Service Popularity-Based Smart Resources Partitioning for Fog Computing-Enabled Industrial Internet of Things

Gaolei Li, *Student Member, IEEE*, Jun Wu, *Member, IEEE*, Jianhua Li, Kuan Wang, and Tianpeng Ye

***Abstract*—Recently, fog computing has gained increasing attention in processing the computing tasks of the industrial Internet of things (IIoT) with different service popularity. In task-diversified fog computing-enabled IIoT (F-IIoT), the mismatch between expected computing efficiency and partitioned resources on fog nodes (FNs) may pose serious traffic congestion even large-scale industrial service interruptions. The existing works mainly studied offloading which type of computing tasks into FNs, but few studies enabled smart resource partitioning of FNs. In this paper, a service popularity-based smart resources partitioning (SPSRP) scheme is proposed for fog computing-enabled IIoT. We first exploit Zipf's law to model the relationship between popularity ranks and computing costs of IIoT services. Moreover, we propose an implementation architecture of the SPSRP scheme for F-IIoT, which decouples the computing control layer from data processing layer of IIoT through a specified SPSRP controller. Besides, a mobility and heterogeneity-aware partitioning algorithm is presented for extending SPSRP scheme to seamlessly support cross-domain resources partitioning. The simulations demonstrate that the SPSRP scheme can bring notable performance improvements on delay time, successful response rate and fault tolerance for fog computing to deal with the large-scale IIoT services.**

***Index Terms*—Fog computing, Industrial Internet of Thing (IIoT), resources partitioning, service popularity, Zipf's law.**

I. Introduction

ARCHITECTURE of Industrial Internet of things (IIoT) is evolving from centralized cloud to distributed fog [1]. Features of fog computing on low-latency and context awareness will be beneficial to energy efficiency, heterogeneous access, and privacy-preservation in the future IIoT systems [2]. Under the guidance of fog computing, it has became a novel trend to aggregate computation resources at network edges to deal with the large-scale computation tasks [3], [4].

The IIoT has been perceived as a vast and dynamic industry territory. Since the connected industrial things increase, IIoT services urge for more and more computation and communication resources, leading to the bottleneck in terms of data latency and traffic overhead. Different from the existing cloud-based IIoT, fog computing advocates to select popular IIoT services to support large-scale IIoT applications. While fog computing provides account of advantages for IIoT, this geo-distributed computing architecture also brings several challenges [5]. First, due to the decentralization nature of fog computing, there is no state observer to globally monitor and control the computing states of each fog-enabled IIoT service. Thus, fog computing urgently needs to be equipped with a novel function that can automatically partition computation resources for popular IIoT services. Second, the mismatch between the expected computing efficiency and the partitioned resources on fog nodes (FNs) may pose serious traffic congestion even large-scale IIoT service interruptions. Third, the heterogeneity of computation resources in industrial FNs makes it more difficult for FNs to efficiently partition their resources.

In response to these challenges, we proposed a service popularity-based smart resources partitioning (SPSRP) scheme for future fog computing-enabled IIoT. The SPSRP scheme made it possible for each FN to actively partition its resources in real-time based on the popularity of IIoT services. As an example, it is common to see that there are many machines in newspaper office for printing books and the bestsellers should be printed as soon/many as possible to meet purchasers requirements and reduce the handover cost of machines. However, the machines cannot acknowledge which book is popular without observing the markets. Proposed approach seems like to add a function into the newspaper office to recognize which book is bestseller and then decide how many machines should be used to print bestseller. Correspondingly, in a decentralized IIoT environment, we must decide whether to forward the computing tasks for popular IIoT services to other FNs or locally handled. The contributions of this paper are summarized as follows.

Manuscript received May 20, 2018; accepted June 4, 2018. Date of publication June 11, 2018; date of current version October 3, 2018. This work was supported by the National Natural Science Foundation of China Under Grant 61431008 and Grant 61571300. Paper no. TII-18-1260. *(Corresponding author: Jun Wu.)*

G. Li and K. Wang are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: gaolei_li@sjtu.edu.cn; wangkuanfyfy@sjtu.edu.cn).

J. Wu, J. Li, and T. Ye are with the School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200240, China, and also with Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai 200240, China (e-mail: junwuhn@sjtu.edu.cn; lijh888@sjtu.edu.cn; ytp_SJTU_2015@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2845844

1) We formulated a mathematical model to model the relationship between service popularity and computing cost, and investigated the approach of resources partitioning in fog computing-enabled IIoT system. We provided the system model of the SPSRP scheme and provided utility function formulations for F-IIoT to solve the joint optimization problem of resources partitioning.
2) We proposed an SPSRP scheme for fog computing-enabled IIoT. We exploited the Zipf's law to calculate the popularity rank of the IIoT service and predicted the computing cost of arriving IIoT services on FNs. We provided a solving method of threshold value for forwarding IIoT services, and applied it to decide whether the arriving IIoT service should be locally handled.
3) We proposed the implementation architecture of the SPSRP scheme, which first decoupled the computing control layer from the computing layer, and provided a programmable interface for IIoT operators. With service popularity and SPSRP controller, we seamlessly associated the communication performances with computing performance, and provided a generalized platform for FNs to uniformly partition the heterogeneous and mobile computation resources. Additionally, we developed several comparison experiments to validate the efficiency and scalability of the proposed scheme.

The remainder of this paper is structured as follows. Section II gives an overview of the related work and the strengths of the SPSRP scheme. Section III describes the system model of the SPSRP scheme. Section IV introduces the basic implementation architecture of the SPSRP scheme and discusses its design principles in detail. Simulations are given in Sections V to evaluate the performance of the SPSRP scheme. Finally, Section VI draws the conclusion and gives the future work.

II. Related Work

The study on introducing fog computing into IIoT is still in the initial stage. In order to satisfy the secure automation control requirements of IIoT, exploiting fog computing to achieve adaptive operations platform has been perceived as a promising approach, which enabled high manageability of IIoT [6]. Different from the cloud-based IIoT that aggregates all IIoT data into a remote data center, fog computing provides a more efficient and scalable platform that enables context-awareness, low latency, energy efficiency, and big data analytics [7], [8].

Resource partitioning is a hot topic in wireless communication filed [9]–[11]. These studies always focused on the management of radio and frequency resources in femtocell and small cells. Due to the heterogeneity nature of wireless communication, the most popular approach for radio resource partitioning was frequency reuse. Singh and Andrews [12] provided an joint analytical framework for users offloading and resource partitioning in co-channel heterogeneous networks, this paper first studied the association between the number of users offloaded into edge elements (e.g., small cells) and performance of resource partitioning. Recently, another study [13] exploited the Stackelberg game model to cooperatively optimize the resource partitioning and data offloading in co-channel two-tier heterogeneous networks. However, different from resource partitioning in wireless communication, FNs in IIoT pay more attention on recognizing which is the most popular delay-sensitive services regardless of the data scale or user number in a domain so that the existing studies on resource partitioning in wireless communication cannot be applied in fog-enabled IIoT.

Moreover, the existing resources partitioning approaches designed for cloud-enabled IIoT also cannot be applied to fog-enabled IIoT directly. For example, Mach and Becvar [14] formulated a load balancing problem between multiple fog servers as the cooperative resource sharing. However, the existing load balancing scheme required all data traffic to pass through an additional load balancer. To improve the efficiency of big data analysis, the literature [15] proposed a computation partitioning model for mobile cloud computing. However, this method only can improve the data processing efficiency in data center, but not adapt to fog computing paradigm due to the decentralization nature of fog computing [16].

Expect for the studies on flows shunting in IIoT, some early proposals in [17] and [18] also tried to develop the autonomous resources allocation platforms for IoT to reduce the service response time under the fog environment. Recently, advocating the underlying IIoT infrastructures to share their resources was also very insightful [19], [20]. However, it was not easy to observe the computing states of all heterogeneous edge devices in real-time [21]. Moreover, this kind of proposals did not give a deeper discussion about how to partition the aggregated computation resources on FNs.

By modeling the relationship between service popularity and computing cost, this paper proposed an SPSRP scheme for F-IIoT. Strengths of the SPSRP scheme mainly contained the following three aspects. First, the SPSRP scheme seamlessly associated the communication performances with computing quality to improve the resources utilization as well as the service response time in F-IIoT. Second, heterogeneous computing resources were uniformly scheduled for processing IIoT services. By using SPSRP scheme, the aggregated computation resources can be smartly partitioned according to popularity ranks of IIoT services. Third, the computing control layer was decoupled from the computing layer and a programmable resources partitioning interface was provided, this novel architecture facilitated F-IIoT to be more scalable to embrace industrial situation awareness.

III. System Model

We consider the system model of the SPSRP scheme for F-IIoT with one cloud and several geo-distributed fog servers as illustrated in Fig. 1. The cloud and geo-distributed fog servers in the system model of the SPSRP scheme are deployed as a hierarchical framework. Each of upstream FN has responsibility to supervise the status of local FNs.

The SPSRP has three key components.

1) *Global fog identifier (GFID):* The SPSRP exploits the GFID to name each of FN. The separation between the GFID and the global service identifier (GSID) provides support for global observability.
2) *SPSRP controller:* The SPARP controller is utilized to monitor and control the computing states of all FNs in

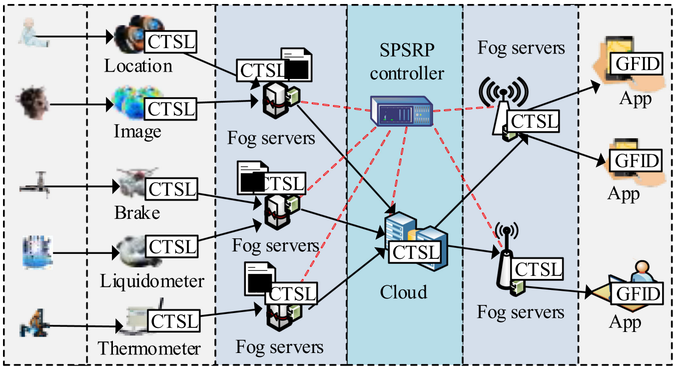


Fig. 1. System model of the proposed SPSRP scheme.

F-IIoT. The SPSRP controller maintains the mapping from GFID to GSID. If a fog node moves from GSID to another location renamed by GSID or the computing resource of FN is exhausted, service providers can redirect the service requests to a new address to find computing resources without any service interrupts.

3) *Computing task stream list (CTSL):* To realize automatic resources partitioning, the CTSL is presented, which includes three basic tuples: $MatchField$, $ActionField$, and $Counter$. The parameters in each tuple can be precustomized by system designers of F-IIoT.

Since the cloud usually aggregates large-scale computing, storage, and network resources, an SPSRP controller can be implemented in cloud to monitor the global states of geo-distributed entities, end-to-end QoS services, and computing services in fog servers. Also, IIoT users can optimize the resources utilization, QoS control, and computing efficiency by adding their own resources scheduling algorithms into SPSRP controller. The states of each underlying infrastructure (e.g., GPS, camera, liquidometer, and thermometer) are identified by geo-distributed FNs. We emphasize that all the edge devices are equipped with SDN protocols [22], which can provide programmable QoS control, e.g., the OpenFlow-enabled virtual switch is deployed on each fog server. Especially, all of the computing tasks on different edge devices are labeled as record items and added into the defined CTSL. Besides, the simple network management protocol (SNMP) acting as IEEE P21451 standard [23] will be an important network management protocol for IIoT applications. The SNMP also can be exploited to achieve low-cost end-to-end QoS monitoring without SDN by aggregating energy status, disk size, memory, running process of each IIoT devices in real-time. In each IIoT device, it might have a management information base to cache the above information and a client to upload these information to state observer. For each client, it was configured on only one network location. When the state observer required to schedule the QoS, the state observer acting as an server obtained the above information from distributed IIoT devices and FNs by publish "get" messages.

The definition of "smart" in this article is that each fog sever can actively partition its resources in real-time based on the popularity rankings of IIoT services. This "smart" provides a guarantee for the stability of F-IIoT, by reducing the mismatch between expected computing efficiency and partitioned resources. Also, this "smart" improves the utilization of edge resources, while it reduces the service delay and disconnection rate in F-IIoT. Besides, this "smart" makes it possible for industrial operators to enforce efficient big data analysis and machine learning. Therefore, a smart resources partitioning mechanism can intelligently partition the resources of fog servers according to the popularity rankings of IIoT services. The SPSRP makes fog servers pick up and process the sensed data of local IIoT services automatically rather than shunt all the data to idle virtual machines in cloud. To achieve this smart SPSRP, we exploit the Zipf's law to calculate the popularity rank of each IIoT service on each FN, propose Algorithm 1 to partition computation resources of each FN, and Algorithm 2 to deal with mobility and heterogeneity of computation resources. By combining Zipf's law, Algorithms 1 and 2, fog-enabled IIoT can smartly partition mobile and heterogeneous computation resources.

A. IIoT Services Popularity Model

Consider there are many different types of IIoT services to be processed in IIoT. An IIoT service is denoted as $E = \{\alpha_{\text{type}}, \beta_{\text{task}}, \gamma_{\text{SLA}}\}$, where $\alpha_{\text{type}}, \beta_{\text{task}}, \gamma_{\text{CQC}}$ denote application type, computing task, and computing quality contract (CQC), respectively. It is common to see that an FN simultaneously serves for multiple IIoT service sessions. Similar to the content caching problem in information-centric networking (ICN), the IIoT service E on the ith FN is modeled through a generalized Zipf function

$$Z_i^E(k_t) = \frac{\Omega}{k_t^{\gamma}}, \qquad k_t = 1, 2, \ldots, K \tag{1}$$

$$k_{t+\Delta t} = Z_i^{E^{-1}}(Z_i^E(k_t) + \lambda_{\Delta t}) \tag{2}$$

where $\Omega = (\sum_{k=1}^{K} \frac{1}{k^{\gamma}})^{-1}$ and $0 \leq \gamma \leq 1$ is the exponent and k_t denotes the popularity ranking of IIoT service E on the ith FN at time t. λ is the number of arrival E type of IIoT services on FN ith during Δt spot. Also, the $Z_i^{E^{-1}}(*)$ is the inverse function of $Z_i^E(*)$.

Originally, Zipf's law was found by observing and analyzing the word frequency distribution. About twenty years ago, the distribution of many Internet services was proven to follow Zipf's law and many existing web caching strategies used Zipf's law to model Internet users' service requests [24], [25]. Recently, popularity-based smart caching for ICN has utilized Zipf' law to model the content distribution [26]. Now, Zipf's law is being applied in many fields such as linguistics, geography, economics, and broadcast TV [27]. Similar to Internet services, the distribution of IIoT services also follows Zipf's law [28]. This paper exploits Zipf's law to predict the computing cost of IIoT services by calculating their popularity rankings. FN gets popularity rankings of IIoT services by analyzing the statistics of past and current logs in real-time.

B. Computing Cost Model

To improve the resources utilization and computing quality, FNs are more willing to locally process popular IIoT services

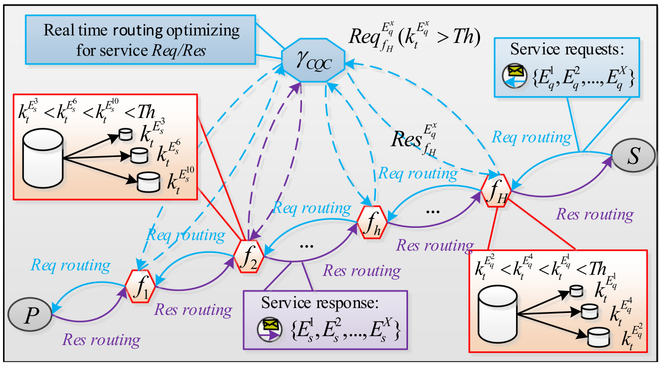


Fig. 2. On-path partitioning for end-to-end IIoT services.

and work with fewer remote control operations (e.g., wake up, sleep, and migration).

Definition 1: For multiple types of IIoT services at time t, the computing cost for one example IIoT service on FN is defined as a the following function:

$$C_i^{E_j} = \frac{C_i^{E_0}}{Z_i^{E_j}(k_t)} \tag{3}$$

where $C_i^{E_0}$ is the fixed original computing cost on FN jth when $Z_i^{E_j}(k_t) = 1$. By combining (1)–(3), the relationship between computing cost and service popularity is a convex function when $\gamma < 1$, while the relationship between computing cost and service popularity is a concave function when $\gamma > 1$. Moreover, for a fixed ΔR, ΔC_2 is larger than ΔC_1 and $\Delta C_2^{'}$ is smaller than $\Delta C_1^{'}$. In the other word, for $\gamma < 1$, the change of service popularity when $k_t < 7$ has a greater impact on the computing cost than the change of service popularity when $k_t > 16$. For $\gamma > 1$, the change of service popularity has a greater impact on the computing cost when $k_t > 30$ than when $k_t < 20$. In this paper, the SPSRP shifts the less popular services on ith FNs into the other FNs to minimize their computing costs under $\gamma > 1$.

C. Utility Function Formulation

In a complex IIoT system, there are many pairs of service provider and service subscriber. Also, there are many multiple routing paths for forwarding data flows of IIoT services from service provider to service subscriber. Therefore, to clarify the utilization of the SPSRP scheme, we formulate two different utility functions. One is on-path partitioning for end-to-end IIoT services; another one is generalization to large-scale IIoT services.

1) On-Path Partitioning for End-to-End IIoT Services: As illustrated in Fig. 2, we consider an end-to-end IIoT service with one type of providers (P) and one type of subscribers (S). The number of P and S is generated randomly. On the service routing path from P to S, there are H FNs. To clarify the strengths of the SPSRP scheme, we introduce the working flow of on-path partitioning for end-to-end IIoT services step by step.

1) At time t, S broadcast a group of service requests $\{E_q^1, E_q^2, \ldots, E_q^X\}$ to f_H in the network.
2) When the f_H receives these service requests, it will calculate the popularity rankings $\{k_t^{E_q^1}, k_t^{E_q^2}, \ldots, k_t^{E_q^X}\}$ of each kind of service request.
3) f_H begins to traverse all of the popularity rankings from $x = 1$ to $x = X$. If $k_t^{E_q^x} \leq Th$, the corresponding IIoT services should be processed by this FN f_H. If $k_t^{E_q^x} > Th$, the corresponding IIoT services will be forwarded to SPSRP controller for real-time routing optimization.
4) When the SPSRP controller receives an IIoT service, it will calculate an optimized routing path for this IIoT service based on the γ_{CQC} and send the next hop back to the FN f_{H-1} (it can be f_h in Fig. 2).
5) f_{H-1} will execute the same operations with f_H. The following FNs on routing path do not need to execute these operations until all IIoT services are accepted by FNs. To guarantee all IIoT services can be accepted by FNs, the mapping from $\{f_1, f_2, \ldots, f_H\}$ to $\{E_q^1, E_q^2, E_q^X\}$ should be a surjection.

Definition 2: For the case of end-to-end IIoT services, we treat $\{f_1, f_2, \ldots, f_H\}$ and $\{E_q^1, E_q^2, \ldots, E_q^X\}$ as two different sets. To guarantee all the computing tasks of each IIoT service are processed by the FNs on the service routing path, the mapping from $\{f_1, f_2, \ldots, f_H\}$ to $\{E_q^1, E_q^2, \ldots, E_q^X\}$ should be a surjection as follows:

$$\{f_1, f_2, \ldots, f_H\} \twoheadrightarrow \{E_q^1, E_q^2, \ldots, E_q^X\} \tag{4}$$

where $\twoheadrightarrow$ denotes the surjection operation. The definition of surjection operation can be found in any book about logic theory. Here, we give its mathematical description: If each possible image is mapped to by at least one argument, a function is surjective. Notationally: $\forall y \in Y, \exists x \in X$ such that $y = f(x)$. In mathematics, the surjection is one of the most important functions distinguished by the manner in which arguments (input expressions from the domain) and images (output expressions from the co-domain) are related or mapped to each other. Especially, the surjection in IIoT requires Th to be defined as a big value, while the consumers require a small Th to provide better user experience. This tradeoff problem can be formulated as a problem of multiple objective optimization as follows:

Objective 1:

$$\min \left[C_{f_h} = \sum_{j=1}^{Th} \frac{C_{f_h}^{E_0}}{Z_{f_h}^{E_j}\left(\text{Sort}_{x=1}^{X}\{\theta_{f_h} \times \Gamma_{f_h}^{E_x}\}\right)} \right. \tag{5}$$

$$\left. + \sum_{j=Th+1}^{X} C_{-f_h}^{E_j} \right] \tag{6}$$

$$\text{S.t.} \quad \sum_{j=1}^{Th} C_{f_h}^{E_j} \leq TR_{f_h} \tag{7}$$

$$\theta_{f_h} = 1 - \eth_{f_h} \times C_{f_h} + \eth_{-f_h} \times C_{-f_h} \tag{8}$$

where Sort$\{\bullet\}$ is an ascending ranking function, θ_{f_h} is the total demands for computing on f_h, and $\Gamma_{f_h}^{E_j}$ is the fraction of f_h's demands for IIoT service E_j that will be served by f_h's computation resources. Therein, $\eth_{f_h}$ is the reflective coefficients of the computing costs' influence on user demands and are positive. Also, we exploit C_{-f_h} to denote the computing costs of all FNs excluding f_h.

Specifically, there is a simple approach to calculate a suitable Th for forwarding service requests. Here, we give a case of exploiting simulate anneal arithmetic (SSA) to calculate the Th. The working flow of SSA-based Th solving algorithm is introduced step by step as follows.

1) IIoT operators specify a big Th as the initial state, initialize the objective value $C^0_{f_h}$, and the iteration number L for each threshold value.
2) Launch a circulation process to execute the step 3)–5) based on $k = 1, 2, 3, \ldots, L$.
3) Calculate (4)–(6) to get a new objective value C'_{f_h}.
4) Calculate the increment $\Delta C = C'_{f_h} - C^0_{f_h}$. If , $\Delta C < 0$ update as the minimize computing cost of FN f_h. Otherwise, update C'_{f_h} as C'_{f_h} the minimize computing cost of FN f_h according to the updating possibility $\rho = \frac{\delta C}{C^0_{f_h}}$.
5) If the stop condition is triggered (there exists multiple calculated computing cost C'_{f_h} that are not updated), break this circulation process.
6) Let $Th = Th - 0.01$, $Th > 0$, and go to step 2).

In a real F-IIoT system, the Th can be updated by IIoT operators periodically. Since the computation resources of FNs are limited and fitful, it is common for FNs to borrow some computation resources from its neighboring FNs. Frequent computation resource mitigations will result in bad computing quality γ_{CQC}. γ_{CQC} of f_h is impacted by ultimate utilization $t^U_{f_h}$, performance degradation, total active time $t^{\text{act}}_{f_h}$, and mismatch fraction for user demands. The SPSRP scheme reduces the real CQC violations in F-IIoT. The CQC violation between service provider and subscriber is formulated as follows.

Objective 2:

$$\min \left[V^{\gamma_{\text{CQC}}}_{R_{sd}} = \frac{1}{H} \sum_{h=1}^{H} \frac{TR_{f_h} - \sum_{j=1}^{Th^*} C^{E_j}_{f_h}}{TR_{f_h}} \right. \tag{9}$$

$$\left. \times \frac{1}{h} \sum_{h=1}^{H} \frac{t^U_{f_h}}{t^{\text{act}}_{f_h}} \times \frac{1}{h} \sum_{h=1}^{H} \frac{\xi^{\text{mis}}_{f_h}}{\xi^{\text{dem}}_{f_h}} \right] \tag{10}$$

where ξ is the mismatch number of IIoT service on f_h, and the ξ is the total computing demands of IIoT services on f_h.

2) Generalization to Large-Scale IIoT Services: The case of on-path partitioning for end-to-end IIoT services can be extended to a generalized scenario where the subscribers and providers are connected through sophisticated service relationships. In order to select the most appropriate services in fog computing-enabled IIoT, the entities that play a vital role are as follows: Subscribers $\{S_1, S_2, \ldots, S_N\}$, providers $\{P_1, P_2, \ldots, P_N\}$, service routing path matrix $R_{N\times N}$, and a group of FNs $F = \{f_{h^*}\}, 1 < h^* < H^*$, H^* is the total number of FNs in the whole network. Therein, $R_{ij} \in \{0, 1\}$, and $R_{ij} = 1$ means that S_j is allowed to subscribe IIoT services published by P_i, while $R_{ij} = 0$ means that S_j is not allowed to subscribe IIoT services published by P_i. In addition, we define an location matrix L^R_f to identify the location of FN f (e.g., if the FN f_{h^*} is located on the service routing path R_{ij}, then $L^{R_{ij}}_{f_{h^*}} = 1$; otherwise, $L^{R_{ij}}_{f_{h^*}} = 0$). Every service routing path with $L^{R_{ij}}_{f_{h^*}} = 1$ incurs a CQC violation. Therefore, we utilize the average CQC violation for all service routing paths in the whole network as the global objective. The average CQC violation is given as follows.

Global Objective:

$$\min \left[V^{\text{avg}}_{f_{h^*}} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} V^{\gamma_{\text{CQC}}}_{R_{ij}} L^{R_{ij}}_{f_{h^*}}}{N^2} \right] \tag{11}$$

$$\text{S.t.} \sum_{j=1}^{Th^*} C^{E_j}_{f_{h^*}} \geq TR_{f_{h^*}} \tag{12}$$

$$Th^* \geq Th \tag{13}$$

where Th^* is the real threshold value of service ranking that users preconfigure on each FN.

The motivation of presenting (11)–(13) is to validate the scientificity and feasibility of this paper. By using these equations, the computing cost of each IIoT service on different FNs can be estimated based on their popularity rankings, and then, the CQC validation can be deduced. Proposed algorithms presents the working flow of the SPSRP scheme. Proposed algorithms are configured on different FNs to enforce the SPSRP scheme, but not to solve the global objective function. The implementation of the SPSRP scheme will exploit these equations and algorithms to optimize the resource partitioning strategy. In summary, the formulated equations are the guidance of proposed algorithms. Proposed algorithms cannot be used to solve the formulated equations, but can partition the resources of each FN. By using Algorithms 1 and 2, we can obtain the minimized computing cost and the minimized CQC validation.

IV. Implementation of SPSRP

In F-IIoT, each edge network device (e.g., base station, IoT gateway, and sensors) can process the sensed data and provide context-aware services for local IIoT users. Therefore, the SPSRP scheme treats all edge network devices as the underlying computing infrastructures. As illustrated in Fig. 3, the basic architecture of the SPSRP scheme decouples the computing control layer from the underlying computing infrastructures, and provides a programmable interface for operators to deploy the novel computing control strategies on distributed fog servers. The computing layer is composed of large-scale geographical FNs that invoke the computing resources according to the decisions of computing control layer. The work time of each FN can be regulated by SPSRP controller dynamically. The SPSRP controller is the key entity in computing control layer, which works on the basis of global computing states. All action sets in CTSL are generated by the SPSRP controller. These actions usually contain several

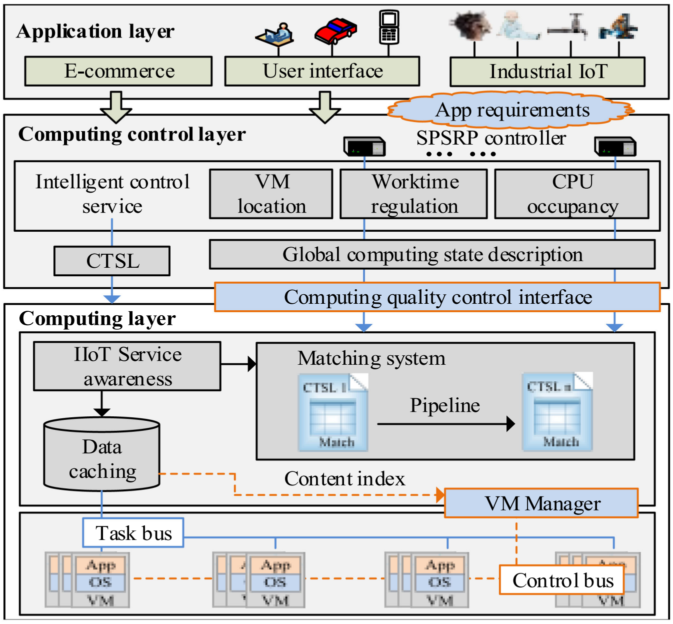


Fig. 3. Basic architecture of the SPSRP scheme.

TABLE I
KEY SYMBOLS AND EXPLAINS

Symbols	Explains
$Z_i^E(k_t)$	Popularity of IIoT service E with ranking on k_t the i_{th} fog node
$\lambda_{\Delta t}$	Number of arrival E type of IIoT service on fog node during Δ spot
E_q^x	The x_{th} request for E type of IIoT services
Th	Specified threshold to distinguish where the IIoT services are processed
$\Gamma_{f_h}^{E_j}$	The fraction of f_h's demands for IIoT service E_j that will be served by f_h's resources
C_{-f_h}	Computing costs of fog nodes excluding f_h
$\eth_{f_h}$	The reflective coefficients of the computing costs' influence on use demands
TR_{f_h}	Total computation resources on fog node f_h
$t_{f_h}^{act}$	The total active time of f_h fog node
$t_{f_h}^{U}$	The total time in which f_h fog node is experiencing ultimate utilization
$V_{R_{sd}}^{\gamma_{CQC}}$	The violation of computing quality contract
$L_{f_{h^*}}^{R_{ij}}$	Location matrix to identify whether the fog node f_{h^*} is on service routing path R_{ij}
$V_{f_{h^*}}^{avg}$	The average CQC violation for all service routing paths in the whole network
$\xi_{f_h}^{mis}$	The mismatch number of IIoT services on f_h
$\xi_{f_h}^{dem}$	Computing demands of IIoT services on f_h

instructions (such as VM creation, shutdown, and sleep) or parameters (memory size, CPU clock speed, and word size).

Different from the load balancing mechanism in cloud that focuses on traffic redirection, the SPSRP scheme makes it possible for fog servers to actively pick up the data flows that they want to process. If the arriving data flows cannot be picked up by fog server, it will be handled by the SPSRP controller. The SPSRP controller can calculate an optimized policy (e.g., computation migration and traffic forwarding) based on the whole view of all fog servers.

A. Functions of Customized FN in SPSRP Scheme

Each FN is a nano data center (nDC). The VM can be deployed on any local network element. To meet the IIoT's special demands on ultralow latency and automated control, it is imperative to reconstruct the functions of FNs in the SPSRP scheme. The functions of reconstructed FN in the SPSRP scheme are introduced as follows.

1) *IIoT service awareness*: IIoT users can deploy the advanced service awareness technologies on FNs to identify the delay-sensitive IIoT services.
2) *Matching system*: Matching system contains a set of CTSLs. One CTSL consists of GFIDs, CQC, and action sets. Each labeled IIoT service will be sent into this matching system and select a appropriate CTSL.
3) *Data caching*: To satisfy the ultralow latency requirements and save bandwidth resources, caching data at nDC is very popular and efficient.
4) *Resource manager*: In FN, the offloaded resources are maintained by resources manager. The IIoT service with higher popularity will be shifted into the higher performance VMs by resource manager.

The decision generation process of FN is formulated as a popularity-aware computation partitioning algorithm, as shown in Table II. By using this algorithm, a service routing path with lowest computing cost defined by (5)–(8) can be resolved. The minimization value $\min[V_{R_{ij}^*}^{\gamma_{\rm CQC}}]$ defined by (9) and (10) can also be calculated. For large-scale IIoT services, the Algorithm 1 will calculate the minimization value $\min[V_{R_{ij}^*}^{\gamma_{\rm CQC}}]$ following (11)–(13).

The working flow of Algorithm 1 is described as illustrated in Table II step by step. The input parameters of Algorithm 1 contain λ_{δ_t}, f_h, k_i, Th, R_{ij}, L_f^R, $\gamma_{\rm CQC}$. Therein, λ_{δ_t}, f_h, R_{ij}, L_f^R can be calculated by fog server based on the IIoT service requests in a real system. Th and $\gamma_{\rm CQC}$ are two constants, which are configured by the IIoT engineer according to the engineering experience in the applied IIoT scenario. k_i is a statistical variable that can be calculated using (2). When the data flows of IIoT services arrive at the FN, the service types of these data flows will be identified and then the IIoT service popularity rankings on this FN will be updated. If the ranking of an arriving IIoT service is less than Th, it will be pushed into the pending list. Otherwise, it will be pushed into the forwarding list ($FWList$). For the IIoT service on the pending list, FN will calculate the computing cost of providing this IIoT service and observe if the computing quality is in the scope of $\gamma_{\rm CQC}$. The FN will select a policy (it may be an identity of a virtual machine) that can minimize (9). For the IIoT service on the $FWList$, the FN will send it to the SPSRP controller for deeper analysis.

TABLE II

Algorithm 1: Popularity-aware computation partitioning algorithm intra fog node.

Input: $\lambda_{\Delta_t}, f_h, k_i, Th, R_{ij}, L_f^R, \gamma_{CQC}$
Output: VM_{ID}, Hop_{next}
1: Identify the service type of λ_{Δ_t}
2: Upgrading the popularity of all IIoT services on f_h
3: **for** $(i = 1, i++, i <= n)$
4: **if** $k_i < Th$, $k_i \in PendingList$; **else** $k_i \in FWList$
5: **end**
6: **if** $PendingList \neq NULL$
7: **for** j=1:length($PendingList$)
8: 1) Calculate the computing cost $C_{f_h}^{E_x}$
9: 2) Find R_{ij}^* the E_x belongs to; 3) Find which $L_f^R = 1$
10: **for** $l = TotalNumberofVMs$
11: Select a VM to process the arrival data flows
12: Calculate the $V_{R_{ij}^*}^{\gamma_C QC}$ and add it to $Array[l]$
13: **end**
14: Find the minimization value $min[V_{R_{ij}^*}^{\gamma_C QC}]$ in $Array[l]$ and output the parameters of corresponding VM
15: **end**
16: **end**
17: **if** $FWList \neq NULL$
18: **for** j=1:length($FWList$)
19: **if** $FWList(j) == CTSL$, {Forwarding to Hop_{next}}
20: **else** {Add $FWList(j)$ into $PendingFlow$,
21: Forwarding to SPSRP controller to get Hop_{next}}
22: **end**
23: **end**
24: **end**

B. Mobile and Heterogeneous Resources Partitioning With SPSRP Controller

Due to the mobility and heterogeneity of computation resources in FNs, IIoT users require to dynamically allocate the computation resources for the incoming computing tasks according to the context requirements or their individual demands. By introducing a SPSRP controller to the IIoT, we can achieve the cross-domain resources partitioning. It is common to see that there are multiple FNs in a city community that are dealing with different kinds of IIoT services. Usually, we cannot use the computing resources of one FN that is preprocessing the sensed temperatures when this FN is dealing with the captured face images. However, the SPSRP controller is independent of the IIoT service's type. All action sets in CTSL are generated by the SPSRP controller. These actions usually contain several instructions and parameters (such as VM creation, shutdown, and CPU clock speed).

The mobility and heterogeneity of computation resources have a great impact on $M_{R_{sd}}^{\text{mig}}$, M_u^{mis}, and $T_{R_{sd}}^U$, thus when it requires the SPSRP controller to generate CTSL for mobile and heterogeneous FNs, the above parameters should be updated according to (9)–(15). The working flow of Algorithm 2 is illustrated in Table III step by step. Algorithm 2 is executed on SPSRP controller. The input parameters of Algorithm 2 contain pending flow, GFID, GFID′, and resource type. The output of Algorithm 2 is CTSL. When SPSRP controller receives data flows, it will identify the data flows' GFID, source FN, and destination FN as well as the available service routing path. Then, the SPSRP controller traverses the computing states of all FNs that can act as the next hop on the available service routing path. Meanwhile, the GFIDs of all FNs that can act as the next hop are added into $AlternativeList$. Only when the arriving IIoT services' type matches with the type of most popular IIoT service and the total computing cost is less enough, the next FN can be specified by the SPSRP controller with CTSL.

TABLE III

Algorithm 2: Mobility and heterogeneity-aware computation partitioning algorithm inter fog node.

Input: $PendingFlow, GFID, GFID', ResourceType$
Output: $CTSL$
1: Identify the service routing path R_{ij} and the GFID of source fog node f_{h_s} that $PendingFlow$ belong to
2: Traverse the computing states of fog nodes that can act as next hop, and add their GFIDs into $AlternativeList$
3: **for** $i = 1 : length(AlternativeList)$
4: **if** $Type(PendingFlow) == Type(AlternativeList(i))$
5: **if** $Cost(PendingFlow) \leq Size(AlternativeList(i))$
6: Add $GFID'$ of $AlternativeList(i)$ to $CTSL$
7: **else**
8: Predict the mitigation cost
9: If the mitigation cost is acceptable, go to step (6); Otherwise, break;
10: **end**
11: **else** {Update the M_u^{mis}; Calculate the computing cost C_s
12: on $AlternativeList(i)$ and add it to $CostList(i)$}
13: **end**
14: **end**
15: Add the instruction of $min(CostList)$ to $CTSL$
16: Distribute $CTSL$ to f_{h_s}

V. Simulation and Discussion

In this section, we would like to evaluate the efficiency of the proposed SPSRP scheme by a series of comparisons on dead time, fault tolerance capability, successful response rate, and delay time of IIoT service retrieval. The iFogSim [29] is a public simulation platform that is generated based on CloudSim, as used in [30] and [31]. This simulation platform contains a fog-based IoT model and several kinds of measurement methods to evaluate the impact of resource management techniques in service latency, energy consumption, and computing cost. In the original document of iFogSim, the FN passively accepts IoT services. Therefore, the efficiency of the first in first process (FIFP) based resources partitioning was considered as the experimental scheme for performance comparison. Moreover, since our experiments are developed based on the iFogSim, we use many existing function libraries on iFogSim. Thus, the software/hardware configurations also can be obtained by looking up [29]. In our simulation experiments, there were total seven different scenarios had been considered as follows.

A. Performance Comparison on Delay Time of IIoT Services

Due to the heterogeneity and mobility of computation resources in FNs, observing the CPU occupancy of one or more

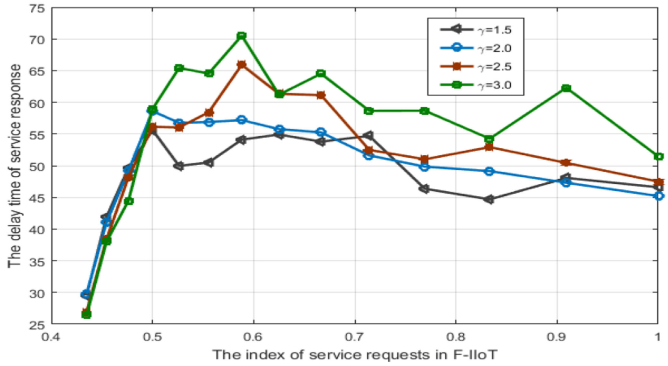


Fig. 4. Performance comparison on delay time of IIoT Services.

VM cannot provide solid and comprehensive performance evaluations. Therein, we focused on the successful response rate, service latency, dead time, and additional computing cost of the proposed scheme. The computing resources of FNs can be aggregated from the network edges of IIoT but not from the data center. Even, the VM can be migrated from the IIoT user's computer. In this scenario, we considered the computation resources of a FN were randomly offloaded from $\{'MCU','Processor','CPU','VM'\}$ with 10, 100, 1000, and 500 unit of computation resources, respectively. The arriving number of IIoT services during per minute was randomly generated within $[60 \backsim 150]$. Fig. 4 shows the delay time of IIoT service response at different values of γ. Note that if the value of γ is approximate to $+\infty$, it means that the IIoT system has not been equipped with any optimization, while all of the curves denote the delay time of service response in fog-enabled IIoT with SPSRP scheme. It can be observed that the SPSRP can reduce the delay time of service response generally by adjusting γ approximate to 1.

B. Successful Response Rate of Service Requests

In F-IIoT, the computing resources on FNs are dynamical. To bring out the strengths of the proposed SPSRP scheme, each type of IIoT services is assigned to at least one FN that has capability to process the corresponding computing tasks. In this case, we consider a peer-to-peer network with 100 IIoT users and 10 FNs. Initially, each FN maintains a $PendingList$. This $PendingList$ records all types of IIoT services that should be sent to the SPSRP controller. The SPSRP controller traverses the computing states of all other FNs to find the most reasonable FN to accept the arriving IIoT services. The successful response rate of service requests is one of the most important indicators to evaluate the performance of the SPSRP scheme. In this case, we measure the successful response rate of service requests with tree different schemes as illustrated in Fig. 5(a). It can be observed that the average successful response rate is about [0.65, 0.7] in load lancing for cloud-based IIoT, [0.75–0.8] in the FIFP scheme for fog-enabled IIoT, and [0.85 0.9] SPSRP scheme for fog-enabled IIoT. This result $([0.850.9] > [0.75 - 0.8] > [0.650.7])$ validates the performance improvements of the proposed scheme.

C. Fault Tolerance and Dead Time Compensation

The SPSRP scheme exploited Zipf's law to model the popularity rankings of IIoT services and predict the possible computing cost of processing these IIoT services on FNs. The FN partitions its own resources for processing the most popular IIoT services by arbitrating whether it should pick up and process the arriving IIoT services. To simulate the abnormal case of possible faults caused by abnormal popularity rankings of IIoT services, we set an additional service requesters in F-IIoT to inject a series of irrelevant service requests continuously to tamper the service rankings. As illustrated in Fig. 5(b), for industry-scale IoT services, the proportion of nonimpacted FNs in the proposed scheme is higher than the FIFP scheme. Therefore, the fault tolerance of the proposed scheme is improved.

The dead time is the completing time slot of resources partitioning subtracts the arriving time slot of service request. There are two different kinds of methods that can be exploited for the SPSRP controller to do the dead time compensation. First, the SPSRP controller can dynamically change the queueing time of service requests on each FN based on the real-time distribution of service requests. A short queueing time of service requests can significantly reduce the dead time, but it has a great impact on the performance of Zipf's law. Second, the SPSRP controller can dynamically change the Th on each FN based on the real-time distribution of service requests. A small Th also can significantly reduce the dead time. In FIFP-based resources partitioning scheme, it is common to do the dead time compensation by changing the queueing time, while in our simulation, we do dead time compensation by changing the Th. In Section III (C), we have introduced an SSA-based Th solving algorithm. With this algorithm, a minimized Th can be obtained by the SPSRP controller. We compare the dead time of the SPSRP scheme and FIFP-based resources partitioning scheme, as shown in Fig. 5(c). It can be observed that the proposed scheme in fog-based IIoT has lowest dead time compared to the FIFP scheme in cloud-based IIoT and fog-based IIoT.

D. Complexity Analysis and Additional Cost

In algorithms, each FN can adaptively pick up and process the most popular IIoT services and smartly partition its resources-based according to the popularity rankings of picked IIoT services. Unpopular IIoT services on a FN will be forwarded to the other FN for efficient processing. In other words, it is no need for each FN to ask for the states of other FNs. Thus, the complexity of proposed algorithms is $\Theta(n)$. The function of proposed algorithms was not to copy the load balancing and VM migration in cloud data to distributed FNs. By using Algorithms 1 and 2 to partition the resources of FNs in IIoT, we can obtain minimized computing cost and minimized CQC validation. All the performance improvements of the proposed scheme were directly beneficial to IIoT users because the service popularity reflected the real demands of IIoT users. In terms of whether it will cause additional computing cost, the answer is inevitable. However, compare to the improvements of proposed algorithms, the additional computing cost caused by complexity of proposed

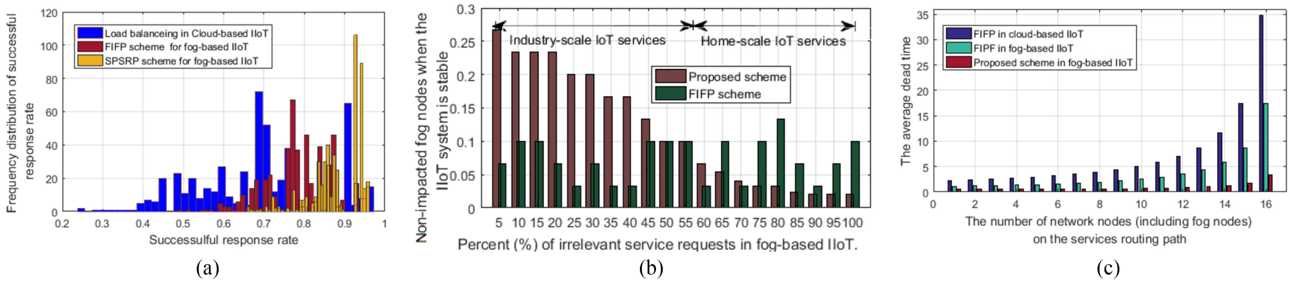


Fig. 5. Simulation results obtained for time-slot.

algorithms is minor. Moreover, the additional computing cost can be handled by resources offloaded from cloud. Besides, the FN selectively deals with the local delay-sensitive IIoT services rather than all of the arriving IIoT services.

VI. Conclusion and Future Work

In this paper, an SPSRP scheme was presented for fog computing-enabled IIoT. The SPSRP scheme modeled the relationship between service popularity and computing cost with Zipf's law. Moreover, the SPSRP scheme decoupled the computing control from data processing and support mobile and heterogeneous computing resource scheduling. Also, we provided an implementation architecture of the SPSRP scheme for real scenario. The simulations on delay time, fault tolerance, and dead time validated the strengths of the SPSRP scheme.

References

[1] C. C. Byers, "Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 14–20, Mar. 2017.

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

[3] S. Mubeen *et al.*, "Delay mitigation in offloaded cloud controllers in industrial IoT," *IEEE Access*, vol. 5, pp. 4418–4430, 2017.

[4] V. Gazis *et al.*, "Components of fog computing in an industrial internet of things context," in *Proc. IEEE Int. Conf. Sens. Commun. Netw. Workshops*, 2015, pp. 1–6.

[5] H. Zhang, Y. Xiao, S. Bu, D. Niyato, FR. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[6] W. Steiner and S. Poledna, "Fog computing as enabler for the Industrial Internet of Things," *e i Elektrotechnik und Informationstechnik*, vol. 133, no. 7, pp. 310–314, 2016.

[7] B. Tang *et al.*,"Incorporating intelligence in fog computing for big data analysis in smart cities," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2140–2150, Oct. 2017.

[8] E. Zeydan *et al.*,"Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.

[9] W. S. Jeon, J. Kim, and D. G. Jeong, "Downlink radio resource partitioning with fractional frequency reuse in femtocell networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 308–321, Jan. 2014.

[10] Y. Dhungana and C. Tellambura, "Multichannel analysis of cell range expansion and resource partitioning in two-tier heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2394–2406, Mar. 2016.

[11] U. Tefek and T. J. Lim, "Relaying and radio resource partitioning for machine-type communications in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1344–1356, Feb. 2017.

[12] S. Singh and J. G. Andrews, "Joint resource partitioning and offloading in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 888–901, Feb. 2014.

[13] M. H. Tai, N. H. Tran, L. B. Le, W. Saad, S. M. A. Kazmi, and C. S. Hong, "Coordinated resource partitioning and data offloading in wireless heterogeneous networks," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 974–977, May 2016.

[14] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surv. Tuts.* vol. 19, no. 3, pp. 1628–1656, Jul.–Sep. 2017.

[15] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[16] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold , K. Sandstrom, and M. Behnam, "Delay mitigation in offloaded cloud controllers in industrial IoT," *IEEE Access*, vol. 5, pp. 4418–4430, 2017.

[17] J. Li, L. Huang, Y. Zhou, S. He, and Z. Ming, "Computation partitioning for mobile cloud computing in a big data environment,"*IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2009–2018, Aug. 2017.

[18] D. R. D., Vasconcelos, R. M. D. C. Andrade, and J. N. D. Souza, "Smart shadow–an autonomous availability computation resource allocation platform for internet of things in the fog computing environment," in *Proc. Int. Conf. Distrib. Comput. Sens. Syst.*, 2015, pp. 216–217.

[19] H. Zhang, Y. Xiao, S. Bu, D. Niyato, FR Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining stackelberg game and matching,"*IEEE Int. Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[20] B. Yin, W. Shen, Y. Cheng, LX. Cai, and Q. Li, "Distributed resource sharing in fog-assisted big data streaming," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[21] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci.*, 2013, pp. 19–26.

[22] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *Acm Sigcomm Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[23] X. Feng, J. Wu, K. Wang, J. Li, and M. Wang, "Security analysis of simple network management protocol based IEEE P21451 internet of things," in *Proc. IEEE 15th Int. Conf. Dependable Autonomic Secure Comput., 15th Int. Conf. Pervasive Intell. Comput., 3rd Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr.(DASC/PiCom/DataCom/CyberSciTech)*, 2017, pp. 326–330.

[24] L. Breslau, P. Cao, F. Li, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proc. 18th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 1999, pp. 126–134.

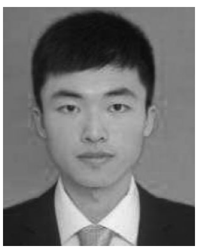
[25] L. A. Adamic and B. A. Huberman, "Zipf's law and the internet," *Glottometrics*, vol. 3, pp. 143–150, 2002.

[26] K. Suksomboon *et al.*, "PopCache: Cache more or less based on content popularity for information-centric networking," in *Proc. IEEE 38th Annu. Conf. Local Comput. Netw.*, 2013, pp. 236–243.

[27] C. Zhu, G. Cheng, and K. Wang, "Big data analytics for program popularity prediction in broadcast TV industries," *IEEE Access*, vol. 5, pp. 24593–24601, 2017.

[28] G. Li, J. Wu, J. Li, T. Ye, and R. Morello, "Battery status sensing software-defined multicast for V2G regulation in smart grid," *IEEE Sens. J.*, vol. 17, no. 23, pp. 7838–7848, Dec. 2017.

[29] H. Gupta, A. Vahiddastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments," *Softw. Pract. Exp.*, vol. 47, no. 1, pp. 1–22, 2016.
[30] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE Int. Things J.*, vol. 4, no. 5, pp. 1185–1192, Oct. 2017.
[31] M. R. Mahmud, F. Koch, and R. Buyya, "Cloud-fog interoperability in IoT-enabled healthcare solutions," in *Proc. Int. Conf. Distrib. Comput. Netw.*, 2018, pp. 1–10.



Gaolei Li (S'15) received the B.S. degree in electronic information engineering from the School of Electronic Information, Si Chuan University, Chengdu, China, in 2015. He is currently working toward the Ph.D. degree in information and communication at the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China.

He was a TPC member of international conference on Internet of Things (iThings2017) and a TC member of the IEEE CSIM. His current research interests include Industrial Internet of things, fog computing, software-defined networking, etc.

Jun Wu (S'09–M'12) received the Ph.D. degree in information and telecommunication studies (GITS) from Waseda University, Tokyo, Japan, in 2011.

From 2011 to 2012, he was a Postdoctoral Researcher with the Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology (AIST), Japan. From 2011 to 2013, he was a Researcher with GITI, Waseda University. He is currently an Associate Professor with the School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China. He also serves for Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, China, and the Vice Director with the National Engineering Laboratory for Information Content Analysis Technology, China. He has hosted and participated in several research projects for the National Natural Science Foundation of China, National 863 Plan and 973 Plan, Japan Society of the Promotion of Science projects, etc. He has authored or coauthored more than 100 papers. His research interests include the advanced computing and communications techniques in smart grids, Internet of Things, industrial security, etc.

He is currently an Associate Editor of the IEEE ACCESS, a Guest Editor for the IEEE SENSORS JOURNAL, and a TPC Member of more than ten international conferences including WICON, ICC, GLOBECOM, etc.

Jianhua Li received the B.S., M.S. and Ph.D. degrees in information security from the School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China, in 1986, 1991, and 1998, respectively.

He is a Professor/Ph.D. Supervisor and the Dean of the School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China, and the Chairman of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, China, and the Vice Director in National Engineering Laboratory for Information Content Analysis Technology, China. He was the Chief Expert in the information security committee experts with the National High Technology Research and Development Program of China (863 Program) of China and a Committee Expert of Information Technique Standardization Committee of Shanghai, China. He was the Leader of more than 30 state/province projects of China, and authored or coauthored of more than 300 papers. His research interests include cyber security, data science, next generation networking, etc.

Dr. Li was the recipient of the second prize of National Technology Progress Award of China in 2005, first prize of National Technology Progress Award of Shanghai in 2003 and 2004, and first prize of National Technology Progress Awards of Shanghai in 2004.

Kuan Wang received the B.S. degree in information and communication engineering from the School of Electronic Information Engineering in Northeastern University, Qinhuangdao, China, in 2016. He is currently working toward the Ph.D. degree in information and communication engineering at the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China.

His research interests include network calculation, Internet of things, information-centric networking, etc.

Tianpeng Ye was born in Jiangsu province of China, in 1993. He received the B.S. degree in communication engineering from Southwest Jiaotong University, Chengdu, China, in 2015 and the M.S. degree in cyber security from the School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China, in 2018.

He has participated in many national projects (e.g., National Natural Science Foundation of China) and accumulates many engineering experience. His research interests include vehicle cloud computing, network function virtualization, fog-enabled radio access network, etc.