Noname manuscript No.
(will be inserted by the editor)

SOSW: *S*calable and *O*ptimal Nearsighted Location Selection for Fog Node Deployment and Routing in *S*DN-based *W*ireless Networks for IoT Systems

Muhammad Ibrar · Lei Wang ·
Gabriel-Miro Muntean · Nadir Shah ·
Aamir Akbar · Khalid Ibrahim Qureshi

Received: date / Accepted: date

Abstract In a fog computing (FC) architecture, cloud services migrate towards the network edge and operate via edge devices such as access points (AP), routers, and switches. These devices become part of a virtualization infrastructure and are referred to as "fog nodes". Recently, software-defined networking (SDN) has been used in FC to improve its control and manageability. The current SDN-based FC literature has overlooked two issues: *(a)* fog nodes' deployment at optimal locations and *(b)* SDN best path computation for data flows based on constraints (i.e., end-to-end delay and link utilization). To solve these optimization problems, this paper suggests a novel approach, called scalable and optimal near-sighted location selection for fog node deployment and routing in SDN-based wireless networks for IoT systems *(SOSW)*. First, the SOSW model uses singular-value decomposition (SVD) and QR factorization with column pivoting linear algebra methods on the traffic matrix of the network to compute the optimal locations for fog nodes, and second, it introduces a new heuristic-based traffic engineering algorithm, called the constraint-based shortest path algorithm (CSPA), which uses ant colony optimization (ACO) to optimize the path computation process for task offloading. The results show that our proposed approach significantly reduces average latency and energy consumption in comparison with existing approaches.

Keywords Wireless Network · IoT · Fog Computing · SDN · Optimization.

M. Ibrar *(Corresponding author)* and K. I. Qureshi
School of Software, Dalian University of Technology, China.
E-mail: mibrar@mail.dlut.edu.cn and khalidibrahim84@gmail.com

L. Wang *(Corresponding author)*
School of Software, Dalian University of Technology; Key Lab of Ubiquitous Network and Service Software of Liaoning Province; and Peng Cheng Laboratory, China.
E-mail: lei.wang@dlut.edu.cn

G-M. Muntean
School of Electronic Engineering, Dublin City University, Ireland.
E-mail: gabriel.muntean@dcu.ie

N. Shah
Department of Computer Science, COMSATS University Islamabad, Wah Campus, Pakistan.
E-mail: nadirshah82@gmail.com

A. Akbar
Department of Computer Science, Abdul Wali Khan University Mardan (AWKUM), Pakistan.
E-mail: amirakbar@awkum.edu.pk

1 Introduction

The fog computing (FC) architecture migrates services (i.e., computing, processing, and storage) from the cloud to the network edge or devices located near end users by implementing these services at the routers, switches, WiFi access points (APs), smart devices, etc. The devices that host these services are named fog nodes [15,27]. The purpose of FC is to minimize the average end-to-end delay for time-critical applications and reduce the core network's bandwidth usage. FC has been used for the Internet of Things (IoT) to support time-critical applications (i.e., smart health and smart cities) [19], which demand substantial computing resources for real-time processing, leading to high energy consumption on resource-constrained devices[1]. For instance, IoT devices offload tasks to fog nodes for processing. Then, fog nodes make the resource-constrained IoT devices virtually limitless and provide better services in terms of latency. Nevertheless, ubiquitous fog node deployment is still unrealistic in the near future because of cost issues [27,38]. Accordingly, in this work, we consider a scenario in which IoT devices offload tasks to fog nodes via multihop wireless networks.

1.1 Motivation

Fog nodes are the primary FC units and can include any network device that uses dedicated servers, processing capabilities, and computational servers to coordinate underlying IoT devices. An FC architecture usually consists of several levels of fog nodes. The processing of a given task may be adequate for a specific layer due to its requirements, like scalability, security, latency, and mobility. The number of layers and the position of nodes depend on the FC architecture. R. Vilata *et al.* proposed an architecture in which fog nodes are placed in the nearest base stations (BSs) for *5G* applications [33]. Kem *et al.* described an FC architecture [17], where end users provide household fog nodes and receive incentives for serving nodes. More recent work [10,12,16,17,23,24,27,33,38] has studied the influence of fog nodes connected to other network devices in an FC architecture. However, the abovementioned authors have not addressed the impact of fog node deployment on different physical locations. Existing works randomly deploy the fog nodes and result in low network performance; importantly, these works do not consider fog node deployment costs.

To address this problem, we intend to deploy the fog nodes in optimal physical locations, minimizing the average end-to-end delay from IoT devices to fog nodes [**?**]. The singular-value decomposition (SVD) and QR factorization with column pivoting linear algebra methods [13] are used to select the optimal locations of the fog nodes because the traffic matrix is a sparse matrix [9]. Additionally, the placement of fog nodes in optimal locations in a network minimizes both the energy consumption and average end-to-end delay of IoT devices during task offloading.

In the FC-IoT architecture, the existing literature has mainly focused on the energy consumption and computational power of IoT devices during task offloading. However, it has overlooked network dynamics and their related performance, such as delay and link utilization, when computing the path from IoT devices to fog nodes. For reliable packet delivery, the IoT uses application layer protocols (i.e., CoAP [26] and MQTT-SN [29]), which results in congestion and extra energy consumption. Therefore, there is a need to consider network performance parameters when addressing task offloading.

As a potential solution, the software-defined networking (SDN) architecture has been proposed for FC-IoT applications [1, 28]. With the help of a controller, SDN obtains an

[1] This is true in terms of storage, energy, and computing

abstract global view of the network, achieves centralized control of the dynamic network conditions, and supports rule-based flow forwarding. Furthermore, the programmable SDN architecture also supports network management simplification, network flexibility, and the management and orchestration of the FC-IoT architecture [14,30], which, in turn, makes the SDN architecture attractive for addressing the task offloading issues in the dynamic wireless FC-IoT architecture [1, 24]. The SDN controller can use network management abstraction and obtain information about heterogeneous wireless devices [6, 36], as well as employ global view information to compute the optimal path for task offloading.

1.2 Contributions

In a successful wireless FC-IoT architecture, the number of fog nodes should be less than the number of APs to limit operating expenses (OPEXs) and capital expenditures (CAPEXs) [24]. Unlike existing work, we consider a multihop FC-IoT paradigm, where the IoT device tasks traverse through multiple intermediate AP nodes to reach the fog node for processing.

This paper proposes a scalable and optimal nearsighted location selection approach for fog node deployment and routing in SDN-based wireless networks for IoT systems *(SOSW)*. The SOSW model solves two problems: fog node deployment and the computation of the best path between the device and fog nodes, based on existing network constraints.

This contributions of this paper are as follows:

- **The deployment of fog nodes** in wireless FC-IoT architecture in optimal locations is an optimization problem. Therefore, the SOSW model proposes a traffic-monitoring scheme using SVD and QR factorization with column-pivoting techniques of linear algebra. This scheme accurately identifies important APs based on traffic load for fog node deployment. Following this identification, we connect the minimal number of fog nodes with the most important APs. The intended solution aims to reduce the end-to-end delay and energy consumption in SDN-based FC-IoT.
- The SOSW model introduces a new heuristic algorithm called the **constraint-based shortest path algorithm (CSPA)** for task offloading in SDN-based FC-IoT. As the CSPA problem is *NP-complete*, we use an ant colony optimization (ACO) algorithm as a solution. The CSPA considers dynamic network conditions like end-to-end delay, link utilization, and energy consumption.

The remainder of this paper is structured as follows. The related work is reviewed in Section 2. Section 3 presents the proposed solution related to deployment and routing in the SDN-based FC-IoT architecture. A performance evaluation is included in Section 4, and conclusions are drawn in Section 5.

2 Related Works

The cost-effective deployment of fog nodes in SDN-based wireless networks is crucial for satisfying delay-sensitive tasks in an FC architecture for IoT systems. Many approaches in the literature have highlighted the concerns regarding fog node deployment for real-time applications in FC. These approaches offer relevant solutions for the deployment of fog nodes, and some of them will be briefly described and discussed below. Fog node location issues in SDN-based wireless networks for FC-IoT have been mostly studied for static fog nodes. However, little attention has been paid to optimal physical location selection solutions for these fog nodes. The primary goal of these solutions is to improve the method of providing services to fog nodes with reduced cost in SDN-based wireless networks for FC-IoT.

In general, fog nodes have heterogeneous space and time characteristics. The deployment of fog nodes at optimal locations is a prior provision and is also expected to provide high-performance services to IoT devices. For instance, Wang *et al.* proposed a fog node deployment solution based on space-time characteristics, called the TSBP [34]. The main objectives of this proposed TSBP model are balancing the load among fog nodes and minimizing the response time by introducing an innovative discrete differential evolution algorithm. The fog smart gateway (FSG) [22] model was proposed to improve the quality of services (QoS), i.e., to minimize service latency. In the FSG approach, the above authors consider the gateway nodes to be fog nodes; the gateway nodes collect the data from the end devices but do not have decision-making and processing capabilities. The proposed model makes the gateway node smarter, as it distributes the tasks of IoT devices among various virtual machines. Through VM processing and traffic aggregation, the FSG model decreases the number of service delays. A three-layer architecture (i.e., thing-fog-cloud) [7] is proposed to minimize the transmission delay and average data rate. The three-layer architecture shows that the performance of a network can be optimized if it increases the number of fog nodes in the fog layer.

Le *et al.* proposed a solution to deploy fog nodes in the local area network (LAN) to optimize the objective parameters, i.e., the response time of services and network data [18]. The authors of [16, 20, 21, 35] deployed cloudlets at optimal locations to reduce the access latency between edge users and cloudlets in large-scale wireless metropolitan area networks. The authors of [21] used particle swarm optimization and heuristic algorithms to address fog node deployment issues. To minimize the number cloudlets, the authors of [16] proposed two algorithms: cloudlet placement based on density and cloudlet placement based on traffic load. Although these two algorithms increase the network performance in terms of load balancing and network latency, the proposed model does not achieve the optimal solution. Sudip *et al.* proposed a heuristic algorithm [24] to control the dynamic task offloading problem in SDN Fog-IoT. The proposed model computes the task locally or on fog nodes to minimize the energy consumption of IoT devices. Additionally, the proposed model also computes the optimal fog node and path for IoT task offloading. However, the above authors deploy fog nodes randomly in the network. Moreover, the authors in [2, 4] proposed a solution to optimize the tradeoff between energy consumption and network usage in mobile-to-fog/cloud-code offloading. Furthermore, they also proposed a self-aware decision-making mechanism to minimize the network latency from a mobile device perspective [3, 5]. However, their code-offloading frameworks did not take into account the QoS parameter, which is to find the best fog node for different types of traffic.

The authors of [27] aimed to address the issue of locating fog nodes to reduce latency and cost. The proposed solution was based on a mixed-integer linear programming formulation concerning multiple workload criteria. These workload criteria were divided into strict and flexible demands. The performance evaluation of the proposed solution showed improved performance in terms of providing services to fog nodes at a reduced cost. Online seRvice caching for mobile edge cOmputing (OREO) [31] was proposed to jointly optimize the caching and task offloading problem in a dense cellular network. Yu *et al.* used a fully polynomial-time approximation (FPA) [38] scheme to study joint data routing and placement with both delay and bandwidth guarantees. However, the proposal did not consider the optimal location for fog node deployment. Another related work also addressed the fog node location problem [10] while considering human mobility and location problems. In [16], demand variables over time were employed to obtain end-user movement. Various traffic load types were also considered—an issue related to the communication between cloud and fog nodes. However, these approaches considered network dynamics such as traffic, delay and link utilization in SDN-based wireless networks for IoT systems.

3 *SOSW*: *S*calable and *O*ptimal Nearsighted Location Selection for Fog Node Deployment and Routing in *S*DN-based *W*ireless Networks for IoT Systems

This section describes the proposed *SOSW* approach in the following two steps:

1) the deployment of fog nodes at an optimal location and

2) the computation of the optimal path-based new heuristic constraint-based shortest path algorithm (CSPA).

Assume that we have a wireless network represented by an undirected graph G (V ∪ F, E), as shown in Fig. 1. In graph G, V describes the set of APs (i.e., V = $\{v_1, v_2, ..., v_n\}$), F = $\left\{f_i, f_j ... f_k\right\}$ shows the set of fog nodes, and E represents the set of links between nodes (i.e., $e_i(v_i, v_j)$), such that v_i, $v_j \in V$, and $v_i \neq v_j$.

3.1 Fog Node Deployment

As the traffic matrix of the network is sparse [9], computing the important APs for fog node deployment can utilize SVD and QR factorization with the column-pivoting method [13].

We assume that the wireless network runs for a long time (i.e., days or weeks) and has a network traffic log. From this traffic log, a matrix, κ, is used to represent the number of flows passing through each AP at different time intervals, i.e., $\kappa \in \mathbb{R}^{t \times s}$. t indicates the total number of rows of κ in sequential timer interval t_i, while s is the total number of columns of κ (i.e., the total number of APs). More precisely, in matrix κ, row r_i denotes a snapshot of the total number of flows transitory through all V at t_i, and column c_j represents the number of flows transitory through an AP ($v_j \in V$) at t_i. Additionally, it is assumed that $t \gg s$. Subsequently, following the obtaining of traffic matrix κ, we compute important APs.

To compute important APs in the network and check their spatial correlation property, first, we perform the SVD technique on traffic matrix κ. The SVD decomposes traffic matrix κ into three matrices, as shown in Eq. (1).

$$\kappa_{t \times s} \; = \; \mathscr{L}_{t \times t} \, \mathsf{C}_{t \times s} \, \mathscr{R}_{s \times s}, \tag{1}$$

where Eq. (1) shows that $\mathscr{L} \in \mathbb{R}^{t \times t}$ and $\mathscr{L}^T \mathscr{L} = I$. $\mathscr{R} \in \mathbb{R}^{s \times s}$ and $\mathscr{R}^T \mathscr{R} = I$, and C is a $t \times s$ diagonal matrix. The left singular matrix is denoted as $\mathscr{L}$, and the columns of $\mathscr{L}$ are known as left singular vectors, represented by $\{\ell_i \mid i \in \{1, 2, 3, ..., t\}\}$, and are orthogonal to each other. The right singular matrix is represented by $\mathscr{R}$, and the columns of $\mathscr{R}$ are known as right singular vectors, represented by $\{\varkappa_i \mid i \in \{1, 2, 3, ..., s\}\}$. These values are also orthogonal to each other. C is a $t \times s$ diagonal matrix, the diagonal values (a.k.a. singular values) of which in descending order are indicated by $\vartheta_i \geq \vartheta_{i+1} \geq 0$ (i.e., $\{\vartheta_i \mid i \in \{1, 2, 3, ..., t\}\}$). These singular values in traffic matrix κ represent the importance level/value of an AP. The most important property of SVD is the relation shown in Eq. (2):

$$\kappa_{t \times s} \; = \; \vartheta_1 \ell_1 \varkappa_1^T + \vartheta_2 \ell_2 \varkappa_2^T + ... + \vartheta_s \ell_s \varkappa_s^T \, . \tag{2}$$

In Eq. (2), note that every singular value may be either zero or positive (i.e., $\vartheta_i \geq 0$). Assume that among all s singular values, there is a range of $\mathfrak{p}$ positive singular values. $\mathfrak{p}$ is referred to as the rank of matrix κ because Eq. (2) shows that each column of matrix κ is equal to a linear combination of $\mathfrak{p}$ left singular vectors $\{\ell_i \mid i \in \{1, 2, 3, ..., \mathfrak{p}\}\}$, and every one of these left singular vectors ℓ_i has a positive singular value, called its coefficient. Every column in matrix κ is a linear combination of the $\mathfrak{p}$ primary vectors, which means that there

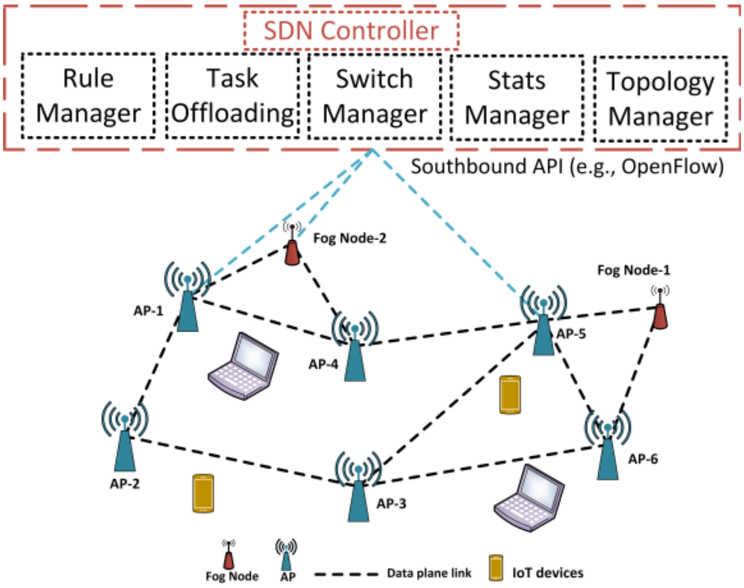

Fig. 1: SDN-based Wireless FC-IoT Network.

are $\mathfrak{p}$ left singular vectors $\{\ell_i \mid \vartheta_i > 0; i \in \{1,2,3,...,\mathfrak{p}\}\}$ from a set of primary orthogonal vectors spanning the column space of matrix κ. More precisely, $\mathfrak{p}$ primary vectors can be recovered using exactly the $\mathfrak{p}$ column of matrix κ. In this paper, we denote these $\mathfrak{p}$ columns in matrix κ as its primary columns. Consequently, by considering matrix $\kappa \in \mathbb{R}^{t\times s}$ of rank $\mathfrak{p}$, we can compute its $\mathfrak{p}$ primary columns, and the $\mathfrak{p}$ primary columns indicate the $\mathfrak{p}$ primary APs in our network. Therefore, the traffic information of every AP in the network can be jointly represented by a linear combination of the traffic information of these $\mathfrak{p}$ primary APs. Additionally, we may completely and accurately represent the matrix $\kappa \in \mathbb{R}^{t\times s}$ using primary $\mathfrak{q}$ columns. The traffic information passed through $\mathfrak{p}$ APs may be useful for correctly and entirely recovering the global traffic matrix by collecting the traffic information passing through the rank $\mathfrak{p}$ of traffic matrix $\kappa \in \mathbb{R}^{t\times s}$.

However, traffic matrix $\kappa \in \mathbb{R}^{t\times s}$ often has a lower rank of $\mathfrak{q}$. This lower rank $\mathfrak{q}$ is referred to as a useful rank. The lower rank $\mathfrak{q}$ can be much smaller than the rank $\mathfrak{p}$ (i.e., $\mathfrak{q} \ll \mathfrak{p}$), which shows that the set of singular values is distributed sparsely. In other words, traffic matrix $\kappa \in \mathbb{R}^{t\times s}$ has more small values $\mathfrak{q}$ and only a few large values $\mathfrak{p}$. For example, Eq. (2) shows that the traffic information contained in κ is disseminated into s terms on the right-hand side of the equation. Each term $\vartheta_i \ell_i \varkappa_i^T$ is the product of both the matrix, $\ell_i \varkappa_i^T$, and a coefficient, ϑ_i.

For instance, first, we consider $\mathfrak{q}$ large singular values and that the remaining singular values are small (i.e., $\mathfrak{q} \ll \mathfrak{p}$) in matrix κ, which means that the first large singular values of $\mathfrak{q}$ represent a significant portion of matrix κ's information (see Eq. (2)). This theory suggests that only the first-largest singular values of $\mathfrak{q}$ (i.e., $\mathfrak{q}$ primary vectors) can represent the entire matrix κ. Considering these $\mathfrak{q}$ primary columns only significantly decrease the computation time compared to those recorded when $\mathfrak{p}$ primary columns or all columns of matrix κ are considered, it is essential to find $\mathfrak{q}$ primary columns (i.e., $\mathfrak{q} \ll \mathfrak{p}$) of matrix κ such that they can be used to approximately represent matrix κ. This problem is NP-complete and is a subset selection problem in linear algebra [13].

Unfortunately, there are no algorithms to give an optimal solution to subset selection problems. However, our proposed method uses the estimated solution proposed in [13] to compute a subset of $\mathfrak{q}$ columns of matrix κ that can represent the space formed by its first $\mathfrak{q}$ primary vectors $\{\ell_i \mid i \in \{1,2,3,...,\mathfrak{q}\}\}$. Now, in Eq. (2), the right-hand side represents the $\mathfrak{p}$ primary vectors and the primary vectors ℓ_i in the first $\mathfrak{q}$ terms. In matrix κ, the APs associated with $\mathfrak{q}$ basic columns are denoted as important (or target) APs. Furthermore, the solution for finding the $\mathfrak{p}$ basic columns in the subset selection problem is described next.

Problem-1 ***Consider given matrix*** $\kappa \in \mathbb{R}^{t\times s}$ ***and integer*** $\mathfrak{q}$***. The subset selection problem is to find matrix*** $\mathscr{H}$***, which is a permutation matrix of*** κ***,*** $\kappa\mathscr{H} = (\kappa_1, \kappa_2)$***, where*** κ_1 ***indicates a*** $t \times \mathfrak{q}$ ***matrix, and*** κ_2 ***represents a*** $t \times (s - \mathfrak{q})$ ***matrix. The main objective is to maximize the information contained in*** κ_1***. Additionally, the goal is to enable all columns to be adequately independent in matrix*** κ_1***.***

To maximize the smallest singular value of κ_1, we have to find the adequately independent columns of κ_1. For this purpose, we first apply SVD on $\kappa \in \mathbb{R}^{t\times s}$, as shown in Eq. (2), then multiply $\mathscr{H}$ (permutation matrix) by $\mathscr{R}$ (the right singular matrix), and obtain Eq. (3). In Eq. (3), $\mathscr{R}_{11} \in \mathbb{R}^{q\times q}$. Assuming that $\mathscr{R}_{11}$ is a nonsingular value, then we have the following lower bound on the smallest singular value $\vartheta_q(\kappa_1)$ of κ_1, as shown in Eq. (4):

$$\mathscr{H}\mathscr{R} = \begin{pmatrix} \mathscr{R}_{11} & \mathscr{R}_{12} \\ \mathscr{R}_{21} & \mathscr{R}_{22} \end{pmatrix}, \tag{3}$$

$$\vartheta_q(\kappa_1) \geq \left(\frac{\vartheta_q(\kappa_1)}{\left\|\mathscr{R}_{11}^{-1}\right\|} \right). \tag{4}$$

In Eq. (4), $\vartheta_q(\kappa_1)$ shows the lowermost ($\mathfrak{q}^{th}$) singular value of matrix κ_1. The maximization of the lowermost bound value of $\vartheta_q(\kappa_1)$ corresponds to the maximization of the value of $\vartheta_q(\kappa_1)$. In Eq. (4), this is equivalent to minimizing $\left\|\mathscr{R}_{11}^{-1}\right\|$. To handle this issue, we used QR factorization with column pivoting [13] on matrix κ. The overall selection process of important APs is described in Algorithm 1. After the computation process, we connect the fog nodes to the important APs and obtain the required configuration.

Algorithm 1: Searching for Important APs

```
Input: traffic matrix $\kappa \in \mathbb{R}^{t\times s}$ and an integer $d$
Result: $\kappa\mathscr{H} = (\kappa_1, \kappa_2)$, where $\kappa_1$ indicates $t \times d$ matrix
Compute the SVD of $\kappa$, where $\kappa = \mathscr{L}\mathfrak{C}\mathscr{R}^T$
Use QR factorization with column pivoting to traffic matrix $\kappa$ using the following equation:
    $\kappa = QR\mathscr{H}^T$, where $Q$, $R$, and $\mathscr{H}$ are matrix, and the permutation matrix is $\mathscr{H}$
Permute all columns of $\kappa$ (i.e., by multiplying $\kappa$ with $\mathscr{H}$. Afterwards, the first $d$ columns of the
    matrix $\kappa$ form $\kappa_1$:
    $\kappa\mathscr{H} = (\kappa_1, \kappa_2)$, where $\kappa_1$ indicates a $t \times d$ matrix and $\kappa_2$ represents a $t \times (s-d)$ matrix.
```

3.2 Constraint-based Shortest Path Algorithm (CSPA)

In the FC-IoT architecture, the paramount constraints are end-to-end delay and bandwidth, which means that any computed path between the IoT device and a fog node must meet these constraints. The cost of the path must be less than or equal to a predefined acceptable threshold. To ensure appropriate bandwidth and end-to-end delay for IoT applications in

FC-IoT, the SDN controller computes the CSPA for IoT applications in this proposed work. The notations are summarized in Table 1.

Table 1: Summary of Notations

Notation	Explanation
G	network graph
V	set of APs, v_i, $u_i \in$ V
E	set of links, $e_i(v_i,u_i)$, $e_i \in$ E
D_{IoT}	set of IoT Device
F	set of flows, $f_i \in$F
c_{e_i}	delay cost function is of a link e_i
d_{e_i}	delay of link e_i
λ	delay of path $\lambda > 0$
u_{e_i}	utilization ratio of link e_i, $u_{e_i} = \frac{\alpha_{e_i}}{B_{e_i}}$, α_{e_i} traffic value, B_{e_i} total bandwidth of e_i

The computation of the CSPA follows a linear programming (LP) approach. The CSPA's cost function is defined to minimize the path cost function from the IoT device to the fog node. In the proposed model, the controller computes the shortest path with the lowest cost. The CSPA model is illustrated as follows:

$$\min \sum_{e_i \in E} c_{e_i} e_i, \quad \forall e_i \in E,\ e_i(v_i,u_i),\ v_i,u_i \in V \tag{5}$$

$$\sum_{u_i|(v_i,u_i)\in E} e_i(v_i,u_i) - \sum_{u_i|(v_i,u_i)\in E} e_i(u_i,v_i) = \begin{cases} 1\ v_i \in D_{IoT} \\ \text{-1}\ v_i \in f_i \\ 0\ otherwise \end{cases}, \tag{6}$$

$$\sum_{e_i \in E} e_i * (-d_{e_i}) - \lambda = -\alpha,\ \ \lambda > 0 \tag{7}$$

$$e_i(u_i,v_i) = 1\ or\ 0.\ \ \forall e_i(u_i,v_i) \in E \tag{8}$$

In the proposed model, the minimization delay cost function is c_{e_i} of link e_i, where $e_i = (u_i,v_i) \in E$, as shown in Eq. 5. The CSPA constraint (see Eq. 6) introduces the *flow-conservation constraints*, ensuring that all tasks are served by the fog node only. Constraint (Eq. 7) shows that the end-to-end delay of the computed path should be less than or equal to the predefined threshold value α. The additional variable λ indicates that each path has a delay, which is greater than 0 (i.e., $\lambda > 0$), and that it should be less than or equal to the predefined threshold value. Consequently, the constraint in Eq. 8 converts the proposed model into an integer linear model. $e_i(u_i,v_i) = 1$ means that a link exists in the path, and $e_i(u_i,v_i) = 0$ indicates that a link is not present in the path.

Similar to other communication networks, in a fog-IoT architecture, link congestion is a factor that dramatically affects time-critical applications in terms of end-to-end delay and packet loss [14]. Therefore, to minimize the end-to-end delay and loss for time-critical applications in FC-IoT, the SDN controller takes into account delay and link utilization parameters while computing the path for task offloading. This is the reason why in the proposed work, the CSPA also considers link utilization along with link delay. The CSPA computes the cost of each link as follows:

$$e_i = \delta * u_{e_i} + (1-\gamma) * d_{e_i}, \tag{9}$$

where u and d are the utilization ratio $u_{e_i} = \frac{\alpha_{e_i}}{B_{e_i}}$ and delay of link e_i, respectively. δ and γ are additional parameters that reflect the importance of delay or utilization. For the sake of simplicity, in the proposed model, we consider utilization and delay to be of equal importance.

In this work, we consider an FC-IoT topology in which the IoT devices are multiple hops away from the fog node. In other words, IoT devices can interact with the fog node through AP connectivity support. In such a multihop topology, the cost function delay can be computed as follows:

$$d = \sum_{task} Tx_d + Pr_d + Pro_d + \sum_{task} Qu_d, \tag{10}$$

where Tx_d is the transmitting time in which the IoT device transmits the data to the associated AP, and Pro_d represents the propagation delay between any fog node and the IoT device. Processing delay (task execution time) is indicated by Pro_d, and Qu_d is the queuing delay at the fog node.

In the proposed model, we consider the *log-distance* path loss model with log-normal shadowing [31], as shown in Eq. 11. The maximum data rate between the IoT device and associated AP is shown in Eq. 12:

$$path - loss_{[dB]} = 140.7 + 36.7log_{10}d_{km} + N(8), \tag{11}$$

$$data - rate_{IoT-AP} = Blog_2(1 + \frac{Pw_{IoT}^{Tx} - Path - loss_{IoT-AP}}{\eta^2}), \tag{12}$$

where Pw_{IoT}^{Tx} is the transmitting power of the IoT device, and η^2 shows the noise power. Unfortunately, the CSPA cannot compute the optimal path with reasonable resources in an extensive network because it is an optimization *NP-complete* problem [24]. In the proposed work, for optimization purposes, we use an ACO algorithm to solve the CSPA problem, which is described below.

3.3 Explanation and Complexity Analysis of CSPA-based ACO Algorithm

The proposed CSPA-based ACO algorithm, detailed in Algorithm 2, starts by receiving network information about the cost function of the links, IoT devices, fog nodes, and max delay (λ) of IoT applications. In each iteration, the algorithm finds a path between the IoT device and a fog node using Eq. 6. The algorithm checks if each path satisfies the maximum delay value of IoT applications, and the pheromone value is updated. Next, the SDN controller compares the computed path with the global solution and performs the following action. If the computed path cost is less than the global solution, then the global solution is replaced with the computed path. Finally, the algorithm returns the minimum cost path (mcp). The proposed algorithm guarantees that this mcp is the optimal path cost. Algorithm 2 consists of statements, loops, and a while loop to compute the minimum cost path (mcp) from the available paths for a given flow f_i. The statements in Algorithm 2 have constant complexity $\mathcal{O}(c)$, and the NetworkX function has $\mathcal{O}(n+c)$ complexity when computing all the available paths for a flow f_i, as shown in Step 4. Consequently, the complexity of Steps 5 and 8 is $\mathcal{O}(n^2)$ because it involves loops. Moreover, the worst-case complexity of Algorithm 2 is $\mathcal{O}((c)+(n+c)+(n^2)) \approx \mathcal{O}(n^2)$.

Algorithm 2: CSPA: ALGORITHM BASED ON ANT COLONY OPTIMIZATION (ACO)

```
Input: G(V, E), $f_i \in$ F, IoT-device (s∈S), path-cost, $\alpha$, $\lambda$
Output: Compute minimum cost path
// constant complexity time $\mathcal{O}(c)$
pop ← initializePheromoneValues() // $\mathcal{O}(c)$
mcp = ∅ /*min-path-cost*/// $\mathcal{O}(c)$
while $s \in S$ do // $\mathcal{O}(n)$
    paths ← $Networkx.all\_simple\_paths(G, s, f_i)$// $\mathcal{O}(n+c)$
    for $f_i \in F$ do // $\mathcal{O}(n^2)$
        compute a path between IoT device (s) and Fog node $(f_i)$
    end
    for $f_i \in F$ do // $\mathcal{O}(n^2)$
        if path satisfies Max-delay $(\lambda)$ then // $\mathcal{O}(c)$
            update pop
        if path-cost < mcp then // $\mathcal{O}(c)$
            mcp = path-cost /*Global Solution*/
    end
end
return mcp
```

4 Experimental Work and Results

4.1 Simulation Settings

To evaluate the performance of the proposed approach, we used the Mininet[2] network emulator and the POX[3] SDN controller. In our experiments, we employed an Intel core i7 PC with a 3.40 GHz CPU and 12 GB of RAM, running Ubuntu 16.04 OS with Linux kernel version 4.4. Table 2 shows a summary of the remaining simulation parameters. In this work, we have used a *Barabasi-Albert* [8] topology for creating the network in Mininet. The APs and fog nodes are placed in a small area (i.e., 1,000 × 1,000). We distributed the IoT devices uniformly within the convergence area.

We have compared our proposed approach with two baseline models: (a) the random offloading random path (RORP) model and (b) the delay-aware greedy path (DGP) [11] model. The RORP model chooses the path randomly between the fog node and IoT device. Additionally, in the RORP model, the fog nodes are deployed randomly. For task offloading, the RORP model selects the closest fog node, i.e., based on the number of hop counts. For the DGP model, the abovementioned authors suggested an efficient offloading algorithm to minimize the delay while considering the energy constraints of end-user devices. For task offloading, the DGP model considers the number of hops between the IoT and fog devices. We consider a comparison between the DGP and RORP models to demonstrate the need for intelligent fog node deployment and appropriate path selection in an SDN-based wireless FC-IoT architecture and to assess the influence of parameters other than delay, i.e., multihop path-based delay and link utilization as well as the efficient deployment of fog nodes in the network in SDN-based wireless FC-IoT. In contrast to these models, our proposed solution deploys the fog nodes at optimal locations in the network, and the path is computed between the pair of IoTs and the fog node based on minimum link utilization and delay.

[2] http://mininet.org

[3] http://github.com/noxrepo/pox

Table 2: Simulation Parameters

Parameters	Value
Number of fog nodes	1-15
APs	40
Tasks	100-1000
Task size (average)	450 KB [31]
Transmitting power of IoT	60 mW [37]
Battery power of IoT	1,000 J [11]
Bandwidth of wireless channel	20 MHz [31]
Noise power	-100 dB [31]
Computation amount for task	1,500-2,500 megacycles [31]

4.2 Results and Discussion

Following the above simulations, we compared our proposed model with the baseline models using the following parameters.

4.2.1 Impact of Optimal Location Selection for Fog Nodes

Fog node placement in a wireless network at an optimal location significantly impacts network performance. Therefore, to demonstrate the impact of optimal location selection for fog nodes, we examined two parameters: average latency (s) and average load by altering the number of fog nodes. Fig. 2 shows that the average latency decreases as the number of fog nodes in the network increases. In the proposed model, the average latency is very low compared to those experienced in the DGP and RORP models. It can be observed that in the proposed model, the average latency (s) almost becomes normal when the number of fog nodes increases from 10 to 15. The reason behind the lowest average latency of the SOSW model is that it installs the fog nodes in the optimal location in the wireless domain for IoT applications. From the results, we can see that the selection of the optimal location for the fog nodes provides efficient services in terms of the delay constraint.

When the number of fog nodes increases from 10 to 15, the average delay when employing the proposed SOSW model is less than 2 seconds, further improving system performance, as expected. The delays experienced by the DGP and RORP models are higher than 3 seconds, even when the number of fog nodes reaches 15. However, according to the OPEX and CAPEX limitations, the deployment of more fog nodes is costly, and it is desirable to achieve good performance with a limited number of such nodes. The SOSW model achieves the lowest latency, as it considers delay and link utilization when it computes the path between the IoT device and fog node. Given the OPEX and CAPEX constraints, in the remaining simulations, we limit our study to 10 fog nodes providing services to IoT devices. Fig. 3 shows the average traffic load (%) on the fog nodes. From the results, we can see that the average traffic load on the fog nodes decreases when the number of fog nodes increases in all models. However, in the proposed SOSW model, the traffic load is lower than that of the DGP and RORP models when the number of fog nodes increases from *6* to *10*. Additionally, the traffic load is balanced in the proposed model because the SOSW model accurately estimates the optimal location for the fog nodes in the FC-IoT architecture. The results show that the proposed SOSW model is 20% and 25% more effective in terms of load balancing compared to the DGP and RORP models, respectively.

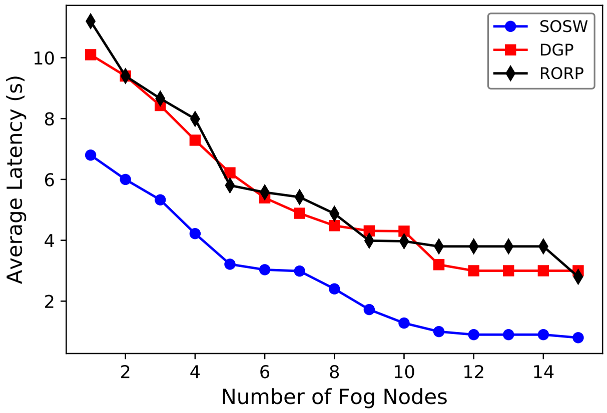

Fig. 2: Average Latency and Number of Fog Nodes

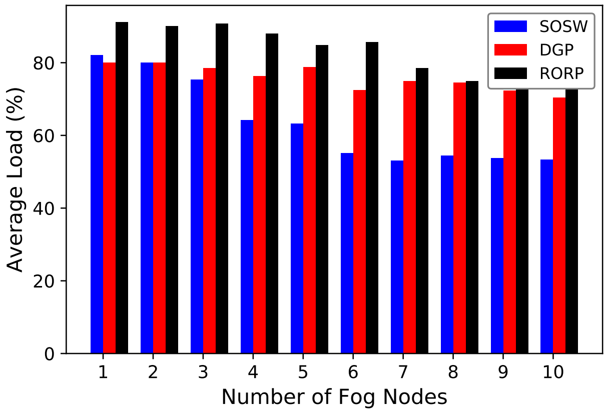

Fig. 3: Average Traffic Load and Number of Fog Nodes

4.2.2 Impact of Number of Flows on Average End-to-end Delay

Following the simulations, we examined the average end-to-end delay (s) w.r.t. the number of tasks. The results illustrated in Fig. 4 show that the proposed model performs better than the baseline RORP and DGP models. From the figure, we can conclude that when the number of tasks increases, the average end-to-end delay also increases in all models. It is obvious that more tasks exist, and the load in a network also increases in terms of bandwidth utilization and load on the fog nodes. Therefore, the average end-to-end delay increases as the congestion in the network increases. However, the figure shows that the average end-to-end delay in the proposed model is much lower than that experienced by the baseline models, which select the path based on hop count and fog nodes deployed randomly. In the proposed model, the fog nodes are deployed at an optimal location, which results in decreases in the average delayes of IoT applications. Additionally, the proposed model also considers link utilization as a paramount parameter in path computation. Notably, the baseline models do not consider the utilization parameter, so they record further delays due to retransmissions and congestion. From the results, it can be seen that the proposed SOSW model provides service to IoT devices with lower delays when the number of tasks/flows reaches 1,000. For instance, for 1,000 flows, the average delay is less than 50 ms. However, when the DGP and RORP models are employed, the average delay computed is greater than 60 ms and 70 ms, respectively. These results show that the SOSW-based optimal deployment of fog nodes and CSP-based path computation significantly improve the overall performance of the SDN-based wireless FC-IoT architecture.

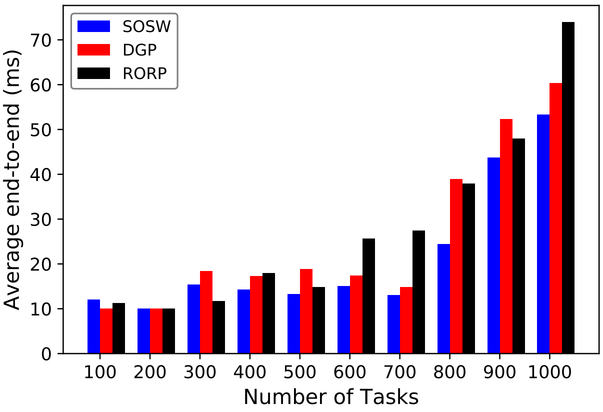

Fig. 4: Number of Tasks vs. Average End-to-end Delay

4.2.3 *Impact of Number of Flows on Average Energy Consumption*

As previously mentioned, IoT devices are resource-constrained in terms of processing, storage, and battery power. Therefore, in the proposed model, we also analyse average energy consumption. Using the Provisioning of Wireless Access Points (CAPWAP), Simple Network Management Protocol (SNMP) [25] or OpenNetMon [32], the SDN controller collects information about the data rate and transmission power. In this work, we use OpenNetMon. The energy required for transmitting the data is computed as $Tx_{data}^{energy} = Tx_{IoT} \times \Gamma_{AP}^{t}$, where Γ_{AP}^{t} shows the time taken to transmit the data to the associated AP.

The results show that the proposed model's performance is better than that experienced by the baseline models in terms of average energy consumption, as shown in Fig. 5. From the results, we can see that energy consumption increases as the number of tasks increases in all models. However, the proposed model performs relatively better than the other models. The DGP and RORP models consume more energy than the proposed model because they consider the shortest path, without considering link utilization. Therefore, the congestion path and retransmissions determine extra energy consumption. Additionally, the proposed SOSW model deploys the fog nodes at an optimal location for IoT devices, which leads to the minimization of the energy consumption of IoT devices. Furthermore, the CSPA module in the SDN controller computes the optimal path by taking into account the end-to-end delay

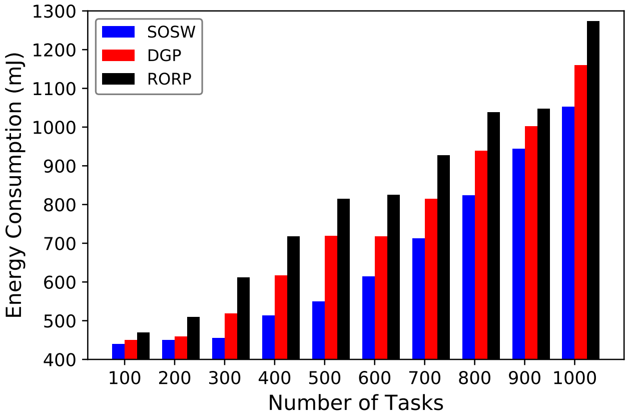

Fig. 5: Number of Tasks vs. Energy Consumption

and utilization ratio of the links. The link utilization ratio minimizes the congestion problem in the FC-IoT architecture. In the SOSW model, energy consumption is less than 1,100 *mJ*; in the DPG model, energy consumption is almost 1,200 *mJ*; and in the RORP model, energy consumption is almost 1,300 *mJ*.

When analysing the above results, it can be concluded that the proposed model decreases the average end-to-end delay and energy consumption because it places the fog node at an optimal location for task offloading. Additionally, the proposed SOSW solution computes the path from the IoT device to the fog node by utilizing the proposed CSPA-based ACO algorithm, which is also beneficial.

5 Conclusions and Future Research Directions

This paper considers two important parameters for optimizing the SDN-based wireless network FC-IoT's performance: end-to-end delay and link utilization. It proposes an innovative solution that, first, places the fog nodes in an optimal location to serve IoT devices and, second, computes the optimal IoT-fog device path for IoT applications based on given resource constraints using the ACO algorithm. The results show that fog nodes' optimal place in the network increases the performance in terms of delay and load balancing. Additionally, the computation of an optimal path based on end-to-end delay, energy consumption, and link utilization significantly decreases average end-to-end delay and energy consumption. In this paper, we have focused on static IoT devices. In future work, we will consider a more realistic topology, including both mobile and fixed IoT devices.

Acknowledgements The work was supported by the "National Natural Science Foundation of China" grant no. 61902052, "National Key Research and Development Plan" grant no. 2017YFC0821003-2, "Science and Technology Major Industrial Project of Liaoning Province" grant no. 2020JH1/10100013, "Dalian Science and Technology Innovation Fund" grants no. 2019J11CY004 and 2020JJ26GX037, Science Foundation Ireland Research Centres Programme grants no. 12/RC/2289_P2 (Insight) and 16/SP/3804 (ENABLE) and "Fundamental Research Funds for the Central Universities" grants no. DUT20ZD210 and DUT20TD107.

References

1. Akbar, A., Ibrar, M., Jan, M.A., Bashir, A.K., Wang, L.: SDN-enabled Adaptive and Reliable Communication in IoT-Fog Environment using Machine Learning and Multi-Objective Optimization. IEEE Internet of Things Journal pp. 1–1 (2020)
2. Akbar, A., Lewis, P.R.: Towards the optimization of power and bandwidth consumption in mobile-cloud hybrid applications. In: 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), pp. 213–218 (2017). DOI 10.1109/FMEC.2017.7946433
3. Akbar, A., Lewis, P.R.: Self-adaptive and self-aware mobile-cloud hybrid robotics. In: 2018 Fifth International Conference on Internet of Things: Systems, Management and Security, pp. 262–267 (2018). DOI 10.1109/IoTSMS.2018.8554735
4. Akbar, A., Lewis, P.R.: The importance of granularity in multiobjective optimization of mobile cloud hybrid applications. Transactions on Emerging Telecommunications Technologies **30**(8), e3526 (2019). DOI https://doi.org/10.1002/ett.3526
5. Akbar, A., Lewis, P.R., Wanner, E.: A self-aware and scalable solution for efficient mobile-cloud hybrid robotics. Frontiers in Robotics and AI **7**, 102 (2020). DOI 10.3389/frobt.2020.00102
6. Anadiotis, A.C.G., Galluccio, L., Milardo, S., Morabito, G., Palazzo, S.: Towards a software-defined network operating system for the iot. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), pp. 579–584. IEEE (2015)
7. Balevi, E., Gitlin, R.D.: Optimizing the number of fog nodes for cloud-fog-thing networks. IEEE Access **6**, 11173–11183 (2018)
8. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. science **286**(5439), 509–512 (1999)
9. Bell, N., Garland, M.: Efficient sparse matrix-vector multiplication on cuda. Tech. rep., Nvidia Technical Report NVR-2008-004, Nvidia Corporation (2008)
10. Chen, L., Liu, L., Fan, X., Li, J., Wang, C., Pan, G., Jakubowicz, J., et al.: Complementary base station clustering for cost-effective and energy-efficient cloud-ran. In: SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, pp. 1–7. IEEE (2017)
11. Chen, M., Hao, Y.: Task offloading for mobile edge computing in software defined ultra-dense network. IEEE Journal on Selected Areas in Communications **36**(3), 587–597 (2018)
12. Consortium, O., et al.: Openfog reference architecture. Retrieved January (2017)
13. Golub, G.H., Van Loan, C.F.: Matrix computations, vol. 3. JHU press (2012)
14. Hakiri, A., Berthou, P., Gokhale, A., Abdellatif, S.: Publish/subscribe-enabled software defined networking for efficient and scalable iot communications. IEEE communications magazine **53**(9), 48–54 (2015)
15. Ibrar, M., Wang, L., Muntean, G.M., Chen, J., Shah, N., Akbar, A.: IHSF: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid SDN-based FC IoT systems. IEEE Internet of Things Journal pp. 1–1 (2020)
16. Jia, M., Cao, J., Liang, W.: Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. IEEE Transactions on Cloud Computing **5**(4), 725–737 (2015)
17. Kim, W.S., Chung, S.H.: User-participatory fog computing architecture and its management schemes for improving feasibility. IEEE Access **6**, 20262–20278 (2018)
18. Lee, J.H., Chung, S.H., Kim, W.S.: Fog server deployment considering network topology and flow state in local area networks. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 652–657. IEEE (2017)
19. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., Zhao, W.: A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet of Things Journal **4**(5), 1125–1142 (2017). DOI 10.1109/JIOT.2017.2683200
20. Ma, L., Wu, J., Chen, L.: Dota: Delay bounded optimal cloudlet deployment and user association in wmans. In: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 196–203. IEEE (2017)
21. Ma, L., Wu, J., Chen, L., Liu, Z.: Fast algorithms for capacitated cloudlet placements. In: 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 439–444. IEEE (2017)
22. Maiti, P., Shukla, J., Sahoo, B., Turuk, A.K.: Qos-aware fog nodes placement. In: 2018 4th International Conference on Recent Advances in Information Technology (RAIT), pp. 1–6. IEEE (2018)
23. Marín-Tordera, E., Masip-Bruin, X., García-Almiñana, J., Jukan, A., Ren, G.J., Zhu, J.: Do we all really know what a fog node is? current trends towards an open definition. Computer Communications **109**, 117–130 (2017)
24. Misra, S., Saha, N.: Detour: dynamic task offloading in software-defined fog for iot applications. IEEE Journal on Selected Areas in Communications **37**(5), 1159–1166 (2019)

25. Qureshi, K.I., Wang, L., Sun, L., Zhu, C., Shu, L.: A review on design and implementation of software-defined wlans. IEEE Systems Journal (2020)
26. Shelby, Z., Hartke, K., Bormann, C.: The constrained application protocol (coap) (2014)
27. da Silva, R.A., da Fonseca, N.L.: On the location of fog nodes in fog-cloud infrastructures. Sensors (Basel, Switzerland) **19**(11) (2019)
28. Sood, K., Yu, S., Xiang, Y.: Software-defined wireless networking opportunities and challenges for internet-of-things: A review. IEEE Internet of Things Journal **3**(4), 453–463 (2015)
29. Stanford-Clark, A., Truong, H.L.: Mqtt for sensor networks (mqtt-sn) protocol specification. International business machines (IBM) Corporation version **1**(2) (2013)
30. Tomovic, S., Yoshigoe, K., Maljevic, I., Radusinovic, I.: Software-defined fog network architecture for iot. Wireless Personal Communications **92**(1), 181–196 (2017)
31. Tran, T.X., Pompili, D.: Joint task offloading and resource allocation for multi-server mobile-edge computing networks. IEEE Transactions on Vehicular Technology **68**(1), 856–868 (2018)
32. Van Adrichem, N.L., Doerr, C., Kuipers, F.A.: Opennetmon: Network monitoring in openflow software-defined networks. In: 2014 IEEE Network Operations and Management Symposium (NOMS), pp. 1–8. IEEE (2014)
33. Vilalta, R., López, V., Giorgetti, A., Peng, S., Orsini, V., Velasco, L., Serral-Gracia, R., Morris, D., De Fina, S., Cugini, F., et al.: Telcofog: A unified flexible fog and cloud computing architecture for 5g networks. IEEE Communications Magazine **55**(8), 36–43 (2017)
34. Wang, J., Li, D., Hu, M.Y.: Fog nodes deployment based on space-time characteristics in smart factory. IEEE Transactions on Industrial Informatics (2020)
35. Wang, S., Zhao, Y., Xu, J., Yuan, J., Hsu, C.H.: Edge server placement in mobile edge computing. Journal of Parallel and Distributed Computing **127**, 160–168 (2019)
36. Yap, K.K., Kobayashi, M., Sherwood, R., Huang, T.Y., Chan, M., Handigol, N., McKeown, N.: Openroads: Empowering research in mobile networks. ACM SIGCOMM Computer Communication Review **40**(1), 125–126 (2010)
37. Yousefpour, A., Ishigaki, G., Gour, R., Jue, J.P.: On reducing iot service delay via fog offloading. IEEE Internet of Things Journal **5**(2), 998–1010 (2018)
38. Yu, R., Xue, G., Zhang, X.: Application provisioning in fog computing-enabled internet-of-things: A network perspective. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 783–791. IEEE (2018)