A Green Data Management Layer in Industrial IoT

A Ghaderi¹, Z Qavany¹ and Z Movahedi¹,²,*

¹School of Computer Engineering, Iran University of Sciences and Technology (IUST), Narmak, Tehran, Iran.
²School of Computer Engineering, Iran University of Sciences and Technology (IUST), Narmak, Tehran, Iran.
*Corresponding author E-mail: zmovahedi@iust.ac.ir

Abstract. All Nowadays, industry 4.0, which aims at transforming industrial manufacturing systems to the smart factories, has gained considerable attention from both industry and academia. The Industrial Internet of Things (IIoT) plays a primordial role to achieve the objectives of Industry 4.0, by enabling the Cyber-Physical Production Systems (CPPS) to communicate and cooperate with each other and with humans both internally and across the participants of the supply chain. A communication model used commonly in IIoT networks consists of the publish-subscribe model, in which the data generated by sensor nodes are cached in a central controller to be subsequently consumed by actuator nodes. However, the traditional centralized data management is inappropriate for real application because of its communication overhead and inability to cope with strict delay requirements of IIoT applications and energy constraints of IIoT networks. To address these issues, we propose a Green Data Management Layer (GDML), which caches data distributedly in some IoT nodes, called proxy nodes, so that the network energy consumption is optimized while the constraints on data access latency and IoT nodes’ cache capacity are met. In this measure, the extra IoT nodes not involved in data transmission nor in data caching are switched off to prolong the battery lifetime of constrained IoT nodes. We modeled the aforementioned optimization problem as an Integer Linear Programming (ILP) and solved it using the CPLEX tool. The obtained results show that the proposed GDML outperforms significantly alternative solutions in terms of overall energy consumption, energy consumption devoted to data transmission, energy consumption devoted to keeping active the involved nodes, while the constraints on data applications latency and the cache size of IoT nodes are satisfied.

Keywords: Industry 4.0, Industrial IoT, Data Management, Energy Efficiency, Green Manufacturing

1. Introduction

Internet of Thing (IoT) is a concept describing the ubiquitous connection of billions of deployed smart objects which sense the surrounding environment, process acquired data, and then send feedback to the environment [1]. As a subset of IoT, Industrial IoT (IIoT) mainly focuses on the transfer and control of mission-critical information, which relies on machine-to-machine (M2M) and industrial
communication technologies [2]. Limiting the definition of IIoT to manufacturing gives another subset of IoT, known as Industry 4.0 [3]. Industry 4.0, refers to the fourth revolution of industry, which is merged with Cyber-Physical Production Systems (CPPS), focusing on smart manufacturing. IIoT and Industry 4.0, are two terms that are often used interchangeably; the aims of both are making industrial networks robust, fast, reliable, and secure. By utilization of IIoT in a smart factory, it would be possible to optimize the production line with better Quality of Service (QoS), energy efficiency, and access latency.

In an IIoT environment, colossal amount of data will be generated. The data which are obtained from sensors are analyzed by controllers to generate valuable information for actuators. The routing of the data from sensors to controllers and actuators is a significant aspect of any industrial network environment for maintaining data access latency constraints and ensuring efficient energy consumption. Due to the growth and different type of data traffic in IIoT networks and IoT devices are energy constraints, managing energy consumption is an essential challenge in IIoT [4]. Energy conservation is not only crucial to green manufacturing, which covers and reduces environmental pollution and economic cost, but it efficiently prolongs the network lifetime [5]. In IIoT, data generated continuously may contain critical information and may be time-critical as well. To this end, machines require rapid responses at times. For example, in various applications such as monitoring chemical plants, aircraft maintenance, mining industries, inspecting underground pipelines, and so on, there is data criticality which small error or delay beyond the acceptable maximum latency might result in unavoidable damages [6], [7].

The more commonly used architecture model for transmitting data in IIoT is Publish/Subscribe (Pub/Sub) [8]. The model consists of three main components: publishers, subscribers, and a broker. Publishers are the lightweight sensors that are connected to the broker to send their data. Subscribers are actuators interested in a specific topic, so they connect to brokers to be informed whenever new data are received. The brokers classify sensory data in topics and send them to subscribers interested in those topics only. Because incoming traffic flows around the central controller (broker), extra communication overhead, maximum data access latency, and high energy consumption, implementing this model centrally might be inapplicable in IIoT.

To adopt the Pub/Sub model for the context of IIoT, several researches have already focused on investigating distributed approaches, which exploit the presence of a few more capable nodes that act as distributed local data storage [9], [10], and [11]. However, the status of research is in this field is still in its infancy. In this approach, instead of storing data in a central controller, data are stored in several intermediary nodes, called proxy nodes. These distributed approaches have some lack in improving overall energy consumed from sensors to proxy nodes and from proxy nodes to actuators, considering the limited amount of storage capacity for proxy nodes and prioritizing the data, which makes it inappropriate for IIoT. However, due to the energy constraint, most the networks are designed in its peak period. In some cases, some proxy nodes are involved neither in data transmission nor in data caching. Therefore, they could be switched off to achieve more network performance during high traffic situations.

To handle the aforementioned shortcomings, this paper proposes a Green Data Management Layer (GDML), which decouples the data management plane from the routing (network) plane. Our application in the IIoT network requires source nodes that produce data (sensor nodes), destination nodes that consume data (actuator nodes), and intermediary nodes that store data as proxy nodes. The proposed GDML offers an efficient method for distributing data between sources and destinations. Hence, The GDML, selects proxy nodes that store data in a place close to sensor and actuator nodes for maintaining critical access latency and minimizing energy consumption. GDML selects proxies from a set of potential proxies that are more potent than ordinary nodes, because of its storage, computational, and communication activities. To reduce wasteful resources in green manufacturing, GDML switched some potential proxies to the idle mode. Furthermore, we improve overall energy consumption in both sides of the industrial network, one side from the sensor to proxy nodes and one side from proxy nodes to the actuator. Also, we distinguish two types of energy consumption: (1) Energy that is consumed by turning on the proxy function of nodes and (2) Energy consumed by transferring data. We observe that in some applications, data should be accessed more rapidly than
other data, so we prioritize the data and consider the presence of urgent data. To manage urgent data latency requirements, we distinguish data in two categories: normal data and urgent data. Additionally, we consider that proxy nodes have a limited amount of cache capacity.

The rest of this paper is organized as follows. In section 2, we provide a brief overview of related works. In section 3, the system model is presented. In section 4, we introduce the proposed GDML. Evaluate settings, and obtained results are discussed in section 5. Finally, in section 6, we conclude the paper and provide some insights for future works.

2. Related Works

In [12], the authors considered a real IIoT model based on WirelessHART to satisfy end-to-end delay in a harsh industrial environment from sensors to actuators. The technology of WirelessHART uses a centralized management approach, which handles the requirements of the network. The centralized approach has some drawbacks, such as prone to the single point of failure problem, high communication overhead around the central controller, and incapability with network dynamicity. Since they did not expect the presence of some proxy nodes, their approach is stuck in the traditional centralized IIoT setting.

The most relevant works to this paper are [9] and [10], in which the authors presented a distributed approach and use proxy nodes to implement a decentralized data distribution across the network. In [9], the authors presented actuator-to-proxy assignments to guarantee a maximum delivery latency to actuator nodes. Although they considered delay aspects to put on average end-to-end data access latency below a given time threshold, their main objective was minimizing the number of proxies. Moreover, they did not consider energy consumption as a constraint, but in [10], the authors considered energy consumption to address the lifetime maximization problem, under latency constraints. To achieve this goal, they considered a problem with the initial limited energy supplies of the nodes and predefined proxy locations in the network. To obtain network energy consumption and to maximize the time until the first node dies, they computing the energy drain on the nodes, which are only located in possible paths that would meet the maximum latency constraint, not all the existing paths. Also, in [11], the authors have extended their previous work by some contributions. They introduced two improvements so that the data distribution can dynamically change before the first node depletes its energy. One improvement is centralized, which uses local area wireless communication to optimize paths based on global knowledge and re-establish the data distribution schemes according to the network energy map. The other improvement is distributed, which periodically rotates the available data distribution paths in a relatively fair manner.

However, in [13], the authors have been investigated a new contribution in the field of energy efficiency for industrial field data. They introduced a method that reconfigured path in a local and distributed manner when a network node failed, or excessive degradation of the performance of wireless links occurred. In addition, their method regulates the return to an operational state of nodes that have been offline in the past. Also, they regulated how the local path reconfiguration should be implemented and how a node can join a new path or modify an already existing path, ensuring that there will be no loops and scarifying latency guarantees to improve energy consumption. All of the works as mentioned above, did not consider a limited amount of cache capacity for proxy nodes. Moreover, they did not consider various traffic types in an industrial environment with various data volume, latency, and importance.

Another relevant work for data distribution in the context of IIoT is [14]. The authors introduced some special devices as software-defined networking controllers and sink nodes that are selected and located in the network to reduce deployment cost and delay. They proposed a method to satisfy time-critical IoT systems requirements and ensuring that each sensor is covered by a certain number of sinks and controllers and considering the capacity of unique devices concerning both the size and frequency of data generated by sensors. Their primary objective is network coverage, while increasing network efficiency by putting a limitation on the maximum number of hops between sensors and sinks and between controllers and sensors is considered as a secondary objective. Even though, they considered random for both deployment and topology; they did not take energy consumption into account.
Last but not least, caching popular data in brokers, which are considered as an application middleware nodes, have been investigated in [15]. The authors proposed a method that re-cache the popular data from heavily loaded brokers into lightly loaded brokers in order to achieve better traffic balancing among brokers, thus reducing the average delay of brokers in transmitting the content of popular cached data to the clients. For reducing the traffic loads of the server, they specified which data are suitable for caching in brokers to solve the problem of energy consumption of servers and which brokers should cache the pertinent data to solve the problem of traffic balancing. For reducing the energy consumption of the network, they only considered the energy which consumes by servers, and they did not consider the overall energy consumption devoted to the brokers and data transmission.

To summarize, implementing IIoT based on a central network controller, which is centralized network management, has been suffered from IIoT requirements. Skimming over the related works exposes that none of those works did focus on the problem of overall energy consumption, which is considered one of the most crucial challenges in IIoT. In [10], they computed the energy drain on the nodes in the admissible paths, which has met the maximum latency constraint, not all existing paths. Even they did not consider some nodes which is involved neither in data caching nor in data transmission for switching to the idle mode. Hence, their energy consumption problem is not appropriate for energy constraint green manufacturing. To the best of our knowledge, there are no existing works to consider various data types in an industrial environment, such as criticality, volume, traffic, and variety.

3. System Model

In this section, we introduce our model in three different aspects. First, we describe the model of our network, while introducing our main assumptions. Then we declare how to model data as data piece and features of each data piece such as urgency and latency. Finally, we state our energy model to figure out how much energy needs to maintain a stable network. To understand better the system model, we propose pseudo-code of the model at the end of this section.

3.1. Network Model

We model an industrial IoT network by a directed graph $G(V, E)$, where $V = \{v_1, ..., v_n\}$ is the set of $n$ nodes, and $E$ is a set of edges that connect nodes with direct communication links. Each node $v_i \in V$ has a limited capacity $m$, which ensures that each node does not store data more than its maximum capacity. Every node $v_i \in V$ can transmit data via industrial wireless technologies to a set of nodes are located in its neighborhood $N_i$. We define $N_p$ as a distance between $v_i$ and every neighbor $v_j \in N_p$, which should be less or equal than the transmission range of node $v_i$. The central network controller $C$, which maintains comprehensive network knowledge, is the first node in the network, $C = v_1$. This knowledge has consisted of several information, such as the location of nodes, maximum capacity of nodes, the length of the shortest path between node $v_i$ and $v_j$, and the total number of proxy nodes, which denoted by $p$. To this end, we use a set of potential proxy nodes $P$, with $p \in P \subset V$ which cache data from sensors that can be accessed on time by actuators.

3.2. Data Model

Generally, we divided data into two terms, urgent data and normal data. We model the data a set of $D = \{d_1, ..., d_n\}$ of data pieces. Each data piece is a tuple $D_i = (s_i, c_i, z_i, u_i)$ consisting of the source node $s_i \in V$, the consumer node $c_i \in V$, the size of the data piece $z_i$, and urgency $u_i \in \{0, 1\}$ which indicates if the data piece $d_i$ is urgent or not.

To maintain data access latency restriction, we introduce $l_{v_i,v_j}$ which indicates that estimated time to send data from node $v_i$ to node $v_j$. As communication in industrial IoT networks, it is vital that in most cases, we have to guarantee low data access latency for the communication between consumer and proxy. Therefore we consider $L_{max}$ as the maximum tolerable latency for each consumer. In urgent data which have higher prioritized data pieces, should be sent rapidly. So we add a second latency threshold $L_{max}$ to the model.
3.3. Energy Model
To maintain energy efficiency, we introduce two kinds of energy consumption: one for the energy of nodes and one for the energy of paths. We model the first type of energy consumption for using a potential proxy node \( P \) as proxy node \( p \). In this work, each \( P \) have two status: active (i.e., communication or caching) and idle. Note that in the idle mode, a \( P \) is ON, but its internal memory is not used and refused to participate in transmitting/receiving data. So \( e_{p}^{on} \) are the energy costs of activating node \( P \) as a proxy node, while other node \( P \) involved neither in data caching nor in data transmission are switched to the idle mode. The energy consumption of an active \( P \) represents the peak energy consumption, and an idle mode for \( P \) consumes a minuscule amount of energy.

We model the second type of energy consumption for the amount of data transmission in the path from sensor to proxy node and from proxy node to actuator. So \( E_{l,p}^{path} \) is the energy consumed per byte for routing of data piece \( d_{l} \) across proxy \( p \) and let \( l_{v_{i},v_{j}} \) the shortest path from node \( v_{i} \) to node \( v_{j} \), obtained by Dijkstra’s Algorithm [16]. The energy consumption for routing can be formulated as the following:

\[
E_{l,p}^{path} = e_{l}^{transmit} + e_{l}^{receive} + \left( \sum_{v_{n} \in (l_{v_{i},v_{j}} \cup l_{i,p}) \setminus \{l_{v_{i},l_{v_{j}}}, l_{v_{j},l_{v_{i}}}, l_{v_{i}}\}} e_{n}^{receive} + e_{n}^{transmit} \right)
\]

Where \( e_{p}^{receive} \) is the energy to receive data in proxy \( p \), and \( e_{p}^{transmit} \) is the energy to transmit data from proxy \( p \).

We now introduce GreenDataManagementLayer (Algorithm 1), which creates a random network model with the nodes ordered in a grid. GreenDataManagementLayer, defines possible proxies, generate data pieces and compute energy consumption.

**Algorithm 1**: GreenDataManagementLayer

**Input**: \( dimX, dimY, dataAmount, urgentDataAmount, cacheSize, e_{on}, E_{send}, E_{receive}, L_{max}, L_{max}^{u} \)

**Line 1**: Creating network model \((G \leftarrow \text{Creating grid of dimantions }^{\text{dim}X*\text{dim}Y}, s \leftarrow \text{random sources, } c \leftarrow \text{random consumers, } u \leftarrow \text{random urgent data})\)

**Line 2**: \( shp \leftarrow \text{Shortest paths using Dijkstra’s algorithm} \)

**Line 3**: \( E_{path} \leftarrow \text{Compute energy consumed by shp} \)

**Line 4**: \( P \leftarrow \text{Proxy for each data piece minimizing the objective function using cplexbip function} \)

**Line 5**: \( E \leftarrow \text{Energy costs of solution paths considering } P \)

**Output**: \( P, E \)

4. Green Data Management Layer
To manage data distribution and jointly decrease the energy consumption and data access latency, we introduced the GDML. The rudimentary element of GDML is a set of potential proxy nodes deployed in the field network. The GDML provides a solution to correctly select and active some potential proxy nodes as a proxy node to cache and communicate data that can be accessed by the consumer on time as well as energy efficiency. Notably, the GDML define a set of \( P \in V \) which are potential proxy nodes, then select proxy \( p \in P \) among them to act their roles.

Next, we want to formulate the problem as an integer program. We define the decision variables \( x_{p} \), which indicates whether node \( p \) is used as a proxy node to cache data, and the variable \( y_{i,p} \), which indicates if node \( p \) is the corresponded proxy for data piece \( d_{i} \). The objective function of the program is to minimize energy consumption. We distinguish between two types of energy consumption: \( e_{p}^{on} \) is consumed by turning on the proxy function of nodes, and \( E_{l,p}^{path} \) is consumed by transferring data. In order to balance the two types of energy consumption, we add the constraint \( \lambda \). Setting \( \lambda = 0.5 \) leads to a focus on both types equally, while \( \lambda = 0 \) focuses the objective function only on the energy consumed by transferring data. The problem can then be formulated as the following integer program:
\[ \min \lambda \cdot \sum_{p \in P} x_p \cdot e_p^{on} + (1 - \lambda) \cdot \sum_{d \in D} \sum_{p \in P} \mathbb{E}^\text{path}_{l_p} \cdot y_{l_p} \cdot v_i \]  
\[ \text{s.t.} \quad \sum_{p \in P} l_p \cdot c_i \cdot y_{l_p} \leq (1 - u_i) \cdot L_{\text{max}} + u_i \cdot L_{\text{max}} \quad \forall d_i \in D \]  
\[ \sum_{d \in D} y_{l_p} \cdot v_i \leq m_p \quad \forall p \in P \]  
\[ \sum_{p \in P} y_{l_p} = 1 \quad \forall d_i \in D \]  
\[ y_{l_p} \leq x_p \quad \forall p \in P, \forall d_i \in D \]  
\[ x_p, y_{l_p} \in \{0,1\} \quad \forall p \in P, \forall d_i \in D \]  

Constraint (3) ensures that the latency thresholds for normal data and urgent data are not exceeded. Constraint (4) guarantees that the cache of each proxy is not overrun. Constraint (5) ensures that each data piece is only assigned to exactly one proxy while constraint (6) guarantees that each node that is assigned to data needs to turn on its proxy functionalities. Finally, constraint (7) sets the decision variables to be binary integer values.

5. Performance Evaluation

In this section, we demonstrate the performance of the proposed method through extensive evaluation by considering different network scales and compare it against the appropriate method [9], which is mentioned in section 2. We analyze the performance of the proposed method based on the optimization problem by CPLEX solver, and we simulate the system model by MATLAB simulation environment.

The criteria we use to verify the simulations are total energy consumption, average access latency for normal data, urgent data, and both, and cache utilization of proxy nodes. We then explain the parameters setting, and finally, observe the analysis for each figure.

5.1. Parameters Setting

We constructed, in simulation, different typical industrial network scales, from small-scale to large-scale. In each scale, we consider the different situations of data presence, from a small amount of data to a large amount of data, respectively, during off-peak and peak periods. We set the number of nodes from 100 to 392 and the percentage of data from 10\% to 50\% of nodes. In each situation, we consider half of the data urgent and half of them normal.

To simulate a real industrial environment, we consider the condition of one of the most resource constraint motes. So we choose 92 KB for internal memory of proxy nodes as a cache capacity [17] and 15 KB for the typical payload size of each data [18]. However, for simulating the real energy consumption of industrial networks, we consider the same condition of the Zolertia Z1 mote, which uses the CC2420 radio transceiver module. Due to radio power consumption [19], we set the energy for transmitting and receiving data and keeping active and idle the proxy nodes respectively 52.2 mW, 56.4 mW, 6 mW, and 1.28 mW. The details of the simulation are exposed in table 1.

| Table 1. Simulation parameters |
|-------------------------------|
| **Parameter**                 | **Value**                  |
| **Topology**                  |                             |
| nodes deployment              | Grid                        |
| Number of nodes \([V]\)       | 100, 174, 246, 320, 392    |
| Number of potential proxy nodes \([P]\) | All inside nodes (without borders) |
| Number of proxy nodes \([p]\) | vary due to the execution problem |
| Number of source and consumer nodes \([s], [c]\) | vary due to the execution problem |
| **Hardware**                  |                             |
Resource constraint mote | Zolertia Z1
---|---
Energy of transmitting data ($e^{\text{transmit}}$) | 52.2 mW
Energy of receiving data ($e^{\text{receive}}$) | 56.4 mW
Energy of keeping active node ($e^{\text{on}}$) | 6 mW
Energy of keeping idle node ($e^{\text{idle}}$) | 1.28 mW
Internal memory ($m$) | 92 KB

**Time**
- One-hop latency ($l^{(h)}$) | 18 ms
- Normal latency threshold ($L_{\text{max}}$) | 100 ms
- Urgent latency threshold ($L^{u}_{\text{max}}$) | 80 ms

**Data**
- Amount of normal data ($D$) | 10%, 20%, 30%, 40%, 50% of $|V|$  
- Amount of urgent data ($u$) | 50% of ($D$)
- Payload size ($z$) | 15 KB

5.2. Analysis

To better evaluation, we compare our optimal solution (GDML) with optimal solution (DML) in [9]. We compare GDML with DML in each criterion such as, energy consumption, average access latency, and cache utilization.

Energy consumption: Since our goal is to minimize energy consumption, we thoroughly evaluate our objective function with other parameters. First, we evaluate the objective function by changing the lambda and data parameters. In figure 1a, we observe that energy costs are high when the lambda parameter is toward zero. Note that paths consume much more energy than active nodes. For the other figures, we set the lambda parameter to 0.5 to focus on both types of energy costs.

In figure 1b, we can see that the GDML is performing really well compared to the DML in a various data presence, especially in higher data presence. It is evident that with the increase in the amount of data, we would have more energy costs. For example, when the amount of data presence is 50% of number of nodes, we have really better energy consumption compared to the reference model.

![Figure 1](image1.png)

(a) Different values of lambda  
(b) Different amount of data

**Figure 1.** Comparison of objective function with lambda and data parameters
In figure 2, we evaluate objective function simultaneously with the lambda and data parameters. The energy consumption is at the highest point when the lambda is set to zero to considering only path costs with the maximum amount of data.

![Figure 2. Comparison of objective function simultaneously with lambda and data parameters](image)

In figure 3, we evaluate objective function with normal and urgent data simultaneously. With an increasing amount of normal data, energy costs are increased, but with an increasing amount of urgent data, energy costs are not really changed.

![Figure 3. Comparison of objective function with normal and urgent data simultaneously](image)
Finally, we evaluate objective function by changing the data parameter on different scales. All figures mentioned above, were on the scale of 100 nodes, but now we evaluate objective function with different scaling. Figure 4 displays energy costs from small scale to large scale with different data presence on each scale. The same observation can be made here. Indeed, when the amount of data kept increasing with the network scalability, much energy is consumed. It noticed that the GDML is more efficient even in the large scale network with 392 nodes and maximum amount of data with 50% of 392 nodes.

![Figure 4. Comparison of objective function with amount of data in different scales](image)

Average access latency: Now, we want to evaluate average access latency in three different aspects. In each aspect, GDML makes sure that average access latency is always below the maximum normal and urgent latency threshold. We compare latency with the amount of data which half of them are normal data, and half of them are urgent data. In figure 5a, we compare average normal latency with the amount of data, and it guarantees that the maximum latency threshold is not violated by average latency.

In figure 5b, we compare average urgent latency with the amount of data. In this compression, average latency should be below the 80 ms, which is the maximum latency threshold for urgent latency. Also, we combine normal latency and urgent latency as a total average latency and compare it with the amount of data in figure 5c.
Cache utilization: Another aspect that we can easily evaluate in the simulation is the cache capacity usage for proxy nodes. We ensure that each proxy node does not store data more than its maximum capacity. So if we have more data presence in the network, we have to balance caching data with concerning internal capacity usage of each proxy node. Figure 6 illustrates that the GDML balanced caching data in which cache utilization is not increased even in the maximum data distribution.

**Figure 5.** Comparison of different aspect of latency with amount of data

**Figure 6.** Comparison of cache utilization with amount of data

5.3. **Discussion**

To better clarify the superiority of our model over the other alternatives models, we now discuss the GDML performance with the state of the art researches especially [9] and [10]. Most researches did not consider a limited amount of cache capacity for proxy nodes, which helps to balanced caching data across proxy nodes with regards to its capacity. So GDML is performing really well compared to the reference model with large amount of data. Another advantage of our model compared to the reference model is the low latency in different conditions such as normal, urgent and total latency. We also modeled the energy consumption in two terms, one for data transmission and the other for data caching. Since we calculate energy consumption in all existing paths and consider some nodes in sleep mode that do not play a role in data caching, we can claim that our model calculates overall energy...
consumption. However, it also has better result than the reference model in each scale with different amount of data. To the best of our knowledge, there are no existing works to consider data criticality, but we modeled urgent data that have higher priority that should be sent rapidly and meet the maximum urgent latency threshold.

6. Conclusion
In this paper, we efficiently managed data distribution in industrial IoT networks by using proxy nodes and specified each data as a data piece to denote source and consumer of data. We consider a method for identifying a set of proxy nodes among potential proxy nodes in which data are cached for the use of consumers in a timely manner. Also, we consider a high-risk industrial network with urgent data that should be responsible in an acceptable latency, which we called urgent latency. We introduced an optimized method for optimizing energy consumption in the proxy nodes and the paths by switching off some extra nodes which are involved neither in data transmission nor in data caching. Additionally, we consider cache utilization of each proxy node. We validate a simulation model and use it for performance evaluation in different network scales. The obtained results show that the proposed method outperforms significantly alternative solutions in terms of overall energy consumption, while the constraints on data application latency and the cache size of proxy nodes are satisfied.

References
[1] Sisinni E, Saifullah A, Han S, Jennehag U and Gidlund M 2018 Industrial internet of things: Challenges, opportunities and directions IEEE Trans. Industr. Inform. 14 4724-34
[2] Aazam M, Zeadally S and Harras K A 2018 Deploying fog computing in industrial internet of things and industry 4.0 IEEE Trans. Industr. Inform. 14 4674-82
[3] Xu L D, Xu E L and Li L 2018 Industry 4.0: state of the art and future trends Int. J. Prod. 56 2941-62
[4] Oh E and Son S Y 2016 Toward dynamic energy management for green manufacturing systems IEEE Commun. Mag. 54 74-79
[5] Dabaghi-Zarandi F and Movahedi Z 2017 An energy-efficient algorithm based on sleep-scheduling in IP backbone networks Int. J. Commun. Syst. 30 e3276
[6] Shu L, Mukherjee M, Pecht M, Crespi N and Han S N 2017 Challenges and research issues of data management in IoT for large-scale petrochemical plants IEEE Syst. J. 12 2509-23
[7] Wu D, Chatzigeorgiou D, Youcef-Toumi K and Ben-Mansour R 2015 Node localization in robotic sensor networks for pipeline inspection IEEE Trans. Industr. Inform. 12 809-19
[8] Salman T and Jain R 2019 A survey of protocols and standards for internet of things preprint arXiv:1903.11549
[9] Raptis T P, Passarella A and Conti M 2018 Performance analysis of latency-aware data management in industrial IoT networks J. Sens. 18 2611
[10] Raptis T P, Passarella A and Conti M 2018 Maximizing industrial IoT network lifetime under latency constraints through edge data distribution IEEE 1st Industrial Cyber-Physical Systems (St. Petersburg) pp 708-13
[11] Raptis T P, Passarella A and Conti M 2020 Distributed Data Access in Industrial Edge Networks IEEE J. Sel. Area. Comm. 38 915-27
[12] Saifullah A, Xu Y, Lu C, and Chen Y 2014 End-to-end communication delay analysis in industrial wireless networks IEEE J. Sel. Area. Comm. 64 1361-74
[13] Raptis T P, Passarella A and Conti M 2020 Energy efficient network path reconfiguration for industrial field data Comput. Commun. 158 1-9
[14] Faragardi H R, Vahabi M, Fotouhi H, Nolte T and Fahringer T 2018 An efficient placement of sinks and SDN controller nodes for optimizing the design cost of industrial IoT systems Software Pract. Exper. 48 1893-919
[15] Sun X and Ansari N 2017 Traffic load balancing among brokers at the IoT application layer
    *IEEE T. Netw. Serv. Man.* **15** 489-502

[16] Dijkstra E W 1959 A note on two problems in connexion with graphs *Numer. Math.* **1** 269-271

[17] Koutsiamanis R A, Papadopoulos G Z, Fafoutis X, Del Fiore J M, Thubert P and Montavont N
    2018 From best effort to deterministic packet delivery for wireless industrial IoT networks
    *IEEE Trans. Industr. Inform.* **14** 4468-80

[18] White Paper 2019 5G for Connected Industries and Automation Second Edition *5G Alliance for
    Connected Industries and Automation (ZVEI – German Electrical and Electronic
    Manufacturers’ Association, Frankfurt am Main)*

[19] Zolertia WSN 2010 platform Z1 Datasheet