Optimal Service Provisioning for the Scalable Fog/Edge Computing Environment

Jonghwa Choi and Sanghyun Ahn *

Department of Computer Science and Engineering, University of Seoul, Seoul 02504, Korea; David13@uos.ac.kr
* Correspondence: ahn@uos.ac.kr; Tel.: +82-(0)2-6490-2446

Abstract: In recent years, we observed the proliferation of cloud data centers (CDCs) and the Internet of Things (IoT). Cloud computing based on CDCs has the drawback of unpredictable response times due to variant delays between service requestors (IoT devices and end devices) and CDCs. This deficiency of cloud computing is especially problematic in providing IoT services with strict timing requirements and as a result, gives birth to fog/edge computing (FEC) whose responsiveness is achieved by placing service images near service requestors. In FEC, the computing nodes located close to service requestors are called fog/edge nodes (FENs). In addition, for an FEN to execute a specific service, it has to be provisioned with the corresponding service image. Most of the previous work on the service provisioning in the FEC environment deals with determining an appropriate FEN satisfying the requirements like delay, CPU and storage from the perspective of one or more service requests. In this paper, we determined how to optimally place service images in consideration of the pre-obtained service demands which may be collected during the prior time interval. The proposed FEC environment is scalable in the sense that the resources of FENs are effectively utilized thanks to the optimal provisioning of services on FENs. We propose two approaches to provision service images on FENs. In order to validate the performance of the proposed mechanisms, intensive simulations were carried out for various service demand scenarios.

Keywords: fog/edge computing; service provisioning; service placement; service offloading; Internet of Things (IoT)

1. Introduction

The Internet of Things (IoT) is a new wave significantly affecting our lives in various areas including smart energy grids, smart factories, smart cities, and smart farms [1]. IoT is accomplished by sensing (monitoring) the target environment, collecting and analyzing the sensory information and actuating based on the feedback information from the analysis. The data produced in the IoT environment may be enormous and require intensive computation such as for deep learning-based prediction. Hence, the IoT data tend to be transmitted to the remote cloud data center (CDC) for storage and computation [2–4]. Many IoT applications have stringent timing requirements which may not be met by cloud computing because of long-distance message transmissions through the network between CDC and end devices (aka IoT devices). For the reduced latency between CDC and end devices, the concept of fog computing has been introduced by Cisco [5]. In fog/edge computing (FEC), fog/edge nodes (FENs) located in the proximity of end devices provide services to them in lieu of CDC without causing a long delay by executing the lightweight virtualized service images pre-allocated from CDC [5–7]. However, due to the limited capacity (CPU, storage, etc.) of FENs, only a subset of service images can be placed on each FEN [8,9].

In the previous work on service provisioning (or placement) in FEC [10,11], a service is placed on the FEN satisfying the given service requirement for the execution of the service, assuming that the corresponding service image is already installed in the FEN.
Service provisioning, service placement and service offloading are interchangeably used for installing service images and executing them on FENs or CDC instead of end devices. However, due to the limited capability of a FEN, it is infeasible to place all the services on an FEN. Even in the case that FENs install service images on demand for offloaded services, the resource of FENs may not be efficiently utilized because of patchwork-like service image placement [12]. In addition, depending on the IoT applications, the type of services requested may vary. For example, in an area occupied by smart factories, the service analyzing the sensed data from a specific machine for the detection of malfunctions will be requested by the factories equipped with the machine. Placing service images on FENs in the per-service request-based manner may result in the improper resource usage of FENs due to redundantly placed service images. If we can optimally provision service images on FENs based on the pre-obtained service demands (service demands and service requests are interchangeably used in this paper), the resource utilization of FENs can be optimized. To our knowledge, this issue has not been addressed to date and thus, we formulate the problem of optimal service provisioning (SP) in the FEC environment. In our work, we adopted “service provisioning” instead of service placement in the sense that service provisioning means service placement with planning.

For the scalability of the FEC environment, FENs are located hierarchically near to end devices and in between end devices and CDC which is the last resort of providing services to end devices. The following two cases (c1) and (c2) are not desirable from the perspective of resource utilization and latency, respectively:

(c1) For a sporadically requested service, the corresponding service images are redundantly placed on multiple FENs.

(c2) For the coverage of multiple requests of a service, the corresponding service image is provisioned on a FEN at a high level of the FEN hierarchy.

If we consider both the resource usages and locations of FENs and the locations of end devices requesting services, we could avoid the above-mentioned cases. The consideration of only the case (c1) (i.e., the resource utilization of FENs) may result in the case (c2) (i.e., long delay), and vice versa. Therefore, in provisioning service images in the FEC environment, we must consider both the resource usages and locations of FENs and the delay requirements and locations of service requesters. In this paper, we propose two SP approaches by extending our previous work [12]; the first one is on the basis of the number of service requests from end devices and the second one is based on the locations of end devices requesting services. We can assert that the proposed FEC environment is scalable because more service images can be accommodated by FENs thanks to the optimal provisioning of service images on them. For the performance evaluation of the proposed mechanisms, we perform simulations for various service requesting scenarios and analyze the performance in terms of the total number of service images provisioned on FENs and the total number of service requests not accommodated by FENs.

The rest of the paper is organized as follows. In Section 2, we describe the related work on the service provisioning, offloading or placement on FENs. In Section 3, the problem of service provisioning FENs is formulated as a 0–1 integer linear programming problem for the proof of the NP-hardness of the problem and then, our two approaches are described in detail. The performance of our mechanisms is evaluated in Section 4 from the simulation results. Finally, Section 5 concludes this paper.

2. Related Work

Services can be offered to end devices via computation offloading in which CDC or FENs perform computation in lieu of end devices. SP on FENs is the pre-allotment of the resources, like CPU, RAM and storage, of FENs to service images for computation offloading [6–8]. For the sake of effective computation offloading, SP on FENs must be properly carried out beforehand.

In the FEC environment, the service provisioning, placement or service offloading has been studied by many researchers with different objectives and considerations, like
power consumption [13], quality of experience (QoE) [14–16], migration [17], network perspective [18], service atomization [19], etc. The authors of [19] proposed strategies for offloading service executions by considering the resources of CDC and FENs. For that, they introduced the concept of service atomization and parallel resource allocation. By service atomization, complex IoT services are divided into smaller atomic services which, then, can be executed on multiple FENs sequentially or parallelly in a distributed way. In their scheme, end devices offload service executions to CDC or FENs according to network and processing demands.

In [18], the application provisioning problem in the fog-cloud environment is studied from a network perspective to guarantee the quality of service (QoS) of application (service) data streams in terms of transmission delay and bandwidth for each application. Application provisioning is to find the host node (i.e., FEN) and data routing for a specific application. The authors formulate the issue as the single application provisioning (SAP) and the multi-application provisioning (MAP) problems. SAP determines the path to the FEN satisfying the bandwidth and delay requirements of data traffics from end devices for one application. MAP is the extension of SAP for multiple applications from the perspective of network link capacity (bandwidth) usage.

In [11], the deployment of multi-component application in the fog hierarchy is defined as the components deployment problem (CDP) which determines candidate FENs to be deployed with the components of an application according to application-specific QoS requirements of end devices on latency and bandwidth, software and hardware capabilities of FENs and business policies. They target to deploy large-scale applications composed of multiple components that can be independently deployable and work together for the infrastructure and prove that the CDP problem is NP-hard by the reduction from the subgraph isomorphism problem (SIP) [20]. The CDP problem is to find the optimal deployment of a single application (service) based on the service requests of end devices in a centralized manner.

Both [18] and [11] determine the FENs to be provisioned with services in a centralized way for a given set of service requests, similar to ours. However, theirs take account of each individual FEN, not of entire FENs, from the perspective of resource usage. The consideration on the capacity of each FEN is just for checking the relevant constraint, but the consideration on the total resource usage of all the FENs is for optimizing the overall resource usage of FENs. Thus, the latter is more appropriate for enhancing the scalability of the fog infrastructure in terms of resource usage.

Related to the resource provision of FENs, the authors of [14,15] proposed a FEC architecture composed of fog colonies each consisting of one fog orchestration control node and multiple fog cells. The fog orchestration control node is a fog cell with extended functionalities like managing fog cells in the same fog colony and other connected control nodes. The fog cell is the software components running on an FEN. The fog orchestration control node keeps all the service implementations (i.e., service images) in its service registry, located in the low-cost abundant storage unit, and deploys the corresponding service image to the fog cell covering the service requester on demand. In this way, the constrained resources of FENs are effectively utilized. The fog cell for a service request is determined based on the resources of FENs and the resource and delay requirements of the service request such that the resource requirement is satisfied with the minimum delay. In the scheme, all the service images are kept in the storage of the fog orchestration control node and deployed on FENs per service request by the control node. This will incur overwhelming communication overhead and delay in deploying service images on demand from fog orchestration control nodes to fog cells, and the requirement on the fog orchestration control node to be installed with all the service images is infeasible or, at least, inefficient if plenty of services are to be deployed in the fog.

In this paper, we aim to formulate and resolve the problem of minimizing the total number of service images provisioned on FENs in order to optimize the overall resource usage of FENs.
3. Service Provisioning for Fog/Edge Nodes

3.1. System Environment and Problem Description

For the simplicity of problem formalization, we assume that there is one CDC and all the service demands are accommodated by the FENs. Each service demand is assumed to require the same amount (e.g., one capacity unit) of resources even though service demands are heterogeneous in reality. The network link capacity is assumed to be sufficient for the delivery of the data generated by the service demands (i.e., we do not consider the network link capacity as a constraint in the problem).

We define the notations needed for the problem formalization in Table 1. $S$ is the set of the services, $\{s_1, \ldots, s_\beta\}$, and $F$ is the set of the FENs, $\{f_1, \ldots, f_\beta\}$, and $U$ is the set of the end devices, $\{u_1, \ldots, u_\gamma\}$, and $D$ is the service demand matrix, $[d_{ik}]$, where $d_{ik}$ is 1 if the service $s_i$ is demanded by the end device $u_k$.

Table 1. Notations for the definition of the optimal service provisioning (SP) problem.

| Notation | Description |
|----------|-------------|
| $S$      | The set of services; $S = \{s_1, \ldots, s_\beta\}$ |
| $F$      | The set of FENs; $F = \{f_1, \ldots, f_\beta\}$ |
| $U$      | The set of end devices; $U = \{u_1, \ldots, u_\gamma\}$ |
| $X$      | The service placement matrix; $X = [x_{ij}]$ |
| $x_{ij}$ | The binary variable indicating whether a service image of the service $s_i$ is provisioned in the FEN $f_j$ or not; $x_{ij} = \begin{cases} 1, & \text{if } s_i \text{ is provisioned on } f_j \\ 0, & \text{otherwise} \end{cases}$ |
| $D$      | The given service demand matrix consisting of the service demands, $d_{ik}$'s |
| $d_{ik}$ | The element of $D$ indicating whether the service $s_i$ is demanded by the end device $u_k$ |
| $c_j$    | The maximum capacity of the FEN $f_j$ |
| $T_i$    | The maximum delay requirement of the service $s_i$ |
| $\tau_{ik}$ | The delay from the FEN $f_j$ to the end device $u_k$ |
| $b_{ik}$ | The binary variable indicating whether the delay requirement of the service demand $d_{ik}$ is satisfied or not; $b_{ik} = \begin{cases} 1, & \text{if the delay requirement of } d_{ik} \text{ is satisfied} \\ 0, & \text{otherwise} \end{cases}$ |

The problem of the optimal service provisioning (SP) in the FEC environment can be defined as follows:

\[
\text{Minimize } \sum_{i \in \{1, \ldots, \alpha\}} \sum_{j \in \{1, \ldots, \beta\}} x_{ij} \tag{1}
\]

Subject to:

\[
\sum_{i \in \{1, \ldots, \alpha\}} x_{ij} \leq c_j, \forall j \in \{1, \ldots, \beta\} \tag{2}
\]

\[
x_{ij} = \begin{cases} 1, & \text{if a service image of } s_i \text{ is provisioned on } f_j \\ 0, & \text{otherwise} \end{cases} \tag{3}
\]

\[
\sum_{i \in \{1, \ldots, \alpha\}} \sum_{k \in \{1, \ldots, \gamma\}} d_{ik} = \sum_{i \in \{1, \ldots, \alpha\}} \sum_{k \in \{1, \ldots, \gamma\}} b_{ik} \tag{4}
\]

\[
b_{ik} = \begin{cases} 1, & \text{if } \exists j \in \{1, \ldots, \beta\} \text{ satisfying } 0 < \tau_{jk} \times x_{ij} \times d_{ik} \leq T_i \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in \{1, \ldots, \alpha\} \text{ and } k \in \{1, \ldots, \gamma\} \tag{5}
\]

\[
b_{ik} \in \{0, 1\}, \forall i \in \{1, \ldots, \alpha\}, k \in \{1, \ldots, \gamma\} \tag{6}
\]

Equation (1) is the objective function of minimizing the total number of the service images provisioned on the FENs. Equation (2) is for the capacity constraint, termed Condition-C, and Equations (5) and (6) are for the delay constraint, termed Condition-D. The Equations (3), (4), (6) and (7) define the binary variables $b_{ik}$ and $x_{ij}$. $x_{ij} = 1$ indicates that
a service image of \( s_i \) is deployed in \( f_j \). Equation (2) checks whether each FEN accommodates service demands within its maximum capacity \( c_j \). Equation (5) checks for whether the delay requirements of all the service demands in \( D \) are satisfied. In Equation (6), the binary variable \( b_{ik} \) is set to 1, for the service demand \( d_{ik} \) with 1, if there exists at least one FEN with a service image of \( s_i \) that satisfies the maximum delay requirement of \( T_i \), i.e., \( 0 < \tau_{ik} \times x_{ij} \times d_{ik} \leq T_i \).

Because the optimal SP problem was formulated as a 0–1 integer linear programming problem, the problem is NP-hard. Thus, we propose two heuristic mechanisms that determine how to deploy service images on FENs according to the collected service demands from end devices.

### 3.2. Logical Fog Network

CDC determines the FENs to be installed with specific service images based on the service demands of the end devices. The locations of the FENs can be anywhere in the given network, determined by the deployment strategy which is out of the scope of this paper. In general, the physical network topology of the FEC environment is a mesh which increases the computational complexity of determining the optimal FENs to be provisioned with service images. Therefore, in order to simplify the problem, we form a logical tree topology of the FENs rooted at CDC, called a logical fog network [12], \( N_{\text{fog}} \), in a hierarchical manner, as shown in Figure 1b for the physical network in Figure 1a in which CDC is labeled as \( f_0 \) for the sake of convenience. Because \( N_{\text{fog}} \) is a subnetwork of the given physical network and the tree topology is a special case of the mesh topology, the SP problem for \( N_{\text{fog}} \) is also NP-hard.

**Figure 1.** Logical fog network: (a) an example of a physical network; (b) the logical fog network of (a).

In \( N_{\text{fog}} \), if an end device is in the coverage area of a FEN, we say that the end device is in the direct coverage of the FEN or the FEN directly covers the end device. If an FEN is an ancestor (including the parent) of the FEN directly covering an end device, the ancestor FEN is said to cover the end node indirectly. In \( N_{\text{fog}} \), all the FENs on the path from CDC to the directly covering FEN of an end device cover the end device.

A service requester is an end device requesting the corresponding service. The data from a service requester can be handled by any FEN, with the service image and satisfying the requirements, both Condition-C and Condition-D, of the service requester on the path from the directly covering FEN up to CDC in \( N_{\text{fog}} \). Each FEN, located in between the
service requester and the service handling FEN, just delivers the data from the service requester to its parent FEN for the proper processing of the data.

We assume that CDC has the information on $N_{\text{fog}}$ including the end devices directly covered by the FENs and the information of the locations and the capacities of the FENs. For the provisioning of service images in $N_{\text{fog}}$, we assume that CDC has unlimited capacity such that any requirements of the end devices, except for latency, can be accommodated—that is, if there exist any service requests not accommodated by any FENs, CDC can cover those service requests.

The logical fog network, $N_{\text{fog}}$, is constructed as follows:

[Step N1] The initial $N_{\text{fog}}$ is null and the FENs are sorted in the decreasing order of the capacity.
[Step N2] CDC is first included in $N_{\text{fog}}$ as the root and the level of CDC is set to 0.
[Step N3] From the sorted list of the FENs obtained at [Step N1], the first k FENs are selected as the child nodes of CDC. $\Gamma$ is the set of the child FENs of CDC. The value of the parameter k has to be properly decided by the fog network designer.
[Step N4] The delay of the link between an FEN in $\Gamma$ and CDC is adjusted to a value less than its original delay (e.g., a half of the original delay).
[Step N5] Each FEN in $F - \Gamma$ is added to $N_{\text{fog}}$ via the minimum delay (shortest) path to CDC.

In [Step N3], the criterion for selecting the FENs in $\Gamma$ is the capacity because an FEN with more capacities can have more service images installed and a higher possibility of covering more end devices. In [Step N5], the criterion for adding the FENs in $F - \Gamma$ to $N_{\text{fog}}$ is the latency because the main purpose of FEC is reducing the latency from end devices. After [Step N5], the obtained $N_{\text{fog}}$ is a logical tree rooted at CDC with the FENs in $\Gamma$ directly connected to CDC and with each FEN in $F - \Gamma$ connected to CDC via the shortest path. The reason for the link cost (delay) adjustment in [Step N4] is to make the possibility of the FENs in $F - \Gamma$ to go through the FENs in $\Gamma$ to get to CDC. From this, we can achieve the load balancing effect by making high-capacity FENs share the burden of computing load with CDC. Through the procedure of [Step N1]∼[Step N5], we can obtain a logical fog network that is appropriate for less service images provisioned thanks to the tree topology and for lower delays from end devices to FENs thanks to the shortest path branches. In the following Section 3.3, we described two mechanisms finding the right FENs, based on $N_{\text{fog}}$, that can accommodate the service images demanded by the end devices with the aim of minimizing the number of service images provisioned.

### 3.3. Service Provisioning Based on Service Demands

In this subsection, we propose two heuristic SP mechanisms, based on $N_{\text{fog}}$, in which the amount of service demands for a FEN is the major factor to be considered in placing a service image on the FEN. The first SP mechanism provisions the highest-level FEN, satisfying the conditions, Condition-C and Condition-D, of the most-demanding service, with the service so that the coverage of the FEN for the service can be maximized. Thus, the first SP mechanism is named as the maximal coverage-SP (MC-SP) mechanism. The MC-SP mechanism performs well if the distribution of the end devices demanding a specific service is uniform in the logical fog network. However, if we consider an industrial area with many smart factories requesting a specific service, the MC-SP mechanism may not perform optimally. In this situation, it is desirable to place the corresponding service image near the area. Thus, in our second mechanism, we take into consideration the locations of the end devices demanding a specific service in finding the right FEN to be placed with service images. The second mechanism is named as the flexible coverage-SP (FC-SP) mechanism. In the Sections 3.3.1 and 3.3.2, the MC-SP and the FC-SP mechanisms are described in detail, respectively. For the simplicity of the description of the mechanisms, we assume that, in $N_{\text{fog}}$, there exists at least one FEN satisfying Condition-C and Condition-D of each service in $S$. 
3.3.1. Maximal Coverage-Service Provisioning Mechanism

Before we describe the procedure of the MC-SP mechanism, the FENs in F are rearranged to \( \tilde{F} = (f'_{1}, \ldots, f'_{\beta}) \) such that the index of an FEN at the level \( q \) in \( N_{fog} \) is smaller than that of a FEN at a level lower than \( q \). That is, \( f'_{1} \sim f'_{k} \) \((k = 1, \ldots, \beta)\) are the children FENs of CDC in \( N_{fog} \) and \( f'_{\beta} \) is at the lowest level of \( N_{fog} \). We define the coverage matrix \( V_{fog} \) representing \( N_{fog} \) such that the element of \( V_{fog}, v_{kj} \), is set to 1 if the end device \( u_{k} \) is covered by the FEN \( f'_{j} \). With given the service demand matrix \( D, W = [w_{ij}] = D \times V_{fog} \) is the \( \alpha \times \beta \) matrix whose element \( w_{ij} \) indicates the total demands on \( s_{i} \) from the end devices covered by \( f'_{j} \). The notations for the SP mechanisms are listed in Table 2.

| Notation | Description |
|----------|-------------|
| \( N_{fog} \) | The logical fog network |
| \( \tilde{F} \) | The ordered list of the FENs in \( F \) from the highest to the lowest level of \( N_{fog} \); \( \tilde{F} = (f'_{1}, \ldots, f'_{\beta}) \) |
| \( V_{fog} \) | The coverage matrix of \( N_{fog} \) consisting of \( v_{kj} \)'s |
| \( v_{kj} \) | The element of \( V_{fog} \) indicating whether the end device \( u_{k} \) is covered by the FEN \( f'_{j} \) or not; \( v_{kj} = \{ 1, \text{ if } u_{k} \text{ is covered by } f'_{j} \}, 0, \text{ otherwise} \) |
| \( W \) | The total demand matrix consisting of \( w_{ij} \)'s |
| \( w_{ij} \) | The element of \( W \) indicating the total demands on the service \( s_{i} \) from the end devices covered by the FEN \( f'_{j} \) |
| \( W_{j} \) | The \( j \)th column of \( W; W_{j} = (W_{j}(1), \ldots, W_{j}(\alpha)) \), where \( W_{j}(q) = w_{qj} \) for \( q \in \{ 1, \ldots, \alpha \} \) |
| \( \tilde{W}_{j} \) | The sorted list of \( W_{j} \) in the decreasing order of the amount of demands; \( \tilde{W}_{j} = (\tilde{W}_{j}(1), \ldots, \tilde{W}_{j}(\alpha)) \), \( \tilde{W}_{j}(q) \geq \tilde{W}_{j}(\tilde{q}) \) if \( q < \tilde{q} \) for \( q, \tilde{q} \in \{ 1, \ldots, \alpha \} \) |
| \( Y_{j} \) | The corresponding service list of \( W_{j} \); \( Y_{j} = (Y_{j}(1), \ldots, Y_{j}(\alpha)) \), \( Y_{j}(q) \in S \) for \( q \in \{ 1, \ldots, \alpha \} \) |

The procedure of the MC-SP mechanism with given \( N_{fog}, \tilde{F} \) and \( W \) is described in Figure 2.

![Figure 2](image_url)

The procedure of the maximal coverage-SP (MC-SP) Mechanism.

In Figure 2, the \( j \)th column of \( W \) is denoted as \( W_{j} \) and the \( i \)th element of \( W_{j} \) as \( W_{j}(i) \). \( \tilde{W}_{j} \) is the sorted list of \( W_{j} \), in the decreasing order of the amount of demands, which is obtained by carrying out the function \( \text{Sort}(W_{j}) \) (see [M2]). \( Y_{j} \) is the corresponding service list of \( \tilde{W}_{j} \) obtained from the function \( \text{Service}(\tilde{W}_{j}) \) (see [M3]). That is, \( Y_{j}(i) \) has the information of the service whose demands imposed on \( f'_{j} \) is the \( i \)th largest among the services in \( W_{j} \). The outer
for-loop checks for each FEN from the highest to the lowest level of $N_{foo}$ and the inner for-loop checks for each service from the list in $Y_j$. That is, for $f'_j$, the larger the demands on a service, the earlier the service is checked for its provisioning in $f'_j$. The condition in [M5] is for excluding those services with no demands from being provisioned in $f'_j$. In [M6]–[M7], the provision of the service $Y_j(i)$ in $f'_j$ is performed if the provision satisfies the Condition-C and the Condition-D of $Y_j(i)$. The functions $ChkCapacity(f'_j, Y_j(i))$ and $ChkDelay(f'_j, Y_j(i))$ check Condition-C and Condition-D, respectively, and return the true or false value according to whether the corresponding requirement is satisfied or not. The function Provision($f'_j, Y_j(i)$) in [M7] provisions the FEN $f'_j$ with the service image of the service $Y_j(i)$. The function Modify($N_{foo}, W, f'_j, Y_j(i)$) in [M8] removes the demands on the service $Y_j(i)$ from the FENs which are the descendants of $f'_j$ because all the service demands are resolved by $f'_j$.

Figure 3 is a simple example showing $N_{foo}$ with the FENs $f_1, \ldots, f_6$ and the end devices $u_1, \ldots, u_9$ and the services $s_1, \ldots, s_6$. The service demands of each end device are listed in its corresponding box and the capacity of $f_j$ is indicated by $c_j$. In this example, to focus more on SP from the aspect of Condition-C, all the FENs are assumed to satisfy Condition-D of $x_i$ for all $i \in \{1, \ldots, 6\}$. For simplicity, $f_0$ is also assumed to satisfy the Condition-Ds of all services. In the figure, we can see the result of applying the MC-SP mechanism, and the service placement matrix $X = [x_{ij}]$, indicated by a black solid line box, where $x_{ij} = 1$ implies that $s_i$ is provisioned on $f_j$. Since all FENs satisfy Condition-D of each service, services are placed from $f_0$ to $f_6$. In MC-SP, the service with more demands has a higher chance to be placed on. In this example, $f_0$ can accommodate at most three services and $s_1, s_5, s_6$ have demands of 4, 3, 3, respectively, so $s_1, s_5, s_6$ are provisioned on $f_0$. After that, the demands on $s_1, s_5, s_6$ are removed from the other FENs. Then, services $s_2, s_4$ are placed on $f_1$ and $s_3$ is placed on $f_2$.

3.3.2. The Flexible Coverage-Service Provisioning Mechanism

The MC-SP mechanism takes into consideration only the hierarchical topology of $N_{foo}$ in the provisioning service images on FENs. This mechanism is the lack of the awareness of the patterns of service demands which may depend on some geographical, industrial, or social characteristics. Thus, in this subsection, we propose the FC-SP mechanism, which adaptively places service images on FENs according to the pattern of service demands. This is well suited for the situation with uneven service demands in a limited area. For this situation, an FEN near to the area is best placed to resolve the demands. Thus, we adopt the lowest-level common ancestor (LCA) FEN, in $N_{foo}$, of the end devices demanding a
service as the candidate to be provisioned with the corresponding service image. If the LCA FEN \( f \) of a service \( s \) is installed with the service image of \( s \), then those end devices having \( f \) as the LCA FEN demanding \( s \) can be commonly supported by \( f \). In the FC-SP mechanism, the pattern of service demands is estimated by using the Shannon entropy \([21]\).

The preference of a service image to be provisioned is determined based on the total amount of demands imposed on the FEN which is currently considered for the service provisioning. The additional notations for the FC-SP mechanism are listed in Table 3.

Table 3. Additional notations for the flexible coverage-SP (FC-SP) mechanism.

| Notation | Description |
|----------|-------------|
| \( Z \)  | The ordered list of \( z_i's; Z = <z_1, \ldots, z_n>, \) where \( z_i \) is the total demands on \( s_i \), \( z_i = \sum_{j \in \{1, \ldots, \beta\}} w_{ij} \) |
| \( \bar{Z} \)  | The sorted list of \( Z \) in the decreasing order of the total demands on services |
| \( \bar{s} \)  | The corresponding service list of \( \bar{Z}; \bar{s} = s_1, \ldots, s_n \), where \( s_i \) is the service on which the total demands of \( z_i \) are imposed |
| \( A^{rev} \)  | The ordered list of LCA FENs satisfying Condition-D of each service in \( \bar{S} \) |
| \( A^{fog} \)  | The ordered list of the services with the same LCA FENs; the \( i \)th element, \( A^{fog}(i) \), is the ordered list of the services, in the decreasing order of the total demands imposed on each service, whose LCA FEN is \( f \) |
| \( \sigma \)  | The current service to be provisioned |
| \( f^{\min} \)  | The minimum entropy service which was already provisioned on the current FEN |
| \( f^p \)  | The parent node of the current FEN |
| \( f^c \)  | The child node of the current FEN |
| \( P^{fog} \)  | The ordered list of visited ancestor FENs starting from \( f \) |

The procedure of the FC-SP mechanism with given \( N_{fog} \) and \( W \) is described in detail in Figure 4. Here, we assume that, in \( N_{fog} \), there exists at least one FEN satisfying Condition-C and Condition-D of each service in \( S \). The pseudocode description in Figure 4b is somewhat lengthy and complicated because lots of notations were used, so we provided a simpler form of description, flowcharts, in Figure 4a.

(a)
The total demands on \( s_i \) is \( z_i = \sum_{j \in \{1, \ldots, \beta\}} w_{ij}, z_i \in Z \). The elements in \( Z \) are sorted in the decreasing order, resulting in the sorted list \( \tilde{Z} \). The corresponding service list of \( \tilde{Z} \) is \( \tilde{S} = s_1, \ldots, s_\alpha \). For each service \( s_i \) in \( \tilde{S} \), the LCA FEN satisfying Condition-D is determined by the function \( \text{LCAncestor}(N_{fog}, W) \) and the ordered list of the LCA FENs for \( \tilde{S} \) is \( A^{\text{env}} \). Then, the function \( \text{FogLCAncestor}(A^{\text{env}}) \) returns \( A^{\text{fog}} \) whose \( i \)th element, \( A^{\text{fog}}(i) \), is the ordered list of the services, in the decreasing order, whose LCA FEN is \( f'_i \). GetFogService \( (A^{\text{fog}}(i)) \) in [F9] returns a service (the current to-be-provisioned service \( \sigma \))
one by one from $A_{\text{log}}^f(i)$ per call. In [F7]–[F10], the services in $A_{\text{log}}^f(i)$ are provisioned on $f'_i$ while the Condition-C of $f'_i$ is satisfied. If the condition in [F12] is met, the service $\sigma$ cannot be provisioned on $f'_i$ because of the capacity shortage. Then, according to the status of $f'_i$, upward provisioning (see [F31]–[F39]) and/or downward provisioning (see [F22], [F41]–[F50]) is performed. In the case that $\sigma$ cannot be provisioned on $f'_i$ and $f'_i$ is located at the highest level of $N_{\text{log}}$, $\sigma$ may replace the already provisioned service with the minimum entropy, $f_{\text{min}}$ found by calling EntropyService($f'_i$), if $\sigma$’s entropy is larger than the minimum entropy, i.e., $f_{\text{min}}$’s entropy (see [F16]–[F20]), or be provisioned on a lower-level FEN (by calling SplitProvision($\sigma$, $f'_i$), i.e., the downward provisioning of [F22]). Or, if $\sigma$ cannot be provisioned on $f'_i$ and $f'_i$ is not located at the highest level of $N_{\text{log}}$, the upward provisioning of [F31]–[F39] is performed first and if the upward provisioning fails (that is, if all the ancestors of $f'_i$ cannot provision $\sigma$), the downward provisioning of [F41]–[F50] is performed. This process is repeated until all the service images demanded by end devices are provisioned on FENs. Here, $p_{\text{log}}$ is the set of FENs fully provisioned with services.

In the FC-SP mechanism, the Shannon entropy of a service is used as a metric for the provisioning priority of the service. In Figure 4, the function Entropy($\sigma$) is called when $\sigma$ cannot be provisioned on $f'_i$ because of Condition-C being not satisfied at $f'_i$. Entropy($\sigma$) returns the Shannon entropy of $\sigma$ by using the following Equation (8) which reflects the degree of the distribution of the end devices requesting $\sigma$:

$$\text{Entropy}(\sigma) = -\sum_{f' \in F'} p_{f'} \log_2 \left(\frac{p_{f'}}{p_{f'}}\right), \text{ where } F' \subseteq F$$  

In Equation (8), first, $F'$ is determined for $\sigma$, which is the set of the descendent LCA FENs of the LCA FEN currently failing in provisioning $\sigma$. $p_{f'}$ is obtained by dividing the total demands on $\sigma$ at $f'$ by the total demands on $\sigma$ at the FENs in $F'$.

Figure 5 shows an example of applying the FC-SP mechanism to $N_{\text{log}}$ of Figure 1b with the result of the service placement matrix $X$. Here, we also assume that all FENs including $f_0$ satisfy Condition-Ds of all services. On $f_0$, services $s_3$ and $s_4$ are provisioned first since each of them has demands of 8 which is the largest. After that, because $s_1$ and $s_2$ have the same amount of total demands of 5, respectively, the Shannon entropy at $f_0$ is calculated for $s_1$ and $s_2$. For that, $F'$ is determined for $s_1$ and $s_2$, respectively. $F'$ of $s_1$ is $\{f_1, f_2\}$ and $F'$ of $s_2$ is $\{f_1, f_3\}$. Then, the entropy of $s_1$ is “$-(3/5 \log_2(3/5) + 2/5 \log_2(2/5))$ (about 0.9709506)” because the total demands on $s_1$ at $f_1$ is 3 and the total demands on $s_1$ at $f_2$ is 2. The entropy of $s_2$ is “$-(4/5 \log_2(4/5) + 1/5 \log_2(1/5))$ (about 0.7219281)” because the total demands on $s_2$ at $f_1$ is 4 and the total demands on $s_2$ at $f_2$ is 1. Because $s_1$ has a higher entropy than $s_2$, $s_1$ is provisioned on $f_0$ and $s_2$ is provisioned on $f_1$ and $f_2$ by the downward provisioning of [F22]. The higher the entropy is, the more the service demands are distributed. Thus, by provisioning more distributed services on FECs at higher levels of $N_{\text{log}}$, the capability of covering service demands can be enhanced.
Figure 5. The service provisioning result of the FC-SP mechanism.

4. Performance Evaluation

Simulations were performed by using the NetworkX package [22] and Python. The simulation network environment is similar to that of [12]. The simulation network is obtained by designing an example of physical fog/edge network based on the administrative/population information obtained from Seoul open data center [23], South Korea, as shown in Figure 6a. The physical fog/edge network is built by locating one CDC (the largest dot in the figure) at the geographic center of Seoul and by locating 449 FENs according to the two-level administrative districts of Seoul. The capacity of a FEN is determined as proportional to the number of households of the corresponding administrative district. In the figure, the dots except for the largest dot (i.e., CDC) are FENs and the size of a dot implies the capacity of the corresponding FEN. The weight of an edge between two FENs is determined proportional to the physical distance between them. The corresponding logical fog network, shown in Figure 6b, was constructed based on the logical fog network construction mechanism described in Section 3.2.

Figure 6. An example fog network for simulations: (a) a physical network with FENs in a mesh topology; and (b) a logical representation of the physical network of (a) in a tree topology [12].

For the performance comparisons of the proposed MC-SP and FC-SP mechanisms, we designed and implemented a mechanism, called the on-demand mechanism, that dynamically places the corresponding service image upon a request based on the logical
In the on-demand mechanism, if an end device receives a request on a service, it checks whether it has any FENs installed with the corresponding service image within 2-hop in the logical fog network. If there is none, it checks whether any of its ancestor FENs have the service image. If none of the ancestor FENs have the service image, it places the corresponding service image on itself. In the case when it is the lack of the resources, it places the service image on the 2-hop FEN which is the closest to itself. If it is not possible, it places the service image on one of its ancestor FENs which is the closest to itself. By comparing our proposed mechanisms with the on-demand mechanism operating on the basis of per-service request, we measure the performance of ours in terms of the resource utilization of FENs.

The number of service types, the distribution of service demands, the maximum capacity of CDC, and the maximum capacity of each FEN are the factors affecting performance. The number of service types requested by end devices is set to 1000 and 2000 service types and the number of end devices is 450. For the generation of service demands, we adopt the long-tailed distribution [24] of service demands in which a small set of service types is heavily requested and the rest of the service types are not frequently requested. The long-tailed distribution is adopted because a set of popular services are heavily used in the real world. For the simulations, the long-tailed distribution of service demands (we call this the L distribution case) is used, where about 10% of 1000 service types are heavily requested by the end devices, as shown in Figure 7 [12], which is obtained by using the zeta distribution [25]. This is achieved by using the function np.random.zipf (1.6, 1000) [26] where 1.6 is the value of the distribution parameter and 1000 is the number of service types. The function returns a value (we call this an L value) for each service type and the L value is used in determining the end devices with a service request for the service type. If the L value of a service type is greater than or equal to 100, all the end devices are assigned with a service request for the service type. Otherwise, the L value is used as the probability of assigning a service request for the service type to each end device. That is, for a service type, a higher L value implies a higher possibility of assigning a corresponding service request to each end device. If we sum up the service requests for all the service types in Figure 7, it becomes 42,210 in total.

Figure 7. The L distribution case with 42,210 service demands for 1000 service types [12].
The measured performance factors are the number of service images placed, the number of non-accommodated service requests, and the average network cost per service request. The average network cost per service request is the average one-way physical distance required for providing a service which is assumed to be proportional to the delay. For the calculation of the average network cost per service request, the physical distance between the end device with a service request and the FEN accommodating the corresponding service request is measured on the logical fog network. The physical distance is measured for each service request and then, all the measured physical distances are summed up. The average network cost per service request is obtained by dividing the total physical distance by the number of service requests. The maximum capacity of CDC is set to 300, 500 and 750 resource units, and one service image placement is assumed to require one unit of resources.

Figures 8–11 show the performances of the proposed MC-SP and FC-SP mechanisms compared with the on-demand mechanism for various maximum CDC capacities of 300, 500 and 750 with 1000 service types for the L distribution case. Figure 8 depicts the graph showing the performance in terms of the number of service images placed on FENs. We can observe that, as the maximum capacity of CDC increases, more service images are placed on CDC, resulting in less service images placed on FENs in all the mechanisms. The on-demand mechanism places significantly more service images than our proposed mechanisms and even with increased CDC capacity, the on-demand mechanism slightly reduces the number of service images. The reason for this is that the on-demand mechanism places a service image near the end device which has requested the service, resulting in many service images near to end devices. On the other hand, the MC-SP and the FC-SP mechanisms reduce the number of service images placed significantly. Hence, we can assert that the FC-SP mechanism considering the service pattern performs the best in the resource utilization of FENs.

![Number of Service Images Placed](image)

**Figure 8.** The number of service image placements for various maximum cloud data centers (CDCs) capacities with 1000 service types for the L distribution case.

Figure 9 is the graph showing the number of service requests for which the corresponding service images are not placed on any FENs for the L distribution case. Thanks to the well-utilized FEN resources, the FC-SP significantly outperforms the MC-SP mechanism and the on-demand mechanism, especially for larger CDC capacities. As we expected, the number of non-accommodated service requests decreases as the maximum CDC capacity increases for all the mechanisms.
In Figure 10, the average network cost of handling a service request is shown for various maximum CDC capacities for the L distribution case. As the maximum CDC capacity increases, the network cost also increases because more service images are placed on the CDC. Both of the proposed mechanisms perform much worse than the on-demand mechanism because service images are placed near the end devices in the on-demand mechanism with less service images placed. As for the proposed mechanisms, the FC-SP mechanism requires less network cost than the MC-SP mechanism. The reason is that the MC-SP mechanism does not consider the pattern of service demands and places service images on FENs at higher levels of the logical fog network for larger coverage.

Figure 9. The number of non-accommodated service requests for various maximum CDC capacities with 1000 service types for the L distribution case.

The performance of the proposed mechanisms is shown in Figure 11 with a varying FEN capacity and number of service types for the L distribution case. We simulated two cases of the FEN capacity, the basic case and the 10 times case. The basic case is the case of the FEN capacity in Figure 6b and the 10 times case is the FEN capacity of 10 times that of Figure 6b. The number of service types is set to 1000 and 2000.

Figure 11a shows the performance in terms of the number of service images placed on FENs, and Figure 11b shows the performance in terms of the number of non-accommodated...
service requests for the L distribution case. For the larger FEN capacity (i.e., the 10 times case), both MC-SP and FC-SP mechanisms place more service images, but the FC-SP mechanism decreases the number of non-accommodated service requests more significantly than the MC-SP. This indicates that the FC-SP mechanism fully utilizes the advantage of the increased FEN capacity. That is, the FC-SP mechanism accommodates more service requests than the MC-SP mechanism by provisioning more service images on FENs. In addition, as the number of service types increases, more service images are placed and more service requests are accommodated in both of the mechanisms. It is intuitive that more service images are placed for more service types, but the MC-SP mechanism shows a more noticeable increase in the number of service images placed with an insignificant decrease in the number of non-accommodated service requests. This implies that the MC-SP does not perform well in the situation of changed FEN capacities.

Figure 11c shows the graphs depicting the average network cost of a service request for the L distribution case. As the FEN capacity increases, the network cost decreases in both of the mechanisms. The FC-SP mechanism shows a more significant decrease in the network cost compared to the MC-SP mechanism, especially for more service types. This indicates that the FC-SP mechanism performs better than the MC-SP mechanism even for the case of more service types. Overall, the FC-SP mechanism is better than the MC-SP in utilizing the FEN resources and in adapting to changing environments such as the changes in CDC capacity, FEN capacity, and the number of service types to support.

For a more concrete evaluation of the performance, we performed simulations for another service demand distribution shown in Figure 12 (we call this the U distribution...
For a more concrete evaluation of the performance, we performed simulations for another service demand distribution shown in Figure 12 (we call this the U distribution case) which is obtained by using the uniform distribution. This is achieved by using the function `np.random.uniform(0, 100, 1000)` [27] where 0 and 100 are the lower and the upper boundary of the output interval, respectively, and 1000 is the number of service types. This function returns a value (we call this a U value) in the half-open output interval [0, 100) for each service type. The U value is used as the probability of assigning a service request for the service type to each end device. That is, the higher the U value of a service type is, the higher the possibility of assigning a service request of the service type to each end device. The U distribution case tends to assign service requests more evenly on service types than the L distribution case. The U distribution case generates 140,525 service requests for the performance evaluation of our mechanisms in a situation with heavy service requests.

![Figure 12. The U distribution case with 140,525 service demands for 1000 service types.](image)

The performance of the proposed mechanisms for the U distribution case is depicted in Figure 13. We can easily see that the FC-SP mechanism outperforms the MC-SP mechanism in the aspect of all the performance factors. This indicates that the FC-SP mechanism performs better than the MC-SP mechanism even for the case with service requests relatively less biased over the service types (i.e., the U distribution case).

![Figure 13. Cont.](image)
Figure 13. The performance with the maximum CDC capacity of 500 and 1000 service types for the U distribution case: (a) the number of service images placed; (b) the number of non-accommodated service requests; and (c) the network cost per service request provided by CDC or a FEN.

From Figure 14, we can observe the performance comparison of our proposed SP mechanisms for the L and the U distribution cases. In Figure 14a,b, it is clearly shown that our mechanisms perform better for the L distribution case than for the U distribution case in terms of the number of service images placed and the percentage of non-accommodated service requests. This can be intuitively expected because the U distribution case has about 3.33 times more service requests than the L distribution case. In addition, we can see that the FC-SP mechanism performs very effectively in the sense that it accommodates more service requests than the MC-SP mechanism with many less service images placed, even in the stressful situation (i.e., the U distribution case). Figure 14c shows that the average network cost per service request for the U distribution case is lower than that for the L distribution case. The reason for this is that, for the L distribution case, the majority of the service requests are likely covered by higher-level FENs because most of the service requests are biased on a few specific service types, resulting in higher average network cost per service request than the U distribution case.

Figure 14. The performance comparison of the L distribution case and the U distribution case with the maximum CDC capacity of 500 and 1000 service types: (a) the number of service images placed; (b) the number of non-accommodated service requests; (c) the network cost per service request provided by CDC or a FEN.
5. Conclusions

Because of the limited resources of FENs, it is not possible to place all the services on an FEN. Moreover, if service images are placed on FENs in a per service request-based way, the resources of FENs may not be efficiently utilized due to the duplicate placement of service images, resulting in less accommodation of service requests. Therefore, in this paper, we proposed two SP mechanisms, the MC-SP mechanism and the FC-SP mechanism, for provisioning service images on FENs on the basis of logical fog network considering the pre-obtained service demands. The MC-SP mechanism provisions service images on FENs based on the number of service requests from end devices and the FC-SP mechanism based on the locations of end devices requesting services. The performance of the proposed mechanisms was evaluated by carrying out through simulations. According to the simulation results, we observed that both of our mechanisms perform better than the On-Demand mechanism which was designed for performance comparison and operates in a simple manner of placing a service image near to the end device requesting the service upon each service request. Therefore, we can say that our mechanisms are scalable because they can save the FEN resources by effectively placing service images on FENs and are good for the environment with plenty of IoT devices deployed.

Author Contributions: Conceptualization, J.C. and S.A.; methodology, J.C. and S.A.; software, J.C.; validation, J.C. and S.A.; formal analysis, J.C. and S.A.; investigation, J.C. and S.A.; resources, J.C. and S.A.; data curation, J.C. and S.A.; writing—original draft preparation, J.C. and S.A.; writing—review and editing, S.A.; visualization, J.C.; supervision, S.A.; project administration, S.A.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the South Korean government (MSIT) (No. NRF-2018R1D1A1B07047339). This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2015-0-00363) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References
1. Internet of Things At-A-Glance. 2016. Available online: https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf (accessed on 2 November 2019).
2. Vaquero, L.M.; Rodero-Merino, L.; Caceres, J.; Lindner, M. A Break in the Clouds: Towards a Cloud Definition. ACM SIGCOMM Comput. Commun. Rev. 2008, 39, 50–55. [CrossRef]
3. Mell, P.M.; Grance, T. The NIST Definition of Cloud Computing; NIST Special Publication 800–145; National Institute of Standards and Technology: Gaithersburg, MA, USA, 2011; Volume 800, pp. 1–7. [CrossRef]
4. Dillon, T.; Wu, C.; Chang, E. Cloud Computing: Issues and Challenges. In Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia, 20–23 April 2010; pp. 27–33.
5. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog Computing and its Role in the Internet of Things. MCC Workshop Mob. Cloud Comput. 2012, 13–16. [CrossRef]
6. Yi, S.; Li, C.; Li, Q. A Survey of Fog Computing: Concepts, Applications and Issues. In Proceedings of the 2015 Workshop on Mobile Big Data, Hangzhou, China, 22–25 June 2015; pp. 37–42.
7. Trakadas, P.; Nomikos, N.; Michailidis, E.T.; Zahariadis, T.; Facca, F.M.; Breitgand, D.; Rizou, S.; Masip, X.; Gkonis, P. Hybrid Clouds for Data-Intensive, 5G-Enabled IoT Applications: An Overview, Key Issues and Relevant Architecture. Sensors 2019, 19, 3951. [CrossRef] [PubMed]
8. Goudarzi, M.; Wu, H.; Palaniswami, M.S.; Buwaya, R. An Application Placement Technique for Concurrent IoT Applications in Edge and Fog Computing Environments. IEEE Trans. Mob. Comput. 2020. [CrossRef]
9. Slaht, F.A.; Desprez, F.; Lebre, A. An Overview of Service Placement Problem in Fog and Edge Computing. ACM Comput. Surv. 2020, 53, 1–35. [CrossRef]
10. Souza, V.B.C.; Ramirez, W.W.; Masip-Bruin, X.; Marin-Tordera, E.; Ren, G.; Tashakor, G. Handling Service Allocation in Combined Fog-Cloud Scenarios. In Proceedings of the IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–5.
11. Brogi, A.; Forti, S. QoS-Aware Deployment of IoT Applications Through the Fog. *IEEE Internet Things J.* 2017, 4, 1185–1192. [CrossRef]

12. Choi, J.; Ahn, S. Scalable Service Placement in the Fog Computing Environment for the IoT-based Smart City. *J. Inf. Process. Syst.* 2019, 15, 440–448.

13. Deng, R.; Lu, R.; Lai, C.; Luan, T.H. Towards Power Consumption-Delay Tradeoff by Workload Allocation in Cloud-Fog Computing. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 3909–3914. [CrossRef]

14. Skarlat, O.; Schulte, S.; Borkowski, M.; Leitner, P. Resource Provisioning for IoT Services in the Fog. In Proceedings of the IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), Macau, China, 4–6 November 2016. [CrossRef]

15. Skarlat, O.; Nardelli, M.; Schulte, S.; Dustdar, S. Towards QoS-aware Fog Service Placement. In Proceedings of the IEEE International Conference on Fog and Edge Computing (ICFEC), Madrid, Spain, 14–15 May 2017.

16. Faticanti, F.; Pellegrini, F.D.; Siracusa, D.; Santoro, D.; Cretti, S. Cutting Throughput with the Edge: App-aware Placement in Fog Computing. In Proceedings of the IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, 21–23 June 2019.

17. Saurez, E.; Hong, K.; Lillethun, D.; Ramachandran, U.; Ottenwälder, B. Incremental Deployment and Migration of Geo-Distributed Situation Awareness Applications in the Fog. In Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems, Association for Computing Machinery (ACM), Irvine, CA, USA, 20–24 June 2016; pp. 258–269.

18. Yu, R.; Xue, G.; Zhang, X. Application Provisioning in FOG Computing-enabled Internet-of-Things: A Network Perspective. In Proceedings of the IEEE Infocom 2018 IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 783–791. [CrossRef]

19. Souza, V.; Masip-Bruin, X.; Marín-Tordera, E.; Sánchez-López, S.; Garcia, J.; Ren, G.; Jukan, A.; Ferrer, A.J. Towards a Proper Service Placement in Combined Fog-to-Cloud (F2C) Architectures. *Future Gener. Comput. Syst.* 2018, 87, 1–15. [CrossRef]

20. Eppstein, D. Subgraph Isomorphism in Planar Graphs and Related Problems. *J. Graph. Algorithms Appl.* 1999, 3, 1–27. [CrossRef]

21. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* 1948, 27, 379–423. [CrossRef]

22. NetworkX—NetworkX Document. Available online: https://networkx.org/ (accessed on 2 November 2019).

23. Seoul Open Data Plaza. Available online: https://data.seoul.go.kr/ (accessed on 2 November 2019).

24. Long Tail—Wikipedia. Available online: https://en.wikipedia.org/wiki/Long_tail (accessed on 2 November 2019).

25. Zeta Distribution—Wikipedia. Available online: https://en.wikipedia.org/wiki/Zeta_distribution (accessed on 2 November 2019).

26. numpy.random.zipf—NumPy v1.20 Manual. Available online: https://numpy.org/doc/stable/reference/random/generated/numpy.random.zipf.html (accessed on 2 November 2019).

27. numpy.random.uniform—NumPy v1.20 Manual. Available online: https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html (accessed on 2 November 2019).