Beef Up the Edge: Spectrum-Aware Placement of Edge Computing Services for the Internet of Things

Haichuan Ding, Yuanxiong Guo, *Member, IEEE*, Xuanheng Li, *Member, IEEE*, and Yuguang Fang, *Fellow, IEEE*

Abstract—In this paper, we introduce a network entity called point of connection (PoC), which is equipped with customized powerful communication, computing, and storage (CCS) capabilities, and design a *data transportation network (DART)* of interconnected PoCs to facilitate the provision of Internet of Things (IoT) services. By exploiting the powerful CCS capabilities of PoCs, DART brings both communication and computing services much closer to end devices so that resource-constrained IoT devices could have access to the desired communication and computing services. To achieve the design goals of DART, we further study the spectrum-aware placement of edge computing services. We formulate the service placement as a stochastic mixed-integer optimization problem and propose an enhanced coarse-grained fixing procedure to facilitate efficient solution finding. Through extensive simulations, we demonstrate the effectiveness of the resulting spectrum-aware service placement strategies and the proposed solution approach.

Index Terms—Internet of Things (IoT), edge computing, spectrum allocation, service placement

1 Introduction

To fulfill the vision of the Internet of Things (IoT) and smart cities, numerous devices are expected to be interconnected for data transmissions and service provisioning. According to recent projections, there will be around 30 billion connected IoT devices in the near future [1]. The interconnection of these devices will not only facilitate various kinds of IoT applications, such as environment monitoring, smart parking, online virtual reality gaming, augmented reality navigation, and smart healthcare, but also pose a great challenge to our already congested communication networks [2]. To deal with the spectrum shortage, governments, academia and industries have already started revisiting spectrum allocation issues, and cognitive radio (CR) technologies are considered as a promising solution. Unfortunately, in most research work, CR is employed for single-hop communications under the premise that end devices are equipped with corresponding communication capabilities, which might severely limit the effectiveness of such emerging technologies due to relatively high costs or implementation complexity, particularly when applied to resource-constrained small devices [3].

To fully exploit the benefits of CR technologies, we advocate a network-level approach where, instead of relying on resource-constrained IoT devices for spectrum sensing and establishing one-hop communications, the network exploits CR technologies to bring networking services closer to end devices. Then, IoT devices can connect to the network with potentially shorter distance and thus lower transmit power, which improves not only the energy efficiency of end devices but also frequency reuse of the spectrums that end devices are allowed or licensed to use. As a result, IoT devices can enjoy the benefits of CR technologies and obtain the desired networking services without having to conduct resource-consuming spectrum sensing and decision making. To achieve this goal, we introduce a network entity, called Point of Connection (PoC), which can tune to IoT devices' radio interfaces to offer them desired network connections. These PoCs can be wirelessly interconnected via, for example, CR technologies. They can collect data from IoT devices and collaboratively deliver the aggregated data to data networks or intended destinations through CR technologies and harvested spectrum resources, in which they act as routers. In addition, according to specific applications, PoCs can be customized with powerful computing capabilities and sufficient storage space so that they can be interconnected to form a localized computing entity at the network edge to offer desired edge computing functions/services, which aligns well with the current initiatives on edge computing [4]. In practice, both a (small) base station in cellular systems and a cognitive radio router (CR router) in cognitive radio networks can be considered as a PoC as

- *H. Ding and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611. E-mail: dhcbit@gmail.com, fang@ece.ufl.edu.*
- *Y. Guo is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078. E-mail: richard.guo@okstate.edu.*
- *X. Li is with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116023, China. E-mail: xhli@dlut.edu.cn.*

Manuscript received 4 Feb. 2018; revised 16 Oct. 2018; accepted 16 Nov. 2018. Date of publication 30 Nov. 2018; date of current version 31 Oct. 2019. (Corresponding author: Yuguang Fang.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2883952

long as they have the aforementioned communication, computing, and storage (CCS) capabilities which are collectively called the CCS capability [3]. Depending on the customers' needs or the premises of interest, PoCs can be customized accordingly. These PoCs can either be fixed at strategic locations, such as roadsides or road intersections, acting as road side units or be mounted on light-weight vehicles, acting as mobile PoCs to carry and forward data [5], [6], [7]. With these PoCs in place, end users can gain communication and/or computing services by searching for them at a closer proximity. Information contents of common interest can also be distributed to the locations where consumers are likely to be. Finally, PoCs can also serve as temporary storage facilities for big data to be opportunistically transferred to the cloud or the intended locations of consumption.

To fully exploit the potential of PoCs for service provisioning, we further develop a *data transportation network (DART)* where a secondary service provider (SSP) coordinates the deployed/recruited PoCs to make connections for end devices so that they can get desired communication/computing services. Noticing the convergence of communications, computing, and storage in service provisioning for the emerging IoT applications, data may have to be transported to the appropriate locations for processing or computing [8], [9]. To facilitate efficient service provisioning, the SSP needs to coordinate PoCs to gather information on available spectrum and computing resources in DART and jointly manage the spectrum and computing resources to support relevant IoT applications. When necessary, the SSP can organize a localized computing entity at the network edge for IoT applications with both its deployed computing resources, such as that colocated with PoCs, and the spectrum/computing resources harvested from other parties [10]. For example, when an "Amber Alert" is issued, the SSP can organize a temporary localized computing entity with PoCs to preprocess and analyze both the video streams coming from surveillance cameras and crowdsourced video streams in order to facilitate license plate identification. In this way, we can not only efficiently utilize the collected data for service provisioning but also reduce the traffic on backbone networks [11], [12], [13].

To effectively support these data processing services in DART, the SSP should assign them to appropriate edge computing facilities so that their input data can be delivered to the places where their demands for computing resources can be satisfied [14], particularly in a smart city environment. As mentioned in [13], [15], [16], [17], a large number of smart-city applications will rely on edge computing services for data processing and intelligence extraction. For example, video surveillance applications depend on edge computing services to analyze the video feeds from surveillance cameras to detect, for example, suspicious people or abnormal events [13], [18]. Traffic management applications need edge computing services to analyze the data collected from roadside sensors, traffic cameras, and vehicles, to understand current traffic conditions and identify reckless driving behaviors [17]. Crowdsensing applications employ edge computing services to remove redundancy and sensitive information in the crowdsourced data [16]. To support these edge computing services, we should not only provide enough computing resources to host the services but also offer enough communication resources to continuously move the input data from end devices to the computing facilities hosting the corresponding services. Unlike the cloud, the computing resources available at each edge computing facility is limited. In addition, the amount of data which can be moved between end devices and each computing facility is restricted by the available spectrum resources. Given the increasing number of smart-city applications and their demands for edge computing services, a critical issue is how the computing and communication resources available at the network edge could be efficiently used to simultaneously support more edge computing services. Since different edge computing services might have different sizes of input data and different requirements for computing resources, the placement of edge computing services will greatly affect resource utilization at the network edge [19]. For example, the inappropriate placement of a service with large-sized input data and low demand for computing resources might prevent other services from being placed in the corresponding computing facility, which results in the waste of the spare computing resources at the computing facility. On the other hand, the inappropriate placement of a service with high demand for computing resources and small-sized input data might force other services to be deployed to the computing facilities that are far from the corresponding end devices, which unnecessarily increases the level of contention in the network. The above observation motivates us to study the placement of edge computing services in DART by jointly considering available computing and communication resources[1] [15], [20]. To articulate our approach, we consider a DART with fixed PoCs which are deployed by an SSP. Under the supervision of the SSP, PoCs gather data from IoT devices and collectively deliver the data, via harvested licensed/unlicensed bands, to PoCs with available computing resources for processing.[2] As aforementioned, the SSP will jointly consider spectrum resources and computing resources for service placement and thus need a spectrum aware service placement (SASP) scheme. By jointly considering spectrum allocation, service placement, and the potential variations in spectrum availability, we cast the SASP schematic design as a stochastic optimization problem and reformulate it as mixed integer linear programming (MILP). We propose an enhanced coarse-grained fixing procedure to facilitate solution finding [21]. Through extensive performance evaluation, we demonstrate the effectiveness of the obtained service placement strategies and the proposed solution approach. As a first phase study, we only focus on theoretical evaluation in this paper and demonstrate the effectiveness of the obtained service placement strategies via simulations. The prototype design and real-world evaluation are left for future work.

1. Although not explicitly addressed in this paper, the storage capability is indispensable for the efficient utilization of the communication and computing capabilities of PoCs. With the storage capability, PoCs can temporarily store received data and wait for spectrum access opportunities for data delivery. Moreover, for service placement, PoCs need to store the associated libraries and databases to support a specific service type.

2. For current study, we only consider the computing resources colocated with or embedded in PoCs.

The major contributions of this paper are summarized as follows:

- In view of resource-constrained end devices, we advocate a network-level approach, which leverages the capabilities of PoCs and network-side management/coordination, to facilitate IoT applications. To achieve this goal, we introduce the concept of PoCs and design DART based on interconnected PoCs to support communication and computing services for IoT applications.
- Noticing the importance and popularity of data processing services at the network edge, we further study an edge computing service placement problem in DART. By jointly considering spectrum allocation, service placement, and the variations in spectrum availability, we formulate the service placement problem as a stochastic optimization problem.
- We reformulate the stochastic optimization problem as MILP and develop an enhanced coarse-grained fixing procedure for efficient solution finding.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. The architecture of DART and the considered scenario are introduced in Section 3. The SASP schematic design is formulated as a stochastic optimization problem, which is further recast as an MILP in Section 4. In Section 5, an enhanced coarse-grained fixing procedure is introduced for efficient solution finding. Performance evaluation is conducted in Section 6. Finally, conclusions are drawn in Section 7.

2 Related Work

Although we have advocated network-level architectural design to take full advantage of CR technologies [3], [5], we only focused on the communications aspect [22], [23]. In this paper, we refine our design and introduce the concept of PoC so that the designed DART architecture can not only provide significantly better communication services but also efficiently support mobile computing and opportunistic storage, commonly observed in various future IoT applications. Different from our previous proposals, PoCs enable the desired connections of end devices for both communication and computing services. Thus, in addition to communication capabilities, PoCs should be endowed with sufficient computing capabilities to facilitate service discovery and provisioning [24]. The systems developed by Microsoft bear some similarity to DART as both of them utilize network connecting devices and under-utilized spectrum resources to serve end devices [25], [26]. However, their design goals are different. Microsoft introduces network connecting devices so that the favorable signal propagation characteristics in TV White Spaces can be exploited to extend network coverage over a large area and provide cost-effective solutions to sensory data collection and Internet service provisioning. In contrast, the design goal of DART is on how to exploit PoCs and the under-utilized spectrum resources, besides TV White Spaces, to support the increasing wireless traffic and facilitate service provisioning. In DART, communication and computing resources should be jointly managed, and thus the SSP needs novel resource management schemes to provide the needed communication and computing services for IoT applications.

How to efficiently place computing services at different edge computing nodes has been studied in [27], [28], [29], [30], [31], [32]. Generally speaking, the service placement schemes are designed by jointly considering the computing resources requested by each service and the communication resources used to deliver the input data. In [28], Yu et al. consider a fog radio access network (F-RAN) where a group of fog nodes are interconnected via backhaul links to serve users' requests, and a long-term service placement problem is formulated to minimize the incurred backhaul traffic. Other than backhaul connections, service providers often need to rent edge computing resources for service provisioning. In view of this, Yang et al. study a cost-aware service placement and load dispatch problem for mobile cloudlets in order to balance the average latency of users' requests and the cost of service providers [30]. Similar problems have been considered in [31], [32]. In [31], Gu et al. design cost-efficient service placement schemes for cellular networks with colocated edge computing resources to support medical cyber-physical systems. By further considering base station association, the corresponding problem is formulated as a mixed integer nonlinear programming, which is linearized and solved via a linear programming based two-phase heuristic algorithm. In [32], Chen et al. investigate a collaborative service placement problem where multiple small cell base stations collaboratively provide edge computing services to end users. By transforming the formulated utility maximization problem into a cost minimization problem, an efficient distributed algorithm is designed to facilitate solution finding. In all the aforementioned scenarios, the input data is delivered via wired transmissions or at most one-hop wireless transmissions, whereas, PoCs in DART could collaboratively harvest licensed/unlicensed spectrum resources and deliver the input data to corresponding edge computing facilities through multi-hop wireless transmissions. To facilitate efficient service provisioning, the SSP should jointly consider service placement, spectrum allocation, flow routing, which is much more challenging than the cases considered in existing work. Particularly, given the uncertain activities of licensed/unlicensed users, the SSP should consider potential variations in spectrum availability when making service placement decisions. Thus, existing schemes might not be efficient in these situations, and how to place computing services in DART needs further exploration.

Our work and existing literature on multi-resource schedulers are focused on different aspects of resource sharing, which are on different time scales. Specifically, we investigate where each edge computing service should be placed during the considered period of time so that more edge computing services can be simultaneously supported. In other words, instead of real-time scheduling, we study a service placement problem which determines the set of services to share a specific group of resources. In contrast, existing work on multi-resource schedulers, such as the Dominant Resource Fairness (DRF) scheduler and the DRFH scheduler, is interested in how to schedule the tasks of different services in real time so that a group of resources are shared among a set of services and each service gets its promised/allocated resources [33], [34]. In other words, given the placement of services, existing work on multi-resource schedulers is focused on real-time task scheduling

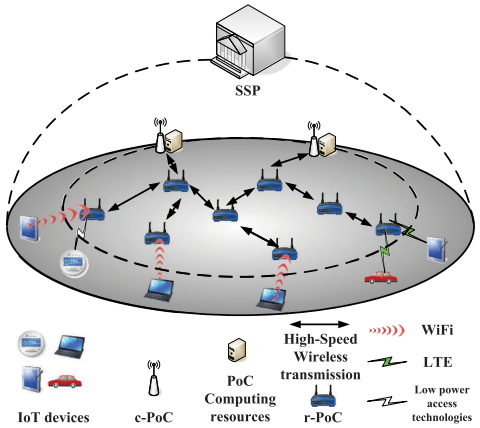


Fig. 1. The network architecture of DART. The symbols between IoT devices and r-PoCs represent data transmissions between these entities.

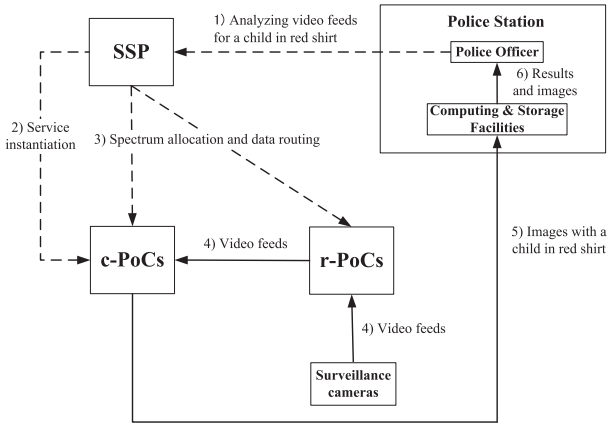


Fig. 2. Edge video analytics to locate a lost child. The arrows represent data flows and the dashed arrows represent control flows.

policies which facilitate resource sharing among the set of services assigned to the same group of resources.

3 Network Architecture and Models

3.1 The Network Architecture of DART

As shown in Fig. 1, DART consists of a set of PoCs deployed/recruited and coordinated by an SSP.

PoCs are network entities which can tune to end devices' radio interfaces to communicate, and have sufficient computing resources and storage space to push/pull computing and data services close to end devices. PoCs in DART can be either fixed or mobile. The fixed PoCs are deployed at strategic locations by the SSP to transport data and provide basic network services. Some fixed PoCs connect to data networks such as the Internet via wired connections or sustainable and reliable wireless connections, serving as agents for the SSP to manage available resources and data transportation and offering the presence of network connections to the backbone networks. These PoCs are also called *connected PoCs*, or c-PoCs for short. Other fixed PoCs that do not have direct network connections serve as relaying nodes for IoT devices and/or collecting points for computing tasks and/or opportunistic caching nodes. These fixed PoCs are called relaying PoCs (r-PoCs). The mobile PoCs are mainly utilized to transport data around, which are generally installed on vehicles. Depending on where they are installed, the communication, computing, and storage capabilities (CCS capabilities for short) of PoCs can be customized accordingly to perform communication and computing tasks. It should be noted that "PoC computing resources" in Fig. 1 refers to the computing resources colocated with or embedded in c-PoCs where edge computing tasks are executed, which is similar to edge servers deployed at base stations or WiFi access points [11], [35].

The SSP could be an independent service provider or an existing service operator such as a cellular operator, but must have its own reliable bands (e.g., cellular bands if the SSP is a cellular operator), called basic bands, which are used to support common control signaling or services with stringent quality-of-service (QoS) guarantee. The network control functions of the SSP are implemented through, for example, the computing resources colocated with or embedded in the c-PoCs. These network control functions manage the operations of PoCs and available communication/computing resources in DART to connect IoT devices to the desired data and edge computing services. The SSP employs PoCs to collect various kinds of information, such as locally available spectrum bands, computing, and storage resources. Based on the collected information, the network control functions make decisions on spectrum allocation, data delivery, and edge computing service placement. The corresponding decisions are sent to PoCs via the SSP's basic bands to coordinate the operations of PoCs so as to support requested services.

To elaborate on our approach, in the following, we study the placement of streaming data processing services in DART with CR routers introduced in [3], [5] as PoCs. As a result, in the following development, the high-speed wireless transmissions between PoCs are achieved by harvesting and exploiting a wide range of under-utilized lincensed/unlicensed spectrum. Before getting into the modeling part, we will give an example of the streaming data processing services considered in this paper.

3.2 An Example of Edge Computing Services

A relevant example is edge video analytics to locate a lost child [16]. When a child has gone missing during a large public event, such as a parade, the local police station can issue service requests to the SSP to analyze video feeds from surveillance cameras to search for a child wearing a specific type of clothes. The corresponding system diagram is shown in Fig. 2. To better illustrate how the SSP processes a service request, in Fig. 2, we consider a DART with an SSP, r-PoCs, and c-PoCs with available computing resources for task execution. Since such a service will involve continuously receiving and processing of the video feed from a camera, the SSP determines where the service should be placed by jointly considering available spectrum and computing resources and coordinates PoCs for service provisioning. Specifically, when the SSP receives the service request, it places the service to the PoC (c-PoC in Fig. 2) which best meets the computing and bandwidth requirements of this service.[3] Then, it makes spectrum allocation

3. Once the SSP makes the service placement decisions, it will directly instantiate the services at the selected PoCs if they have the copies of the corresponding services [12]. Otherwise, it will extract the services from a service repository and send them to the selected PoCs for service instantiation [36].

and data routing decisions for PoCs in order to establish a connection between the source (a surveillance camera in Fig. 2) and the selected PoC for input data delivery. Once such a child is found, the corresponding images will be sent to the the local police station, through data networks, for further analysis to determine if it is the lost child.

In this example, each service request corresponds to analyzing the video feed from a specific surveillance camera. Thus, its bandwidth requirement corresponds to the bandwidth requirement of a surveillance camera, and its computing requirement is the processing power required to analyze the corresponding video feed. According to [37], the bandwidth requirement of a surveillance camera is closely related to the video resolution, the frame rate, the image quality level, the complexity of the scene, and the adopted compression algorithm, which makes it difficult to exactly predict the bandwidth requirement. In practice, such bandwidth requirement can be obtained through real-world measurements. When H.264 is used for video compression, the estimated bandwidth requirements of a surveillance camera under different parameter settings are provided in [37]. For example, a video with high image quality, resolution 720×576, and frame rate of $6\,\text{fps}$ requires a bandwidth of $0.5\,\text{Mbps}$. When the frame rate of this video increases to $30\,\text{fps}$, its bandwidth requirement will increase to $1\,\text{Mbps}$. The computing requirement can be estimated from the resolution of the video, the frame rate and the adopted processing algorithms. From [38], the computing requirement can be estimated through the product of the size of input data and the number of CPU cycles needed to process 1-bit of the input data. For example, if the adopted algorithm needs 10 CPU cycles to process 1-bit of the input data, we will need 9.95×10^7 CPU cycles to process an image with resolution of 720×576. Thus, a video feed with resolution 720×576 and frame rate of $30\,\text{fps}$ requires the processing power of $2.99\,\text{GHz}$.

3.3 Network Configuration and Related Models

We consider a DART where N fixed PoCs, indexed as $\mathcal{N} = \{1, \ldots, N\}$, are deployed by the SSP. The c-PoCs with computing resources available for executing computing tasks are collected in a set $\mathcal{V} \subseteq \mathcal{N}$. These PoCs are connected to data networks via wired connections or sustainable and reliable wireless connections. For each PoC $i \in \mathcal{V}$, denote its available computing capability (i.e., CPU cycles per second) as Δ_i. There are K streaming data processing service requests, such as the requests for edge video analytics mentioned in Section 3.2, and the SSP attempts to support as many services as possible by assigning these services to appropriate c-PoCs.[4] The kth service is characterized by a tuple (s_k, r_k, δ_k), where s_k is the source PoC where the input data of the kth service is collected, r_k is the data rate required to move the input data of the kth service to assigned PoCs, and δ_k is the computing capability required by the kth service[5] [14].

4. For the current study, we use c-PoCs and their colocated computing resources interchangeably.

5. δ_ks are submitted to the SSP by the clients when requesting edge computing services.

TABLE 1
The List of Important Notations and Definitions

Notation	Definition
N	The number of fixed PoCs
$\mathcal{N}$	The set of fixed PoCs
$\mathcal{V}$	The set of c-PoCs with colocated computing resources
Δ_i	The computing capability of the computing resources colocated with PoC $i \in \mathcal{V}$
K	The number of stream processing service requests
s_k	The PoC where the input data of the kth service is collected
r_k	The bandwidth requirement of the kth service
δ_k	The computing requirement of the kth service
$\mathcal{M}_i$	The set of spectrum bands available at PoC i
w_{ij}^m	The equivalent bandwidth of band $m \in \mathcal{M}_i \cap \mathcal{M}_j$ available to the SSP
θ_{ki}	$= 1$ if the kth service is placed at PoC $i \in \mathcal{V}$ and is 0 otherwise
f_{ij}^k	The flow rate of the kth flow allocated to link (i, j)
y_{ij}^m	$= 1$ if band m is allocated to link (i, j) and is 0 otherwise

The data transmissions and interfering relationships between different PoCs are characterized via the transmission range R_T and the interference range R_I, respectively. We call the link from PoC i to PoC j as link (i, j). Link (i, j) exists only when the distance between PoC i and PoC j is less than R_T. When the distance between PoC i and PoC l is larger than R_I, the interference incurred by the transmission of PoC i is negligible at PoC l.

Due to spatial variations in spectrum availability, the set of available spectrum bands at different PoCs might be different. Denote the set of spectrum bands available at PoC i as $\mathcal{M}_i$. $\mathcal{M}_i$ is a subset of $\mathcal{M} = \{1, \ldots, m, \ldots, M\}$, where M is the total number of spectrum bands in the considered system. Thus, the bands in $\mathcal{M}_i \cap \mathcal{M}_j$ can be utilized to support the data transmissions over link (i, j). Since these bands are harvested via CR technologies, the SSP needs to vacate these bands when primary users (PUs) reclaim them. Noticing that the SSP might not be able to accurately predict the activities of PUs and thus the duration when these bands can be used for its transmissions, we model the equivalent bandwidth of band $m \in \mathcal{M}_i \cap \mathcal{M}_j$ available to the SSP as a random variable w_{ij}^m and assume w_{ij}^m's are independent and identically distributed (i.i.d.) for illustrative purposes.

Based on these models, we will formulate the SSP's spectrum aware service placement (SASP) schematic design as a stochastic optimization problem in the next section.

4 Problem Formulation

As aforementioned, the SSP should jointly consider available communication and computing resources when determining its service placement strategy. In this paper, we employ a 0-1 integer variable θ_{ki}, $i \in \mathcal{V}$, to represent the SSP's service placement strategy, and θ_{ki} equals 1 only when the kth service is assigned to PoC $i \in \mathcal{V}$. Table 1 lists the important notations used in this paper.

4.1 Flow Routing and Spectrum Allocation Constraints

For effective service placement, the SSP must ensure the input data of different services can be delivered to the assigned c-PoCs. In this paper, we consider a case where

each edge computing service can be placed in at most one PoC in $\mathcal{V}$, i.e.,

$$\sum_{i\in\mathcal{V}}\theta_{ki}\leq 1, k\in\{1,\ldots,K\}. \tag{1}$$

Since the SSP's placement decision for each service is not known in advance, the kth service could be placed at any PoCs in $\mathcal{V}$, which implies that any PoCs in $\mathcal{V}$ could potentially be the destination for input data delivery. In view of (1), we introduce a virtual sink node d_k as the virtual destination for the data of the kth service to facilitate problem formulation. We add a directed link from PoC $i\in\mathcal{V}$ to d_k with capacity $\theta_{ki}\Upsilon$, where Υ is large enough so that the link between d_k and PoC $i\in\mathcal{V}$ will not become the bottleneck when $\theta_{ki}=1$. It should be noted that d_k's and the corresponding directed links are virtual nodes and virtual links added to ease the formulation of the flow routing constraints. No actual data flow will go through the added directed links, nor will any spectrum resources be allocated to these added links.

Noticing that the capacity of the link from PoC $i\in\mathcal{V}$ to d_k is positive only when $\theta_{ki}=1$, i.e., the kth service is placed at PoC $i\in\mathcal{V}$, the amount of data delivered from s_k to d_k equals that delivered from s_k to the c-PoC where the kth service is placed. As a result, with d_k, the SSP can determine the amount of data potentially delivered from PoC s_k to the PoCs in $\mathcal{V}$, which depends on both the available harvested spectrum resources and the SSP's data routing strategies, according to the following flow routing and link scheduling constraints.

We call the flow incurred by delivering the input data of the kth service as the kth flow, denoted as f^k. Let f_{ij}^k be the flow rate of the kth flow allocated to the link from PoC i to PoC j. Since PoC s_k is the source node for the kth flow, we have

$$\sum_{i=s_k,j\in\mathcal{T}_{s_k}} f_{ij}^k = r_k, \tag{2}$$

$$\sum_{\{i|s_k\in\mathcal{T}_i\},j=s_k} f_{ij}^k = 0, \tag{3}$$

where $\mathcal{T}_{s_k}$ is the set of PoCs within the transmission range of PoC s_k. $\{i|s_k\in\mathcal{T}_i\}$ is the set of PoCs with PoC s_k staying in their transmission range.

For other PoCs, the kth flow must satisfy the following flow balance constraints

$$\sum_{\{i|j\in\mathcal{T}_i\}} f_{ij}^k = \sum_{l\in\mathcal{T}_j\cup\Xi_k} f_{jl}^k, \forall j\in\mathcal{N}, j\neq s_k, j\neq d_k, \tag{4}$$

where $\Xi_k=\{d_k\}$ if $j\in\mathcal{V}$ and $\Xi_k=\emptyset$ if $j\notin\mathcal{V}$.

Since d_k is the destination of the kth flow and we only consider the link from PoC $i\in\mathcal{V}$ to d_k, the flow routing constraint at d_k can be formulated as

$$\sum_{i\in\mathcal{V},j=d_k} f_{ij}^k = r_k. \tag{5}$$

Noticing that d_k is the destination for the kth flow, we have $f_{id_k}^{k'}=0, \forall i\in\mathcal{V}, k'\neq k$.

Clearly, the amount of flow carried over each link cannot exceed the achievable data rate of the corresponding link, which is closely related to the SSP's spectrum allocation strategy. We introduce a $0-1$ integer variable $y_{ij}^m\in\{0,1\}$, $i,j\in\mathcal{N}$, $m\in\mathcal{M}_i\cap\mathcal{M}_j$, to represent the SSP's spectrum allocation decisions. $y_{ij}^m=1$ only when band m is allocated to link (i,j). To ensure efficient data delivery, a PoC cannot simultaneously transmit to or receive from multiple PoCs on the same band, which leads to the following constraints

$$\sum_{\{j\in\mathcal{T}_i|m\in\mathcal{M}_j\}} y_{ij}^m \leq 1, \forall i\in\mathcal{N}, m\in\mathcal{M}_i, \tag{6}$$

$$\sum_{\{j|i\in\mathcal{T}_j,m\in\mathcal{M}_j\}} y_{ji}^m \leq 1, \forall i\in\mathcal{N}, m\in\mathcal{M}_i, \tag{7}$$

where $\{j\in\mathcal{T}_i|m\in\mathcal{M}_j\}$ represents the set of PoCs in $\mathcal{T}_i$ with available band m, $\{j|i\in\mathcal{T}_j, m\in\mathcal{M}_j\}$ is the set of PoCs with PoC i in their transmission range and available band m.

To avoid self-interference, a PoC cannot simultaneously transmit and receive on the same band, i.e.,

$$\begin{aligned} y_{ji}^m + \sum_{\{j'\in\mathcal{T}_i|m\in\mathcal{M}_{j'}\}} y_{ij'}^m \leq 1, \\ \forall j\in\mathcal{N}, i\in\mathcal{T}_j, m\in\mathcal{M}_i\cap\mathcal{M}_j, \end{aligned} \tag{8}$$

where $\mathcal{M}_i\cap\mathcal{M}_j$ is the set of available bands for link (i,j).

Besides above constraints, the SSP should ensure the operations of different links will not interfere with each other. That is, if link (i,j) is scheduled, all other links that interfere the operation of link (i,j) cannot be scheduled. Thus, we have the following link scheduling constraints

$$\begin{aligned} y_{ji}^m + \sum_{\{l\in\mathcal{T}_{j'}|m\in\mathcal{M}_l\}} y_{j'l}^m \leq 1, \forall j\in\mathcal{N}, i\in\mathcal{T}_j, m\in\mathcal{M}_i\cap\mathcal{M}_j, \\ j'\in\left\{j'\middle|i\in I_{j'}, j'\neq j, m\in\mathcal{M}_{j'}\right\}, \end{aligned} \tag{9}$$

where $I_{j'}$ is the set of PoCs within the interference range of PoC j'.

Notice that f^k's are feasible only when the corresponding flow rate over each link can be supported by the allocated spectrum resources. For link (i,j), $i\in\mathcal{N}, j\in\mathcal{T}_i$, if the values of w_{ij}^m's are known, the SSP can ensure the feasibility of f^k's by imposing the following constraints

$$\sum_{k=1}^{K} f_{ij}^k \leq \sum_{m\in\mathcal{M}_i\cap\mathcal{M}_j} y_{ij}^m c w_{ij}^m, \forall i\in\mathcal{N}, j\in\mathcal{T}_i, \tag{10}$$

where c is the spectral efficiency of link (i,j) over any band $m\in\mathcal{M}_i\cap\mathcal{M}_j$.

Unfortunately, due to the uncertain activities of licensed/unlicensed users, the values of w_{ij}^m's are not known in advance, which implies that (10) cannot be directly applied in our formulation. To incorporate the uncertainty of w_{ij}^m in the formulation, we quantify the achievable data rate of link (i,j) by the following chance constraint

$$\mathrm{P}\left(\sum_{k=1}^{K} f_{ij}^k \leq \sum_{m\in\mathcal{M}_i\cap\mathcal{M}_j} y_{ij}^m c w_{ij}^m\right) \geq \alpha, \forall i\in\mathcal{N}, j\in\mathcal{T}_i, \tag{11}$$

where α is the confidence level for the chance constraint. (11) means that f_{ij}^k's are considered to be feasible if it can be supported by the spectrum resources allocated to link (i,j) with at least probability α.

4.2 Optimal Spectrum Aware Service Placement

Given the above constraints, the SSP can make optimal spectrum placement decisions by jointly considering the available communication and computing resources. Noticing that the computing resources requested by the placed services cannot exceed the capabilities of the corresponding PoCs, we have the following constraints

$$\sum_{k=1}^{K} \theta_{ki}\delta_k \leq \Delta_i, \forall i \in \mathcal{V}. \tag{12}$$

To support as many services as possible, the SSP needs a service placement strategy to maximize the number of supported services while satisfying the communication and computing resource constraints.[6] The desired strategy can be obtained from the following optimization problem

$$\begin{aligned}
&\textbf{OPT}: \text{maximize } \vartheta = \sum_{k=1}^{K}\sum_{i\in\mathcal{V}} \theta_{ki} \\
&\text{s.t.: } (1) \sim (9), (11), (12), \\
&\quad f_{ij}^k \geq 0, i \in \mathcal{N}, j \in \mathcal{T}_i, k \in \{1,\ldots,K\}, \\
&\quad f_{id_k}^k \geq 0, i \in \mathcal{V}, k \in \{1,\ldots,K\}, \\
&\quad f_{id_k}^{k'} = 0, i \in \mathcal{V}, k, k' \in \{1,\ldots,K\}, k \neq k', \\
&\quad \theta_{ki} \in \{0,1\}, k \in \{1,\ldots,K\}, i \in \mathcal{V}, \\
&\quad y_{ij}^m \in \{0,1\}, i \in \mathcal{N}, j \in \mathcal{T}_i, m \in \mathcal{M}_i \cap \mathcal{M}_j.
\end{aligned} \tag{13}$$

Clearly, since **OPT** aims to efficiently utilize available communication and computing resources to support as many services as possible, it will consider placing the services next to the corresponding IoT devices when possible. Once placed, the services will not be reallocated for a period of time. As time goes on, the placement might not be optimal and the SSP can reallocate these services based on **OPT**. When the SSP should reallocate these services is an interesting research problem and will be explored in our future work.

Due to the stochastic constraint, it is challenging for us to take advantage of existing solution approaches to solve **OPT**. In the next section, we will reformulate **OPT** to facilitate solution finding.

4.3 Problem Reformulation

For simplicity, let $\chi_{ij} = \sum_{m\in\mathcal{M}_i\cap\mathcal{M}_j} y_{ij}^m w_{ij}^m$ and $\chi_{ij}(\alpha)$ be the largest value which satisfies $\mathrm{P}\big(\chi_{ij} \geq \chi_{ij}(\alpha)\big) \geq \alpha$. Then, (11) is equivalent to

$$\sum_{k=1}^{K} f_{ij}^k \leq c\chi_{ij}(\alpha), i \in \mathcal{N}, j \in \mathcal{T}_i. \tag{14}$$

Since w_{ij}^m's are assumed to be i.i.d. for our case study, the distribution of χ_{ij} is a sum of i.i.d. random variables and the number of these random variables is determined by $\sum_{m\in\mathcal{M}_i\cap\mathcal{M}_j} y_{ij}^m$. Let χ_{ij}^τ be the sum of τ i.i.d. random variables with the same distribution as w_{ij}^m and $\chi_{ij}^\tau(\alpha)$ be the largest value which satisfies $\mathrm{P}(\chi_{ij}^\tau \geq \chi_{ij}^\tau(\alpha)) \geq \alpha$. Then, $\chi_{ij}(\alpha)$ can be expressed as

$$\chi_{ij}(\alpha) = \sum_{\tau=1}^{|\mathcal{M}_i\cap\mathcal{M}_j|} \eta_{ij}^\tau \chi_{ij}^\tau(\alpha), i \in \mathcal{N}, j \in \mathcal{T}_i, \tag{15}$$

where $|\mathcal{M}_i \cap \mathcal{M}_j|$ is the cardinality of $\mathcal{M}_i \cap \mathcal{M}_j$. η_{ij}^τ is a 0-1 integer variable and equals 1 only when $\sum_{m\in\mathcal{M}_i\cap\mathcal{M}_j} y_{ij}^m = \tau$. This implies that η_{ij}^τ's should satisfy the following constraints

$$\sum_{\tau=1}^{|\mathcal{M}_i\cap\mathcal{M}_j|} \tau\eta_{ij}^\tau = \sum_{m\in\mathcal{M}_i\cap\mathcal{M}_j} y_{ij}^m, i \in \mathcal{N}, j \in \mathcal{T}_i, \tag{16}$$

$$\sum_{\tau=1}^{|\mathcal{M}_i\cap\mathcal{M}_j|} \eta_{ij}^\tau \leq 1, i \in \mathcal{N}, j \in \mathcal{T}_i, \tag{17}$$

where (17) ensures that at most a single η_{ij}^τ, $\tau = 1,\ldots,|\mathcal{M}_i \cap \mathcal{M}_j|$, equals 1. (16) and (17) guarantee that η_{ij}^τ equals 1 only when $\sum_{m\in\mathcal{M}_i\cap\mathcal{M}_j} y_{ij}^m = \tau$.

Thus, the chance constraint in (11) is equivalent to a set of linear constraints shown in (14), (15), (16), and (17). By replacing (11) in **OPT** with (14), (15), (16), and (17), we can obtain an equivalent optimization problem of **OPT** as

$$\begin{aligned}
&\textbf{OPT1}: \text{maximize } \vartheta = \sum_{k=1}^{K}\sum_{i\in\mathcal{V}} \theta_{ki} \\
&\text{s.t.: Other constraints in } \textbf{OPT} \text{ except } (11), \\
&\sum_{k=1}^{K} f_{ij}^k \leq c \sum_{\tau=1}^{|\mathcal{M}_i\cap\mathcal{M}_j|} \eta_{ij}^\tau \chi_{ij}^\tau(\alpha), i \in \mathcal{N}, j \in \mathcal{T}_i, \\
&\sum_{\tau=1}^{|\mathcal{M}_i\cap\mathcal{M}_j|} \tau\eta_{ij}^\tau = \sum_{m\in\mathcal{M}_i\cap\mathcal{M}_j} y_{ij}^m, i \in \mathcal{N}, j \in \mathcal{T}_i, \\
&\sum_{\tau=1}^{|\mathcal{M}_i\cap\mathcal{M}_j|} \eta_{ij}^\tau \leq 1, i \in \mathcal{N}, j \in \mathcal{T}_i, \\
&\eta_{ij}^\tau \in \{0,1\}, i \in \mathcal{N}, j \in \mathcal{T}_i, \\
&\tau \in \{1,\ldots,|\mathcal{M}_i \cap \mathcal{M}_j|\},
\end{aligned}$$

where f_{ij}^k, θ_{ki}, y_{ij}^m, and η_{ij}^τ are decision variables. Clearly, the objective function and the constraints in **OPT1** are linear. Noticing that θ_{ki}, y_{ij}^m and η_{ij}^τ are integer variables, **OPT1** is a mixed integer linear programming (MILP), which is NP-hard in general, and the number of integer variables in the MILP increases with the size of the network. Besides, as pointed out in [21], due to the spectrum allocation constraints ((6), (7), (8), and (9)), solving the formulated MILP could be equivalent to searching the maximum independent set of a graph, which is not only NP-hard but extremely difficult to approximate as well. Particularly, as mentioned in [21], the maximum independent set problem is hard due to the existence of odd holes and odd anti-holes in the corresponding graphs and the spectrum allocation constraints in the formulated MILP could give rise to a graph with many odd holes and odd anti-holes. In other words, the spectrum allocation constraints in the MILP hinder us to develop approximation algorithms to solve the formulated MILP with analytical performance guarantee. In the following section, we will introduce a heuristic algorithm, called basic coarse-grained fixing procedure, to find the solution.

6. Due to resource constraints, the SSP might not be able to support all these services. In this case, the SSP can, for example, harvest extra computing resources from other systems for service provisioning, which will be addressed in future work.

4.4 A Basic Coarse-Grained Fixing Procedure for Solution Finding

Clearly, the difficulty in solving **OPT1** primarily comes from the spectrum allocation constraints where the number of involved integer variables, y_{ij}^m, increases as the size of the considered network grows. As long as the values of y_{ij}^m's are fixed, the values of other variables can be easily fixed via off-the-shelf optimization software. Thus, the key to solve **OPT1** is to develop good heuristics to fix the values of y_{ij}^m's.

Motivated by the coarse-grained fixing procedure developed in [39], we attempt to fix the value of y_{ij}^m's by relaxing y_{ij}^m's and η_{ij}^τ's to $[0,1]$ and solving a sequence of relaxed optimization problems. After solving each relaxed optimization problem, we set the values of certain y_{ij}^m's to either 0 or 1 according to their values in the obtained solution. For y_{ij}^m's with values close to 1 in the obtained solution, if they are set to 0, the achievable data rates of the corresponding links will be significantly reduced and the value of the objective function is more likely to be adversely affected. Thus, if y_{ij}^m's have values close to 1 in the obtained solution, we will fix them to 1. Specifically, we pick the y_{ij}^m's, which have not been fixed yet, with the values larger than 0.5 in the obtained solution and set them to 1 to obtain another optimization problem which will be solved as the next optimization problem. If all such y_{ij}^m's are less than 0.5 in the obtained solution, the y_{ij}^m with the largest value is set to 1. Once the value of y_{ij}^m is set to 1, we can also fix a set of other y_{ij}^m's based on the conflicting relationships by the spectrum allocation constraints. When all y_{ij}^m's are fixed, we solve the resulting optimization problem to obtain the final results. The above procedure is called the basic coarse-grained fixing procedure and is shown in Algorithm 1.

Algorithm 1. Basic Coarse-Grained Fixing Procedure

Input: The parameters of **OPT1**.
Output: A solution to **OPT1**.
1: Relax y_{ij}^m's and η_{ij}^τ's to the interval $[0,1]$ and collect y_{ij}^m's not yet fixed in a set φ;
2: Solve the relaxed optimization problem and collect y_{ij}^m's with values larger than 0.5 in a set ψ;
3: **if** $\psi \cap \varphi \neq \emptyset$ **then**
4: Fix y_{ij}^m's in ψ to 1;
5: **else**
6: Search for the y_{ij}^m with the largest value in φ and set the obtained y_{ij}^m to 1;
7: **end if**
8: Based on the fixed y_{ij}^m's, try to fix other y_{ij}^m's according to the constraints (6), (7), (8), and (9). Remove the fixed y_{ij}^m's from φ;
9: **if** $\varphi = \emptyset$ **then**
10: Go to Line 14;
11: **else**
12: Reformulate the relaxed optimization problem with the lately fixed y_{ij}^m's and go to Line 2;
13: **end if**
14: Reformulate **OPT1** with all fixed y_{ij}^m's and solve the reformulated optimization problem to obtain the maximum objective value ϑ and the corresponding solution x;
15: **return** ϑ and x;

5 An Enhanced Coarse-Grained Fixing Procedure for Solution Finding

While fixing y_{ij}^m's with the basic coarse-grained fixing procedure, we might encounter the case where two conflicting variables, say y_{ij}^m and $y_{ij'}^m$, have values around 0.5. For example, we have the following scenario: y_{ij}^m is 0.52 and $y_{ij'}^m$ is 0.48. In this case, it is difficult to determine which variable will have larger impact on the value of the objective function merely based on their values in the obtained solution. If we set y_{ij}^m to 1 according to the aforementioned procedure, $y_{ij'}^m$ will be automatically set to 0. Noticing that $y_{ij'}^m$ in the obtained solution is very close to 0.5, directly fixing y_{ij}^m to 1 might significantly impair the achievable rate of link (i,j') and thus adversely affect the final performance. To facilitate efficient spectrum allocation, we introduce a revised optimization problem which is obtained from the original relaxed problem by reducing the bandwidth per harvested band. With a reduced bandwidth, the solution to the original relaxed problem becomes infeasible. In other words, the flow rates allocated to different links can no longer be supported based on the y_{ij}^m's obtained by solving the original relaxed optimization problem. Since the flow rates on different links have different impacts on the objective function, to maintain the performance, we should try to keep the flow rates of the links which affect the objective function the most. Thus, more spectrum resources should be allocated to these important links, which might push y_{ij}^m's away from 0.5 and thus facilitate more efficient spectrum allocation. Based on this idea, we further introduce an enhanced coarse-grained fixing procedure as shown in Algorithm 2 for **OPT1** by noticing that the set of y_{ij}^m's, which are fixed to 1 during parameter fixing procedure, will be affected by the parameter-fixing decisions in the initial step.

In the enhanced coarse-grained fixing procedure, **OPT1** is first solved by following the basic coarse-grained fixing procedure to obtain a solution x_1 and the corresponding value of the objective function ϑ_1. Then, **OPT1** is solved again through the basic coarse-grained fixing procedure but with a different set of y_{ij}^m's fixed in the initial step. This set of y_{ij}^m's are determined by solving a revised version of **OPT1** with the bandwidth of each harvested band reduced and the integer variables relaxed to $[0,1]$. By following this procedure, we obtain another solution x_2 to **OPT1** and the corresponding value of the objective function ϑ_2. Finally, the solution to **OPT1** is chosen from x_1 and x_2 based on the values of ϑ_1 and ϑ_2.

6 Performance Evaluation

To evaluate the effectiveness of our SASP scheme as well as the proposed heuristic algorithms, we consider a DART with $N = 10$ PoCs and a scenario where the SSP receives $K = 10$ computing service requests and makes service placement decisions for received service requests to maximize the number of supported services. These 10 services have the same source PoC, and two of the 10 PoCs are c-PoCs with available computing resources for task execution. The PoCs are randomly distributed in a $500 \times 500\ \mathrm{m}^2$ area to form a connected network. We randomly pick a PoC from these PoCs to serve as the source of the K service requests and select two other PoCs to be the c-PoCs with

available computing resources for task execution. For simplicity, we assume the computing resources colocated with each c-PoC have the same computing capability of 10 GHz,[7] i.e., $\Delta_i = \Delta = 10$ GHz, $\forall i \in \mathcal{V}$. The rate requirement of each service request is randomly drawn from the interval $[0.5, 1]$ Mbps according to a uniform distribution and the processing power required by each service is within $[1, 2]$ GHz [18], [41]. The transmission range and the interference range are 150 and 250 m, respectively. For illustrative purpose, the same set of spectrum bands is assumed to be available at each PoC and all bands are assumed to have the same bandwidth of w MHz as well as the spectral efficiency of 1 bit/s/Hz. Due to the uncertain activities of licensed/unlicensed users, for each band, the ratio of the available bandwidth to the total bandwidth follows a truncated exponential distribution on interval $[0, 1]$ as shown in [42] with $\lambda = 2$. The confidence level for the chance constraint α is set to 0.8. All the evaluations are conducted using MATLAB on a laptop with 2.6 GHz Intel(R) Core(TM) i5-3320M CPU and 8 GB RAM.

Algorithm 2. Enhanced Coarse-Grained Fixing Procedure

Input: The parameters of **OPT1**.
Output: A solution to **OPT1**.
1: Solve **OPT1** via the basic coarse-grained fixing procedure to obtain a solution x_1 and the corresponding value of the objective function ϑ_1;
2: Obtain a revised version of **OPT1** by reducing the bandwidth of each harvested band and updating the corresponding constraints in **OPT1**;
3: Relax y_{ij}^m's and η_{ij}^τ's to the interval $[0, 1]$ and collect y_{ij}^m's not yet fixed in a set φ';
4: Solve the relaxed version of the revised optimization problem and collect y_{ij}^m's with values larger than 0.5 in a set ψ';
5: Repeat the procedure from Line 3 to Line 8 of Algorithm 1 with φ replaced by φ' and ψ replaced by ψ';
6: **if** $\varphi' = \emptyset$ **then**
7: Repeat the procedure in Line 14 of Algorithm 1 and obtain a solution x_2 and the corresponding value of the objective function ϑ_2;
8: Go to Line 13;
9: **else**
10: Reformulate **OPT1** with the fixed y_{ij}^m's;
11: Repeat the procedure from Line 1 to Line 14 of Algorithm 1 for the reformulated problem and obtain a solution x_2 and the corresponding value of the objective function ϑ_2;
12: **end if**
13: **return** $\vartheta = \max\{\vartheta_1, \vartheta_2\}$ and $x_{\arg\max\{\vartheta_1, \vartheta_2\}}$;

6.1 Results and Analysis

Based on above settings, we compare the performance of the enhanced coarse-grained fixing procedure with that of the optimal solution and the basic coarse-grained fixing procedure. To proceed, we first solve **OPT1** via the enhanced coarse-grained fixing procedure to obtain the maximum number of supported services. Specifically, in step 2 of Algorithm 2, we reduce the bandwidth of each harvested band by setting it to $w - 1$. Then, the same problem is optimally solved via the branch and bound algorithm. Finally, we solve **OPT1** via the basic coarse-grained fixing procedure. The results are shown in Table 2, where M is the number of harvested bands. For simplicity, we assume the same set of harvested bands is available at each PoC. From Table 2, our enhanced coarse-grained fixing procedure (Algorithm 2) can achieve almost the same performance as that of the optimal solution and outperforms the basic coarse-grained fixing procedure (Algorithm 1), which demonstrates the effectiveness of the enhanced coarse-grained fixing procedure.

TABLE 2
Maximum Number of Supported Services

	$w(MHz)$ / Solution	2	3	4	5	6	7	8	9
$M = 4$	Optimal	4	5	6	6	6	6	7	8
	Algorithm 1	4	5	5	4	5	6	7	4
	Algorithm 2	4	5	6	6	5	6	7	8
$M = 5$	Optimal	5	6	6	7	7	8	8	8
	Algorithm 1	5	4	5	4	7	6	7	8
	Algorithm 2	5	6	5	7	7	8	7	8

To further evaluate the effectiveness of the enhanced coarse-grained fixing procedure, we have conducted experiments with $N = 20$ PoCs and $K = 20$ computing service requests. To incorporate the case where the computing resources colocated with different c-PoCs might have different computing capabilities, Δ_i ($i \in \mathcal{V}$) is drawn from the interval $[10, 15]$ GHz. Specifically, we randomly generate 10 network topologies. For each of these topologies, 20 PoCs are randomly deployed in a 500×500 m^2 area to form a connected network and the source of the service requests and the c-PoCs with computing resources are randomly chosen from these 20 PoCs. The other parameter settings are the same as those introduced at the beginning of this section. When $N = 20$, it is difficult to optimally solve the formulated MILP and obtain the corresponding maximum number of supported services in reasonable time. Thus, we relax the 0-1 integer variables in the formulated MILP to $[0, 1]$ and solve the relaxed problem to obtain an upperbound of the maximum number of supported services as a benchmark to evaluate the performance of the enhanced coarse-grained fixing procedure. For each of the 10 topologies, we solve the relaxed problem to obtain an upperbound and solve the formulated MILP with Algorithm 2. After that, we take the average value over these 10 topologies as the final results shown in Fig. 3. As shown in Fig. 3, the maximum number of services supported under the enhanced coarse-grained fixing procedure is close to the upperbound. Noticing that the maximum number of services supported under the enhanced coarse-grained fixing procedure serves as a lower bound of that supported under the optimal solution, the results in Fig. 3 demonstrate the effectiveness of the enhanced coarse-grained fixing procedure. Thus, in the following, we will study the impact of various parameters on service placement based on the enhanced coarse-grained fixing procedure.

In Fig. 4, we evaluate how the availability of communication resources affects the maximum number of supported services, namely, ϑ. The parameter settings are the same as those introduced at the beginning of this section. It can be observed from Fig. 4 that ϑ increases with either the number

7. This computing capability could come from a multi-core CPU. For example, IntelCore2 Quad Q6600 processor has a CPU speed of 4×2.4 GHz $= 9.6$ GHz ≈ 10 GHz [40].

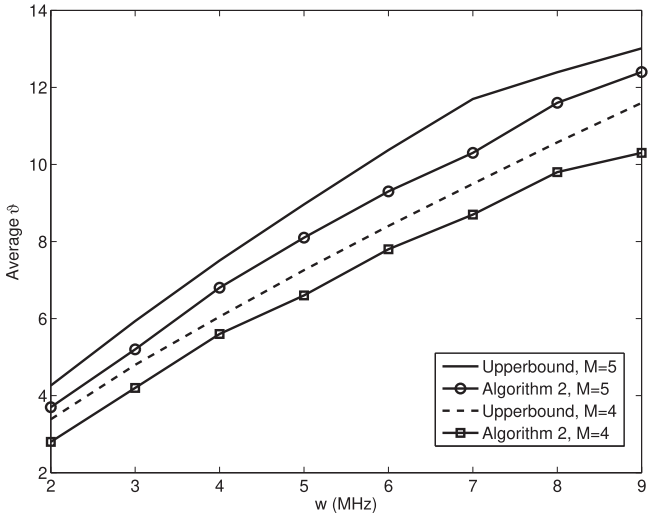


Fig. 3. The average number of supported services under the proposed algorithm versus the upperbound.

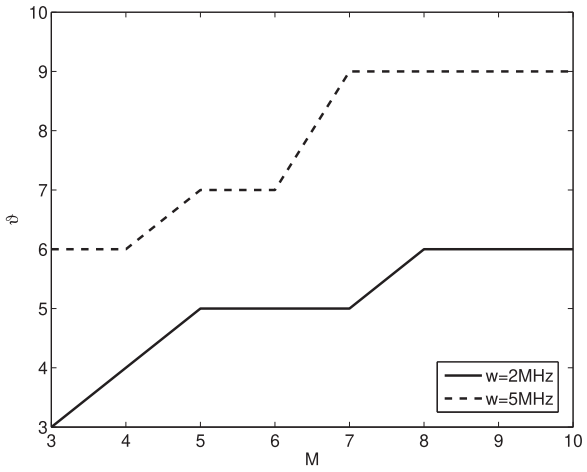


Fig. 4. The maximum number of supported services versus the number of harvested bands. ϑ is the maximum number of supported services.

of harvested bands M or the bandwidth of each harvested band. In both cases, the SSP has more communication resources to handle the service requests so that their rate requirements are satisfied, which leads to the increase in ϑ. This result implies that, by exploiting PoCs for spectrum harvesting, the SSP can potentially support more edge computing services.

As aforementioned, the SSP should jointly consider the communication resources, represented by the number of harvested bands M, and computing resources, represented by Δ, when making service placement decisions. In view of this, we study how the maximum number of supported services varies with the availability of communication and computing resources in Fig. 5. The parameter settings are the same as Fig. 4, and the only difference is that w is set to 5 MHz. From Fig. 5, more services can be supported when more communication and computing resources are available, which is consistent with the design goal of the placement algorithm. It can be observed from Fig. 5 that, when Δ is small, the number of harvested bands M does not have significant impact on the maximum number of supported services ϑ. In this case, ϑ is limited by the availability of computing resources instead of

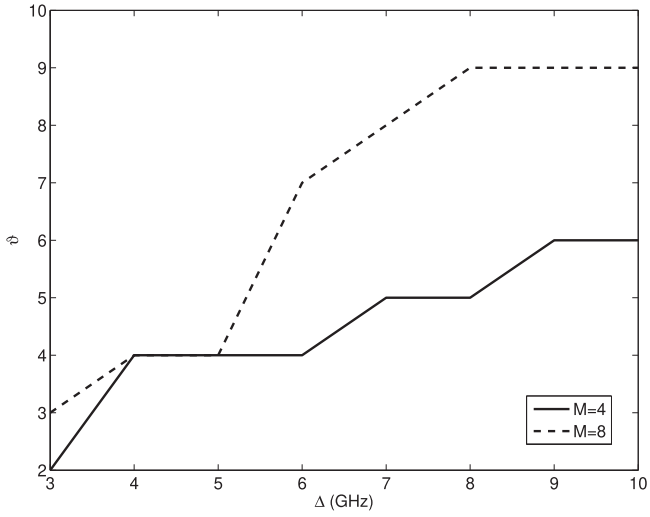


Fig. 5. The impact of spectrum and computing resources on the maximum number of supported services. ϑ is the maximum number of supported services.

communications resources, and thus an increase in M does not make much difference. This observation can also be demonstrated from the fact that, once Δ is large enough, there is a considerable rise in ϑ when M increases from 4 to 8. A large Δ implies that sufficient computing resources are available for edge computing services and ϑ is limited by the available communication resources. When compared with the case where $M = 4$, with $M = 8$, the SSP has more communication resources to facilitate efficient utilization of the available computing resources. This is the reason why, with Δ increasing, a more significant increase in ϑ can be observed when $M = 8$. After Δ reaches a certain value, for example, $\Delta = 8$ for the case where $M = 8$, the SSP cannot accommodate more services due to the lack of communication resources and ϑ stops further increasing as shown in Fig. 5. The results in Fig. 5 indicates that the maximum number of supported services depends on the availability of both the spectrum resources and computing resources. This observation will be further demonstrated in Fig. 8.

In Fig. 6, we investigate how the confidence level for the chance constraints, α, affects the maximum number of supported services ϑ. The parameter settings are the same as those introduced at the beginning of this section other than $M = 8$. Intuitively, α reflects how the SSP utilizes the harvested bands. A larger α implies that the SSP exploits the harvested bands in a more conservative way, and a smaller α means the SSP exploits the harvested bands more aggressively. Thus, it is not surprising that, under the same circumstance, the SSP can accommodate more computing services with a smaller α, as shown in Fig. 6. However, in this case, the SSP needs to handle the PUs' uncertain activities during the service provisioning process with a higher probability, as shown in Fig. 7. The probability of overestimation in Fig. 7 is the probability that the achievable data rate over at least one link has been overestimated. According to Section 4, the SSP estimates the achievable data rate of each link based on (11) and α to facilitate service placement decision making. In other words, the SSP considers the data rate to be achievable over a link as long as the constraints in (10) are satisfied with probability α. Thus, $1 - \alpha$ is

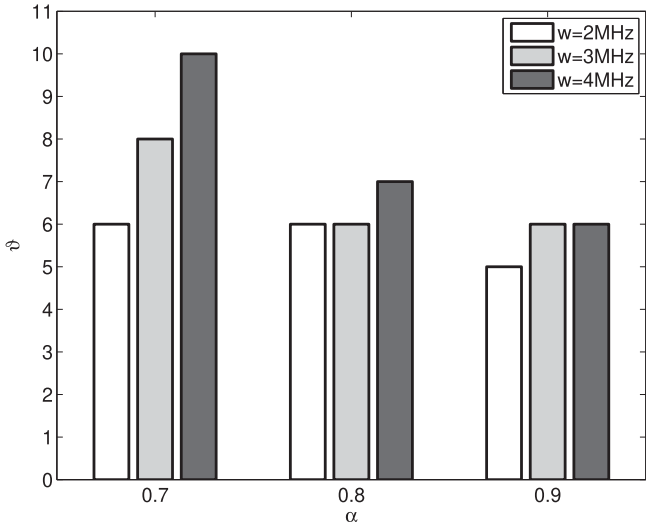


Fig. 6. The maximum number of supported services versus the confidence level for the chance constraints.

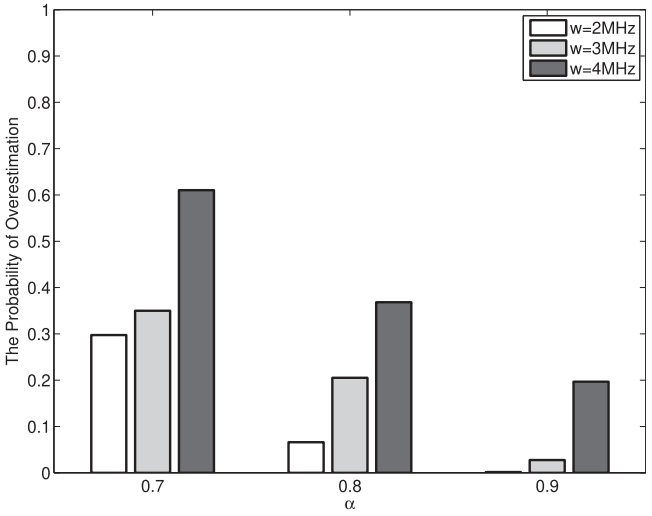


Fig. 7. The probability of overestimation under different values of α.

the probability that the SSP overestimates the data rate of the corresponding link, which is actually achievable during the service provisioning process. With a smaller α, the SSP will overestimate the achievable data rate of each link with a higher probability and thus will need to harvest extra spectrum resources and adjust data routing during the service provisioning process with a higher probability. From Fig. 7, the probability of overestimation decreases as the bandwidth of the harvested band, w, decreases. Notice that the achievable data rate of a link is closely related to the equivalent bandwidth of each harvested band, w_{ij}^m. As mentioned in Section 3.3, w_{ij}^m equals the product of the bandwidth w and the ratio of the available bandwidth to the entire bandwidth, and the randomness of w_{ij}^m is resulted from the ratio of the available bandwidth to the entire bandwidth. Given an estimate of the ratio of the available bandwidth to the entire bandwidth, a larger w will result in a larger discrepancy between the estimated w_{ij}^m and its actual value, which explains the results in Fig. 7. In practice, the SSP should carefully choose the value of α in order to efficiently utilize the harvested bands. Since how to determine the value of α is out of the scope of this paper, we will address this problem elsewhere and regard α as a known constant in this paper.

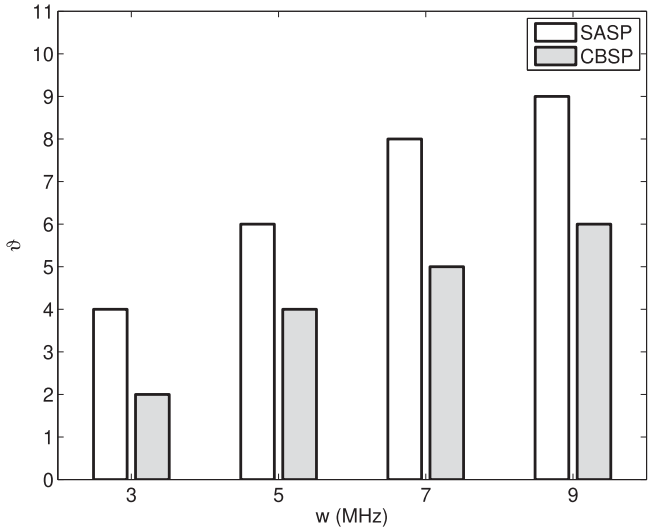


Fig. 8. The performance of the SASP scheme versus the performance of the CBSP scheme. ϑ is the maximum number of supported services.

In Fig. 8, we investigate the effectiveness of our SASP scheme by comparing its performance with a computation based service placement (CBSP) scheme. The CBSP scheme is an adapted version of the service placement scheme introduced in [30], where the edge computing services are assigned to different nodes based on the computing resources requested by each service and that available at each node. In Fig. 8, we compare the performance of our SASP scheme with that of the CBSP scheme. To obtain the maximum number of supported services under the CBSP scheme, we first determine the placement of the services through the CBSP scheme and then allocate spectrum resources to maximize the number of services which can be supported. The parameter settings are the same as those introduced at the beginning of this section, and the only differences are that the computing resources colocated with PoCs have a computing capability of 15 GHz and the number of harvested bands is $M = 5$. To investigate the importance of simultaneously considering communication and computing resources for service placement, we consider the case where the 4th and the 5th bands are not available around the PoC which is randomly picked from the two c-PoCs with colocated computing resources. From Fig. 8, the performance of our SASP scheme outperforms that of the CBSP scheme, which demonstrates the effectiveness of our SASP scheme as well as the importance of the joint consideration of communication and computing resources for service placement. Unlike the CBSP scheme, our SASP scheme explicitly takes spectrum availability information into consideration when making service placement decisions. Thus, it could avoid placing too many services to the same PoC where the available spectrum resources is not enough for input data delivery, which explains the superiority of our SASP scheme over the CBSP scheme.

7 Conclusion

In this paper, we design a network architecture, DART, based on interconnected PoCs to facilitate IoT applications. To exploit the benefits of DART, we study an edge computing

service placement problem and design a spectrum-aware service placement scheme through a mixed integer linear programming. Through simulation results, we demonstrate that available spectrum resources and computing resources should be jointly considered to achieve optimal performance and the obtained spectrum-aware placement strategy can facilitate effective service placement. Through DART, we advocate a network-level approach to jointly managing communication, computing, and storage resources to provide services for resource-limited IoT devices such that resource-constrained IoT devices could gain access to the desired communication and computing services. We hope this work will inspire more research efforts into the network-level orchestration of in-network resources to more effectively address various challenges in IoT applications.

Acknowledgments

This work was partially supported by the US National Science Foundation under grants CNS-1717736 and CNS-1409797. The work of X. Li was supported by the National Natural Science Foundation of China (NSFC) under Grant 61801080, and by the Fundamental Research Funds of Dalian University of Technology under Grant DUT18RC(3)012.

References

[1] A. Nordrum, "Popular internet of things forecast of 50 billion devices by 2020 is outdated," 2016. [Online]. Available: https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

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

[3] H. Ding, Y. Fang, X. Huang, M. Pan, P. Li, and S. Glisic, "Cognitive capacity harvesting networks: Architectural evolution towards future cognitive radio networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1902–1923, Jul.–Sep. 2017.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Jun. 2016.

[5] H. Ding, C. Zhang, Y. Cai, and Y. Fang, "Smart cities on wheels: A newly emerging vehicular cognitive capability harvesting network for data transportation," *IEEE Wireless Commun. Mag.*, vol. 25, no. 2, pp. 160–169, Apr. 2018.

[6] H. Ding and Y. Fang, "Virtual infrastructure at traffic lights: Vehicular temporary storage assisted data transportation at signalized intersections," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12452–12456, Dec. 2018.

[7] H. Ding, X. Li, Y. Cai, B. Lorenzo, and Y. Fang, "Intelligent data transportation in smart cities: A spectrum-aware approach," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2598–2611, Dec. 2018.

[8] 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.

[9] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, Jan.–Mar. 2018.

[10] B. Amento, B. Balasubramanian, R. J. Hall, K. Joshi, G. Jung, and K. H. Purdy, "FocusStack: Orchestrating edge clouds using location-based focus of attention," in *Proc. IEEE/ACM Symp. Edge Comput.*, Oct. 2016, pp. 179–191.

[11] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing - a key technology towards 5G," ETSI White Paper, no. 11, Sep. 2015.

[12] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 38–43, Mar. 2017.

[13] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *Comput.*, vol. 50, no. 10, pp. 58–67, Oct. 2017.

[14] W. A. Aljoby, T. Z. Fu, and R. T. Ma, "Impacts of task placement and bandwidth allocation on stream analytics," in *Proc. IEEE 25th Int. Conf. Netw. Protocols*, Oct. 2017, pp. 1–6.

[15] S. Yang, "IoT stream processing and analytics in the fog," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 21–27, Aug. 2017.

[16] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos, "Edge analytics in the internet of things," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 24–31, Apr.–Jun. 2015.

[17] P. Ta-Shma, A. Akbar, G. Gerson-Golan, G. Hadash, F. Carrez, and K. Moessner, "An ingestion and analytics architecture for IoT applied to smart city use cases," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 765–774, Apr. 2018.

[18] T. Zhang, A. Chowdhery, P. V. Bahl, K. Jamieson, and S. Banerjee, "The design and implementation of a wireless video surveillance system," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2015, pp. 426–438.

[19] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 207–215.

[20] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman, "Live video analytics at scale with approximation and delay-tolerance," in *Proc. USENIX Conf. Netw. Syst. Des. Implementation*, Mar. 2017, pp. 377–392.

[21] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2003, pp. 66–80.

[22] X. Li, H. Ding, M. Pan, Y. Sun, and Y. Fang, "Users first: Service-oriented spectrum auction with a two-tier framework support," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 2999–3013, Nov. 2016.

[23] H. Ding, C. Zhang, X. Li, J. Liu, M. Pan, Y. Fang, S. Chen, Y. Fang, C. Zhang, M. Pan, et al., "Session-based cooperation in cognitive radio networks: A network-level approach," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 685–698, Apr. 2018.

[24] G. Lewis, S. Echeverría, S. Simanta, B. Bradshaw, and J. Root, "Tactical cloudlets: Moving cloud computing to the edge," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2014, pp. 1440–1446.

[25] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, A. Kapoor, S. N. Sinha, M. Sudarshan, and S. Stratman, "FarmBeats: An IoT platform for data-driven agriculture," in *Proc. USENIX Conf. Netw. Syst. Des. Implementation*, Mar. 2017, pp. 515–529.

[26] S. Roberts, P. Garnett, and R. Chandra, "Connecting Africa using the TV white spaces: From research to real world deployments," in *Proc. IEEE Int. Workshop Local Metropolitan Area Netw.*, Apr. 2015, pp. 1–6.

[27] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-aware resource allocation for edge computing," in *Proc. IEEE Int. Conf. Edge Comput.*, Jun. 2017, pp. 47–54.

[28] Y.-J. Yu, T.-C. Chiu, A.-C. Pang, M.-F. Chen, and J. Liu, "Virtual machine placement for backhaul traffic minimization in fog radio access networks," in *Proc. IEEE Int. Conf. Commun.*, May 2017.

[29] M. Taneja and A. Davy, "Resource aware placement of data stream analytics operators on fog infrastructure for internet of things applications," in *Proc. IEEE/ACM Symp. Edge Comput.*, Oct. 2016, pp. 113–114.

[30] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1440–1452, May 2016.

[31] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost-efficient resource management in fog computing supported medical CPS," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 108–119, Jan.–Mar. 2017.

[32] L. Chen and J. Xu, "Collaborative service caching for edge computing in dense small cell networks," 2017. [Online]. Available: https://arxiv.org/pdf/1709.08662.pdf

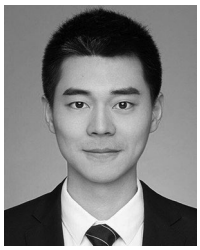
[33] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types," in *Proc. USENIX Conf. Netw. Syst. Des. Implementation*, Mar. 2011, pp. 323–336.

[34] W. Wang, B. Liang, and B. Li, "Multi-resource fair allocation in heterogeneous cloud computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 10, pp. 2822–2835, Oct. 2015.

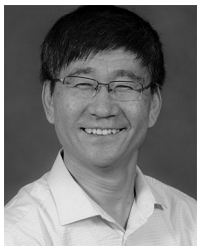
[35] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct.–Dec. 2009.
[36] A. Lertsinsrubtavee, A. Ali, C. Molina-Jimenez, A. Sathiaseelan, and J. Crowcroft, "PiCasso: A lightweight edge computing platform," in *Proc. IEEE 6th Int. Conf. Cloud Netw.*, Sep. 2017, pp. 1–7.
[37] Fortinet, "Understanding ip surveillance camera bandwidth," White Paper, May 2017.
[38] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Oct.–Dec. 2017.
[39] M. Pan, C. Zhang, P. Li, and Y. Fang, "Spectrum harvesting and sharing in multi-hop CRNs under uncertain spectrum supply," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 369–378, Feb. 2012.
[40] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: At the leading edge of mobile-cloud convergence," in *Proc. 6th Int. Conf. Mobile Comput. Appl. Services*, Nov. 2014, pp. 1–9.
[41] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, and H. Balakrishnan, "Glimpse: Continuous, real-time object recognition on mobile devices," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2015, pp. 155–168.
[42] H. Yue, M. Pan, Y. Fang, and S. Glisic, "Spectrum and energy efficient relay station placement in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 5, pp. 883–893, May 2013.

Haichuan Ding received the BEng and MS degrees in electrical engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 2011 and 2014, respectively. He is currently working toward the PhD degree at the University of Florida. From 2012 to 2014, he was with the Department of Electrical and Computer Engineering, University of Macau, as a visiting student. His current research is focused on cognitive radio networks, vehicular networks, and security and privacy in distributed systems.

Yuanxiong Guo (M'14) received the BEng degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2009, and the MS and PhD degrees in electrical and computer engineering from the University of Florida, Gainesville, Florida, in 2012 and 2014, respectively. Since 2014, he has been an assistant professor with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, Oklahoma. His current research interests include cybersecurity, data analytics, and resource management for networked systems including Internet of things, cyber-physical systems, and cloud/edge systems. He is a recipient of the Best Paper Award in the IEEE Global Communications Conference 2011. He has been serving as an editor of the *IEEE Transactions on Vehicular Technology* since January 2018. He is a member of the IEEE.



Xuanheng Li (S'13) received the BS degree in electronic and information engineering from the Dalian University of Technology, in 2012, and the PhD degree in communication and information systems from the Dalian University of Technology, in 2017. From October 2015 to October 2017, he was with the Wireless Networks Laboratory (WINET), University of Florida, as a visiting student. Since 2018, he has worked as a lecturer with the School of Information and Communication Engineering, Dalian University of Technology. He received the Best Paper Award at IEEE GLOBECOM 2015. His research interests include cognitive radio networks, IoT, spectrum sharing, interference alignment, and underwater communications. He is a member of the IEEE.



Yuguang Fang (F'08) received the MS degree from Qufu Normal University, Shandong, China, in 1987, the PhD degree from Case Western Reserve University, in 1994, and the PhD degree from Boston University, in 1997. He joined the Department of Electrical and Computer Engineering, University of Florida, in 2000, and has been a full professor since 2005. He held a University of Florida research foundation professorship (2006-2009), a Changjiang Scholar chair professorship with Xidian University, China (2008-2011) and with Dalian Maritime University (2015-present), and a guest chair professorship with Tsinghua University, China, from (2009-2012). He was the editor-in-chief of the *IEEE Transactions on Vehicular Technology* (2013-2017), and the *IEEE Wireless Communications* (2009-2012). He is a fellow of the IEEE and AAAS.