

**Original scientific paper**

## **GREEN COMPUTING FOR IOT – SOFTWARE APPROACH\***

**Haris Turkmanović<sup>1</sup>, Ivan Popović<sup>1</sup>, Dejan Drajić<sup>2,3</sup>, Zoran Čiča<sup>2</sup>**

<sup>1</sup>University of Belgrade, School of Electrical Engineering, Department of Electronics,

<sup>2</sup>University of Belgrade, School of Electrical Engineering,

Department of Telecommunications

<sup>3</sup>Innovation Centre of School of Electrical Engineering

**Abstract.** *More efficient usage of limited energy resources on embedded platforms, found in various IoT applications, is identified as a universal challenge in designing such devices and systems. Although many power management techniques for control and optimization of device power consumption have been introduced at the hardware and software level, only few of them are addressing device operation at the application level. In this paper, a software engineering approach for managing the operation of IoT edge devices is presented. This approach involves a set of the application-level software parameters that affect consumption of the IoT device and its real-time behavior. To investigate and illustrate the impact of the introduced parameters on the device performance and its energy footprint, we utilize a custom-built simulation environment. The simulation results obtained from analyzing simplified data producer-consumer configuration of IoT edge tier, under push-based communication model, confirm that careful tuning of the identified set of parameters can lead to more energy efficient IoT end-device operation.*

**Key words:** *Green IoT, Energy saving, Real-Time IoT, Push communication technology, Embedded systems*

### 1. INTRODUCTION

Many technological achievements in recent years, especially in field of information and communication technologies, have enabled the usage of a wide range of IoT applications and devices. Healthcare systems, smart cities, home automation and security, wearable devices, and agriculture are just some of the applications whose rapid development is facilitated by the advancement of various IoT communication technologies [2].

According to research [3], global number of connected IoT devices is expected to grow nearly 10% per year, where the number of IP connections by the year 2023. is expected to

---

Received March 6, 2022; revised April 28, 2022; accepted May 8, 2022

**Corresponding author:** Haris Turkmanović

University of Belgrade, School of Electrical Engineering, Department of Electronics, Bulevar kralja Aleksandra 73, 11120 Beograd, Serbia

E-mail: [haris@etf.bg.ac.rs](mailto:haris@etf.bg.ac.rs)

\* An earlier version of this paper was presented at the 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS 2021), October 20-22, 2021, in Niš, Serbia [1]

be three times higher than the total world population. This accelerated growth of network traffic and the increased number of connected IoT devices lead to elevated global energy consumption and pollutions related to CO<sub>2</sub> emission [4]. It has been predicted that soon IoT devices and systems will be leading energy consumers in the domain of information and communication technologies [5]. The utilization of energy-efficient technologies to reduce energy consumption as well as CO<sub>2</sub> emission has become mandatory in the design of Green IoT systems (GIoT).

There are various definitions of IoT systems which commonly include different edge IoT devices distributed all over the IoT network. These devices produce data and autonomously communicate with other parts of IoT system without direct human intervention [6]. Design of an IoT system that relies on the use of modern IoT communication technologies is very complex because of different challenges such as security [7], scalability [8], data management, real-time performance [9], and others [10]. The scope of GIoT systems is even more complex because it involves different set of green technologies in product life cycle. Green technologies are targeting hardware and software design, green production, green utilization and green disposal of IoT devices.

IoT edge devices are usually designed as battery-operated embedded devices that have constrained resources and capabilities, such as limited memory and CPU processing power. Hardware and software design requirements of such devices are mostly related to efficient usage of limited available energy. These requirements are important to prolong IoT application operational runtime and enable utilization of green technologies. Therefore, the limitation in terms of the available energy resources represents the main issue in designing energy-efficient IoT applications [11-13] and GIoT systems. The base for designing such system is green computing which main goal is to reduce IoT devices' energy consumption without degrading their performance [14].

In this paper, we introduce a set of application-level parameters that shape the communication and operational behavior of IoT edge devices but also affect their energy consumption and performance. The conducted study aims to explore the way to utilize a software engineering approach in controlling this set of parameters to achieve the balance between the energy consumption of IoT edge devices, its operational runtime [15], and the required real-time performance of IoT services and applications. To quantify the contribution of the selected software engineering approach, a simulation environment is developed as a flexible open-source framework for analyzing the behavior, performance, and energy footprint of arbitrary distributed IoT systems and applications [16-17].

Achievements and contributions of our proposed software engineering approach in designing GIoT edge devices are the following:

- introduction of application-level software parameters that enable fine-tuning of IoT application performance vs. energy consumption of the IoT devices located at the network edge
- the extension of the simulation framework with the set of parameters for modeling consumption of IoT system processing and communication elements, enables comprehensive analysis of the energy requirements of an arbitrary IoT system and/or its components.
- the analysis of real-time performance and energy consumption confirmed the trade-off potential of the proposed software approach for driving the operation of GIoT edge devices.

The rest of the paper is organized as follows. Section II presents related work regarding the existing energy optimization methods. Section III presents a brief overview of the simulation framework with details related to the implementation of energy calculation algorithms. In section IV, we present the simulation model used in our analysis while section V presents and discusses obtained results. Section VI concludes the paper.

## 2. RELATED WORK

Energy consumption-related problems have been attracting a lot of attention from the scientific and research community. Data involved in processing and communication within IoT applications are usually generated by battery-powered edge IoT devices that generate data by sensing their environment. Until today, various energy optimization methods, which are addressing different energy-intensive aspects of IoT systems, have been developed to prolong IoT edge device operational time and to provide enhanced real-time performance of GIoT applications. The rest of the section provides an overview of existing optimization methods, their classification, and approaches related to the energy-aware design of resource-constrained embedded devices found in various IoT applications and systems.

Although the standard methodologies in designing low-power embedded systems involve the range of approaches from simple usage of low power products to complex algorithms for scheduling system workload, there is no single universally accepted methodology that fits all applications needs. All optimization techniques can be classified into two major categories: hardware and software energy optimization techniques. Within this research we be explored only the software optimization techniques which, based on [18] can be further classified as data center-based, cloud computing-based, and virtualization-based techniques. A more detailed classification of software optimization techniques is done in [4] where they are classified into nine different groups. Based on [19], all software optimization techniques can be classified into three groups: instruction level, compiler level, and operating system level. The taxonomy presented in this paper is an entry point for the analysis of energy optimization techniques applicable for IoT edge devices. In paper [20] is presented overview of such techniques and it has been discussed influence that certain techniques have on edge device energy consumption and overall IoT system consumption as well.

IoT system edge devices are, in a certain sense similar to sensing node devices used within Wireless Sensor Networks (WSN). Therefore, in the case of energy optimization of IoT edge devices, some optimization techniques already considered in the case of WSN can be utilized. A review of energy optimization techniques in [21-22] gives a systematic classification of the solutions that can be used to preserve energy in WSN. Moreover, these papers introduce the division of battery-powered sensor devices on basic subsystem components which consume energy: *sensing subsystem*, *processing subsystem* and *communication subsystem*. This division makes sensing node energy analysis more systematic, and it can be also applied in case of GIoT edge devices. It is pointed out that the consumption of each subsystem must be considered equally during the energy profiling analysis of IoT edge nodes.

In IoT applications based on wireless communication technologies, communication subsystems in most cases consume significantly more energy compared to processing or sensing subsystems. Different software methods are focused to reduce the consumption of

the communication subsystem. Based on research presented in [23] all these methods can be classified into two groups: duty-cycling based methods and in-processing methods. Duty-cycling-based methods reduce energy consumption by disabling communication components in a case when they are not used. In-processing methods use various data compression and/or data aggregation techniques to reduce the amount of data involved in communication.

The amount of data involved in communication increases with the number of edge nodes participating in the communication. Different research has shown that there is a certain similarity between the data produced by sensor nodes [24-25]. Data aggregation methods exploit this feature to reduce the overall amount of data in IoT applications. Instead of forwarding the data instantaneously, data are first collected and then aggregated using functions like sum, average, threshold. In [26] some of the data aggregation methods are presented. It is shown that utilization of these methods has a huge impact on the reduction of sensor node energy consumption, but also decreases real-time performance since delay in data delivery time over IoT applications is increased. Therefore, in the case of applying methods based on data aggregation, it is important to use two metric parameters: energy consumption and data delay to describe overall IoT application performance. In [27] analytical model is presented which enables calculation of energy consumption and packet delivery time in case of aggregation optimization methods usage. This model allows determining parameters values such as buffering time and maximum number of buffered packets. Additionally, this work shows that the utilization of aggregation methods may lead to a significant decrease in energy consumption.

In [28], it is shown that the energy consumption of the processing subsystem may increase compared to the communication subsystem when complex memory-intensive compression algorithms are used. However, a study conducted in [28] modifies already developed techniques, such as Discrete Cosine Transform (DCT) and Discrete Wavelet transform (DWT), that can be applied on sensor nodes to compress data. These techniques are modified in such a way that utilization of memory and processing capability is not too high compared to the original algorithm. It has been shown that among other things, utilization of these modified techniques leads to reduction of energy consumption and prolonged device operational runtime.

When data aggregation methods are used it is very important to decide when to communicate to send aggregated data to the consumer node. Paper [30] named this parameter as transmission period. It is shown that this parameter has a significant impact on sensor node performance in terms of energy, data accuracy, and data freshness. The specific approach is developed in this work which gives a possibility to balance between energy saving and data availability at the higher tiers of hierarchically organized IoT systems.

In some practical applications, sensing subsystem can consume significantly more energy compared to other edge device system parts. Work presented in [31] establishes an approach based on smart sensing policy which achieves less energy consumption of sensing subsystem compared to a usage of standard fixed sensing period policy. This policy used a learning model based on a backpropagation neural network. It has been concluded that this policy may reduce consumption by up to 50%. In [32] adaptive sampling algorithm is proposed which can dynamically estimate optimal sampling frequency. The performance of this algorithm is estimated in simulation of snow monitoring application. Obtained simulation results show that this algorithm may reduce the energy consumption of the sensing subsystem up to 97%.

Beside the techniques that are directly related to our research, there are other techniques that also affect power consumption. In [33] power management techniques are categorized as

dynamic voltage and frequency scaling, subthreshold design, asynchronous circuit design and power-gating. In case of edge devices that utilize real-time operating systems there are different OS-level techniques that impact task scheduling [33-34]. Within IoT multimedia applications control of parameters such as frames per seconds (FPS) is also found as a common approach to lower energy consumption [36]. In the domain of e-healthcare applications there are several solutions offering energy-efficient frameworks using Internet of Medical Things (IoMT) protocol to optimize the communication overhead and overall energy consumption while transmitting the healthcare data [37-38].

Although most reviewed solutions and approaches investigate the impact of individual parameters on the power or energy consumption, neither of them analyzes trade-off potential and more complex tuning of device operation through the control of a group of parameters. On the other hand, our study presented in this paper provides a comprehensive analysis of the performance and energy consumption properties of IoT edge devices, during their operation under the different setup of the selected application-level parameters.

### 3. MATERIALS AND METHODS

The first part of this section presents general aspects of traffic engineering relevant for IoT system energy consumption and performance analysis. Set of application-level software parameters, that enable tuning of performance and consumption properties of IoT devices, is also introduced. Metric associated with the quantification of these properties is presented within the first part of this section, while utilization of simulation framework for energy analysis is described in the rest of the section.

#### 3.1. Overview of the approach

There are many different possibilities of IoT system realizations, but in most cases, it is possible to identify four main elements: 1) the intelligent devices where data are produced – *producer devices*, 2) the gateways that extract data, aggregate data, and/or perform protocol translation, 3) the network used to establish communication between devices and 4) the device which receives data – *consumer device*. In the simplest representation of an IoT system, it is possible to consider that system consists only of producer and consumer device.

The communication between producer nodes and the rest of the distributed IoT system determines producer nodes' energy consumption and the real-time performance of the IoT application. Two traffic engineering strategies can be applied when designing an edge-tier IoT system: pull and push [39]. The messaging patterns of these two strategies are illustrated in Figure 1.

In case of pull strategy, data generated on the producer node side are sent to the consumer node side only when the consumer node sends a pull request to the producer node. Pull strategy is suitable for implementation in a case where the consumer node is interested in partial data from certain producer nodes (there is a correlation between data sent

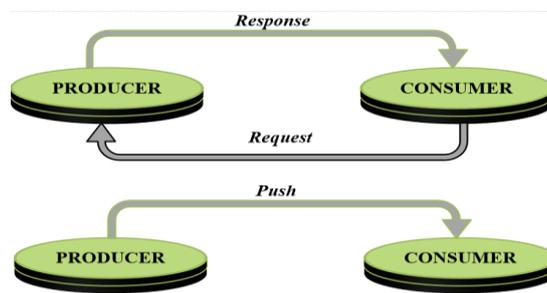


Fig 1. a) pull b) push traffic strategy

from different producer nodes to the same consumer nodes). Contrary, push traffic strategy involves sending data or notifications from the producer node periodically or when a particular event occurs on the producer node. Push strategy forces real-time performance of IoT applications [40], although in some IoT applications it is also possible to combine both communication strategies.

Since we observe real-time IoT applications, the push traffic technology is considered as the reference for our research, since it supports a higher number of parameters on the edge-device node's side. Table 1 gives overview of the parameters that are available from an application point of view for tailoring the device operation for push communication strategy.

**Table 1** Overview of software available parameters for push strategy

Parameter	Description
Sampling time (ST)	Defines how often data are generated on the producer node.
Aggregation rate (AR)	Defines the level of data reduction on the producer node.
Transmission period (TP)	Defines the period for sending data from the Producer to the consumer node.

The performed analysis explores how much these three parameters impact energy consumption and overall real-time performance of IoT applications. To quantify this impact, two metrics are introduced:

- **Energy consumption(E)** – expressed in milliamperes per hour.
- **Average data delivery time (ADT)**– the time interval that elapses from the moment of data generation to the moment of data processing at the destination node.

### 3.2. Tools and procedures

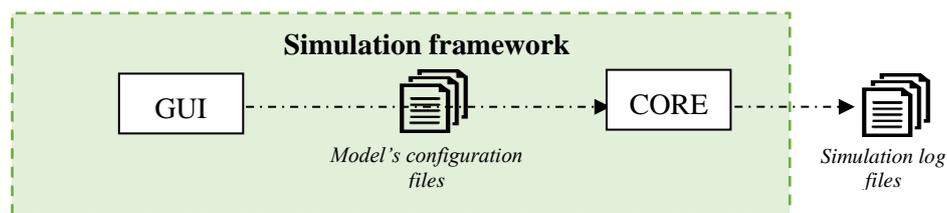
The simulation framework used within this work enables the creation of arbitrary IoT system topologies and analysis of various IoT application performance parameters. Results obtained by simulation provide a detailed overview of data availability across the entire IoT system at any point in time. By analyzing obtained results, it is possible to quantify various IoT application performance parameters such as IoT system consumption, real-time performance, and scalability of IoT system architecture. In paper [17], it is already described how to exploit developed simulation framework to quantify scalability of the IoT architecture. In this section, we describe in more details the main aspects of the simulation framework important for better understanding of how to quantify the influence of a certain set of parameters on IoT system energy consumption. The created simulation framework is available as an open-source solution [16] and it can be further developed and adapted to satisfy any requirements which are not supported by the current framework version.

The simulation framework comprises simulation core and a graphic user interface. The simulation core is in charge to implement all functionalities related to the simulation of IoT system behavior on different levels of IoT system architecture. These functionalities rely on component's model which exists within IoT system in the general case: node model – which represents a device that generates data or consumes data, link model – which represents a connection between IoT system components, and protocol – which encloses all information related to data created and consumed within IoT system.

Current version of simulation framework used within this analysis supports only the simplest models of IoT system components that exist in general case such as processing

devices, links and protocols. These models support only basic parameters configuration. Model included within current simulation framework version does not support modeling of packet dropouts, connection losses and packet retransmissions which can be significant for overall IoT application quality of service analysis. Within each model, it is possible to configure a certain set of parameters.

For the easy process of configuring the model's parameter, the graphical user interface is developed. Communication between the framework core and the graphic user interface is established by using the model's configuration file. At the end of the simulation, different log files are created. By examining and analyzing the content of these files it is possible to understand how IoT systems behave. A general overview of the simulation framework is presented in Figure 2.



**Fig. 2** Simulation framework architecture

The node, link, and protocol model support different parameters. The current version of the simulator supports three-node models: producer, gateway, and consumer. Node consumption, processing time, adjacent nodes, aggregation level, compression rate, and transmission period are parameters that can be configured for each node model. Additionally, in the case of the producer node model, it is possible to define the amount of data produced on the node but also it is possible to define the data sampling rate. The model of the link supports the configuration of the following parameters: link speed, link consumption (transmit and receive), and link speed deviation. Data are exchanged between nodes using a protocol model where it is possible to define protocol overhead and optionally it is possible to enable a handshaking mechanism. A more detailed description of parameters is given in [17].

Energy consumption calculation within the simulation framework is implemented based on current overall node consumption (CONC) expressed in mA. To calculate the charge consumed by a node for specific action, CONC is multiplied by the time required for executing specific action on the node. Calculation algorithms print the cumulative sum of consumed charge over time (CSC), expressed in mA per time resolution – R, to the node log file. Node energy consumption is directly proportional to CSC value, and it is easy to calculate it directly if information about node voltage power supply is available. Based on this information, it is easy to profile nodes based on energy consumption.

CONC value is determined by the current node operation mode as well as the type of the links used to communicate with adjacent nodes. Two operating modes are supported by each node model: active and low power mode. Node is in active mode when data are processing on node or data are transmitting/receiving from/on a node. If there is no any action on the node, it is in low power mode. For each of these modes, within the node's model configuration file, it is possible to configure current node consumption (CNCm) by

setting parameter value related to specific node mode  $m$  ( $CNCA$  – current node consumption when node is in active mode,  $CNCLP$  – current node consumption when it is in low power mode). Link models enable configuration of current link consumption  $CLCs$  during different states  $s$  such as transmission  $CLCT$  and receiving data  $CLCR$ .

The following equation is used for  $CONC$  calculation:

$$CONC = CNC_m + CLC_s(1)$$

where  $CNC_m$  and  $CLC_s$  take value depending on current actions on the node as presented in Table 2:

**Table 2** Value of  $CONC$  depends on action on the node

Action on the node	$CONC =$
Low power mode	$CNCLP$
Processing received data	$CNCA$
Processing received data and receiving new data from another node	$CNCA + CLCR$
Receiving data	$CNCLP + CLCR$
Transmitting data	$CLCT$

It needs to be mentioned that the improvements of available simulation's models to correspond with practical MCU based device implementations is seen as a part of future work. Goal of this future work will be to extend the simulation model to accurately represent both, device and communication power and performance behaviors.

To illustrate the working principle of developed algorithms and to illustrate the potential of developed simulation framework in terms of profiling node's energy consumption, we examine the behavior of simple node  $N$  which is connected to the rest of the IoT system over link  $L$ . Information relevant to this example is presented in Tables 3 and 4.

**Table 3** Node  $N$  parameters values

Parameter name	Value	Unit
Processing speed	50	[B/s]
Data production rate	15	[s]
Data size	50	[B]
$CNCLP$	10	[mA]
$CNCA$	90	[mA]

**Table 4** Link  $L$  parameters values

Parameter name	Value	Unit
Link speed	12.5	[B/s]
$CLCR$	400	[mA]
$CLCT$	400	[mA]

Figure 3 shows part of the node log file obtained after completed simulation. The shown part of the log file includes actions on the nodes inside the time interval [45s, 72s].

```

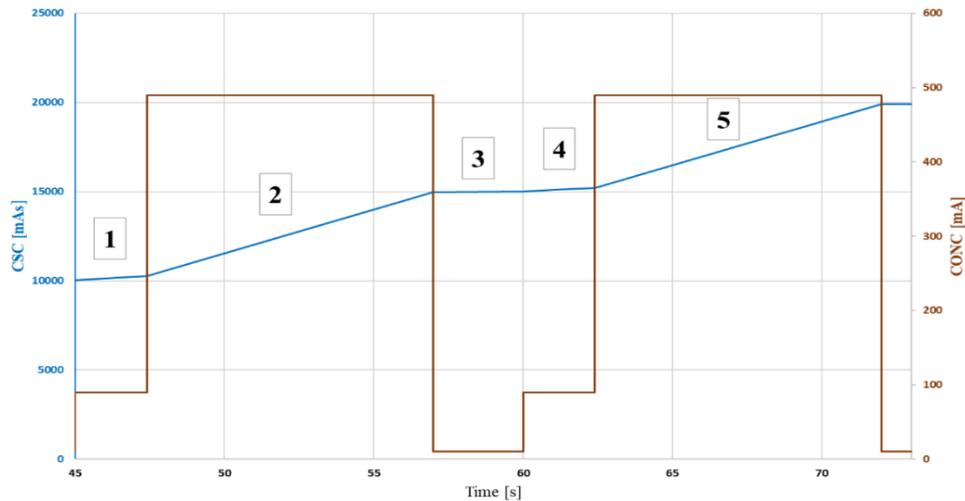
20 45.000000,A,50,C_Req,4,50,0,100,0,0,90.000000,10050.000000
21 45.000000,A,50,P_D_S_Req,4,50,0,100,0,0,90.000000,10050.000000
22 46.000000,A,50,P_D_E_Req,4,50,0,150,0,1,90.000000,10140.000000
23 46.000000,A,70,P_O_S_Req,4,70,0,150,0,1,90.000000,10140.000000
24 47.400000,A,70,P_O_E_Req,4,70,0,150,0,1,90.000000,10266.000000
25 47.400000,A,120,T_S_Req,4,120,0,150,0,1,490.000000,10266.000000
26 57.000000,A,120,T_E_Req,4,0,0,150,0,0,490.000000,14970.000000
27 57.000000,LP,ENTER,10.000000,14970.000000
28 60.000000,LP,EXIT,10.000000,15000.000000
29 60.000000,A,50,C_Req,7,50,0,150,0,0,90.000000,15000.000000
30 60.000000,A,50,P_D_S_Req,7,50,0,150,0,0,90.000000,15000.000000
31 61.000000,A,50,P_D_E_Req,7,50,0,200,0,1,90.000000,15090.000000
32 61.000000,A,70,P_O_S_Req,7,70,0,200,0,1,90.000000,15090.000000
33 62.400000,A,70,P_O_E_Req,7,70,0,200,0,1,90.000000,15216.000000
34 62.400000,A,120,T_S_Req,7,120,0,200,0,1,490.000000,15216.000000
35 72.000000,A,120,T_E_Req,7,0,0,200,0,0,490.000000,19920.000000
36 72.000000,LP,ENTER,10.000000,19920.000000

```

**Fig. 3** Part of node's log file

The node log file is given in form of a CSV file where each value in a single row represents information about node parameter value at a specific point in time. More information about specific values is given in [17] while in this analysis we focus only on the values important for energy analysis such as timestamp (1st value), CONC (next to last value), and CSC (last value). Obtained values are extracted and visualized in Figure 4.

The analysis shown in Figure 4. illustrates the charge and the consumption of the selected IoT node for the selected time interval [45-73s]. Time intervals 1 and 4 include all node actions which occur on the node log file within intervals [45-47.4s] and [60-62.4s], respectively. These actions are mostly based on data processing of created data. Time intervals 2 and 5 represent actions on the node within [47.4-57s] and [62.4-72s] where the action of processed data transmission is executing. After node sends data, there are no more actions on the node, and the node goes to low-power mode. This node state is observed in time interval 3 within [57-60s] as found from a log file.

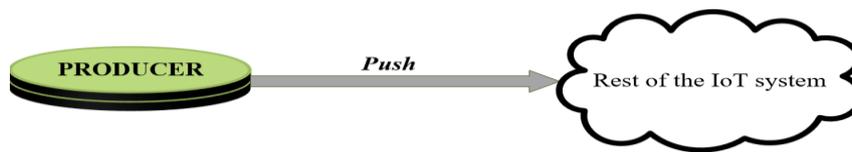


**Fig. 4** Node charge consumption and CONC values within time interval [45s-73s]

## 4. CASE STUDY

This section gives a description of the experiment setup, including IoT system topology and node and link configuration, and the simulation results illustrating the impact of introduced application-level parameters on node operation and consumption. The parametric analysis and the discussion of the associated trade-off properties are also given.

In our analysis, data communication at the edge-tier of the IoT system is modeled as an interaction between the data producer node (IoT edge device) and corresponding consumer (destination) node located in the higher hierarchy of the rest of the IoT system. Data from edge devices are pushed toward data consumer device through a link used to establish communication between producer and consumer device. This IoT system is illustrated in Figure 5. while parameters used in simulation are presented in Tables 5. and 6.



**Fig. 5** Illustration of IoT system used in our analysis

**Table 5** Producer node parameters value

Parameter name	Value	Unit
Processing speed	1	[MB/s]
Data size	100	[B]
Data overhead	70	[B]
CNC <sub>LP</sub>	20	[mA]
CNC <sub>A</sub>	110	[mA]

**Table 6** Link parameters value

Parameter name	Value	Unit
Link speed	18	[kB/s]
Maximum transmission unit (MTU)	1500	[B]
CLC <sub>T</sub>	410	[mA]
CLC <sub>R</sub>	410	[mA]

Edge devices can be considered as simple MCU-based embedded system which gathers data by sensing its environment, performs simple data processing, like data aggregation, and provides physical connectivity with the rest of the IoT system. From software's perspective is only possible to control parameters such as data sampling rate, aggregation rate, and transmission rate. The range of values of these three parameters is shown in Table 7.

**Table 7** Range of parameters values

Parameter name	Range	Unit
Data sampling time	0.1-10	[s]
Aggregation rate	1-100	-
Transmission period	1-10	[s]

The analysis of the results obtained by variation of these three parameters' values in a presented range is given in the next section.

## 5. RESULTS ANALYSIS

Results obtained by simulation are presented in Figure 6. Results are normalized to the operating point with the coordinates  $Q_0(AR_0, TP_0, ST_0) = (10, 10s, 1s)$ . The normalized results are adopted to illustrate the potential of adjusting the parameter values on observed system properties given on different scales. Each parameter value at  $Q_0$  is selected as a midpoint of the parameter range given in the logarithmic scale. Furthermore, parameter range is chosen to avoid boundary conditions of system operation where sampling rate interval is comparable with the data processing time and/or communication latency. Based on the obtained results, it is possible to quantify the impact of these parameters on reducing energy consumption, but also on reducing the average time of data availability on consumer nodes. Normalized energy consumption (E) and optimization cost function (O) values are presented on each graphics' left side, while normalized average data delivery time (ADT) values are presented on the right side.

From figure 6-I it is noticeable that increasing the aggregation rate within the first half of the observed range [0.1-1] reduces data payload size which leads to the reduction of the total energy consumption (~0.26). Within the second half of the observed range [1-10], increasing aggregation rate to a lesser extent contributes to a further reduction of energy consumption (~constant) because payload size becomes negligible to protocol header size. From the same graph, it can be also noticed that the increase in aggregation rate does not cause a significant change in data delivery time (~0.03). This impact is expected because the change of the aggregation rate does not change the outcome in terms of the data availability time, but it changes only the form of the exchanged data since the original data are embedded within the aggregated data format.

The change in the transmission period has a significantly greater impact on the reduction of energy consumption compared to the impact of the aggregation rate parameter, because of the reduced activity of the IoT communication subsystem. It can be seen in Figure 6-II that due to the increase in transmission period, energy consumption decreases almost linearly along with the entire observed range. On the other hand, there is a proportional degradation of data availability time and corresponding real-time performance. The effect of the sampling rate parameter is shown in Figure 6-III. By controlling this parameter, we can achieve certain energy-saving up to half of the observed range [0.1 – 1], like in the case of the aggregation rate parameter. However, in contrast to the other two parameters in the second half of the observed range [1 – 10], it is possible to achieve significantly better characteristics in the domain of data availability at the consumer node side.

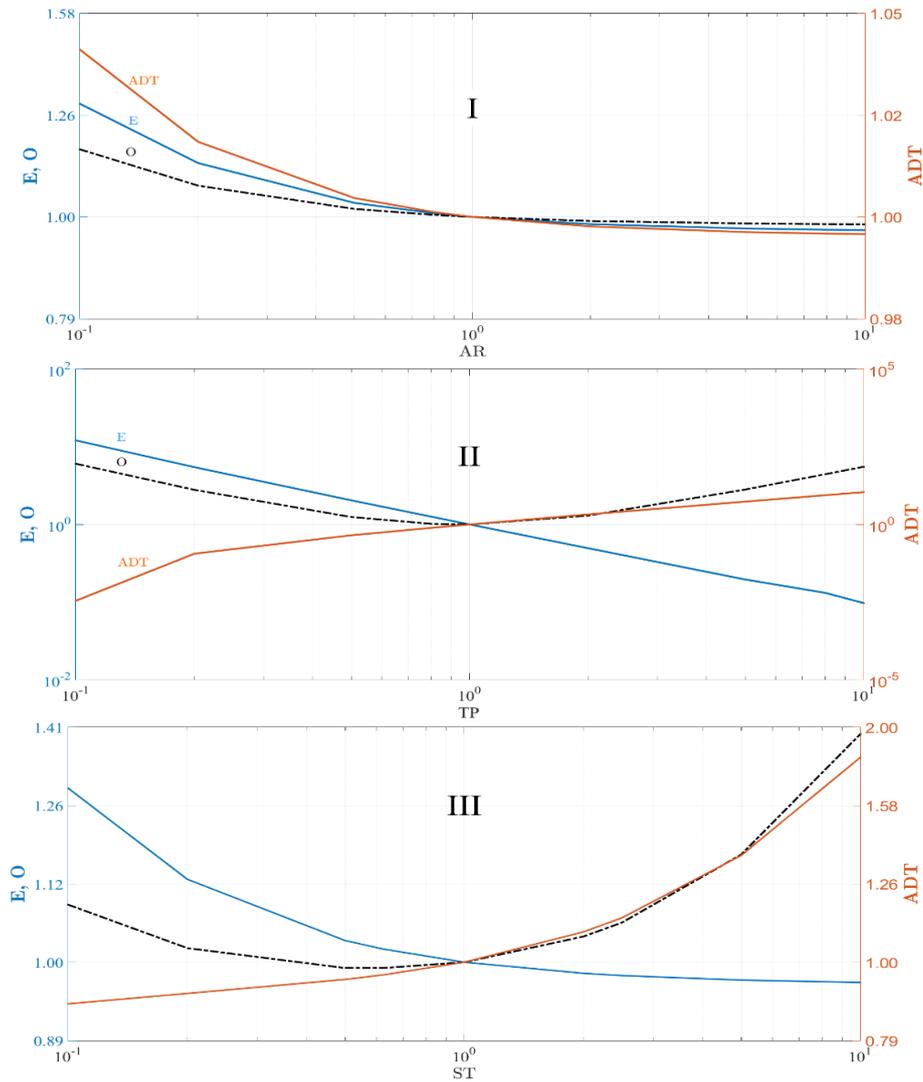
To quantify the trade-off that can be achieved by tuning certain parameters, we introduce the optimization cost function defined as:

$$O = k \cdot E + q \cdot ADT \quad (2)$$

where parameters take a value within a range [0, 1] and relation between k and q is defined within following equality:  $k = 1 - q$ .

The purpose of this cost function is to establish the relation between the power consumption and performance domains to find the optimal operating point for IoT edge device. The cost function provides background for tuning the certain parameter within IoT device at the edge tier to optimize power consumption and/or overall IoT system real-time performance. Operating at the best performance, without the concerns about the consumption means operating point with maximal sampling rate and communication rate without data aggregation.

Thus, optimizing only performance imply that the value for  $q$  is set to 1 while  $k$  equals 0. If both  $k$  and  $q$  are higher than zero than we can talk about the trade-off in power-performance domain. Analysis conducted in this paper considers that both requirements are equally important, and consequently both parameters' values are set to 0.5.



**Fig. 6** Influence of Aggregation Rate (I), Transmission Period (II) and Sampling time (III) on Energy Consumption (left scale – blue) and Average Data Delivery Time (right scale – orange) vs trade-off optimization norm (left scale - black)

By analyzing cost functions presented on Figure 6, it is possible to find optimal operating point by tuning only single parameters. Following the shape of the cost function  $O$ , presented in Figure 6-I, decreasing the value of  $AR$  below the  $AR_0$  results in a significant increase in the optimization cost function's value. Alternatively, increasing the value of  $AR$  above  $AR_0$  has a minor effect on the cost function's value. It's obviously that optimal  $AR$  value is located at the end of the observed range. As visible from Figure 6-II, varying the value of the transmission period ( $TP$ ) parameter away from  $TP_0$  degrades the value of  $O$ , since its optimal value of  $TP$  parameters is found around  $TP_0$ . On the other hand, as observable from figure 6-III it is feasible to identify that optimal  $ST$  value is located left from  $ST_0$  where cost function has minimum value.

Finding optimal operating point in 3D space of system parameters is found from the criterion for minimizing cost function. If both relationships for quantifying performance and power consumption are depending on operating point parameters according to linear equation in opposite direction, then it is expected that optimal parameters are found at the middle between boundary values. As the dependences are not linear as obvious from Figure 6, then it is expected more complex relationship between optimization criterion and system parameters.

## 6. CONCLUSION

The energy requirements and the performance in the operation of the IoT edge device are analyzed through the investigation of the typical data producer-consumer relationship. As the more generalized option, the IoT edge device was considered a typical data producer which operates under a push-based communication model. IoT edge device operation, under the influence of the identified set of parameters, was investigated utilizing the custom-built simulation environment. The simulation results have shown that the control of parameters such as sampling rate, aggregation rate, and transmission period at the data producer side can lead to the more optimal behavior of IoT systems in the power-performance domain, where the optimization criteria can be tuned to fulfill the particular application requirements. Simulation results confirmed the trade-off potential, where adjusting parameters often have opposite effects on the power requirement of the IoT edge device node and the resulting real-time performance of the IoT application. This trade-off potential was quantified by the introduced cost function, which defines the relationship between both, power, and performance domains, in linear form. By introducing the cost function, it has been shown that it is possible to find the optimal operating point where IoT system real-time and edge device energy consumption performance will be optimized in case where power consumption and performance are equally important to optimize. The exact position of optimal operating point in the 3D space of system parameters is complex to estimate without comprehensive parametric analysis since the complex relationship between system parameters and system power consumption and performance. In general, to lower energy consumption, in the same time compromising real-time performance, presumes less frequent sampling with higher aggregation rate and lower communication rate.

The utilization of this approach can result in the development of an algorithm that would control introduced parameters to achieve optimal compromise and enable the design of GIoT applications. The design and the implementation of an algorithm that controls the introduced set of parameters to achieve optimal operation of the edge devices, in the same way enabling the deployment of GIoT applications, is seen as a part of future work.

**Acknowledgment:** *This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.*

#### REFERENCES

- [1] H. Turkmanović, I. Popović, D. Drajić and Z. Čiča, "Launching Real-time IoT Applications on Energy-aware Embedded Platforms", In Proceedings of the 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), pp. 279-282, 2021.
- [2] R. Lu, X. Li, X. Liang, X. Shen and X. Lin. "GRS: The Green, Reliability, and Security of Emerging Machine to Machine Communications", *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 28-35, April 2011.
- [3] Cisco, "Cisco Annual Internet Report 2018-2023", March 2020.
- [4] A. S. H. Abdul-Qawy, N. M. S. Almurisi and S. Tadisetty, "Classification of Energy Saving Techniques for IoT-based Heterogeneous Wireless Nodes", *Procedia Comput. Sci.*, vol. 171, pp. 2590-2599, 2020.
- [5] X. Liu and N. Ansari, "Toward Green IoT: Energy Solutions and Key Challenges", *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 104-110, March 2019.
- [6] P. K. Verma, R. Verma, A. Prakash, A. Agrawal, K. Naik, R. Tripathi, M. Alsabaan, T. Khalifa, T. Abdelkader and A. Abogharaf, "Machine-to-Machine (M2M) communications: A survey", *J. Netw. Comput. Appl.*, vol. 66, pp. 83-105, 2016.
- [7] T. Xu, J. B. Wendt and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities", In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 417-423, 2014.
- [8] A. Damian and L. Kung-Kiu, "Evaluating IoT service composition mechanisms for the scalability of IoT systems", *Future Gener. Comput. Syst.*, vol. 108, pp. 827-848, 2020.
- [9] B. Diène, J. J. P. C. Rodrigues, O. Diallo, E. H. M. Ndoeye and V. V. Korotaev, "Data management techniques for Internet of Things", *Mech. Syst. Signal Process.*, vol. 138, April 2020.
- [10] C. C. Sobin, "A Survey on Architecture, Protocols, and Challenges in IoT", *Wirel. Pers. Commun.*, vol. 112, pp. 1383-1429, 2020.
- [11] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II, vol. 2, April 2005, pp. 8-13.
- [12] A. Ali, G. A. Shah and J. Arshad, "Energy-efficient techniques for M2M communication: A survey", *J. Netw. Comput. Appl.*, vol. 68, pp. 42-55, June 2016.
- [13] A. Azari and G. Miao, "Energy-efficient MAC for cellular-based M2M communications", In Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP), December 2014, pp. 128-132.
- [14] M. Muniswamaiah, T. Agerwala and C. C. Tappert, "Green computing for Internet of Things", In Proceedings of the 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2020, pp. 182-185.
- [15] H. Turkmanović and I. Popović, "A systematic approach for designing battery management system for embedded applications", In Proceedings of the Zooming Innovation in Consumer Technologies Conference (ZINC), May 2021, pp. 85-90.
- [16] H. Turkmanovic, <https://github.com/turkmanovic/LSNSimulator.git>, GitHub/turkmanovic, LSNSimulator.
- [17] H. Turkmanović, I. Popović, Z. Čiča and D. Drajić, "Simulation framework for performance analysis in multi-tier IoT Systems", In Proceedings of the 29th Telecommunications Forum (TELFOR), 2021, pp. 1-4.
- [18] U. B. K. Ramesh, S. Sentilles and I. Crnkovic, "Energy management in embedded systems: Towards a taxonomy", In Proceedings of the First International Workshop on Green and Sustainable Software (GREENS), 2012, , pp. 41-44.
- [19] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid and H. Yu, "Green IoT: An Investigation on Energy Saving Practices for 2020 and Beyond", *IEEE Access*, vol. 5, pp. 15667-15681, 2017.
- [20] A. Haider, T. Umair, H. James, Z. Xiaojun, L. Liu, Z. Yongjun, B. Faycal, A. Abbes, F. Kaniz, A. Niko, "A survey on system level energy optimisation for MPSoCs in IoT and consumer electronics", *Comput. Sci. Rev.*, vol. 41, p. 100416, Aug. 2021.
- [21] G. Anastasi, M. Conti, M. Francesco and A. Passarella, "Energy conservation in wireless sensor networks: A survey", *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537-568, May 2009.
- [22] R. Soua and P. Minet, "A survey on energy efficient techniques in wireless sensor networks", In Proceedings of the 4th Joint IFIP Wireless and Mobile Networking Conference, October 2011, pp. 1-9.

- [23] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, K. Araki, "Practical data compression in wireless sensor networks: A survey", *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 37-59, January 2012.
- [24] D. Parker, M. Stojanovic and C. Yu, "Exploiting temporal and spatial correlation in wireless sensor networks", In Proceedings of the Asilomar Conference on Signals, Systems and Computers, November 2013, pp. 442-446.
- [25] Y. Zhou, L. Yang, L. Yang and M. Ni, "Novel Energy-Efficient Data Gathering Scheme Exploiting Spatial-Temporal Correlation for Wireless Sensor Networks", *Wirel. Commun. Mobile Comput.*, vol. 2019, p. 4182563, 2019.
- [26] S. Randhawa and S. Jain, "Data Aggregation in Wireless Sensor Networks: Previous Research, Current Status, and Future Directions", *Wireless Pers Commun.*, vol. 97, pp. 3355-3425, July 2017.
- [27] S.-Y. Tsai, S.-I. Sou and M.-H. Tsai, "Reducing Energy Consumption by Data Aggregation in M2M Networks", *Wireless Pers Commun.*, vol. 74, pp. 1231-1244, Jan. 2014.
- [28] I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks", In Proceedings of the IEEE International Conference on Communications (IEEE Cat. No. 04CH37577), vol. 6, 2004, pp. 3640-3645.
- [29] T. Sheltami, M. Musaddiq and E. Shakshuki, "Data compression techniques in Wireless Sensor Networks", *Future Gener. Comput. Syst.*, vol. 64, pp. 151-162, Nov. 2016.
- [30] I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks", In Proceedings of the IEEE International Conference on Communications (IEEE Cat. No. 04CH37577), vol. 6, 2004, pp. 3640-3645.
- [31] W. Kim and I. Jung, "Smart Sensing Period for Efficient Energy Consumption in IoT Network", *Sensors*, vol. 19, no. 22, p. 4915, Nov. 2019.
- [32] C. Alippi, G. Anastasi, C. Galperti, F. Mancini and M. Roveri, "Adaptive Sampling for Energy Conservation in Wireless Sensor Networks for Snow Monitoring Applications", In Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems, October 2007, pp. 1-6.
- [33] M. Hempstead, M. J. Lyons, D. Brooks and G.Y. Wei, "Survey of Hardware Systems for Wireless Sensor Networks", *J. Low Power Electronics*, vol. 4, pp. 1-10, April 2008.
- [34] W. H. Cheng, W. Isaac, Y. Cheng-Wen, A. Alagan and O. Mohammad, "Energy-efficient tasks scheduling algorithm for real-time multiprocessor embedded systems", *J. Supercomput.*, vol. 62, pp. 967-988, Nov. 2012.
- [35] S. Li and J. Huang, "Energy Efficient Resource Management and Task Scheduling for IoT Services in Edge Computing Paradigm", In Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications and IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), December 2017, pp. 846-851.
- [36] C. H. Lin, J. C. Liu and C. W. Liao, "Energy analysis of multimedia video decoding on mobile handheld devices", *Comput. Stand. Interfaces*, vol. 32, no. 1-2, pp. 10-17, Jan. 2010.
- [37] S. A. Alvi, G. A. Shah, W. Mahmood, "Energy efficient green routing protocol for Internet of Multimedia Things", In Proceedings of the IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), May 2015, pp. 1-6.
- [38] S. Tanzila, H. Khalid, A. Imran and R. Amjad, "Secure and energy-efficient framework using Internet of Medical Things for e-healthcare", *J. Infect. Public Health*, vol. 13, no. 10, pp. 1567-1575, July 2020.
- [39] A. Lindgren, F. B. Abdesslem, B. Ahlgren, O. Schelén and A. M. Malik, "Design choices for the IoT in Information-Centric Networks", In Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference (CCNC), January 2016, pp. 882-888.
- [40] R. C. Sofia and P. M. Mendes, "An Overview on Push-Based Communication Models for Information-Centric Networking", *Future Internet*, vol. 11, no. 3, p. 74, March 2019.