Service Provisioning for IoT Applications with Multiple Sources in Mobile Edge Computing

Jing Li†, Weifa Liang†, Zichuan Xu‡, and Wanlei Zhou¶
† The Australian National University, Canberra, ACT 2601, Australia
‡ Dalian University of Technology, Dalian, 116020, China
¶ University of Technology of Sydney, NSW 2007, Australia
Emails: jing.li5@anu.edu.au, wliang@cs.anu.edu.au, z.xu@dlut.edu.cn, wanlei.zhou@uts.edu.au

2020 IEEE 45th Conference on Local Computer Networks (LCN) | 978-1-7281-7158-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/LCN48667.2020.9314795

***Abstract*—We are embracing an era of Internet of Things (IoTs). However, the latency brought by unstable wireless networks and computation failures caused by limited resources on IoT devices seriously impacts the quality of service of user experienced. To address these shortcomings, the Mobile Edge Computing (MEC) platform provides a promising solution for the service provisioning of IoT applications, where edge-clouds (cloudlets) are co-located with wireless access points in the proximity of IoT devices, and the service response latency can be significantly reduced. Meanwhile, each IoT application usually imposes a service function chain enforcement for its data transmission, which consists of different service functions in a specified order, and each data packet transfer in the network from the gateways of IoT devices to the destination must pass through each of the service functions in order.**

In this paper, we study IoT-driven service provisioning in an MEC network for various IoT applications with service function chain requirements, where an IoT application consists of multiple data streams from different IoT sources that will be uploaded to the MEC network for aggregation, processing, and storage. We first formulate a novel cost minimization problem for IoT-driven service provisioning in MEC networks. We then show that the problem is NP-hard, and propose an IoT-driven service provisioning framework for IoT applications, which consists of streaming data uploading from multiple IoT sources to the MEC network, data stream aggregation and routing, and Virtual Network Function (VNF) instance placement and sharing in cloudlets in the MEC network. In addition, we devise an efficient algorithm for the problem, built upon the proposed service framework. We finally evaluate the performance of the proposed algorithm through experimental simulations. Experimental results demonstrate that the proposed algorithm is promising, compared with the lower bound on the optimal solution of the problem and another comparison heuristic.

I. Introduction

The Internet of Things (IoTs) is paving the way for many new emerging technologies, such as smart grid, industry 4.0, self-driving, smart cities, etc. The number of connected devices is set to increase from 700 million to 3.2 billion by 2023 [4]. Conventional IoT devices usually transmit their data to remote clouds for storage and processing. Due to the remoteness of clouds and increasingly congested core network, this incurs prohibitive long transmission delays, thereby violating the real-time data processing requirement of many IoT applications, Thus, traditional clouds are no longer suitable for IoT applications [20].

With the fast development of 5G and Beyond-5G (B5G), mobile edge computing (MEC) promises to greatly reduce the data processing delay for IoT services, by deploying computing resource (e.g., cloudlets) within the proximity of IoT devices [1]. A typical IoT application usually needs to collect sensory data from multiple sources (e.g., IoT devices) and aggregate the streaming data of multiple sources at some intermediate nodes and ultimately route the aggregated data stream to a destination for further processing and storage. To ensure the security and privacy of data stream routing in the MEC network, various network service functions such as firewalls, Intrusion Detection Systems (IDSs), proxies, and load balancers need to be deployed. For example, IoT applications usually need a sequence of network service functions for data aggregation and filtering, to guarantee the real-time in-network processing (summation, averaging, maximum or minimum) of IoT data streams. Such a sequence of network service functions is referred to as a *Service Function Chain* (SFC). Conventional network functions are implemented by dedicated hardware - middleboxes, making their deployment and maintenance very expensive and not agile. Network Function Virtualization (NFV) [5], [6], [8] as a new technology promises to provide inexpensive and flexible network services, and implements these network service functions as software in VMs or containers in cloudlets. NFV makes virtual service provisioning in the MEC network becomes affordable, and easy to adopt and maintain.

To enable service provisioning in an MEC network for IoT applications with multiple sources and SFC requirements, it poses significant challenges. First, the provisioning of an IoT service requires joint consideration of many complicated data processing procedures, such as the data uploading from multiple sources through different gateways to a single destination, data stream aggregation and routing in the backhaul of the MEC network, and the processing of data streams by a specified sequence of Virtual Network Functions (VNFs). To save both computing and bandwidth resource usages, we usually need to build a routing tree rooted at the destination and spanning each of the sources for an IoT application with multiple sources, instead of uploading the data stream of each source to the destination separately. We then need to aggregate the uploading data streams at tree nodes and install appropriate numbers of VNF instances of service functions at these tree nodes. Building such a routing tree involves a non-trivial interplay between VNF placement, aggregation node selection, and routing path selection, which is the first fundamental challenge. Second, the computing and bandwidth resources in an MEC network are usually very precious and the cost of using such resources is expected to be minimized. The operational cost of IoT applications depends on not just resource usage costs but also load-balance costs. This requires a non-trivial interplay among the minimization of operational cost and the service chain requirements of IoT applications, while finding a data forwarding tree for multiple sources for each IoT application. Third, data streams of each IoT applications are allowed to be merged or aggregated at intermediate nodes of the routing tree before reaching its destination. VNFs for such data merging and aggregation usually can be shared among multiple IoT applications. How to further reduce the operational cost of IoT applications by allowing VNF sharing is another challenge.

The novelty of the work in this paper lies in formulating a novel IoT-driven service provisioning problem in an MEC network, where multiple data stream sources of an IoT application have their streaming data to be uploaded to the MEC for processing and storage, while each data stream must pass through a given service function chain prior to reaching the destination. We propose an IoT-driven service provisioning framework in an MEC network through VNF instance placement and sharing, in-network aggregation on different data streams. We aim to minimize the operational cost of implementing an IoT application in terms of the computing cost, communication cost and workload balancing cost. We also devise an efficient algorithm for data stream uploading, routing and processing in the network, built upon the proposed framework. To the best of our knowledge, this is the very first work on IoT-driven service provisioning in MEC for IoT applications with multiple sources through network slicing and efficient resource allocation and optimization.

The main contributions of this paper are presented as follows. We first consider IoT-driven service provisioning in an MEC network for IoT applications with service function chain requirements, by formulating a novel cost minimization problem of IoT-driven services. We also show that the problem is NP-hard. We propose an IoT-driven service provisioning framework in MEC for each IoT application with multiple data sources, which includes data stream uploading from multiple IoT device subnetworks through multiple wireless access points (gateway nodes), data stream aggregation and routing, and VNF instance placement and sharing in cloudlets in the MEC network, while meeting the service function chain requirement of the IoT application. We then devise an efficient algorithm for the defined optimization problem with the aim to minimize the (total) operational cost of IoT applications in terms of the cost of various resource consumptions of the MEC such as computing resource, communication resource and the workload of each cloudlet. We finally evaluate the performance of the proposed algorithm through experimental simulations. Experimental results demonstrate that the proposed algorithm is promising, compared with the lower bound on the optimal solution of the problem (a minimization problem) and another heuristic for the problem.

The rest of the paper is organized as follows. Section II summarizes the related work on user service provisioning with service function chain requirements in MEC. Section III introduces notions, notations, and the problem definition. Section IV shows that the defined problem is NP-hard. Section V proposes a novel framework of IoT-driven service provisioning in an MEC network for IoT applications with SFC requirements. Section VI devises an algorithm for the cost minimization problem. Section VII evaluates the proposed algorithms empirically, and Section VIII concludes the paper.

II. Related work

With the emergence of complicated and resource-hungry mobile applications in IoT and smart cities, implementing user tasks in cloudlets of a mobile edge-cloud network becomes an important approach to shorten the response delays of users, reduce mobile device energy consumption, and improve user experience [15].

There are a few investigations of the provisioning of NFV-enabled network services for IoT applications. For example, Xu *et al.* [28] studied the QoS-aware VNF placement of service function chains in MEC for IoT applications with a single source. Xu *et al* [27] also considered the operational cost minimization problem for the implementation of IoT applications with SFC requirements. They once again focused on IoT application placement in MEC, by proposing randomized and heuristic algorithms for the problem. Although we will deal with the operational cost minimization problem of implementing IoT applications, the work in this paper is essentially different from the two mentioned works [27], [28]. We here deal with IoT applications with multiple sources and the data streams from these sources need to be aggregated or merged in the MEC, while the demanded numbers of NFV instances of each service function in the SFC need to be placed in cloudlets, and the load balancing at each cloudlet must be considered. We aim to construct a data routing tree, rather than a routing path in the mentioned works, for the IoT applications. Song *et al.* [21] considered the QoS-based task allocation in MEC for IoT application by proposing efficient algorithms, however, they did not incorporate the SFC requirement for IoT application into consideration. Yu *et al.* [29] studied the problem of IoT service provisioning with the objective to meet computing, network bandwidth and QoS requirements of an IoT application. They however did not consider the processing of IoT traffic with SFC enforcement. Mouradian *et al.* [19] proposed an architecture of NFV- and SDN-based distributed IoT gateways for large-scale disaster management.

There are extensive studies of user unicast and multicast request admissions through resource provisioning and allocations in MEC networks [2], [3], [7], [12], [13], [22], [24], [30]. For example, Jia *et al.* [10] considered the assignment of user requests to different cloudlets in a Wireless Metropolitan Area

Network with the aim to minimize the maximum delay among offloaded tasks, by developing heuristics for the problem. They [12] also studied workload balancing among cloudlets to reduce the maximum response delay of user requests. Xu *et al.* [24] devised approximation algorithms to efficiently offload user requests to different cloudlets under different conditions. Xia *et al.* [23] considered opportunistic task offloading under link bandwidth, mobile device energy, and cloudlet computing capacity constraints. Jia *et al.* [11], [14] considered NFV-enabled unicast request admissions with and without the end-to-end delay requirement, for which they developed efficient heuristic and approximation and online algorithms. Ma *et al.* [18] considered the profit maximization problem in MEC by dynamically admitting NFV-enabled unicast requests with QoS requirements, for which they developed an efficient heuristic and an online algorithm with a provable competitive ratio if the QoS requirement can be ignored. Although they considered the sharing of existing VNF instances among different unicast requests. It can be seen that the problem of NFV-enabled unicast request admissions in [14], [18] is a special case of the NFV-enabled multicast request admissions where the destination set contains only one node. Recently, there are several studies by extending the NFV-enabled unicast routing to NFV-enabled multicast routing in MEC environments [13]. For example, Xu *et al.* [26] considered the cost minimization of admitting a single NFV-enabled multicast request with the QoS requirement in MEC, where the implementation of the service chain of each request is consolidated to a single cloudlet. They aim to minimize the admission cost by placing no more than constant numbers of VNF instances of the service chain of the request in different branches of the found pseudo-multicast tree for the request. Ma *et al.* [16], [17] studied admissions of NFV-enabled multicasting requests under both static and dynamic admission scenarios, by proposing approximation and online algorithms for the problems with provable performance guarantees.

Although there are some similarities between the IoT service provisioning problem in this paper and the NFV-enabled multicasting problem in MEC [17], [25], due to the fact that both aim to build a routing tree for their solutions. However, tackling the IoT service provisioning problem is much more difficult, compared with the NFV-enabled multicasting problem. The main difference between them lies in the following. In the multicasting case, a packet from the source (the tree root) will broadcast to all leaves (destinations) such that a VNF instance of each service function in the SFC is installed along the path from the root to each leaf and the data traffic along all links in the tree is same. In contrast, the data routing tree for an IoT application is that all leaves are data sources, their data streams will be converged to the tree root and the data streams from its children of a node in the tree will be aggregated at the tree node, the number of VNF instances instantiated at a node in the path from a leaf to the tree root is determined by the accumulative volume of data stream at the node; or the values of flows at different tree links are different. Thus, the construction of a data routing tree is much more challenging, due to that the volume of data stream of each tree edge is not fixed, the weight (the volume) of a tree edge is dynamically determined by the number of children and the data volume of each child. Since different volumes on the tree edges, the accumulative volume of data stream at each tree node is not evenly distributed. Thus, the number of VNF instances of a service function at different nodes is different to process the accumulative volume of data steams.

III. Preliminaries

In this section we first introduce the system model, we then give notions and notations, and we finally present problem definitions.

A. System Model

Consider that an MEC network is an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of links between nodes. Each node $v \in V$ is an Access Point (AP). Associated with each AP, there is a co-located cloudlet $v \in V$ (edge cloud) with computing capacity $C_v > 0$, and the AP and its co-located cloudlet are connected through a high-speed optical cable, and the communication delay between them thus is negligible. Assume that the MEC network supports a set of services that each service is expressed as a service function. Denote by $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$, the set of service functions in the system, where the implementation of service function $f_i \in \mathcal{F}$ consumes the amount c_{f_i} of computing resource (in cloudlets), we term one implementation of f_i as one of its VNF instances and each VNF instance of f_i has its data processing capacity μ_{f_i}, where $1 \leq i \leq |\mathcal{F}|$. Figure 1 is an illustrative example of an MEC network.

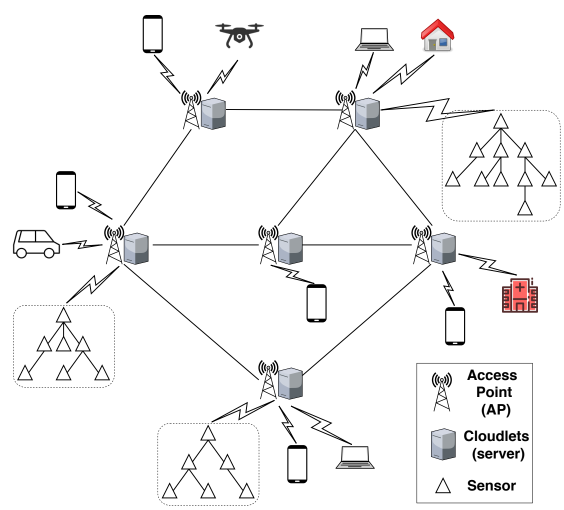


Fig. 1. An illustrative example of an MEC network that consists of 6 APs with 6 cloudlets co-located with the APs. There is an IoT application that consists of three streaming data sources with each coming from a different sensor subnetwork.

B. Problem definitions

Given an MEC network $G(V, E)$, we consider an IoT application, where it provides continuing surveillance to monitor multiple geographical areas in a metropolitan region, and different types of IoT sensory devices are deployed in the monitoring areas. The generating data streams from these devices can be aggregated within the IoT subnetwork and then be uploaded to the MEC through gateways (the access points of the MEC). We assume that there is a destination in the MEC for the aggregated data storage where all updated data from the gateways (APs) of the application will be stored for user ad hoc queries. When the stream data from each gateway is routed to the destination node d, the data stream may be merged or aggregated with the data streams from the other gateways. However, each data stream from a gateway to the destination must pass through the specified VNF instances of the SFC in order to ensure that the security and privacy of the streaming data and network performance. Thus, the VNF instances of each function of the SFC must be instantiated in cloudlets of the routing paths and some of the VNF instances can be shared among data streams from different gateways of the IoT application. For the sake of convenience, in the rest of our discussion, we assume that there are sufficient computing and bandwidth resources in the MEC to accommodate the VNF instances and data stream transfers for the IoT application. In summary, we aim to build a VNF network slicing for the IoT application such that the operational cost of implementing the IoT application is minimized, in terms of both computing and communication resource consumptions. We refer to this IoT-driven optimization problem as *the cost minimization problem of IoT-driven services in MEC*. An illustrative example of such an IoT application is given in Figure 1.

More specifically, the problem is defined as follows. Given an MEC network $G = (V, E)$, where V is the set of access points (APs) and E is the set of optical links that connect APs. Cloudlets are co-located with all APs. Consider an IoT application with a service function chain (SFC) requirement, where its data streams from its IoT devices will be aggregated at K given gateways (APs) independently and then are uploaded to the MEC for further processing and storage. Assuming that the data stream rates at the K gateways are $\rho_1, \ldots, \rho_K$, respectively. All streaming data finally will be merged or aggregated at a given destination node d (the home of sensory data of the IoT application) in the MEC. The routing path of the stream data from each gateway to the destination must pass through all VNF instances of the SFC in their specified order. We assume that each VNF instance of a service function $f \in \mathcal{F}$ consumes the amount c_f computing resource and has the processing rate μ_f.

Definition 1. Given an MEC $G = (V, E)$, and an IoT application with a Service Function Chain (SFC) requirement $\langle f_1, f_2, \ldots f_L \rangle$, in which there is a subset $S = \{s_1, s_2, \ldots, s_K\} \subset V$ of sources and one destination d. Each source s_i has a data rate ρ_i with $1 \leq i \leq K$, *the cost minimization problem of IoT-driven services in MEC* is to find a data routing tree T in G rooted at $d \in V$ and spanning all nodes in S, and place the demanded number of VNF instances of each f_j in the SFC with $1 \leq j \leq L$ at some nodes in T such that (1) the operational cost of implementing the IoT application is minimized; (2) the data stream in the tree path from each s_i to the destination d must pass through the VNF instances of functions of the SFC in order; (3) there is a defined *data aggregation function* $g(v)$ at each non-leaf node v in T that consists of l variables: $g(v_1), g(v_2), \ldots, g(v_l))$, which are l children $v_1, \ldots, v_l$ of v, where the value of function $g(\cdot)$ can be the sum of the data rates of all data streams of its children, or the average of the data rates of its children, depending on the IoT application; and (4) the workload at each cloudlet (in terms of the total amount of computing resource consumed for hosting the VNF instances of different functions) is as balanced as possible.

The total operational cost of an implementation of an IoT application in the MEC consists of the computing cost, the communication cost, and the load-balancing cost, which are defined as follows.

The communication cost of transmitting a volume $\rho_{u,v}$ of data in a routing path in G between two nodes u and v is

$$C_{comm}(u, v) = \rho_{u,v} \cdot l(u, v) \cdot b_{comm}, \tag{1}$$

where $\rho_{u,v}$ is the accumulative volume of data streams transmitted along a shortest routing path in G between nodes u and u with length $l(u, v)$, and b_{comm} is the price of unit data transfer along a link.

Assume that the number of VNF instances of service function f_i in the SFC of the IoT application is deployed at node v in the data routing tree T with the accumulative data steam $g(v) = \rho_v$, then, the computing cost at node v for accommodating the VNF instances of f_i is

$$C_{comp}(v, f_i) = \lceil \frac{\rho_v}{\mu_{f_i}} \rceil \cdot c_{f_i} \cdot c_{comp}, \tag{2}$$

where c_{comp} is the price of a unit computing resource consumption at any cloudlet $v \in V$ in the network.

The cost minimization problem of IoT-driven services in MEC thus is to construct a data routing tree T in G such that the total operational cost of implementing the IoT application is minimized, in terms of computing and communication resource consumptions, and workload balance among cloudlets. Specifically, the mentioned costs for the construction of tree T are formulated as follows.

Computing cost: For a given node $v \in V(T)$, if f_j has VNF instances at v, then $C_{comp}(v, f_j) = \lceil \frac{\rho_v}{\mu_{f_j}} \rceil \cdot c_{f_j} \cdot c_{comp}$. If node v is a leaf node and assuming that node v is source s_i, then $\rho_v = \rho_i$; otherwise, assuming that v has l children $v_1, \ldots, v_l$, then $\rho_v = g(v) = G(g(v_1), \ldots, g(v_l))$, where $G(\cdot)$ is an aggregation function at node v, e.g., G can be a summation function and $\rho_v = \sum_{i=1}^{l} \rho_{v_i}$.

Communication cost: If $(u, v) \in E(T)$ and v is a leaf node of T, then $C_{comm}(u, v) = \rho_v \cdot l(u, v) \cdot b_{comm}$; otherwise ($u$ is the parent of v in T), $C_{comm}(u, v) = g(v) \cdot l(u, v) \cdot b_{comm}$.

Load-balance cost: As the workload W_v of each cloudlet $v \in V$ is the actual computing consumption of cloudlet v that is proportional to the processing delay of data streams passing through the VNF instances at node v, it is well-known that a heavily loaded cloudlet usually has a much longer processing delay, compared to a light-loaded cloudlet, thereby resulting in a longer service delay, a penalty cost of this delay should be taken into account. We thus refer to *the workload utility* W_v/C_v at each cloudlet $v \in V$ as its load balance cost, which is defined as follows.

$$cost_{load}(v) = \beta_{load} \cdot \frac{W_v}{C_v} = \frac{\beta_{load} \cdot \sum_{I(v,f_i)=1} \lceil \frac{g(v)}{\mu_{f_i}} \rceil \cdot c_{f_i}}{C_v} \quad (3)$$

where β_{load} is a coefficient of the workload cost, and $I(v, f_i)$ is an indication function, which is 1 if there is any VNF instance of f_i in cloudlet v; otherwise 0, where $1 \leq i \leq L$ and $v \in V$.

The optimization objective of the cost minimization problem of IoT-driven services in MEC thus is to find a data routing tree T in G for the IoT application such that the operational cost $c(T)$ of T is minimized, where

$$\begin{aligned} c(T) = & \sum_{v \in V(T)} \sum_{j=1}^{L} \lceil \frac{\rho_v}{\mu_{f_j}} \rceil \cdot c_{f_j} \cdot c_{comp} + \\ & \sum_{(u,v) \in E(T)} \rho_{u,v} \cdot l(u,v) \cdot b_{comm} + \sum_{v \in V(T)} \frac{W_v}{C_v} \cdot \beta_{load} \\ = & \sum_{v \in V(T)} \sum_{j=1}^{L} C_{comp}(v, f_i) + \sum_{(u,v) \in E(T)} C_{comm}(u,v) \\ & + \sum_{v \in V(T)} cost_{load}(v) \quad \text{by Eq.(1), Eq.(2), and Eq.(3).} \end{aligned} \quad (4)$$

Notice that the solution of the problem consists of not only a data routing tree T but also the placement of the number of VNF instances of each function f_i in the SFC at each node v in T.

So far, we have formulated a novel IoT-driven service provisioning in an MEC for an IoT application, through performing NFV-enabled network slicing in the MEC to accommodate the IoT service.

IV. NP-HARDNESS OF THE DEFINED PROBLEM

In this section, we show that the defined problem is NP-hard.

Theorem 1: The cost minimization problem of IoT-driven services in an MEC in $G(V, E)$ is NP-hard.

Proof We consider a very special case of the problem of concern, where load-balance at each cloudlet will not be considered, and the data stream rates of all sources of an IoT application are identical, i.e., $\rho = \rho_1 = \rho_2 = \ldots = \rho_K$, the aggregated function $g(\cdot)$ at each node is the average of the sum of data rate of the children of the node, i.e., $g(v) = \frac{\sum_{i=1}^{l} g(v_i)}{l} = \rho$ where node v has l children $v_1, v_2, \ldots, v_l$, and all VNF instances of each service functions in the SFC are instantiated at the destination node. Thus, this special cost minimization problem of IoT enabled IoT services then is equivalent to the Steiner tree problem in G, that is to find a Steiner tree rooted at the destination node d and spanning all source nodes in $S \subset V$ such that the sum of the edges in the tree is minimized. It is well known that the Steiner tree problem is NP-hard, the cost minimization problem of IoT-enabled services in MEC thus is NP-hard, too.

V. A NOVEL FRAMEWORK FOR IOT-DRIVEN SERVICE PROVISIONING IN MEC

In this section we provide a generic IoT-driven service provisioning framework for IoT-driven applications in an MEC. We consider an IoT application that consists of multiple sources, where each service is a base station of a sub-sensor network in which the streaming sensory data from the sensors are collected at the source (or the gateway) and will be uploaded to the MEC network for further processing and storage. We assume that there is a destination node in the MEC for the aggregated data storage.

To ensure the security and privacy of transferring data stream, each data stream from its source to the destination must pass through a specified service function chain as the requirement of the IoT application. For example, in a metropolitan region, there are many public parks, assume that a sensor network is deployed for each park, and the monitored data stream from each of the sensor networks will be uploaded to the MEC through its nearby AP, while all collected data will be finally stored at the destination for further processing and interrogated by ad hoc queries.

A naive approach for sensory data collection from sources to the destination is to build a routing path in the MEC from each source (an AP) to the destination, and place the demanded number of VNF instances of each function in the SFC along the cloudlets in the path to meet the data rate of the source. As this IoT-driven service in an MEC is expected for a long run, which implies that the owner of the IoT application will pay the IoT service with the cost that is proportional to the resource occupied by the service, or a network slicing for the IoT application is needed. However, running IoT service through this naive method may not be economical, because the data streams from different sources can be aggregated or merged through exploring temporal-spatial data correlations, A much less accumulative volume of data stream from these sources can be routed to the destination, thereby saving the communication cost. On the other hand, the certain number of VNF instances for each function in the SFC for the data rate of each source must be instantiated along the nodes in its routing path, while the number of VNF instances of each function in the SFC for the aggregated data stream could be significantly reduced if data stream aggregation or merge operation at each intermediate node is performed, implying that less computing resources for VNF instance instantiations are needed.

To provide a cost-effective IoT-driven service in MEC for an IoT application, a data routing tree rooted at the destination and spanning all source nodes in the MEC can be built to reduce both computing and communication costs through data aggregation at paths and VNF instance instantiations, where each non-leaf node v in the tree except the destination

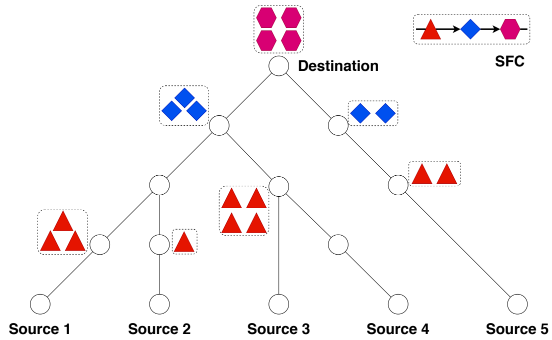


Fig. 2. An illustrative example of a data routing tree for an IoT application with three service functions in its SFC, and there are 5 sources $s_1, s_2, \ldots, s_5$, and the data stream from each source to the destination (the tree root)must pass through the number of VNF service instances of each service function in the SFC.

has a parent node and a set of children nodes. There is an aggregation function $g(\cdot)$ at v that aggregates the data streams from its children, e.g., a simple aggregation function is the rate sum of the data streams of its children. Also, less numbers of VNF instances of some service functions in the SFC can be instantiated at node v. However, the data stream along the unique routing path in the tree from each source to the destination must pass through its demanded number of VNF instances in the SFC. Thus, both communication and computing resources for the IoT application can be shared among the data streams from its different sources through the data routing tree. Figure 2 is an illustrative example of a data routing tree for an IoT application.

VI. Algorithm for the Cost Minimization Problem

Since the cost minimization problem of IoT-driven services in MEC is NP-hard, we aim to develop a heuristic algorithm for the problem. In this section we first provide an efficient algorithm for the problem, based on the built service framework. We then analyze the properties of the solution, and we finally analyze the time complexity of the proposed algorithm.

A. Overview of the proposed algorithm

The proposed algorithm proceeds greedily. That is, for each source node $s \in S$, a data routing tree $T(s)$ is constructed by adding first source node $s \in S$ to the tree, followed adding other source nodes to $T(s)$ greedily, one by one. Each time, a source node with the minimum cost increment compared with the previous cost of $T(s)$ will be chosen to add to the tree until all source nodes are added to the tree. Thus, a data routing tree $T(s_0)$ with the minimum total operational cost is chosen from the $|S|$ trees, which will be the solution to the problem. Specifically, we first choose a node, e.g., $s_{i_1} \in S$, as the very first node to add the data routing tree $T(s_{i_1})$. To this end, we find a shortest path P_1 in G from s_{i_1} to the destination d, and place the VNF instances of each function $f \in SFC$ to the nodes along path P_1 while balancing the workload of the nodes in the path (the detailed VNF instance placement will be discussed later), where the number of VNF instances of each function in the SFC is determined by the data rate ρ_{s_1} of source s_{i_1}. Let $S_1 = S \setminus \{s_{i_1}\}$; the rest source nodes then is added to the tree $T(s_{i_1})$ iteratively, one by one. Having finished the first $(k-1)$th iterations, a partial data routing tree $T(s_{i_1})$ that contains $k-1$ source nodes has been constructed. Within iteration k with $2 \leq k \leq K$, a source node s_{i_k} in $S_k = S \backslash \{s_{i_1}, s_{i_2}, \ldots, s_{i_{k-1}}\}$ with the *minimum total cost gain* will be added to tree $T(s_{i_1})$. Specifically, let $V(T_{k-1})$ be the set of nodes in $T(s_{i_1})$ after the first $k-1$th iterations, for each node $u \in V_{T_{k-1}}$, find a shortest path in $G \setminus T(s_{i_1})$ between s_{i_k} and u for any $s_{i_k} \in S \setminus \{s_1, s_{i_2}, \ldots s_{i_{k-1}}\}$, then compute the potential total cost gain (by increasing the number of VNF instances or sharing existing VNF instances, or instantiating new VNF instances, and the communication cost from source s_{i_k} to the destination d), then choose one source node $s_{i_{k_0}}$ with the minimum total cost gain to be added to $T(s_{i_1})$. Notice that choosing such a node s_{i_0} with the minimum total cost gain is difficult as this subproblem is NP-hard too, we instead choose a node with the minimum total cost gain approximately. This procedure continues until all source nodes in S are added to tree $T(s_{i_1})$. Thus, there are $|S|$ data routing trees constructed, and among the $|S|$ routing trees, one tree $T(s_0)$ with the minimum total cost is chosen as the solution to the problem.

B. The construction of a routing tree rooted at d and source s being the first joined in

Let $T(s)$ be a data routing tree where source s is the first source joining into the tree, and let $T(s,k)$ be a partial routing tree of $T(s)$ by adding the first k source nodes into it so far, where $1 \leq k \leq K$. Then, some VNF instances of each function in the SFC have been instantiated into some nodes in the tree. We claim that the data stream along the tree path from each source node to the destination must pass through its demanded number of VNF instances of each function in the SFC and they pass through only once, i.e., no VNF instances of each function is wrongly placed, or placed in multiple nodes in the path.

Let $T(s,k)$ be a partial routing tree of $T(s)$ containing the first k source nodes. When $k=1$, the partial routing tree $T(s,1)$ is a shortest path P_1 in G from s to d where $(v_i, v_{i+1}) \in P_1$ with $v_1 = s$, $v_{|P_1|+1} = d$, and v_{i+1} being the parent of v_i in the tree. Each edge $e_i = (v_i, v_{i+1})$ in P_1 has a communication cost $b_{comm} \cdot l(v_i, v_{i+1}) \cdot g(v_i) = b_{comm} \cdot l(v_i, v_{i+1}) \cdot \rho_s$, and the number of VNF instances of each service function f_j for source s is $\lceil \frac{\rho_s}{c_{f_j}} \rceil$, which will be deployed to some of the nodes along path P_1, i.e., the placement order of these VNF instances in P_1 from s to d is $f_1, f_2, \ldots, f_L$. We will deal with the VNF instance placement later, by incorporating the workload balancing among cloudlets in P_1. It can be seen that the claim holds for this initial tree construction.

We assume that $T(s,k-1)$ has been constructed, and the $k-1$ data streams from the first $k-1$ added sources to the tree pass through the demanded numbers of VNF instances of each function in the SFC, through their unique routing paths in the tree, and for each edge $(v_i, v_{i+1}) \in T(s,k-1)$, the accumulative data stream on the edge is $g(v_i)$, where $g(\cdot)$ is an aggregation function at node v. We now construct $T(s,k)$ by adding a new source node s'_k to the tree.

Let $P_k(u, s'_k)$ be a shortest path in $G \backslash T(s,k-1)$ between a tree node u and a source node s'_k that has not been contained in $T(s,k-1)$ yet. Let $P^T_k(u,d)$ be the unique tree path in $T(s,k-1)$ between node u and d and the VNF instances of functions of the subchain $\langle f_j, f_{j+1}, \ldots, f_K \rangle$ of the SFC in $P^T_k(u,d)$, i.e., no VNF instances of function f_l with $1 \leq l < j$ are deployed in any node in $P^T_k(u,d)$. Then, for the VNF instances of $f_{j'}$ placed at a node v in path $P^T_k(u,d)$, the newly added number of VNF instances of $f_{j'}$ at v is calculated as follows.

$$N(v, f_{j'}, s'_k) = \lceil \frac{\rho_{s'_k} - r_{j'}}{\mu_{f_{j'}}} \rceil, \quad (5)$$

where $r_{j'}$ is the residual processing capacity in the last VNF instance of $f_{j'}$ with $j \leq j' \leq L$. A more efficient way is to recompute $g(v)$ of node v in the tree by taking into account source node s'_k. That is, if $v \neq u$ in $P^T_k(u,d)$, then the number of VNF instances of $f_{j'}$ is $\lceil \frac{g(v)}{\mu_{f_{j'}}} \rceil$. Otherwise, $g(u)$ is recomputed, by incorporating the data stream $\rho_{s'_k}$ as its new child from the branch $P_k(u, s'_k)$, then the total number of VNF instances of $f_{j'}$ at u is $\lceil \frac{g(u)}{\mu_{f_{j'}}} \rceil$.

For the data stream from s'_k to d along a routing path from s'_k to d which consists of two segments $P^T_k(u,d)$ and $P_k(s'_k, u)$. Although the data stream along the tree path $P^T_k(u,d)$ has been processed by the VNF instances in the subchain $\langle f_j, \ldots, f_L \rangle$ in $P^T_k(u,d)$, the data stream along path $P_k(s'_k, u)$ from node s'_k to node u has not been processed by the VNF instances of each function in the subchain $\langle f_1, f_2, \ldots, f_{j-1} \rangle$ of the SFC. We thus need to place the number of VNF instances for each function of this subchain in the nodes in $P_k(s'_k, u)$ in their specified order, where the rate of data stream on each edge in $P_k(s'_k, u)$ is $\rho_{s'_k}$). In the following, we show how to place the number of VNF instances of each of functions in $\{f_1, f_2, \ldots, f_{j-1}\}$ to which cloudlets (nodes) in $P_k(s'_k, u)$ to meet the processing capacity of the data stream while balancing the workload among the cloudlets, by developing a dynamic program solution for this workload-aware VNF instance placement problem.

C. Workload-aware VNF instance placement in a routing path

We here consider the workload-aware VNF instance placement on the nodes in a routing path. We assume that there is a data stream $\rho_{s'_k}$ from source node s'_k with data rate $\rho_{s'_k}$, the number of VNF instances of each function $f_{j'}$ in the SFC needs to be placed to a node in path $P_k(u, s'_k)$ in order, where $j-1 \leq j' \leq 1$. We aim to place the VNF instances such that the maximum utility ratio among the nodes is minimized, by formulating the following workload-aware VNF instance placement problem.

Let $u_0, u_1, u_2, \ldots, u_p$ be the node sequence in path $P_k(s'_k, u)$ starting from u with $u_0 = u$ and $u_p = s'_k$ and $p \geq 1$, where the computing workload and computing capacity of each cloudlet node $u_i \in V$ are W_{u_i} and C_{u_i}, respectively, or *the utility ratio* of node u_i is

$$\gamma_{u_i} = \frac{W_{u_i}}{C_{u_i}}. \quad (6)$$

There is a subchain $\langle f_{j-1}, f_{j-2}, \ldots, f_1 \rangle$ of the SFC for the data stream from source s'_k with data rate $\rho_{s'_k}$. Equivalently, the number of VNF instances of $f_{j'}$ to be placed along path $P_k(u, s'_k)$ is $\lceil \frac{\rho_{s'_k}}{\mu_{f_{j'}}} \rceil$, and the amounts of computing resource consumed by the VNF instances of $f_{j'}$ are

$$w_{f_{j'}} = \lceil \frac{\rho_{s'_k}}{\mu_{f_{j'}}} \rceil \cdot c_{f_{j'}}, \quad (7)$$

where $\mu_{f_{j'}}$ the data processing capacity of an VNF instance of function $f_{j'}$ in any cloudlet and $1 \leq j' \leq j-1$.

Let $y_1, y_2, \ldots, y_q$ represent the function sequence $\langle f_{j-1}, f_{j-2}, \ldots, f_1 \rangle$ with $y_1 = f_{j-1}$ and $y_q = f_1$. Each $y_{j'}$ has a workload $w_{y_{j'}}$ by Eq. (7) with $1 \leq j' \leq j-1$. *The workload-aware VNF instance placement problem* is to place the VNF instances of functions in the subchain $\langle y_1, y_2, \ldots, y_q \rangle$ to the nodes in path $u_1, u_2, \ldots, u_p$ such that the maximum utility ratio among the nodes is minimized, assuming that there is sufficient computing resource at each cloudlet u_i to accommodate all VNF instances of function $f_{j'}$ in the subchain with $1 \leq i \leq p$ and $1 \leq j' \leq q$.

We here devise an optimal algorithm for the workload-aware VNF instance placement problem, using dynamic programming as follows.

Let $U_i = u_1, u_2, \ldots, u_i$ be the subsequence of $u_1, \ldots, u_p$ and $Y_j = y_1, y_2, \ldots, y_j$ the subsequence of $y_1, \ldots, y_q$ with $1 \leq i \leq p$ and $1 \leq j \leq q$. Denote by $B(i,j)$ the optimal workload utility ratio of placing the VNF instances of functions in Y_j to nodes in the node subsequence U_i. Then,

$$B(i,j) = \min \begin{cases} \gamma_{u_1} + \frac{\sum_{l=1}^{j} w_{y_l}}{C_{u_1}} & i = 1, \\ \{\gamma_{u_{i'}} + \frac{w_{y_1}}{C_{u_{i'}}} \mid 1 \leq i' \leq i\} & j = 1, \\ B(i-1, j) & i > 1, \\ B(i-1, j-1) + (\gamma_{u_i} + \frac{w_{y_j}}{C_{u_i}}) & j > 1, \end{cases}$$

where γ_{u_i} is defined in Eq. (6), $B(1,j) = \gamma_{u_1} + \frac{\sum_{l=1}^{j} w_{y_l}}{C_{u_1}}$ implies that there is only one cloudlet and the VNF instances of the chain will be hosted by the cloudlet; while $B(i,1) = \min\{\gamma_{u_{i'}} + \frac{w_{y_1}}{C_{u_{i'}}} \mid 1 \leq i' \leq i\}$ indicates that the VNF instances of a single function is added to such a cloudlet that results in the minimum increase of the utility ratio among the cloudlets. $B(i,j) = B(i-1,j)$, implying that the VNF instances of the subchain are placed to the first $i-1$ nodes in the node sequence U_i.

The optimal solution to the workload-aware VNF instance placement problem thus is $B(p,q)$. The time complexity of the proposed dynamic program algorithm is $\mathcal{O}(p \cdot q) = \mathcal{O}(|V| \cdot L)$ as $p \leq |V|$ and $q \leq |SFC| = L$. The detailed algorithm is given in `Algorithm` 1.

Algorithm 1 Algorithm for the workload-aware VNF instance placement problem.

Input: A node sequence $u_1, u_2, \ldots, u_p$, a function sequence $y_1, y_2, \ldots y_q$, and the date rate $\rho_{s'_k}$ of a source s'_k, assuming that the current workload W_{u_i} and capacity C_{u_i} of each cloudlet node u_i is given.
Output: the number of VNF instances of each function y_j is placed to a node in the node sequence such that the function order does not change and the maximum load ratio among the nodes is minimized.

for $i \leftarrow 1$ to p **do**
  for $j \leftarrow 1$ to q **do**
    if $i = 1$ **then**
      $B(i,j) \leftarrow \gamma_{u_1} + \frac{\sum_{l=1}^{j} w_{y_l}}{C_{u_1}}$;
    else
      if $j = 1$ **then**
        $B(i,j) \leftarrow \{\gamma_{u_{i'}} + \frac{w_{y_1}}{C_{u_{i'}}} \mid 1 \le i' \le i\}$;
      else
        $B(i,j) \leftarrow \min\{B(i-1,j),\ B(i-1,j-1)+(\gamma_{u_i}+\frac{w_{y_j}}{C_{u_i}})\}$;
      end if;
    end if;
  end for;
end for;
return An assignment of VNF instances of y_j to node u_i and the maximum load ratio $B(p,q)$ among the nodes is the minimum one.

It can be seen that the data stream from s'_k to the destination d passes through the demanded VNF instances of each function f_l in the SFC with $1 \le l \le L$. While the data streams of the first $(k-1)$ sources in tree $T(s,k)$ pass through their demanded numbers of VNF instances of each function in the SFC, following the proposed tree construction.

The total cost gain $\Delta(s,u,s'_k)$, by connecting source s'_k to $T(s,k-1)$ through the tree node u to form the next potential routing tree $T(s,u,s'_k)$, is calculated as follows.

$$\begin{aligned}
&\Delta(s,u,s'_k)\\
=&c(T(s,s'_k,k)) - c(T(s,k-1))\\
&T(s,s'_k,k) = T(s,k-1) \cup P_k(s'_k,u)\\
=&\sum_{l=1}^{L}(\text{\# new VNFs } f_l)\cdot c_{f_l}\cdot c_{comp} + \rho_{s'_k}\cdot l(s'_k,d)\cdot b_{comm}\\
&+\sum_{v\in V(T)}\frac{W'_v - W_v}{C_v}\cdot\beta_{load},
\end{aligned} \tag{8}$$

where $c(T)$ is the operational cost of a data routing tree T in terms of the defined computing cost, the communication cost, and the load balance cost. W_v and W'_v are the workloads of cloudlet $v \in V(T)$ before and after source s'_k joins in tree $T(s,k-1)$, and $W'_v - W_v \le \rho_{s'_k}$.

The detailed algorithm for the construction of a potential data routing tree $T(s,u,s'_k)$ rooted at d with the first joining in source s, and source s'_k is the k source joining in the tree via a tree node u is given in Algorithm 2, where $1 \le k \le K$.

D. Algorithm for the cost minimization problem

In the following, we propose an efficient algorithm for the cost minimization problem of IoT-enabled services in MEC, by making use of Algorithm 2 as its subroutine. The detailed algorithm for the cost minimization problem is presented in Algorithm 3.

Algorithm 2 A potential routing tree construction $T(s,u,s'_k)$.

Input: An MEC network $G(V,E)$ and an IoT application with an $SFC = \langle f_1, f_2, \ldots, f_L\rangle$ and K sources $s_1, s_2 \ldots, s_K$ and a destination $d \in V$, each source s_i has a data rate ρ_i with $1 \le i \le K$.
Output: A potential partial routing tree rooted at d and spanning k source nodes with the first joining in source s, and assuming that the first $(k-1)$ sources are added already and the partial routing tree is $T(s,k-1)$, a routing tree (s,u,s'_k) is constructed with the minimum cost where source s'_k is joining to the tree through a path $P_k(s'_k,u)$ and u is the tree node.

$c(T(s,u,s'_k)) \leftarrow c(T(s,k-1))$; /* the total cost of a solution */
if $k = 1$ **then**
  Find a shortest path P_1 in G from s'_k to d;
  Place the demanded number of VNF instances of f_j to the nodes along path P_1 with $1 \le j \le L$ by invoking Algorithm 1;
else
  Update the number of VNF instances of each function f_j at each node v in the tree path $P_k^T(u,d)$ from u to d by Eq. (5) having considering to add extra $\rho_{s'_k}$ data stream to each edge along path $P_k^T(u,d)$;
  /* assuming f_{j-1} is the first service function in the subchain from f_1 to f_{j-1} whose VNF instances do not appear in a node in $P_k^T(u,d)$; */
  Place the demanded number of VNF instances of each function $f_{j'}$ with $1 \le j' \le j-1$ to the nodes of a shortest path $P_k(s'_k,u)$ in $G \setminus T(s,k-1)$ from s'_k to u' with the data rate $\rho_{s'_k}$ by invoking Algorithm 1, where (u,u') is the tree edge with endpoints u and u', and u' is a child of u in the tree, by balancing the workload among the nodes in the path;
  $T(s,u,s'_k) \leftarrow T(s,k-1) \cup P_k(s'_k,u)$;
  $\Delta(s,u,s'_k) \leftarrow c(T(s,u,s'_k)) - c(T(s,k-1))$; /* compute the cost gain */
end if;
return The solution $T(s,u,s'_k)$ and its cost $c(T(s,u,s'_k))$.

E. Analysis of the proposed algorithm

In the following, we first observe some important property of the data routing tree. Recall that $f_1, f_2, \ldots f_L$ are the service functions in the SFC of the IoT application.

Lemma 1: Let $e_1, e_2, \ldots, e_q$ be the routing path in the data routing tree T from a source node s to the destination d, and let $v_1, v_2, \ldots, v_{q+1} = d$ are the corresponding nodes in the path. We have

- (i) $\rho_{e_1} \le \rho_{e_2} \le \ldots \le \rho_{e_q}$ where ρ_e is the volume of data stream on edge e;
- (ii) The VNF instances of each function $f_i \in SFC$ are installed in the nodes in its specified order as well with $1 \le i \le L$. In other words, it does not allow the following situations to happen, if the VNF instances of f_k deployed in node v_j, then the VNF instances of f_l cannot be deployed to any node $v_{j'}$ with $j' > j$ if $l < k$, we term this as the VNF deployment order by the SFC;
- (iii) The VNF instances of any service function $f \in SFC$ cannot appear in multiple nodes in any routing path in the tree from a source to the destination, i.e., it is prohibited that VNF instances of function f are deployed into two different nodes v_i and v_j in the routing path with $i \ne j$. Otherwise, the data stream data will pass through the VNF instances of f twice from its source to the destination.

Proof (i) Let $e_i = (v_i, v_{i+1})$ with $s = v_1$ and $d = v_{q+1}$, then v_{i+1} is the parent of v_i in a data routing tree T, $\rho_{e_{i+1}} = g(v_i)\cdot b_{comm}\cdot l(v_i, v_{i+1})$. Following the construction of T, we

Algorithm 3 Heuristic algorithm for the cost minimization problem of IoT-enabled services in MEC.

Input: An MEC network $G(V,E)$ and an IoT application with an $SFC = \langle f_1, f_2, \ldots, f_L \rangle$ and K sources $s_1, s_2 \ldots, s_K$ and a destination $d \in V$, each source s_i has a data rate ρ_i with $1 \le i \le K$.

Output: A data routing tree rooted at d and spanning all source nodes, and different numbers of VNF instances of the SFC are placed in the nodes of the tree such that the total operational cost outing of the IoT application is minimized.

$cost \leftarrow \infty$; /* the total operational cost of the solution */
$T \leftarrow \emptyset$;
for $s \in S$ **do**
    /* The construction of a routing tree $T(s)$ with the minimum cost;*/
    Find a shortest path P_1 in G from s to d; $T(s) \leftarrow P_1$;
    place the VNF instances of service function in the SFC along the nodes in P_1 with the data rate ρ_s;
    $T(s) \leftarrow P_1$;
    $S' \leftarrow S \setminus \{s\}$;
    while $S' \neq \emptyset$ **do**
        for each $u \in T(s)$ **do**
            for each $s' \in S'$ **do**
                Find a shortest path $P_{u,s'}$ in $G \setminus T(s)$ between u and s'; and explore the sharing of existing VNF instances and instantiating new VNF instances of SFC in tree $T(s) \cup P_{u,s'}$ by calling `Algorithm` 2 ($T(s,u,s')$);
                Compute the total cost gain $\Delta(s,u,s')$ if path $P_{u,s'}$ is added to $T(s)$;
            end for;
        end for;
        $u_0, s'_0 \leftarrow \min \arg_{u \in T(s), s' \in S'} \{\Delta(s,u,s')\}$;
        $T(s) \leftarrow T(s) \cup P_{u_0,s'_0}$; /* adding a source s_0 with the minimum cost gain to the data routing tree for the IoT application */
        $S' \leftarrow S' \setminus \{s'_0\}$;
    end while;
    if $c(T(s'_0)) < cost$ **then**
        $T \leftarrow T(s'_0)$;
        $cost \leftarrow c(T(s'_0))$;
        $s_0 \leftarrow s'_0$;
    end if;
end for;
return the solution $T(s_0)$ with the total operational cost $c(T(s_0))$.

have $g(v_{i+1}) \ge g(v_i)$ as v_i is a child of node v_{i+1} in the tree, and the aggregation function at any node v will aggregate the data streams from the data streams of all its children. Thus, $\rho_{e_{i+1}} = g(v_i) \ge \rho_{e_i} = g(v_i)$.

By Property (ii) of tree T, if some VNF instances of a function f_i are instantiated at node v_i, then any of its other VNF instances cannot be deployed to other nodes in the subtree rooted at v_i; otherwise, Property (iii) is violated, i.e., the data stream from a descendent leaf node of v_i will pass through the VNF instances of the service function twice at two different cloudlets (at that node and at v_i), and this is prohibited by the SFC requirement of the IoT application.

Lemma 2: Let $T(s,k)$ is a partial data routing tree constructed that contains the first $k-1$ source nodes by `Algorithm` 2, where source s is the first one added to the tree, for all k with $1 \le k \le K$. Then, the data stream along the tree path from each of the sources to the destination will pass through the demanded number of VNF instances of each service function in the SFC, and the solution delivered is feasible.

Proof The claim can be shown by induction. Following `Algorithm` 2, when $k=1$, the tree is a shortest path from s to d and the number of VNF instances of each service function in the SFC are placed to the nodes in the path, and the claim clearly holds. We assume that the first $k-1$ sources have been added to the data routing tree and the claim holds for the data stream from each of the sources. Now, we show the claim still hold when adding the kth source into the tree. Following the algorithm, the number of VNF instances of each service function in the SFC is increased or instantiated accordingly to meet the data rate of data stream from the kth source. The claim still holds.

Lemma 3: The lower bound C_{opt_lb} on the total operational cost of an optimal solution for the cost minimization problem of IoT enabled services in MEC is

$$C_{opt\_lb} = \sum_{j=1}^{L} \lceil \frac{\sum_{s_i \in S} \rho_{s_i}}{\mu_{f_j}} \rceil \cdot c_{f_j} \cdot c_{comp} + \sum_{s_i \in S} l(s_i,d) \cdot \rho_{s_i} \cdot b_{comm} + \sum_{v \in V} \frac{\sum_{l=1}^{L} \lceil \frac{\sum_{s \in S} \rho_s}{\mu_{f_l}} \rceil \cdot c_{f_l}}{\sum_{v \in V} C_v} \cdot \beta_{load}, \tag{9}$$

if there is no aggregation at each node in the data routing tree T on the accumulative data stream on it, i.e., for any non-leaf node v in T with l children $v_1, v_2, \ldots, v_l$, $g(v) = \sum_{i=1}^{l} g(v_i)$, otherwise (node v is a leaf node), $g(v) = \rho_v$, where S is the set of K sources of an IoT application, $s_i \in S$ is a source with data rate ρ_{s_i}, $l(s_i,d)$ is the shortest path in the MEC G between nodes s_i and d, and the length of the SFC of the IoT application is L, the unit cost for computing and bandwidth are c_{comp} and b_{comm}, respectively, and β_{load} is the unit weight cost of workload at any cloudlet.

Proof As the data stream from each source $s_i \in S$ must reach the destination d, the most economic way is to route the data along a shortest path between the two nodes and the length of the path is $l(s_i,d)$, while the bandwidth cost per unit data rate is b_{comm}. Thus, the minimum communication cost is $\sum_{s_i \in S} l(s_i,d) \cdot \rho_{s_i} \cdot b_{comm}$. On the other hand, as each data stream must pass through the VNF instances of each function f in the SFC of length L while each VNF instance has its processing capacity μ_f, the lower bound on the number of VNF instances for f_j thus is $\lceil \frac{\sum_{s_i \in S} \rho_{s_i}}{\mu_{f_j}} \rceil$. Thus, the lower bound on the total computing cost for the problem is

$$\sum_{j=1}^{L} \lceil \frac{\sum_{s_i \in S} \rho_{s_i}}{\mu_{f_j}} \rceil \cdot c_{f_j} \cdot c_{comp}. \tag{10}$$

The lower bound estimation of the workload on each cloudlet is difficult. We here assume that the total workload is evenly distributed among the cloudlets in the MEC network such that all cloudlets have identical workloads. That is, the lower bound on the load balancing cost of all cloudlets thus is calculated as follows.

$$\sum_{l=1}^{L} \lceil \frac{\sum_{s \in S} \rho_s}{\mu_{f_l}} \rceil \cdot c_{f_l} = \varphi \cdot \sum_{v \in V} C_v, \tag{11}$$

where φ is the average workload factor among the cloudlets.

From Eq. (11), the value of φ is obtained. The lower bound on the load balancing cost among cloudlets thus is

$$\sum_{v \in V} \varphi \cdot \beta_{load} = \sum_{v \in V} \frac{\sum_{l=1}^{L} \lceil \frac{\sum_{s \in S} \rho_s}{\mu_{f_l}} \rceil \cdot c_{f_l}}{\sum_{v \in V} C_v} \cdot \beta_{load}. \tag{12}$$

The lemma then follows.

Theorem 2: Given an MEC network $G(V, E)$ and an IoT application that consists of K gateways $s_1, \ldots, s_K$ and a destination $d \in V$, where each gateway $s_i \in V$ has a data rate ρ_i with $1 \leq i \leq K$, there is an SFC requirement $f_1, \ldots, f_L$ for the IoT application and a defined data aggregation function $g(\cdot)$, the cost minimization problem of IoT-driven services in G is to find a data routing tree T in G for implementing the IoT application such that the total operational cost is minimized, in terms of the computing cost, communication cost, and workload balancing cost. There is an efficient algorithm, `Algorithm` 3, which deliver a feasible solution for the problem. The algorithm takes $\mathcal{O}(n^2 \cdot |S|^2 + n^3)$ time, where $n = |V|$ and $|S| = K$ with $S = \{s_1, s_2, \ldots, s_K\}$.

Proof We first show the solution delivered is feasible by Lemma 1 and Lemma 2. That is, the data stream from each source $s' \in S$ to the destination must pass through the demanded number of VNF instances of each function in the SFC prior to the destination d. Let a data routing tree $T(s)$ rooted at d is chosen as the solution of the problem, it can be seen that the construction of $T(s)$ proceeds iteratively. Within each iteration, a source node joins in the tree. Let $T(s, k)$ be the partial data routing tree of $T(s)$ that contains the first k source nodes. Clearly, for tree $T(s, 1)$, the data stream ρ_s is routed along a shortest path (i.e., $T(s, 1)$) in G from s to d, and the demanded number of VNF instances of each service function in the SFC is instantiated in the nodes along the path. Assume that all data streams from the sources in the partial data routing tree $T(s, k-1)$ meet the condition: their demanded numbers of VNF instances of each service function in the SFC are placed in nodes of $T(s, k-1)$ already. When we consider the k source s'_k joining in $T(s, k-1)$ to form tree $T(s, k)$, assume that node u is the tree node at which a branch from s'_k to u is attached to $T(s, k-1)$. It can be seen that the data stream with data rate $\rho_{s'_k}$ along the path $P_k(s'_k, u) \cup P_k^T(u, d)$ from s'_k to d must pass through the demanded VNF instances of each service function in the SFC, following the construction of $T(s, k)$ by `Algorithm` 2. The solution delivered by `Algorithm` 3 thus is feasible.

We then analyze the time complexity of the proposed algorithm, Âlgorithm 3. There are $|S|$ potential data routing trees to be constructed, while the construction of a data routing tree starting with a source takes $\mathcal{O}(|V|^2 \cdot |S| + |V|^3)$ time, which includes VNF instance placements along the tree paths, by invoking the dynamic algorithm - `Algorithm` 1, and for any pair of nodes (one is in the partial tree and another is a source that has not been joining in the tree yet). Thus, the running time of `Algorithm` 3 takes $\mathcal{O}(n^2 \cdot |S|^2 + n^3)$ as there are $|S|$ potential data routing trees to be constructed.

Remarks: The proposed algorithm assumed that all VNF instances of any IoT application can be hosted by a single cloudlet, this assumption is reasonable in practice as the resource demands by a single IoT application should be far less than the resource of a cloudlet provided. However, it is not uncommon that the resources in MEC are very limited, the proposed algorithm can be easily modified for this resource restriction case. That is, if the residual resource of a cloudlet is less than the resource demands of an IoT application, that cloudlet will not involve resource allocation for the IoT application by hosting any of the VNF instances of the IoT application (i.e., it cannot accommodate any VNF instance in the data routing tree for the IoT application), the cloudlet, however, is still part of the data routing tree for the IoT application to maintain the network connectivity, i.e., there may be routing paths passing through the cloudlet.

VII. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm for the service provisioning for IoT applications in an MEC network. We also investigate the impact of important parameters - the number of service function in a service chain L and the load balance coefficient β_{load}, on the performance of the proposed algorithms.

A. Environment settings

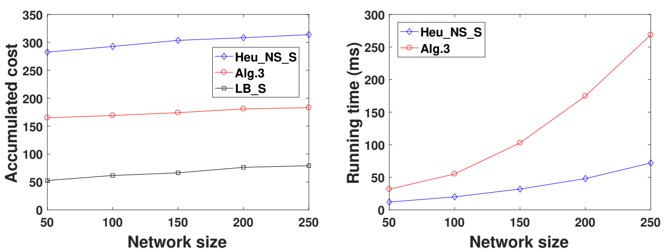
We consider an MEC network that consists of 100 APs, with each having a co-located cloudlet. We generate the topologies of MEC networks using GT-ITM [9]. The capacity of each cloudlet is set from $30,000$ to $60,000$ MHz [26]. We assume there are 10 types of VNFs, and the computing resource consumption of each VNF instance ranges from 20 to 100 MHz [16]. We further assume the processing rate of each VNF instance varies from 30 to 80 packet per millisecond, where each data packet is of size 64 KB [17]. The number of different types of SFCs is set as 20, and the length of each varies from 3 to 7 [18]. For each request, the number of data sources is set from 4 to 8, and the data rate of each source ranges from 2 to 10 packet per millisecond [18]. The communication cost of each link is randomly drawn from \$0.05 to \$0.4 per data packet [17]. The computing resource consumption cost is set at \$0.1 per MHz [26]. The coefficient of the workload balancing cost (i.e., β_{load}) is set at $2,000$.

We evaluate the proposed algorithm `Algorithm` 3 (referred to as `Alg.3`) against the lower bound on the optimal solution of the problem in Lemma 3 (referred to as `LB_S`). We also consider another benchmark for the problem, where there is a minimum cost routing path from each source to the destination, and the VNF instances of each function in the SFC will be instantiated along the path. In this heuristic, neither computing resources (VNF instances) nor bandwidth resource is shared among data streams from different sources. We refer to this heuristic as algorithm `Heu_NS_S`.

The value in each figure is the mean of the results out of 20 MEC network instances of the same size. The actual running time of each algorithm is based on a desktop with a 3.60 GHz Intel 8-Core i7-7700 CPU and 16 GB RAM. Unless specified, the above parameters are adopted by default.

B. Performance evaluation of algorithms for the cost minimization problem of IoT enabled services

We first studied the performance of `Alg.3` against the lower bound `LB_S` of the optimal solution of the problem and algorithm `Heu_NS_S`, respectively, by varying the network size from 50 to 250. Fig. 3(a) and 3(b) depict the total cost and running time of two comparison algorithms. It can be seen from Fig. 3(a) that when the network size is 250, the total cost achieved by `Alg.3` is 58.3% of that by algorithm `Heu_NS_S`, while the lower bound `LB_S` of the optimal cost is 43.2% of that by `Alg.3`. This is because `Alg.3` establishes an efficient data routing tree that jointly considers VNF instance sharing and communication path sharing among different data streams of the IoT application and the workload balancing among cloudlets.



(a) The total cost (b) The running time

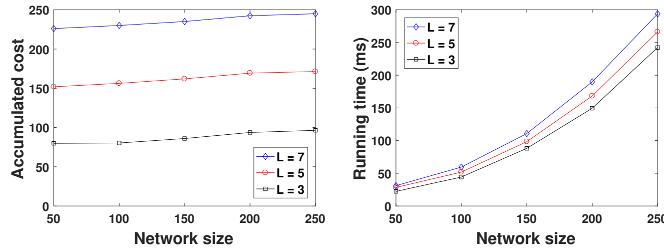
Fig. 3. Performance of different algorithms for the cost minimization problem, by varying the network size from 50 to 250.

C. Impact of important parameters on the performance of the proposed algorithm

We finally investigated the impact of important parameters on the performance of the proposed algorithms for the defined problem including the length L of SFCs, the number of sources K, and the coefficient of the workload balancing cost β_{load}.

We first studied the impact of the length L of SFCs on the proposed algorithms for the cost minimization problem. Fig. 4(a) and Fig. 4(b) depict the total cost and running time of the solution delivered by algorithm `Alg.3` for different network sizes when $L = 3$, 5 and 7, respectively. It can be seen from Fig.4(a) that with the network size of 250, the total cost of the solution delivered by algorithm `Alg.3` when $L = 3$ is 39.4% of itself when $L = 7$. This can be justified that with a small value of L, the IoT applications consumes less computing resource, thus, both the computing cost and the workload-balancing cost decrease.

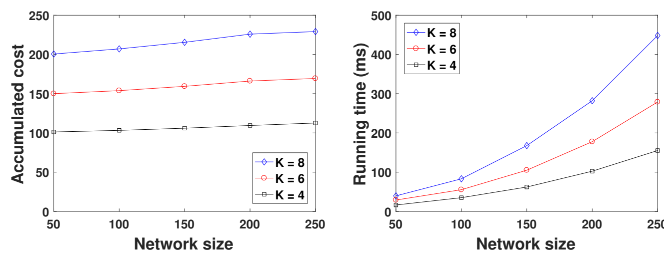
We then investigated the impact of the number of sources K on the proposed algorithms. Fig. 5(a) and Fig. 5(b) depict the total cost and running time of the solution delivered by algorithm `Alg.3` for different network sizes when $k = 4$, 6,



(a) The total cost (b) The running time

Fig. 4. The impact of length L of the SFC on the performance of the proposed algorithm.

and 8. It can be seen from Fig.5(a) that with the network size of 250, the total cost of the solution delivered by algorithm `Alg.3` when $K = 4$ is 49.2% of itself when $K = 8$. This can be justified that with a small value of K, the IoT applications consumes not only less computing resource, but also less data rates. Thus, the communication cost, the computing cost, and the workload-balancing cost decrease.



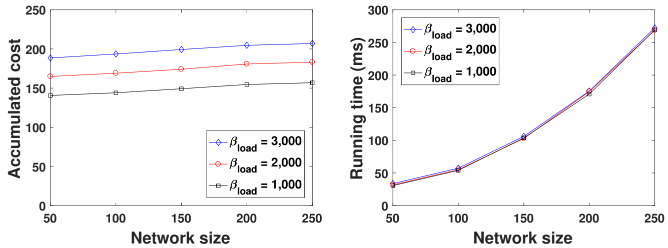
(a) The total cost (b) The running time

Fig. 5. The impact of the number of sources K on the performance of the proposed algorithm.

We finally investigated the impact of the coefficient of the workload balancing cost β_{load} on the proposed algorithms. Fig. 6(a) and Fig. 6(b) depict the total cost and running time of algorithm `Alg.3` with different network sizes when $\beta_{load} = 1,000$, 2,000, and 3,000, respectively. From Fig. 6(a), we can see that the total cost of algorithm `Alg.3` when $\beta_{load} = 1,000$ is 75.1% of itself when $\beta_{load} = 3,000$. This is because more cost is caused to make sure the workload-balance among the cloudlets in the MEC network, with the increase on the value of β_{load}. It can also be seen from Fig. 6(a) that the performance gap of algorithm `Alg.3` from $\beta_{load} = 2,000$ to $\beta_{load} = 3,000$ is 91.1% of itself from $\beta_{load} = 1,000$ to $\beta_{load} = 2,000$. The rationale behind this is that a lager β_{load} will lead to a more balanced workload among the cloudlets, thus, the total workload utilization ratio (W_v/C_v) of each cloudlet is reduced.

VIII. Conclusion

In this paper, we studied IoT-driven service provisioning in an MEC network for multi-source IoT applications with service function chain requirements. We first formulated the cost minimization problem of IoT-driven services, and showed that the problem is NP-hard. We then proposed a novel IoT-driven service provisioning framework for IoT applications, which includes uploading stream data from multiple IoT sources, in-network data stream aggregation and routing, and



(a) The total cost (b) The running time

Fig. 6. The impact of β_{load} on the performance of the proposed algorithm.

VNF instance placement and sharing in cloudlets in the network. We also devised an efficient algorithm for the defined problem. We finally evaluated the performance of the proposed algorithm through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising, compared with the lower bound on the optimal solution of the problem and another heuristic for the problem.

REFERENCES

[1] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie. Mobile edge computing: a survey. *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450 – 465, 2018.

[2] O. Alhussein, P. T. Do, J. Li, Q. Ye, W. Shi, W. Zhuang, X. Shen, X. Li, and J. Rao. Joint VNF placement and multicast traffic routing in 5G core networks. *Proc. of Globecom*, IEEE, 2018.

[3] A. Ceselli, M. Premoli, and S. Secci. Mobile edge cloud network design optimization. *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1818 – 1831, 2017.

[4] Cisco annual Internet report (2018-2023) white paper. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2020.

[5] R. Cohen, L. Eytan, J. Naor, and D. Raz. On the effect of forwarding table size on SDN network utilization. *Proc. of INFOCOM*, IEEE, 2014.

[6] R. Cohen, L. Eytan, J. Naor, and D. Raz. Near optimal placement of virtual network functions. *Proc. of INFOCOM*, IEEE, 2015.

[7] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molish. Approximation algorithms for the NFV service distribution problem. *Proc. of INFOCOM'17*, IEEE, 2017.

[8] A. Gushchin, A. Walid, and A. Tang. Scalable routing in SDN-enabled networks with consolidated middleboxes. *Proc. of HotMiddlebox*, ACM, 2015.

[9] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/, 2019.

[10] M. Jia, J. Cao, and W. Liang. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725 – 737, 2017.

[11] M. Jia, W. Liang, M. Huang, Z. Xu, and Y. Ma. Throughput maximization of NFV-enabled unicasting in Software-Defined Networks. *Proc. of Globecom'17*, IEEE, 2017.

[12] M. Jia, W. Liang, Z. Xu, and M. Huang. Cloudlet load balancing in wireless metropolitan area networks. *Proc. of INFOCOM'16*, IEEE, 2016.

[13] M. Jia, W. Liang, and Z. Xu. QoS-aware task offloading in distributed cloudlets with virtual network function services. *Proc. of MSWiM*, ACM, 2017.

[14] M. Jia, W. Liang, M. Huang, Z. Xu, and Y. Ma. Routing cost minimization and throughput maximization of NFV-enabled unicasting in Software-Defined Networks. *IEEE Transactions on Network and Service Management*, vol.15, no.2, pp. 732 – 745, 2018.

[15] Y. Li, W. Liang, W. Xu, and X. Jia. Data collection of IoT devices using an energy-constrained UAV. *Proc. of 34th IPDPS'20*, pp.644–653, IEEE, 2020.

[16] Y. Ma, W. Liang, and J. Wu. Online NFV-enabled multicasting in mobile edge cloud networks. *Proc. of 39th Intl Conf on Distributed Computing Systems* (ICDCS'19), IEEE, 2019.

[17] Y. Ma, W. Liang, J. Wu and Z. Xu. Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks. *IEEE Transactions on Parallel and Distributed Systems*, vol.31, no.2, pp. 393–407, 2020.

[18] Y. Ma, W. Liang, Z. Xu, and S. Guo. Profit maximization for admitting requests with network function services in distributed clouds. *IEEE Transactions on Parallel and Distributed Systems*, vol.30, no. 5, pp. 1143 – 1157, 2019.

[19] C. Mouradian, N. Jahromi, and R. Glitho. NFV and SDN-based distributed IoT gateway for large-scale disaster management. *IEEE Internet of Things Journal*, vol.5, no.5, pp.4119 – 4131, 2018.

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

[21] Y. Song, S. S Yau, R. Yu, X. Zhang, and G. Xue. An approach to QoS-based task distribution in edge computing networks for IoT applications. *Proc. of the International Conference on Edge Computing* (EDGE), IEEE, pp.32 –39, 2017.

[22] H. Soni, W. Dabbous, T. Turletti, and H. Asaeda. NFV-based scalable guaranteed-bandwidth multicast service for software-defined ISP networks. *IEEE Transactions on Network and Service Management*, vol.14, no.5, pp. 1157-1170, 2017.

[23] Q. Xia, W. Liang, and W. Xu. Throughput maximization for online request admissions in mobile cloudlets. *Proc. of 38th Annual IEEE Conference on Local Computer Networks* (LCN'13), IEEE, 2013.

[24] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo. Efficient algorithms for capacitated cloudlet placements. *IEEE Transactions on Parallel and Distributed Systems*, vol.27, no.10, pp. 2866 – 2880, 2016.

[25] H. Ren, Z. Xu, W. Liang, Q. Xia, P. Zhou, O. Rana, A. Galis, and G. Wu. Efficient algorithms for delay-aware NFV-enabled multicasting in mobile edge clouds with resource sharing. *IEEE Transactions on Parallel and Distributed Systems*, vol.31, no.9, pp. 2050 – 2066, 2020.

[26] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis. Efficient NFV-enabled multicasting in SDNs. *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2052 – 2070, 2019.

[27] Z.Xu, W. G, Q. Xia, W. Liang, O. F. Rana, and G. Wu. NFV-enabled IoT service provisioning in mobile edge clouds. *IEEE Transactions on Mobile Computing*, to be published, 2020, doi: 10.1109/TMC.2020.2972530

[28] Z. Xu, Z. Zhang, W. Liang, Q. Xia, O. F. Rana, and G. Wu. QoS-aware VNF placement and service chaining for IoT applications in multi-tier mobile edge networks. *ACM Transactions on Sensor Networks*, vol.16, no.3, Article 23:1–23:27, 2020.

[29] R. Yu, G. Xue, and X. Zhang. Application provisioning in fog computing-enabled internet-of-things: a network perspective. *Proc. of INFOCOM'18*, IEEE, pp.783 –791, 2018.

[30] S. Q. Zhang, Q. Zhang, H. Bannazadeh, and A. L. Garcia. Network function virtualization enabled multicast routing on SDN. *Proc. of ICC'15*, IEEE, 2015.