Cloud-Edge Collaborative Optimization for Information Layer of Energy Internet

1st Yutong Li
Beijing National Research Center for Information Science and Technology
Tsinghua University
Beijing, P. R. China

2nd Haochen Hua
College of Energy and Electrical Engineering
Hohai University
Nanjing, P. R. China

3rd Jian Guo, 4th Junwei Cao*
Beijing National Research Center for Information Science and Technology
Tsinghua University
Beijing, P. R. China
jcao@mail.tsinghua.edu.cn

***Abstract*—Accompanied with the energy crisis and global warming, the concept of Energy Internet (EI) emerges, which aims to make full use of renewable energy. Among many possible structures, a particularly interesting EI structure has multiple microgrids interconnected by energy routers. In order to ensure that energy routers can control and transmit energy efficiently and reliably among microgrids, information technology plays an important role. Focusing on the information layer of the considered EI structure, a novel edge-cloud coordination architecture is proposed in this paper. A task allocation strategy is developed to improve the reliability and efficiency of the information layer. In order to reduce the information delay as well as ensuring the reliability of information transmission, a joint optimization problem is formulated and solved by particle swarm optimization algorithm. The effectiveness of the proposed method are demonstrated by simulations.**

***Index Terms*—edge computing, energy internet, energy router, microgrid, smart grid**

I. Introduction

Nowadays, in order to cope with the challenges of environmental pollution, energy crisis and global warming, a new energy system, energy Internet (EI), which can make full use of renewable energy, has been widely concerned by both academia and industry [1]- [4]. The EI, which is advocated to be built in a bottom-up mode, is regarded as an upgraded version of the smart grid [5]- [7]. It is different from the conventional power system which is built in the top-down and centralized mode. In the EI, the main power system is the backbone and the microgrid (MG) is regarded as the local area network. MGs can be a kind of small self-sufficient power networks composed of traditional power generation devices, renewable energy, local energy storage devices and a variety of loads [8]- [10].

This work was funded in part by the National Key Research and Development Program of China (Grant No. 2017YFE0132100), Tsinghua-Toyota Joint Research Institute Cross-discipline Program, and the BNRist Program (Grant No. BNR2020TD01009).

Compared with the traditional power system, the exchange of information and energy in the EI scenario is more important [11]. As the core component of EI, the energy router (ER) plays a key role in the implementation and management of information and energy exchange [12]. In order to ensure that the renewable energy can be appropriately converted into the required energy in real time and that energy can be used efficiently, information technology shall be widely used in power information collection, power quality monitoring and energy management [13]. Therefore, on the information level, effective information communication and computing capabilities can improve the production, utilization and transmission efficiency of energy [14], [15].

However, in order to achieve the goal of self-sufficiency of each MG in the EI, various controllers, power monitoring equipment and electrical equipment shall be equipped to EI scenarios, and the terminals of Internet of Thing (IoT) are intelligent and diversified [16]. The intellectualization of EI also calls for the deployment of key functions such as model prediction, fault prediction and power control [17]. These functions have very high requirements for the computing power of the running carrier. The traditional method is to offload these tasks to the cloud with strong computing power.

Due to the fact that cloud computing can effectively integrate network resources and provide powerful information processing capabilities, it is feasible to solve the information processing problems in EI by using cloud computing [14]. However, the massive data generated by a large number of terminal devices would normally lead to the instability of the network [18]. Meanwhile, many computing tasks are very sensitive to the delay of information [19]. All of these could seriously affect the security and reliability of EI, thus affecting the management and control of energy and the user's experience of service (EoS) [20]. Therefore, in order to make up for the deficiency of cloud computing and improve the reliability of the information layer in EI, we turn our attention to a new computing paradigm–edge computing (EC).

EC is a new computing paradigm that places some of the resources of cloud computing, such as computing, storage, network, etc., in the edge which is closer to the devices [21], [22]. By deploying a series of edge servers near the edge of the devices, the computing tasks of the devices can be processed locally, which greatly reduces the time delay caused by the remote network transmission, so as to improve the stability of the energy layer in EI [11], [23]. Despite the fact that EC has notable advantages, it has defects, such as the limited computational capacity on the edge side, which can be made up by the powerful computational capacity of the cloud server [24]. Therefore, it is significant to design a cloud-edge collaborative architecture in EI scenarios to optimize task scheduling and improve the communication and computing efficiency of the information layer.

In recent years, when the technique of EC is considered in EI, the related research has attracted much attention, and significant advances on this topic have been made; see, e.g., [18], [25] and [26]. In [25], considering the transmission delay and computation delay of information, an EC based workload allocation method is proposed to reduce the service delay of the power system. Compared with [25], the workload allocation method proposed by [18] additionally considers the minimization of energy consumption. In [26], an energy-efficient computing offloading method is proposed to minimize the long-term system cost. In [13], a multi-tier communication architecture is proposed for energy trading systems. On the other hand, the reliability of EC system have also received much attention [11]. In [11], an energy management framework is proposed to support the reliable operation of EC in power system. However, few work has considered both the reliability of information communication and task scheduling simultaneously.

In order to deal with the problem of high delay and low reliability caused by the access of a large number of power equipment, EC is introduced into the considered EI scenario in this paper. The structure of EI is generally divided into information layer and energy layer. It is worth noting that information communication plays a central nervous role in EI [14]. In this EC architecture, we formulate an optimization problem to reduce the delay and ensure the reliability in the information layer, thus improving the reliability of energy layer. The particle swarm optimization algorithm is used to solve this optimization problem.

The main contributions of this paper are as follows:

1) A novel EC architecture in EI scenario is proposed, and a task splitting strategy based on cloud-edge collaboration is applied in this paper to make full use of computing resources on both cloud and edge. Most of the existing literatures only consider reducing the delay of information transmission and computation [27], whereas the reliability of information transmission has not been considered. In this paper, these factors have been considered simultaneously.

2) A joint optimization problem for delay and reliability is established. Considering that particle swarm optimization algorithm has the characteristics of strong global optimization ability and fast search speed, this optimization problem is transformed into a form which can be solved by the particle swarm optimization algorithm.

The rest of the paper is organized as follows. Section II describes the modeling for the delay of information transmission, information computation and information reliability. Section III formulates the joint optimization problem. In Section IV, the particle swarm optimization algorithm is introduced to solve the optimization problem numerically. Section V provides some numerical simulations and summarizes the outcomes. Finally, the conclusion is presented in Section VI.

II. System Modeling

In this section, we describe the EC architecture and formulate the delay model of data communication and task computation. After that, we formulate the reliability model of communication in the information layer.

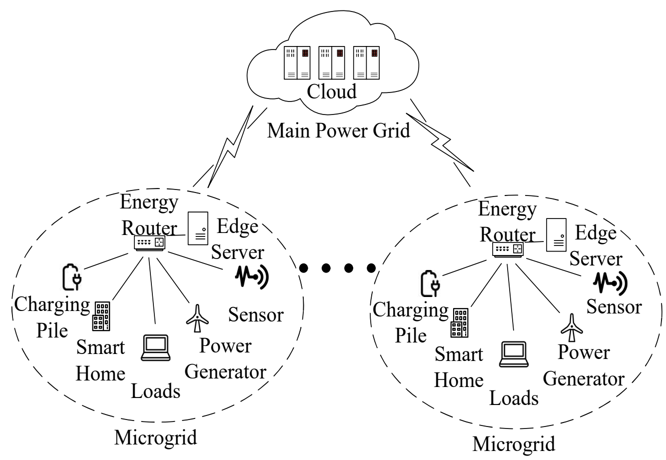

Fig. 1. EC architecture

A. Edge Computing Architecture and System Model

The EC architecture considered in this paper is illustrated in Fig. 1. The EC architecture includes cloud, edge and end. The cloud side is composed of traditional cloud computing center. The edge side is composed of the ER and edge server. The end includes but is not limited to loads, power state sensor and various electrical equipment. As the core component of information and energy management device in EI, the ER acts as the edge controller in this architecture. The task request of the end device at a certain time is sent to the ER where it is located first, and then the ER divides each task into two parts, as shown in Fig. 2. In Fig. 2, dashed lines and solid lines represent energy flow and information flow, respectively. Considering the limited computing power of the edge side, the ER offloads one part of the tasks to the cloud for computing, and the other part to the edge side closer to the end device.

Let us denote $\mathcal{J}$ as the set of ERs, where $\mathcal{J} = \{1, 2, \ldots, J\}$. There are $\mathcal{I}_j$ end devices, which are connected with the j-th ER. In this paper, the task, which is initiated by the i-th end device connected with the j-th ER, can be denoted by a vector $W_{j,i} = [T_{j.i}, E_{j,i}]$, where $T_{j.i}$ represents the data-size (in bits) to be transmitted, and $E_{j,i}$ represents the number of CPU cycles needs to be performed.

B. Delay Modeling

In this paper, we consider four kinds of delay as illustrated in the following.

1) Transfer delay of information from the end device to the ER: As shown in Fig. 2, the computation tasks, such as load forecasting, fault detection based on artificial intelligence algorithm and wind power forecasting, have very high requirements for hardware computing resources, so they need to be offloaded to the cloud server or edge server for computation. In the architecture proposed in this paper, all tasks are first transmitted to the ER. The transmission delay from the i-th end device to the j-th ER is denoted as $d_{j,i}^{tran,ee}$. According to [25], the transmission delay can be modeled as follows.

$$d_{j,i}^{tran,ee} = \frac{r_{j,i}}{c}, \tag{1}$$

where $r_{j,i}$ is channel distance between the i-th end device and the j-th ER, and c is the travel speed.

2) Computation delay of task at the edge server: As mentioned before, the computation of these tasks requires high computing resources and is time-consuming, so we also consider the computation delay at the edge server. As shown in Fig. 2, the task transferred from the end device to the ER is divided into two parts. One part is executed by local edge server, and the other part is executed by cloud server. Let us denote $\lambda_{j,i}$ as the task splitting ratio which represents the task proportion executed at the ER. Thus, $1 - \lambda_{j,i}$ represents the task proportion executed at the cloud. According to [27], [28], the computation delay of task at edge server can be modeled as follows.

$$d_{j,i}^{com,e} = \frac{\lambda_{j,i} E_{j,i}}{f_{j,i}^{e}}, \tag{2}$$

where $f_{j,i}^{e}$ represents the computation resource that the j-th ER allocates to the i-th end device.

3) Transfer delay of information from the ER to the cloud: In this paper, all ERs are assumed to be connected with the cloud through different backhaul links. Although backhaul link has high bandwidth, due to the big amount of transmission data, the transmission delay of this part also needs to be considered [27]. As shown in Fig. 2, after a task is divided into two parts, one of these two parts needs to be transferred from ER to the cloud. According to [27], the transmission delay from the ER to the cloud can be modeled as:

$$d_{j,i}^{tran,ec} = \frac{(1 - \lambda_{j,i}) T_{j,i}}{W_j}, \tag{3}$$

where W_j is denoted as the backhaul communication capacity for the i-th end device connetcted with the j-th ER.

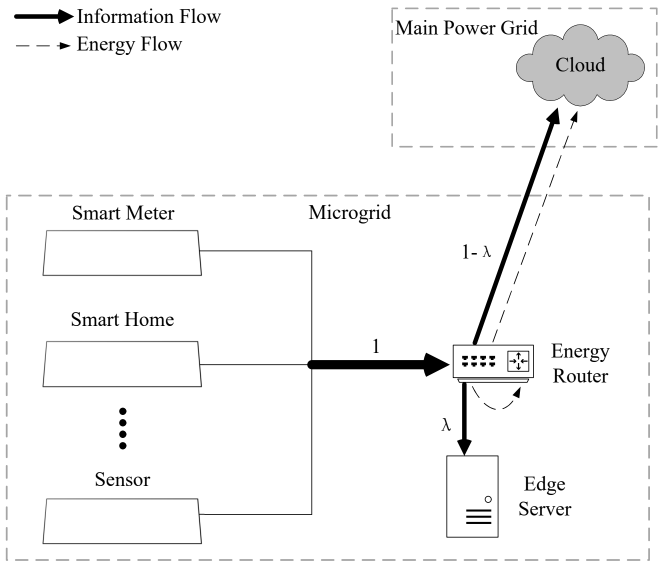

Fig. 2. The procedures for processing and transferring energy and information.

4) Computation delay of the task at the cloud: As mentioned before, the computation of these tasks requires high computing resources and is time-consuming, so the computation delay at the cloud server also needs to be considered. As shown in Fig. 2, when transmitted to the cloud, such part is performed by cloud server. The number of CPU cycles required to perform each part of task can be modeled as $(1 - \lambda_{j,i})E_{j,i}$. According to [27], the computation delay of the cloud can be modeled as:

$$d_{j,i}^{com,c} = \frac{(1 - \lambda_{j,i}) E_{j,i}}{f_{j,i}^{c}}, \tag{4}$$

where $f_{j,i}^{c}$ represents the computation resource that the cloud allocates to this task which is offloaded to the j-th ER from the i-th end device.

C. Reliability Modeling

Ensuring the reliability of information communication is the key to maintaining the security of energy production, transfer and other processes in EI. Once there is a fault in the process of information transmission, it could cause the energy layer to lose the control of information, thus increasing the instability of this system. Therefore, reliability is an important index that must be paid attention to. In order to express the reliability of information communication, the reference [29] adopts the error probability of communication. According to [29], the error probability of each communication link can be modeled as follows.

$$e_j = 1 - (1 - \eta)^{\theta}, \tag{5}$$

where η represents the target block error rate, and it is necessary that the block error rate is less than 10^{-7} [30]. In (5), θ represents the ratio of the size of communication data to transport block. The definition of transport block can be found in, e.g., [31].

III. PROBLEM FORMULATION

In this paper, a task allocation strategy based on cloud-edge collaboration is proposed to improve the reliability and efficiency of the information layer in EI. For improving the the efficiency of the information transmission and computation, we focus on minimizing the delay of information transmission and computation.

We assume that the transmission and computation of tasks in the ER are performed simultaneously, so we take the maximum of $d_{j,i}^{com,e}$ and $d_{j,i}^{tran,ec}+d_{j,i}^{com,c}$ as the overall delay of the i-th end device connected to the j-th ER [27]. Let us denote the overall delay of the i-th end device connected to the j-th ER as $d_{j,i}^{ec}$ which can be expressed as

$$d_{j,i}^{ec} = \max\left\{d_{j,i}^{com,e}, d_{j,i}^{tran,ec} + d_{j,i}^{com,c}\right\}. \tag{6}$$

Then, since each task needs to be transferred from end device to ER first, the overall delay of the task can be described as

$$d_{j,i}^{sum} = d_{j,i}^{tran,ee} + d_{j,i}^{ec}. \tag{7}$$

When the reliability of information transmission is not satisfactory, the number of tasks transferred from ER to the cloud in EI should be reduced accordingly, in order to ensure the safety of EI system. In this paper, the amount of tasks transferred to the cloud is adjusted by changing the value of $\lambda_{j,i}$. Therefore, the $r_{j,i}$, which is

$$r_{j,i} = e_j(1-\lambda_{j,i}), \tag{8}$$

should be minimized.

With the increase of communication error probability, the amount of tasks transferred to the cloud server should be decreased. Therefore, considering the reliability of information, the optimization objective is formulated in (9).

$$\min_{\left\{\lambda_{j,i}, f_{j,i}^{c}, f_{j,i}^{e}\right\}} \sum_{j=1}^{J}\sum_{i=1}^{I_j} d_{j,i}^{sum} + k r_{j,i}, \tag{9}$$

$$\text{subject to} \quad \sum_{j=1}^{J}\sum_{i=1}^{I_j} f_{j,i}^{c} \le F^c,\ f_{j,i}^{c} \ge 0, \tag{10}$$

$$\sum_{i=1}^{I_j} f_{j,i}^{e} \le F_j^e,\ f_{j,i}^{e} \ge 0, \forall j \in J, \tag{11}$$

$$0 \le \lambda_{j,i} \le 1, \forall i \in I_j, \forall j \in J, \tag{12}$$

where (10) and (11) mean that the total computing resources of the cloud and ER are limited, and the upper limits are F^c and F_j^e, respectively. The task splitting radio $\lambda_{j,i}$ meets the constraint (12).

IV. PROBLEM SOLUTION

According to the results in [27], we simplify this complex optimization problem into a form that can be solved by particle swarm optimization algorithm. The optimal task splitting strategy $\lambda_{j,i}$ is derived by the method in Appendix C of [27]. In this way, the optimal task splitting strategy $\lambda_{j,i}$ can be expressed as:

$$\lambda_{j,i}^{*} = \frac{T_{j,i} f_{j,i}^{e} f_{j,i}^{c} + E_{j,i} W_j f_{j,i}^{e}}{E_{j,i} W_j (f_{j,i}^{e} + f_{j,i}^{c}) + T_{j,i} f_{j,i}^{e} f_{j,i}^{c}}. \tag{13}$$

By substituting (13) into (9), the splitting ratio $\lambda_{j,i}$ can be eliminated. Then, the objective function can be solved by the particle swarm optimization algorithm. The detailed process of the particle swarm optimization algorithm is as follows:

1. Initialize the particle swarm, including maximum iterations, size of particle swarm, initial velocity and position of each particle.

2. Calculate fitness values for each particle.

3. Calculate the best position $pbest_i$ and the best global position $gbest_i$.

4. Update the speed and position according to the following equations:

$$\begin{aligned} v_{i+1} =& w \cdot v_i + c_2 \cdot rand_1 \cdot (pbest_i - x_i) \\ &+ c_2 \cdot rand_2 \cdot (gbest_i - x_i), \end{aligned} \tag{14}$$

$$x_{i+1} = x_i + v_{i+1}. \tag{15}$$

5. Check whether the number of iterations reaches the maximum value. If so, stop the iteration; otherwise return to Step 2.

V. NUMERICAL SIMULATION

To verify the performance of the proposed method, numerical simulation is provided in this section. The detailed parameters are listed in Table I.

TABLE I
PARAMETERS FOR SIMULATION

Parameter	Value
$E_{j,i}$	$[500, 700]$ M Cycles
F_j^e	10000 cycles/s
F^c	50000 cycles/s
c	$[100, 1000]$ km/s
k	0.01
e_j	0.1
$T_{j,i}$	$[0.01, 0.03]$ Mbits
W_j	$[50, 100]$ Mbps
$r_{j,i}$	$[50, 500]$ m

The comparison of performance between cloud-edge scheme and cloud-only scheme is described in Fig. 3. The cloud-only scheme means all tasks must be transfered to the cloud for computation. In this simulation, it is assumed that each ER connects 5 end devices. In this figure, it is obvious that the cloud-edge scheme is better than the cloud-only scheme. With the increase of the number of MGs, the growth rate of the average system delay with cloud-edge scheme is slowing down, while the growth rate of the average system delay with cloud-only scheme is almost unchanged.

The effect of the capacity of edge server on the optimal splitting ratio is described in Fig. 4. In this simulation, it is

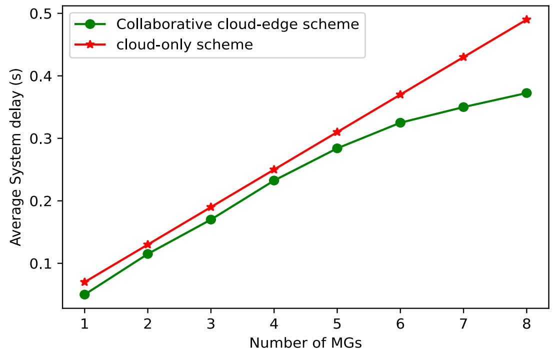

Fig. 3. Average delay affected by the number of edge nodes.

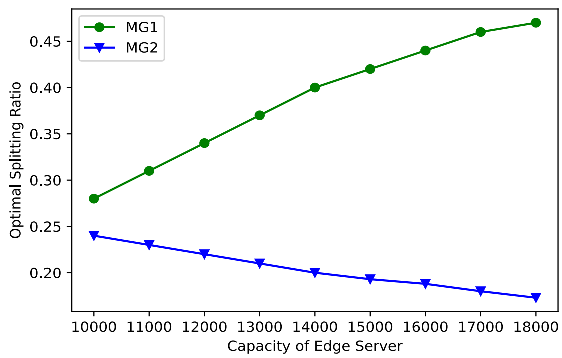

Fig. 4. The optimal splitting ratio affected by the capacity of edge server.

assumed that the capacity of edge server in MG2 is 7000 cycles per second, and each ER is connected with 4 end devices. It can be seen that when the capacity of edge server in MG1 increases, the optimal splitting ratio of MG1 increases, while that of MG2 decreases. This is because when the capacity of edge server in MG1 increases, the ER of MG1 leaves more tasks to be processed locally. Meanwhile, the

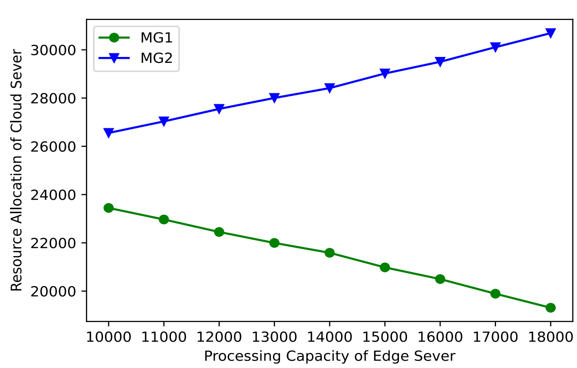

Fig. 5. The resource allocation of cloud affected by the capacity of edge server.

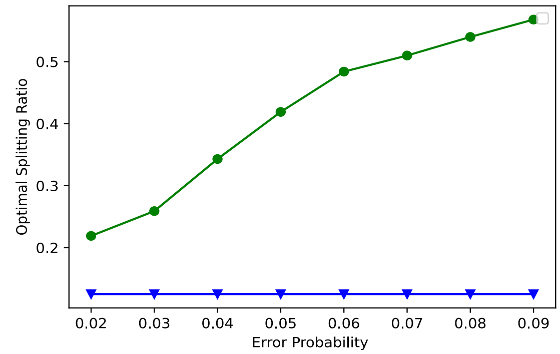

Fig. 6. The optimal splitting ratio affected by the error probability of communication.

computing burden in the cloud is less, so the ER of MG2 transfers more tasks to the cloud for computation. Accordingly, when the number of tasks transfered by an MG is reduced, the computing resources allocated to the MG by the cloud is also reduced. The effect of the capacity of edge server on the resource allocation of cloud is described in Fig. 5.

Once there is a fault in the process of information transmission, it could cause the energy layer to lose the protection from the information layer, thus increasing the instability of EI system. In order to reduce the effect caused by the fault in the process of information transmission, when the fault rate becomes relatively high, the amount of long-distance information transmission should be reduced. As shown in Fig. 6, the optimal splitting ratio increases with the increase of the error probability. In this simulation, the value of parameter k is set to be 5. The blue line represents the optimal splitting ratio when the error probability is zero.

Based on the simulation results above, the feasibility of the proposed method is evaluated. The simulation results demonstrate the cloud-edge collaboration method proposed in this paper is is better than the cloud-only method in reducing the delay of information.

VI. Conclusion

In this paper, the delay and reliability of information in EI is investigated. In order to ensure the reliability of information transmission and reduce information delay, a task splitting strategy based on cloud-edge collaboration is proposed. Then, a joint optimization problem is formulated to reduce the delay of information and to ensure the reliability of information transmission. The effectiveness and feasibility of the proposed method are successfully verified by numerical simulations.

Since the energy flow in EI has not been fully considered in this work, our future research shall focus on the integration of information flow and energy flow in EI scenarios.

Acknowledgment

This work was funded in part by the National Key Research and Development Program of China (Grant No.

2017YFE0132100), Tsinghua-Toyota Joint Research Institute Cross-discipline Program, and the BNRist Program (Grant No. BNR2020TD01009).

References

[1] K. Wang, J. Yu, Y. Yu, Y. Qian, D. Zeng, S. Guo, Y. Xiang, J. Wu, "A Survey on energy Internet: architecture, approach, and emerging technologies," IEEE Systems Journal, vol. 12, no. 3, pp. 2403-2416, Sept. 2018.

[2] H. Hua, Y. Qin, C. Hao, and J. Cao, "Optimal energy management strategies for energy Internet via deep reinforcement learning approach," Applied Energy, vol. 239, pp. 598-609, Apr. 2019.

[3] L. Cheng, T. Yu, H. Jiang, S. Shi, Z. Tan and Z. Zhang, "Energy Internet access equipment integrating cyber-physical systems: Concepts, key technologies, system development, and application prospects," IEEE Access, vol. 7, pp. 23127-23148, 2019.

[4] H. Hua, J. Cao, G. Yang, and G. Ren, "Voltage control for uncertain stochastic nonlinear system with application to energy Internet: non-fragile robust H_∞ approach," Journal of Mathematical Analysis and Applications, vol. 463, no. 1, pp. 93-110, 2018.

[5] Y. Qin, H. Hua, and J. Cao, "Energy Internet system control and optimization: A stochastic risk-sensitive control approach," in Proc. 15th IEEE International Conference on Control & Automation, Edinburgh, UK, July. 2019, pp. 79-84.

[6] H. Hua, Y. Qin, C. Hao, and J. Cao, "Stochastic optimal control for energy Internet: A bottom-up energy management approach," IEEE Transactions on Industrial Informatics, vol. 15, no. 3, pp. 1788-1797, Mar. 2019.

[7] C. Bai, L. Zhang, R. Li, Y. Jiao, Y. Zhang and X. Han, "The construction of user centric service system of energy internet," in Proc. 2018 IEEE International Conference on Energy Internet (ICEI), Beijing, 2018, pp. 7-12.

[8] M. Ye, "Distributed nash equilibrium seeking for games in systems with bounded control inputs," IEEE Transactions on Automatic Control, accepted, DOI: 10.1109/TAC.2020.3027795.

[9] M. Ye, G. Hu, "Game design and analysis for price based demand response: an aggregate game approach," IEEE Transactions on Cybernetics, 47 (3), 720-730, 2017.

[10] H. Hua, Y. Qin, H. Xu, C. Hao, and J. Cao, "Robust control method for DC microgrids and energy routers to improve voltage stability in energy Internet," Energies, vol. 12, no. 9, Art. no. 1622, 2019.

[11] W. Li, T. Yang, F. C. Delicato, P. F. Pires, Z. Tari, S. U. Khan, A. Y. Zomaya, "On enabling sustainable edge computing with renewable energy resources," IEEE Commun. Mag., vol. 56, no. 5, pp. 94–101, May 2018.

[12] H. Hua, Y. Qin, J. Geng, C. Hao, and J. Cao, "Robust mixed H_2/H_∞ controller design for energy routers in energy Internet," Energies, vol. 12, no. 3, Art. no. 340, 2019.

[13] M. H. Yaghmaee Moghaddam, and A. Leon-Garcia, "A fog-based internet of energy architecture for transactive energy management systems," IEEE Internet Things J., vol. 5, no. 2, pp. 1055–1069, Apr. 2018.

[14] J. Wang, K. Meng, J. Cao, Z. Cheng, L. Guo, C. Lin, "Information technology for energy Internet: A survey," Journal of Computer Research & Development, vol. 52, no. 3, pp. 1-18, 2015.

[15] S. Tan, C. Jiang and Y. He, "A research on information value in energy internet based on risk theory," in Proc. 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, 2017, pp. 1-6.

[16] A. C. Swastika, R. Pramudita, and R. Hakimi, "IoT-based smart grid system design for smart home," in Proc. ICWT, Jul. 2017, pp. 49–53.

[17] S. Chen ,H. Wen, J. Wu, W. Lei, W. Hou, W. Liu, A. Xu, Y. Jiang, "Internet of Things based smart grids supported by intelligent edge computing," IEEE Access, vol. 7, pp. 74089-74102, 2019.

[18] M. Guo, L. Li and Q. Guan, "Energy-efficient and delay-guaranteed workload allocation in IoT-edge-cloud computing systems," IEEE Access, vol. 7, pp. 78685-78697, 2019.

[19] J. Cao, Y. Wan, H. Hua, and Y. Qin, "Delay analysis for end-to-end synchronous communication in monitoring systems," Sensors, vol. 18, no. 11, 3615, 2018.

[20] L. Huang, Y. Liu, J. Wu, G. Li, and J. Li, "Software-defined dynamic QoS provisioning for smart metering in energy Internet using fog computing and network calculus," IET Cyber-Physical Systems: Theory & Applications, vol. 3, no. 3, pp. 142–149, Sep. 2018.

[21] W. Shi and S. Dustdar, "The promise of edge computing," Computer, vol. 49, no. 5, pp. 78–81, May 2016.

[22] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, X. Yang, "A survey on the edge computing for the Internet of Things," IEEE Access, vol. 6, pp. 6900-6919, 2018.

[23] Y. Qin, H. Hua, and J. Cao, "Short-term energy cache regulation for energy router: A robust H-infinity approach," in Proc. 3rd IEEE International Conference on Energy Internet, Nanjing, China, May 2019, pp. 161-166.

[24] S. Bitam, S. Zeadally, and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm," Enterprise Information Systems, vol. 12, pp. 1–25, 2017.

[25] X. Niu, S. Shao, C. Xin, J. Zhou, S. Guo, X. Chen, F. Qi, "Workload allocation mechanism for minimum service delay in edge computing-based power Internet of Things," IEEE Access, vol. 7, pp. 83771–83784, 2019.

[26] Y. Nan, W. Li, W. Bao, F. C. Delicato , P. F. Pires, Y. Dou, A. Y. Zomaya, "Adaptive energy-aware computation offloading for cloud of things systems," IEEE Access, vol. 5, pp. 23947–23957, 2017.

[27] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," IEEE Trans. Veh. Technol., vol. 68, no. 5, pp. 5031–5044, May 2019.

[28] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," IEEE J. Select. Areas Commun., vol. 34, no. 5, pp. 1757-1771, May 2016.

[29] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," IEEE Access, vol. 6, pp. 12825–12837, 2018.

[30] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile internet," IEEE J. Select. Areas Commun., vol. 34, no. 3, pp. 460–473, Mar. 2016.

[31] M. Mezzavilla, M. Miozzo, M. Rossi, N. Baldo, and M. Zorzi, "A lightweight and accurate link abstraction model for the simulation of lte networks in ns-3," in Proc. 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2012, pp. 55–60.