}^Q = \begin{cases} \lambda_{xy}^Q & if\ a = x \\ -\lambda_{xy}^Q & if\ a = y \\ 0 & otherwise \end{cases} \quad \forall x \in R,\quad \forall y \in R,\ \forall a \in R : x \neq y \quad (17)$$

$$\sum_{b\in N_a\cap R : a\neq b} \lambda_{xyab}^S - \sum_{b\in N_a\cap R : a\neq b} \lambda_{xyba}^S = \begin{cases} \lambda_{xy}^S & if\ a = x \\ -\lambda_{xy}^S & if\ a = y \\ 0 & otherwise \end{cases} \quad \forall x \in R, \forall y \in R,\quad \forall a \in R : x \neq y \quad (18)$$

$$\lambda_{ab}^Q = \sum_{x\in R}\sum_{y\in R : x\neq y} \lambda_{xyab}^Q \quad \forall a \in R,\quad b \in N_a \cap R : a \neq b \quad (19)$$

$$\lambda_{ab}^S = \sum_{x\in R}\sum_{y\in R : x\neq y} \lambda_{xyab}^S \quad \forall a \in R,\quad b \in N_a \cap R : a \neq b \quad (20)$$

Constraints (15) and (16) calculate the transient traffic between relays due to P2P traffic (task requests and the results traffic). Constraints (17) and (18) represent the flow conservation of the traffic between the source relay (requester's (object) neighbor) and the destination relay (serving peer's neighbor or host) through the intermediate relays. Constraints (19) and (20) calculate the traffic flows through each intermediate relay.

$$I_j^{DM} = \sum_{i\in O, k\in K_j^P : i\neq j} U_{ijk}.M_k \quad \forall j \in P \quad (21)$$

$$I_i^{DC} = \sum_{j\in P, k\in K_j^P : i\neq j} U_{ijk}.C_k \quad \forall i \in O \quad (22)$$

$$I_j^{DM} \leq L^{DO} \quad \forall j \in O \quad (23)$$

$$I_i^{DC} \leq L^{DO} \quad \forall i \in O \quad (24)$$

$$I_j^{DM} \leq L^{DR} \quad \forall j \in R \tag{25}$$

Constraint (21) calculates the download rate of each peer by summing the received traffic demand of each requested task from other objects selected to serve them. Constraint (22) calculates the download rate of the reduced traffic (resulting information) received by each object. Constraints (23), and (24) limit the download rate of each object to its maximum value, while constraint (25) limits the download rate of each relay to its maximum value.

$$I_i^{UM} = \sum_{j \in P, k \in K_j^P : i \neq j} U_{ijk}.M_k \quad \forall i \in O \tag{26}$$

$$I_j^{UC} = \sum_{i \in O, k \in K_j^P : i \neq j} U_{ijk}.C_k \quad \forall j \in P \tag{27}$$

$$I_i^{UM} \leq L^{UO} \quad \forall i \in O \tag{28}$$

$$I_j^{UC} \leq L^{UO} \quad \forall j \in O \tag{29}$$

$$I_j^{UC} \leq L^{UR} \quad \forall j \in R \tag{30}$$

$$\sum_{j \in P, k \in K_j^P : i \neq j} U_{ijk} \leq X_i \quad \forall i \in O \tag{31}$$

$$\sum_{i \in O, k \in K_j^P : i \neq j} U_{ijk} \leq X_i \quad \forall j \in O \tag{32}$$

Constraint (26) calculates the upload rate of each object by summing the uploaded task traffic demands. While constraint (27) calculates the upload rate of each peer that results from sending the reduced traffic (the resulting information from task processing). Constraints (28), and (29) limit the upload rate of each object to its maximum value while constraint (30) limits the upload rate of each relay to its maximum value. Constraints (31) and (32) limit the number of upload slots of each object.

III. MILP MODEL EVALUATION AND RESULTS

Our IoT nodes, depicted in Fig. 1, consist of 25 objects and 25 relays distributed over an area of 30m × 30m [19]. The objects are distributed randomly while the relays are distributed uniformly, every 6m as shown in Fig. 2. The Zigbee protocol is used to implement wireless communication between the network elements. It is defined by IEEE 802.15.4 [20]. We have considered operation of the transceivers at 250 kb/s data rate [21] and all the other parameters meet IEEE 802.15.4 standard. It should be noted that Zigbee networks or wireless sensor networks cannot be generalized to IoT networks. But they can be considered as part of the IoT network once the traffic of these networks is aggregated through IP networks. We have assumed that the Zigbee network is a segment of a large network that is the IP network.

Table 3 lists the model input parameters. We have used the Arduino 101 as an IoT object as it is one of the most power efficient processors with a higher clock speed compared to other types of Arduino [22]. Arduino 101 is referred to as Genuino 101 outside USA [23]. We used the Raspberry pi 3 in the relays, with processing capability of 1.2 GHz [24]. It is

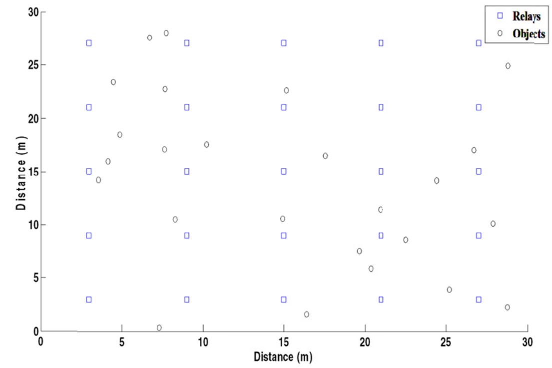

FIGURE 2. IoT objects and relays distribution.

worth mentioning that the wireless communication between the raspberry pi and Arduino is implemented using auxiliary units (such as the XBee shield) that supports the Zigbee protocol. The XBee shield is connected to the GPIOs pins of the Raspberry pi or to the analog or the digital pins of the Arduino depending on the XBee shield design. In addition, Running VMs and non-Raspbian applications on the Raspbian operating system can be implemented using third party software such as Quick EMUlator (QEMU) or ExaGear. However, such an implementation is beyond the scope of this paper and we will consider a real-time experiment in the future. We assumed the traffic demand of the first task is 250 bit/s representing applications with small traffic volume in the range 0-250 bit/s. We assumed other values of traffic close to this one in a consistent way to comply with the link capacity limit constraint and to be very close to practical IoT applications. The data rates thus considered were 240b/s representing a heartbeat sensor and 2.4 kb/s associated with blood glucose level sensors and temperature readings [25]. The range of traffic values considered resulted in heterogeneous tasks that have to be tackled by our optimization model [26], [27]. In Bit Torrent, the typical value for the maximum number of upload slots for each peer is 4 [28]. We have considered, in our P2P communication system, a range of different numbers of upload slots from 1 to 10 slots per object. We found that the average value of upload slots that ensures the highest percentage of executed tasks is 4.

As alluded earlier, we have considered three scenarios. The first scenario is the relays only scenario. This restricts the processing of all requested tasks to 10 VMs out of 25 possible locations. This scenario is implemented by setting the number of end-to-end connections between the objects to zero, to ensure that no objects respond to any task request, i.e. equation (33):

$$\sum_{j \in O, k \in K_j^P} U_{ijk} = 0 \quad \forall i \in O \tag{33}$$

The second scenario is the objects only scenario which restricts the processing of the requested tasks by the IoT to objects only. This scenario is implemented by setting the total

TABLE 3. MILP model input parameters.

Parameter Description	value
Energy per bit consumed by the electronics of the transceiver (E^{elec})	50 nJ/bit [29]
Maximum download rate of each peer (objects and relays) (L^{DO} and L^{DR})	10 kb/s, 25 kb/s [26,29,30]
Maximum upload rate of each (objects and relays) (L^{UO}& L^{UR})	5 kb/s, 25 kb/s [29,31,32]
The processor capacity of object (ψ_j^O)	32 MHz [22]
The processor Capacity of relay (ψ_j^R)	1.2 GHz [24]
CPU maximum power consumption in objects (Ω_j^O)	347 mW [22]
CPU maximum power consumption in relay (Ω_j^R)	3.7 W [24]
Transmission amplifier power coefficient (ϵ)	255 pJ/bit.m^2 [29]
The requested workload for each task (W_k), $k \in K$	0.01 GHz, 0.012 GHz, 0.015 Hz, 0.02 GHz, 0.05 GHz, 0.1 GHz, 0.2GHz, 0.3 GHz, 0.4 GHz, 0.5GHz [33]
Traffic generated by each task request (M_k), $k \in K$	250b/s, 500b/s, 750b/s, 1000b/s, 1250b/s, 1750b/s, 2000b/s, 2250b/s, 2500b/s, 2750b/s [25-27]
Traffic generated by each task result after reduction (C_k), $k \in K$	25b/s, 100b/s, 225b/s, 400b/s, 625b/s, 1050b/s, 1400b/s, 1800b/s, 2125b/s, 2475b/s
Maximum number of upload slots for each peer (X_i)	4 [28]
IoT nodes distribution area	30m × 30m [21]
Range of task weight (F) for all scenarios	{0, 0.1, 0.2, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8}
Scale factor with large value (M)	1000000

number of VMs to zero (it actually means the number of relays hosting VMs equals zero), i.e. equation (34):

$$\sum_{j\in VM} V_j = 0 \tag{34}$$

The last scenario allows cooperation between the relays hosting VMs and the objects in order to process the requested tasks. Fig. 3 shows the processing induced power consumption of the three scenarios. The x axis represents the range of different values of task weights F multiplied by the variable U as shown in (7) (the objective function). This range is used to scale the number of connections to be comparable to the amount of consumed power.

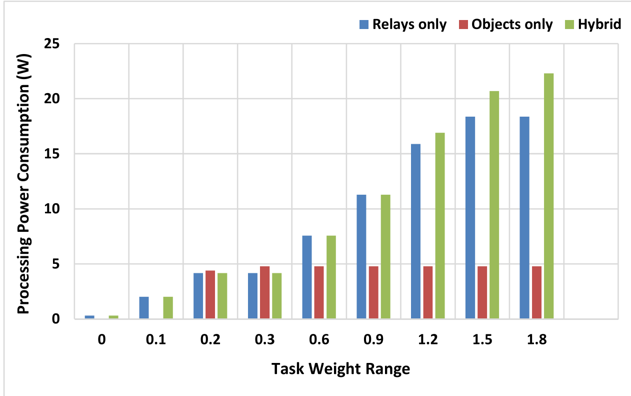

FIGURE 3. Total processing induced power consumption in the three scenarios.

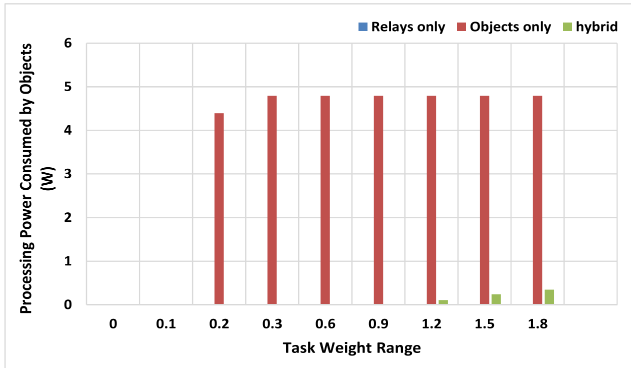

FIGURE 4. Processing induced power consumption by objects in the three scenarios.

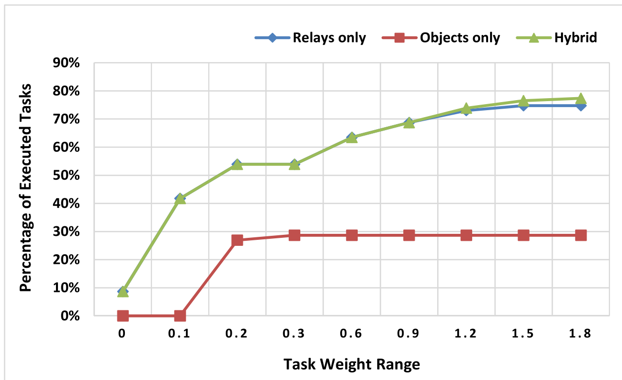

FIGURE 5. Percentage of executed tasks by the three scenarios.

Fig. 3 shows that the hybrid and relays only scenarios consume the same amount of processing induced power at task weight values in the range ($F = 0 \sim 0.9$) as there are no tasks executed by the objects in the hybrid scenario at these values as shown in Fig. 4. The inefficiency of the objects-processors used and the effect of the power optimization at

TABLE 4. Tasks execution map at task weight (F=1.8).

Task ID	Total No. of Task Requests	Total No. of Served Tasks		
		Objects Only Scenario	relays Only Scenario	Hybrid Scenario
k_1	15	12	15	15
k_2	10	6	10	10
k_3	15	10	15	15
k_4	8	5	8	8
k_5	14	0	14	14
k_6	11	0	11	11
k_7	9	0	6	6
k_8	13	0	5	6
k_9	11	0	2	3
k_{10}	9	0	0	1

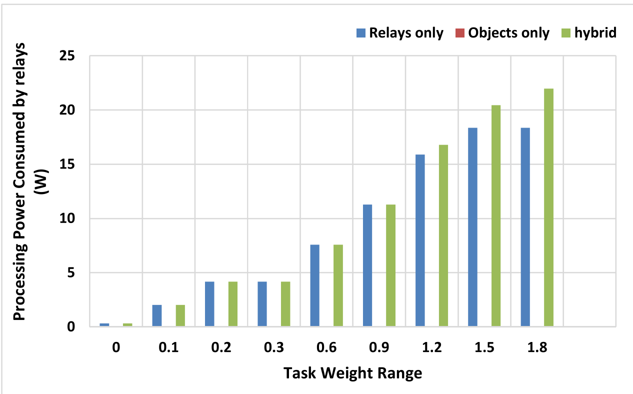

FIGURE 6. Processing induced power consumption by relays in three scenarios.

such low scale factor result in blocking the requested tasks instead of implementing them by the objects. The power inefficiency of a processor used in object processing only is clearly illustrated in Fig. 5 at task weight values ($F = 0.2$ and $F = 0.3$). At these values, the objects only scenario executes less tasks than the other two scenarios (about half), but it consumes more processing power than both of them as shown in Fig. 3.

Starting at $F = 0.3$ and up to the highest task weight, the objects only scenario consumes the same amount of processing induced power. The low utilization of the P2P layer in the objects only scenario is attributed to two reasons. Firstly, the effect of the fairness constraint and secondly, the most effective reason is the low capacity of the processors in the IoT objects. This low capacity is clearly seen in Table 4 as the objects in the objects only scenario drop tasks (5 to 10) as the workloads of these tasks are larger than the processor capacity of the objects (ψ_j^O) as shown in Table 3.

A general trend followed by both hybrid and the relays only scenarios towards higher power consumption for higher task weights can be seen in Fig. 3. Starting from task weight $F = 1.2$ a small gap is observed between the two scenarios and this grows as the task weight increases. This gap is caused by the higher power consumption of the hybrid scenario compared to the relays only scenario because of the internal processing of the objects in the hybrid scenario. Due to the limited number of upload slots available for each object, an object tends to process its requests internally instead of using the free upload slots. Accordingly, internal processing allows the objects to send more task requests with higher workloads to relays to be processed. Therefore, the relays in the hybrid scenario consume more processing induced power than the relays in the relays only scenario as shown in Fig. 6.

To clarify, we consider task k_9 in Table 4 as an example. In the hybrid scenario, task k_9 is requested by objects 8, 15 and 25 and in the relays only scenario are only requested by objects 15 and 25 but not by 8. This means that the request by object no. 8 is blocked. Therefore, by checking object 8, we notice:

1. Object 8 in the hybrid scenario processes internally task request k_2 and sends k_1, k_8 and k_9 task requests to other peers. The total generated traffic as a result of sending all these requests is 5000 b/s which is the maximum limit of the upload capacity of each object (the traffic generated by each task request is illustrated in Table 3).
2. Obviously, in the relays only scenario, internal processing is not allowed, therefore object 8 sends requests k_1, k_2 and k_8 to relays hosting VMs while task request k_9 is blocked. The total upload traffic due to requests is 3000 b/s which leaves only 2000 b/s of allowed traffic that can be uploaded by object 8. This (ie 2000 b/s) is not enough to transmit k_9 and that results in blocking this request instead of sending it to be served. In addition, blocking k_9 by object 8 in particular is due to the power optimization and its impact on the behavior of the object. Since the object tries to send tasks with the lowest processor workload and lowest traffic demand requirements to be served by other peers, this results in blocking k_9.

As a result of the power optimization, there is a general pattern followed by the objects in our network when they send their requests to be served by other peers. First, to make sure that the objects requests are satisfied using the lowest processing and network power consumption, objects search for the nearest available relays hosting VMs starting with ones that are directly connected to the object (objects' neighbors) then the circle of search is increased to include other relays starting from the nearest to the furthest. The implication is that the results in Fig. 7 show that the traffic induced power consumed by relays is more than the power used by the objects. This difference increases with increase in the task weight in both hybrid and relays only scenarios. In the hybrid scenario, when the model starts serving more tasks than the relays hosting VMs can handle because of traffic and processing capacity constraints, the objects serve tasks using their own processors (internal processing) as shown in Fig. 4. This starts at $F = 1.2$ and continues beyond. Given that it is internal processing, it is of interest to understand the drivers behind the increase in the traffic induced power consumption in the relays. In this

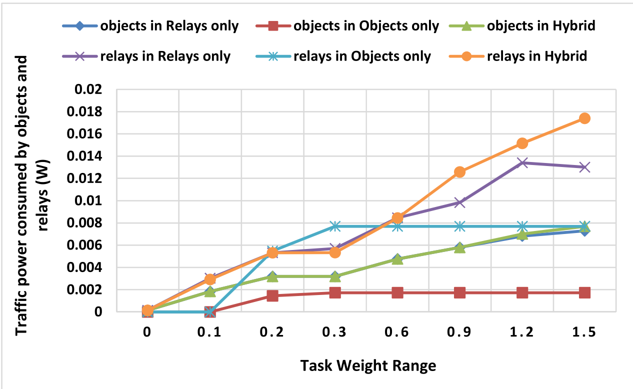

FIGURE 7. Traffic induced power consumption by objects and relays in three scenarios.

scenario, the internal processing affects the relays behavior resulting in serving more tasks with higher workload. Sending task requests with high workloads to relays hosting VMs results in consuming more traffic induced power by relays. In the objects only scenario, the objects either serve task requests using their own resources if they able to, or send the requested tasks to other objects to be served. Sending tasks to other objects while satisfying the fairness constraint can lead to sending the requests to remote objects. This results in higher network power consumption in the relays. Consequently, the traffic induced power consumption in the relays in the objects only scenario is higher than the power consumption in the objects only scenario. It is even higher than the power consumption in the relays in other scenarios as illustrated in Fig. 7 at task weight value $F = 0.3$. However, as discussed earlier, the low capacity of the processor used in IoT objects results in low and constant serving tasks rate for other values of task weight range. This leads in turn to a constant consumption of traffic induced power for all devices in our network in the objects only scenario.

After considering the processing and traffic induced power consumption of the three scenarios, it is clearly seen that the hybrid scenario consumes the highest amount of total power compared to the other scenarios. Moreover, the relays only scenario consumes a comparable power with 8% power saving. Finally, the objects only scenario has the least total power consumption with power saving up to 62% compared to the hybrid scenario.

IV. ENERGY EFFICIENT P2P IoT NETWORKS HEURISTIC and RESULTS

In this section, an energy efficient virtualized IoT P2P networks (EEVIPN) heuristic algorithm has been developed for real time implementation and to verify the MILP model results. The EEVIPN heuristic is illustrated in Fig. 8. It considers the hybrid scenario as it is the generic scenario that can be used to build other scenarios such as the relays only and the object only scenarios. To determine the total power consumption (TPC), the heuristic determines the type and the optimum place of the peer to be used to serve the processing

Inputs: $\boldsymbol{O} = \{\mathbf{1} \ldots \boldsymbol{NO}\}$
$\boldsymbol{K} = \{\mathbf{1} \ldots \boldsymbol{NK}\}$
$\boldsymbol{R} = \{\mathbf{1} \ldots \boldsymbol{NR}\}$
Output: No. of Served Tasks
Total Power Consumption (TPC)

For each task k $\in K$ Do
  For each object requesting a task $i \in O$ Do
    For each candidate relay hosting VM that can serve a requested task $j \in R$ Do
      If all serving constraints are met Then
        U(i,j,k)=1
        Calculate P_j^{rp}
      End If
    End For
  End For
End For
For each task k $\in K$ Do
  For each object requesting a task $i \in O$ Do
    For each candidate object that can serve a task $j \in O$ Do
      Case $(i = j)$
        If all serving constraints are met Then
          U(i,j,k)=1
          Calculate P_j^{op}
        End If
      End Case
      Case $(i \neq j)$
        If all serving constraints are met Then
          Do Tit for Tat
          U(i,j,k)=1
          Calculate P_j^{op}
        End If
      End Case
    End For
  End For
End For
For Each IoT object $i \in O$ Do
  Calculate P_i^{otr}
End For
For Each relay sending and receiving traffic to and from other relays $(a \in R)$
  Calculate P_a^{rtr} based on minimum hop path between node pair (x,y)
End For
For each relay receiving task requests from objects and sending task results to objects $(a \in R)$
  Calculate P_a^{rtr}
End For
  Calculate no. of served tasks

$$NST = \sum_{i \in O} \sum_{j \in P} \sum_{k \in K} U(i,j,k)$$

  Calculate total power consumption

$$TPC = \sum_{j \in O} P_j^{op} + \sum_{j \in VM} P_j^{rp} + \sum_{i \in O} P_i^{otr} + \sum_{a \in R} P_a^{rtr}$$

FIGURE 8. P2P IoT heuristic.

tasks according to the serving constraints of each peer. The serving constraints can be summarized as follows:

i. The processing task should not have been served by any other peer before.
ii. The upload traffic of each candidate peer should not exceed the maximum limit.
iii. The download traffic of each candidate peer should not exceed the maximum limit.

iv. The upload slots of each object should not exceed the specified maximum number.
v. The number of candidate relays hosting VMs should not exceed the specified maximum number of serving relays.
vi. There should be sufficient processor capacity in each candidate peer to accommodate the processing task workload.

Recall that these are the general serving constraints and could be changed according to the type of the serving peer. For example, if the candidate peer is a relay then all the serving constraints should be considered. If the candidate peer is an object (not the task requester), constraint (v) will not be applied. For the internal processing scenario, the heuristic should check constraints (i) and (vi) because the requested task is served by the requester object internally and as a result, there will be no external data processing neither traffic flow.

For each task requested by an object, the heuristic first checks all the candidate relays hosting VMs in the network. Starting with relays, is an attempt by the heuristic to mimic the MILP model behavior at the lowest values of task weight, by looking for candidate relays as serving peers. The heuristic first checks relays hosting VMs due to the power efficiency of their processors compared to the power efficiency of the objects only processors. It also checks the relays first due to their high ability to serve all types of requested processing tasks. The serving constraints of the first candidate relay are investigated by the heuristic. If all these constraints are met, then the link between the requester and the serving relay is set. The requested task is served and the processing power P_j^{Rp} of each relay is calculated. The heuristic loops for the rest of the relays hosting VMs for all requested tasks by all objects. It finally calculates all the processing induced power of all serving relays. If the requested task is served by an object, there are two cases, the first case represents internal processing. In the second case, the object serves another object. In this case, the Tit-for-Tat constraint (the fairness constraint, equation (9)) should be applied to guarantee equal reciprocity between the two objects intending to serve each other. In both cases, if all serving constraints are met then the link between the requester and the serving object is set. The candidate object serves the requested processing task and the processing induced power consumed by the object-processer P_j^{op} is calculated. After checking all the possible serving peers for all requested tasks by all requesting objects, the traffic induced power consumption of each object P_i^{otr} is calculated. In addition, the power consumption of each relay P_a^{rtr} caused by cross traffic between the requesting objects and the serving peers is calculated. The traffic induced power consumption of each relay is composed of two basic parts. The first represents the power consumption due to traffic flowing between relays. The heuristic tries to route the traffic between node x (the directly connected relay to the requesting object) and node y (the directly connected relay to the serving object or hosting the serving VM) by using a minimum hop algorithm in order to minimize the traffic induced power consumption of each relay. The other part of P_a^{rtr} is the network power consumption due to the traffic flowing between relays and the request generator and serving objects. Finally, the heuristic calculates the number of served tasks by all peers NST and the total power consumption TPC.

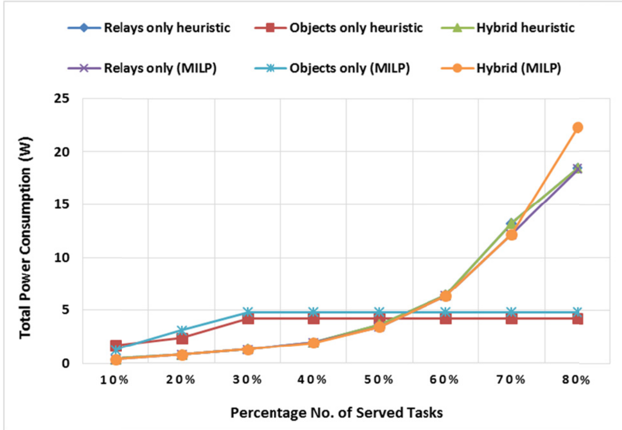

FIGURE 9. Total power consumption evaluated using heuristic and MILP model.

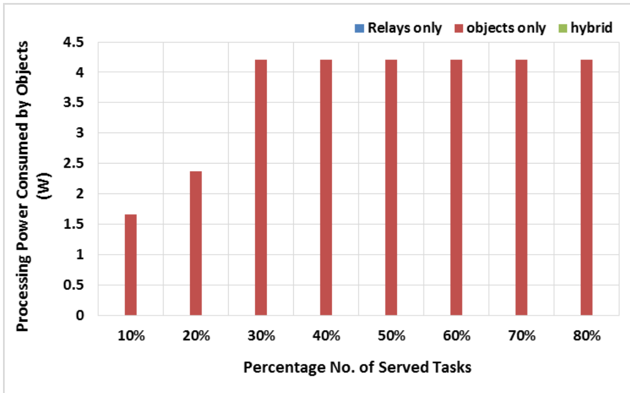

FIGURE 10. Processing induced power consumption of objects.

Fig. 9 presents the total power consumption of both MILP and EEVIPN heuristic versus the percentage of served tasks. It is clearly seen that the power consumption of the MILP and EEVIPN heuristic are comparable. The highest percentage of served tasks that can be achieved is 77% by the hybrid scenario in the MILP model. Therefore, we do not show results beyond 80% of served tasks as these cases will consume the same amount of power. It should be noted that in the hybrid scenario the MILP model consumes higher power than the heuristic when serving higher than 70% of the requested tasks because of the higher VMs utilization as clearly shown in Fig. 11. The higher utilization of VMs results from the internal processing by the objects at higher percentage of tasks execution as mentioned before in the discussion of the results in Fig. 3. There are no tasks served by the objects in the hybrid scenario in the heuristic as illustrated by Fig. 10. In the hybrid scenario, tasks with small workloads are served by relays as the heuristic starts task assignment with relays.

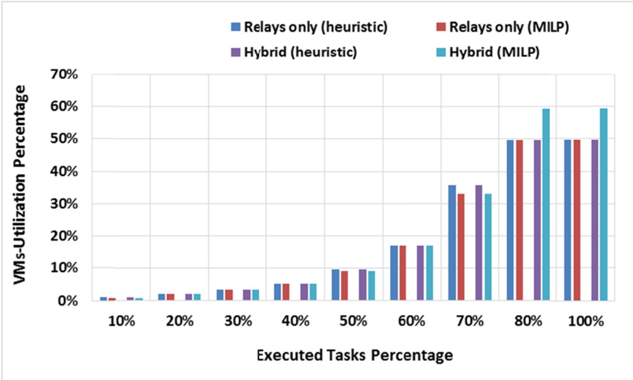

FIGURE 11. VMs- utilization in hybrid and relays only scenarios.

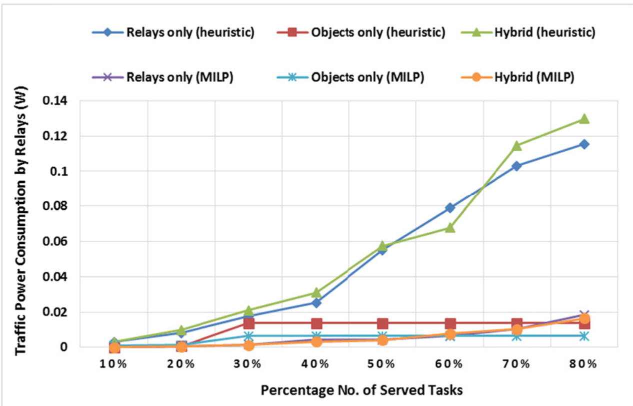

FIGURE 12. Traffic induced power consumption of Relays.

After that, the heuristic tends to assign the remaining tasks (unserved) to objects where the tasks have workload requirements higher than the objects capabilities. As such, objects are not exploited in this scenario. Moreover, this results in both the hybrid scenario and relays only scenario (in heuristic) following the same behavior in executing tasks. This results in the two scenarios consuming the same amount of power as clearly shown in Fig. 9. Fig. 9 shows that the objects only scenario (heuristic) consumes higher power than MILP model. This small difference is attributed to the impact of the network power consumption and specifically the power consumed by the relays as shown in Fig. 12. In the MILP model (objects only scenario), if the tasks are not served internally by the objects then the model optimizes the choice of the serving objects according to the fairness constraint in addition to the distances from the requesting objects to the serving objects in order to reduce the power consumption. In the heuristic, the search for serving objects is carried out sequentially regardless of their locations. This results in the relays consuming more power especially in cases where the tasks are sent to remote serving objects. A similar observation can be made about the difference between the power consumed by relays (due to traffic) in both hybrid and relays only scenarios. In the heuristic, the relays consume higher traffic induced power than in the MILP. This is similar to the objects only scenario. It is also caused by sending the requests far apart in order to be served by the candidate serving relays.

V. COMPLEXITY ANALYSIS

An EEVIPN heuristic algorithm has been developed to overcome the non-deterministic polynomial time hardness (NP-hard) problem associated with our linear programming as proven in [34]. The EEVIPN heuristic is based on two main processes: (i) VMs accommodation in relays to serve objects requested tasks, and (ii) routing the traffic between network nodes. To accommodate VMs in relays and serve objects requested tasks, three loops are needed with $(\boldsymbol{k} \times \boldsymbol{O} \times \boldsymbol{R})$ iterations, where k is the number of requested tasks, O is the number of objects, and R is the number of relays. The number of relays is equal to the number of objects in the developed heuristic algorithm and they are very large compared to the total number of tasks. Therefore, the time complexity to serve the objects requests can be expressed as $O(\boldsymbol{N}^2)$ where N is the total number of relays/objects. On the other hand, routing the traffic between network nodes is based on the minimum hop algorithm which has time complexity $O(N)$ [35]. Therefore, the total time complexity is a polynomial time complexity expressed as $O(\boldsymbol{N}^2)$. The heuristic results were evaluated when EEVIPN algorithm was executed in an Intel ®Core i5, 2.7 GHz processor with 16 GB RAM.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated the energy efficiency of an IoT virtualization framework with P2P network and edge computing. This investigation has been carried out by considering three different scenarios. A MILP was developed to maximize the number of processing tasks served by peers and minimize the total power consumption of the network.

Our results show that the hybrid scenario serves up to 77% (57% on average) processed task requests, but with higher energy consumption compared with other scenarios. The relays only scenario can serve 74% (57% on average) of the processing task requests with 8% of power saving and 28% (22% on average) of task requests can be successfully handled by applying the objects only scenario with 62% power saving. The results also revealed the low percentage of addressed task requests in the objects only scenario resulting from the capacity limit of the IoT objects' processors. In addition, the small difference between the serving percentage of hybrid scenario and relays only scenario resulted from the allowed internal processing of objects in the hybrid scenario.

For real time implementation, we have developed the EEVIPN heuristic based on the MILP model concepts. The heuristic achieved a comparable power efficiency and comparable number of executed tasks to the MILP model. The hybrid Scenario in the heuristic executes up to 74% of the total tasks (MILP 77%), up to 74% of tasks by the relays only scenario (MILP 74%) while the objects only scenario executes up to 21% of the tasks (MILP 28%).

It should be noted that due to channel impairment and/or network congestion, link failures may occur, and hence retransmissions may become necessary. These retransmissions can have an impact on power consumption and therefore it is of interest to study the impact of resilience on energy consumption.

ACKNOWLEDGMENTS

The first author Dr. Zaineb Al-Azez would like to thank Dr. Ahmed Al-Quzweeni for many helpful discussions and the Higher Committee for Education Development in Iraq (HCED) for funding her PhD scholarship. All data are provided in full in the results section of this paper.

REFERENCES

[1] F. Ganz, D. Puschmann, P. Barnaghi, and F. Carrez, "A practical evaluation of information processing and abstraction techniques for the Internet of Things," *IEEE Internet Things J.*, vol. 2, no. 4, pp. 340–354, Aug. 2015.

[2] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[4] S. H. Shah and I. Yaqoob, "A survey: Internet of Things (IOT) technologies, applications and challenges," in *Proc. IEEE Smart Energy Grid Eng. (SEGE)*, Oshawa, ON, Canada, Aug. 2016, pp. 381–385.

[5] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient virtual network embedding for cloud networks," *J. Lightw. Technol.*, vol. 33, no. 9, pp. 1828–1849, May 1, 2015.

[6] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "BitTorrent content distribution in optical networks," *J. Lightw. Technol.*, vol. 32, no. 21, pp. 3607–3623, Nov. 1, 2014.

[7] X. Dong, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Green IP over WDM networks with data centers," *J. Lightw. Technol.*, vol. 29, no. 12, pp. 1861–1880, Jun. 11, 2011.

[8] X. Dong, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "On the energy efficiency of physical topology design for IP over WDM networks," *J. Lightw. Technol.*, vol. 30, no. 12, pp. 1931–1942, Jun. 1, 2012.

[9] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[10] N. I. Osman, T. El-Gorashi, L. Krug, and J. M. H. Elmirghani, "Energy-efficient future high-definition TV," *J. Lightw. Technol.*, vol. 32, no. 13, pp. 2364–2381, Jul. 1, 2014.

[11] A.Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Distributed energy efficient clouds over core networks," *J. Lightw. Technol.*, vol. 32, no. 7, pp. 1261–1281, Apr. 1, 2014.

[12] A. M. Al-Salim, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient big data networks: Impact of volume and variety," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 458–474, Mar. 2018.

[13] Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Virtualization framework for energy efficient IoT networks," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Niagara Falls, ON, Canada, Oct. 2015, pp. 74–77.

[14] Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient IoT virtualization framework with passive optical access networks," in *Proc. 18th Int. Conf. Transparent Opt. Netw. (ICTON)*, Trento, Italy, Jul. 2016, pp. 1–4.

[15] F. Jalali, A. Vishwanath, J. de Hoog, and F. Suits, "Interconnecting Fog computing and microgrids for greening IoT," in *Proc. IEEE Innov. Smart Grid Technol.-Asia (ISGT-Asia)*, Melbourne, VIC, Australia, Nov. 2016, pp. 693–698.

[16] R. Morabito and N. Beijar, "Enabling data processing at the network edge through lightweight virtualization technologies," in *Proc. IEEE Int. Conf. Sens., Commun. Netw. (SECON Workshops)*, London, U.K., Jun. 2016, pp. 1–6.

[17] U. Urosevic and Z. Veljovic, "Distributed MIMO solutions for peer-to-peer communications in future wireless systems," in *Proc. 24th Telecommun. Forum (TELFOR)*, Belgrade, Serbia, Nov. 2016, pp. 1–4.

[18] J. Huang, Y. Meng, X. Gong, Y. Liu, and Q. Duan, "A novel deployment scheme for green Internet of Things," *IEEE Internet Things J*, vol. 1, no. 2, pp. 196–205, Apr. 2014.

[19] P. Szczytowski, A. Khelil, and N. Suri, "DKM: Distributed k-connectivity maintenance in wireless sensor networks," in *Proc. IEEE 9th Annu. Conf. Wireless Demand Net. Syst. Services (WONS)*, Courmayeur, Italy, Jan. 2012, pp. 83–90.

[20] J. T. Adams, "An introduction to IEEE STD 802.15.4," in *Proc. IEEE Aero Space Conf.*, Mar. 2006, pp. 1–8.

[21] R. Peng, M.-H. Sun, and Y.-M. Zou, "ZigBee routing selection strategy based on data services and energy-balanced ZigBee routing," in *Proc. IEEE Asia–Pacific Conf. Services Comput. (APSCC)*, Dec. 2006, pp. 400–404.

[22] Thomas Lextrait. *Arduino: Power Consumption Compared*. Accessed: Jan. 2019. [Online]. Available: https://tlextrait.svbtle.com/arduino-power-consumption-compared

[23] *Arduino and Genuino Products*. Accessed: Jan. 2019. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoard101

[24] *Raspberry Pi3 Model B Technical Specification*. Accessed: Jun. 2018. [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/14ba/0900766b814ba5fd.pdf

[25] N. Javaid, S. Faisal, Z. A. Khan, D. Nayab, and M. Zahid, "Measuring fatigue of soldiers in wireless body area sensor networks," in *Proc. 8th IEEE Int. Conf. Broadband Wireless Comput., Commun. Appl. (BWCCA)*, Compiegne, France, Oct. 2013, pp. 227–231.

[26] W. Xu, W. Liang, X. Jia, and Z. Xu, "Maximizing sensor lifetime in a rechargeable sensor network via partial energy charging on sensors," in *Proc. 13th IEEE Int. Conf. Sens. Commun. Netw.*, Jun. 2016, pp. 1–9.

[27] H. C. Tung *et al.*, "A mobility enabled inpatient monitoring system using a ZigBee medical sensor network," *Sensors*, vol. 14, no. 2, pp. 2397–2416, 2014.

[28] A. Q. Lawey, T. El-Gorashi, and J. M. H. Elmirghani, "Impact of peers behaviour on the energy efficiency of BitTorrent over optical networks," in *Proc. 14th Int. Conf. Transparent Opt. Netw. (ICTON)*, Coventry, U.K., Jul. 2012, pp. 1–8.

[29] F. Comeau and N. Aslam, "Analysis of LEACH energy parameters," *Procedia Comput. Sci.*, vol. 5, pp. 933–938, Jan. 2011.

[30] *Cisco 910 Industrial Router*. Accessed: Jan. 2019. [Online]. Available: https://www.cisco.com/c/en/us/support/routers/910-industrial-router/model.html

[31] *Intel Atom Z510 Specifications*. Accessed: Jun. 2018. [Online]. Available: http://www.cpu-world.com/CPUs/Atom/IntelAtom%20Z510%20AC80566UC005DE.html

[32] *SENSORO Alpha Product Suite-Alpha Node-4AA*. Accessed: Jun. 11, 2018. [Online]. Available: https://www.sensoro.com/static/node4aa_en.pdf

[33] Z. Ren, J. Dong, Y. Ren, R. Zhou, and X. You, "Workload characterization on a cloud platform: An early experience," *Int. J. Grid Distrib. Comput.*, vol. 9, no. 6, pp. 259–268, 2016.

[34] D. P. Dobkin and S. P. Reiss, "The complexity of linear programming," *Theor. Comput. Sci.*, vol. 11, no. 1, pp. 1–18, May 1980.

[35] R. G. Gallager, "Distributed minimum hop algorithms," M.I.T. Lab Inf. Decis. Syst., Cambridge, MA, USA, Tech. Rep. LIDS-P1175, Jan. 1982.

ZAINEB T. AL-AZEZ received the B.Sc. and M.Sc. degrees in computer engineering from Nahrain University, Baghdad, Iraq, in 2005 and 2008, respectively, and the Ph.D. degree in communication networks from the School of Electronic and Electrical Engineering, University of Leeds, U.K., in 2018. From 2009 to 2014, she was an Assistant Lecturer with Al-Mansour University College, Baghdad. Her current research interests include energy efficiency in the Internet of Things networks, cloud computing, and virtualization.

AHMED Q. LAWEY received the B.Sc. and M.Sc. degrees (Hons.) in computer engineering from Nahrain University, Iraq, in 2002 and 2005, respectively, and the Ph.D. degree in communication networks from the University of Leeds, U.K., in 2015. From 2005 to 2010, he was a Core Network Engineer with ZTE Corporation for Telecommunication, Iraq Branch. He is currently a Lecturer in communication networks with the School of Electronic and Electrical Engineering, University of Leeds. His current research interests include energy efficiency in optical and wireless networks, big data, cloud computing, and the Internet of Things.

TAISIR E. H. EL-GORASHI received the B.S. degree (Hons.) in electrical and electronic engineering from the University of Khartoum, Khartoum, Sudan, in 2004, the M.Sc. degree (Hons.) in photonic and communication systems from the University of Wales, Swansea, U.K., in 2005, and the Ph.D. degree in optical networking from the University of Leeds, Leeds, U.K., in 2010. She is currently a Lecturer in optical networks with the School of Electrical and Electronic Engineering, University of Leeds. Previously, she held a position of Postdoctoral Researcher with the University of Leeds (2010–2014), where she focused on the energy efficiency of optical networks investigating the use of renewable energy in core networks, green IP over WDM networks with datacenters, energy efficient physical topology design, energy efficiency of content distribution networks, distributed cloud computing, network virtualization, and Big Data. In 2012, she was a BT Research Fellow, where she developed energy efficient hybrid wireless-optical broadband access networks and explored the dynamics of TV viewing behavior and program popularity. The energy efficiency techniques developed during her postdoctoral research contributed three out of the eight carefully chosen core network energy efficiency improvement measures recommended by the GreenTouch consortium for every operator network worldwide. Her work led to several invited talks at GreenTouch, Bell Labs, Optical Network Design and Modelling Conference, Optical Fiber Communications Conference, International Conference on Computer Communications and EU Future Internet Assembly, and collaboration with Alcatel Lucent and Huawei.

JAAFAR M. H. ELMIRGHANI (M'92–SM'99) received the B.Sc. degree (Hons.) in electrical engineering from the University of Khartoum, in 1989, the Ph.D. degree in the synchronization of optical systems and optical receiver design from the University of Huddersfield, U.K., in 1994, and the D.Sc. degree in communication systems and networks from the University of Leeds, U.K., in 2014. He is currently the Director of the Institute of Communication and Power Networks, School of Electronic and Electrical Engineering, University of Leeds, U.K. In 2007, he joined Leeds. Prior to that, he was the Chair in optical communications with the University of Wales, Swansea (2000–2007). He has founded, developed, and directed the Institute of Advanced Telecommunications and the Technium Digital (TD), a technology incubator/spinoff hub. He has provided outstanding leadership in a number of large research projects at the IAT and TD. He has coauthored *Photonic switching Technology: Systems and Networks* (Wiley) and has published over 450 papers. He has research interests in optical systems and networks. He is a Fellow of the IET, a Chartered Engineer, and a Fellow of the Institute of Physics. He was the Chairman of the IEEE Comsoc Transmission Access and Optical Systems Technical Committee and was the Chairman of the IEEE Comsoc Signal Processing and Communications Electronics Technical Committee. He is and has been on the Technical Program Committee of the 34 IEEE ICC/GLOBECOM conferences (1995–2016), including 15 times as the Symposium Chair. He has given over 55 invited and keynote talks over the past eight years. He received the IEEE Communications Society Hal Sobol Award, the IEEE Comsoc Chapter Achievement Award for excellence in chapter activities (both in international competition in 2005), the University of Wales Swansea Outstanding Research Achievement Award, in 2006; and received in international competition: the IEEE Communications Society Signal Processing and Communication Electronics outstanding service award, in 2009; and the Best Paper Award at the IEEE ICC'2013. Related to Green Communications, he received the IEEE Comsoc Transmission Access and Optical Systems outstanding Service Award 2015 in recognition of Leadership and Contributions to the Area of Green Communications, the GreenTouch 1000x award, in 2015 for pioneering research contributions to the field of energy efficiency in telecommunications, the IET 2016 Premium Award for Best Paper in IET Optoelectronics, and has shared the 2016 Edison Award in the collective disruption category with a team of six from GreenTouch for their joint work on the GreenMeter. He has been awarded in excess of č22 million in grants to date from EPSRC, the EU and industry, and has held prestigious fellowships funded by the Royal Society and by BT. He was awarded all four prizes in the department of the University of Khartoum for academic distinction. He was an Editor of the *IEEE Communications Magazine*. He was the founding Chair of the *Advanced Signal Processing for Communication Symposium* which started at the IEEE GLOBECOM'99 and has continued since at every ICC and GLOBECOM. He was also the Founding Chair of the first IEEE ICC/GLOBECOM Optical Symposium at GLOBECOM'00, the Future Photonic Network Technologies, Architectures and Protocols Symposium. He chaired this Symposium, which continues to date under different names. He was the Founding Chair of the first Green Track at ICC/GLOBECOM at GLOBECOM 2011, and is the Chair of the IEEE Green ICT initiative within the IEEE Technical Activities Board (TAB) Future Directions Committee (FDC), a pan IEEE Societies initiative responsible for Green ICT activities across IEEE, since 2012. He was the Co-Chair of the GreenTouch Wired, Core and Access Networks Working Group, an Adviser to the Commonwealth Scholarship Commission, a member of the Royal Society International Joint Projects Panel, and a member of the Engineering and Physical Sciences Research Council (EPSRC) College. He is currently an Editor of the *IET Optoelectronics* and the *Journal of Optical Communications*, and was an Editor of the IEEE Communications Surveys and Tutorials and the IEEE Journal on Selected Areas in Communications Series on Green Communications and Networking. He was an IEEE Comsoc Distinguished Lecturer (2013–2016).