Mukherjee, M.; Kumar, V.; Lloret, J.; Zhang, Q. (2020). Revenue Maximization in Delay-Aware Computation Offloading Among Service Providers With Fog Federation. IEEE Communications Letters. 24(8):1799-1803. https://doi.org/10.1109/LCOMM.2020.2992781

Revenue Maximization in Delay-aware Computation Offloading among Service Providers with Fog Federation

Mithun Mukherjee, *Senior Member, IEEE*, Vikas Kumar, *Student Member, IEEE*,
Jaime Lloret, *Senior Member, IEEE*, and Qi Zhang, *Member, IEEE*

***Abstract*—In this letter, we study the computational offloading scheme for the delay-aware tasks of the end-users in the fog computing network. We consider a fog federation of different service providers where an individual fog node allocates its computing resources to the end-user in its proximity, while a fog manager coordinates the load balancing among the fog nodes over the entire network. At first, an individual fog node aims to maximize its revenue by selling the computational resources to the end-user in a distributed manner without any global knowledge of the network. To further maximize the overall revenue considering all fog nodes in the fog federation, the fog manager utilizes the remaining computing resources of the underloaded fog nodes. The extensive simulation results show the revenue improvement leveraging fog federation over entire network while maintaining the same and even better delay-performance for the end-users.**

***Index Terms*—Fog computing, fog federation, pricing, computational offloading, delay-sensitive tasks**

I. INTRODUCTION

In recent years, there is a rich literature on the computational offloading in fog and mobile edge computing [1]–[3] for the delay-sensitive tasks of the end-users. Several network models, such as standalone fog node, horizontal collaboration among fog nodes, and vertical collaboration with the remote cloud have been studied with their own advantage and limitations. An end-user avails the computational resources to process the delay-sensitive tasks from the fog node with a computational cost [3]–[7]. To this end, an optimal task data offloading strategy is proposed in [4], that maximizes the benefits of the mobile edge servers while meeting the delay constraints of the computation tasks. Moreover, a uniform and differentiated pricing scheme with Stackelberg game was suggested between end-users and a single mobile edge server [8]. This offloading strategy suggested an optimal decision when to offload the task from the multiple end-users to a single fog (or edge) server.

Motivation: To maximize the revenue from the end-users, the fog node of a specific service provider first tries to exploit its computational resources to the end-users under its own coverage. At the same time, when the end-user decides to offload to the fog node that has sufficient amount of computational resource (in term of total CPU cycles), the optimum value of the offloaded tasks mainly depends on the communication resource allocation (i.e., uplink and downlink transmission rate between end-user and the fog node) and allocated CPU speed. Moreover, due to the local CPU speed as well as the amount of input task data of the end-users and the uneven number of end-users in proximity, a significant amount of computational resources may remain unused in the fog node. This motivates us to study *how to maximize the unused computational resources of the fog node while satisfying the delay-sensitive task provisioning*.

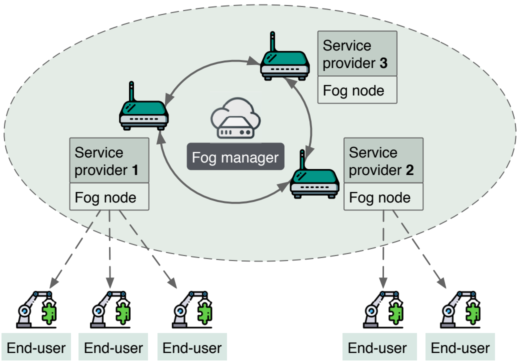


Fig. 1. Illustration of network model with service providers in fog federation.

Contribution: In this work, we introduce a fog federation, where the individual fog node of a service provider serves the end-users within its computational resources and a fog manager that co-ordinates fog resources among service providers. Our design objective is to maximize the revenue for the fog nodes while satisfying the delay-requirement from the end-users. In this letter, we propose a price-based task offloading strategy where the end-users decide to offload the task data based on the task offloading cost and the available computational resources in the fog node. After performing this decentralized strategy, the fog manager comes into the offloading scenario to solve the following two major key issues: how to leverage the remaining computational resources of the fog nodes a) *to maximize the revenue* for the fog nodes and b) *to give another offloading opportunity to the end-users* that were not able to offload due to the computational resource limitation of the primary fog node.

Corresponding author: Mithun Mukherjee (m.mukherjee@ieee.org)

M. Mukherjee is with the School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: m.mukherjee@ieee.org).

V. Kumar is with the Bharat Sanchar Nigam Limited, Patna 800001, India (e-mail: vikas.kumar@bsnl.co.in).

J. Lloret is with the Universitat Politecnica de Valencia, 46022 Valencia, Spain, (e-mail: jlloret@dcom.upv.es)

Q. Zhang is with the DIGIT, Department of Engineering, Aarhus University, 8000 Aarhus, Denmark, (e-mail: qz@eng.au.dk)

II. NETWORK MODEL

We consider a fog federation network of service providers, as illustrated in Fig.1, comprised of N number of fog computing nodes and one centralized fog manager (FM). We envision that FM may be selected among the service providers by their internal service level agreement (SLA) or government agency may be responsible for set up, installation of FM, and unbiased computing and offloading operation. In our *fog federation*, we consider a network model where the different service providers independently allocate and schedule their own computational resources to their respective end-users, however, a centralized FM may exist in the network. For simplicity, we assume a single fog node per service provider, however, fog collaboration among the fog nodes of a service provider enables additional computing resources with increasing resource scheduling complexity. Denote K_i as the total number of end-users being served by the fog node of a service provider i. We assume that any end-user is *directly* connected to only one fog node[1] (termed as, *primary* fog node), however if required, this primary fog node further offloads the end-user's task data to another neighbouring fog node that may belong to another service provider.

A. Delay Model

Local processing at the end-user side. Denote $D_{k,i}$ as the input data for the kth end-user of the fog node, i. Due to the limited computation resource, the kth end-user offloads $l_{k,i}$ amount of data to its primary fog node, i. Now, the time required to locally process the remaining $(D_{k,i} - l_{k,i})$ amount of task data at the kth end-user under the fog node, i, is expressed as $t^{\mathsf{user}}_{\mathsf{cpu},k,i} = C_{k,i}(D_{k,i} - l_{k,i})/f^{\mathsf{user}}_{k,i}$, where $C_{k,i}$ and $f^{\mathsf{user}}_{k,i}$ are the number of CPU cycles required per bit to compute the input task data and the CPU clock speed, respectively, for the kth end-user of the fog node, i.

Offloading to the fog node. When the kth end-user offloads its task data to the primary fog node, i, the offloading time, denoted as $t^{\mathsf{OL}}_{k,i}$, depends on the following: a) uploading the task data from the end-user to the fog node, b) processing the kth end-user's data at the fog node level, and c) downloading the results from the fog node side to the end-user. Denote $r^{\mathsf{UL}}_{k,i}$ and $r^{\mathsf{DL}}_{k,i}$ as the uplink and downlink transmission rate, respectively, between the kth end-user and the fog node, i. Thus, the uploading and downloading time are expressed as $t^{\mathsf{UL}}_{k,i} = l_{k,i}/r^{\mathsf{UL}}_{k,i}$ and $t^{\mathsf{DL}}_{k,i} = \alpha_{k,i}\, l_{k,i}/r^{\mathsf{DL}}_{k,i}$, where $\alpha_{k,i} > 0$ is an application-specific ratio of output and input data size for the kth end-user offloaded at the fog node, i. Moreover, the time required to process kth end-user's offloaded data, i.e., $l_{k,i}$ at the fog node, i will be $t^{\mathsf{fog}}_{\mathsf{cpu},k,i} = C_{k,i}\, l_{k,i}/f^{\mathsf{fog}}_{k,i}$, where $f^{\mathsf{fog}}_{k,i}$ is the CPU clock speed assigned to the kth end-user by the fog node, i. We take the assumption that the fog node equally divides its CPU clock speed, i.e., f^{fog}_{i}, to the end-users, thus $f^{\mathsf{fog}}_{k,i} = f^{\mathsf{fog}}_{i}/K_i$. Thus, the offloading time from the kth end-user to the fog node, i, i.e., $t^{\mathsf{OL}}_{k,i}$ is expressed as

$$t^{\mathsf{OL}}_{k,i} = t^{\mathsf{UL}}_{k,i} + t^{\mathsf{fog}}_{\mathsf{cpu},k,i} + t^{\mathsf{DL}}_{k,i}\ . \tag{1}$$

It is important to note that the end-user's offloaded data to the fog node, i are limited by the following constraint

$$\mathbf{C1}: \quad \sum_{k=1}^{K_i} l_{k,i}\, C_{k,i} \le \mathbb{V}_i\ , \tag{2}$$

where $\mathbb{V}_i$ is the maximum amount of computing resources (i.e., total CPU cycles per slot) available at the fog node, i.

Assuming parallel task processing at the end-user's and the fog node's side, the time required to process the kth end-user's task data will be $t_{k,i} = \max\{t^{\mathsf{user}}_{\mathsf{cpu},k,i}, t^{\mathsf{OL}}_{k,i}\}$.

B. Cost and Revenue

Computational cost at the end-user. Due to the computational resource limitation, the end-users often aim to offload their delay-sensitive task to the primary fog node. The primary fog node sets a price for the computational resources allocated to each end-user. Denote $\mu_{k,i}$ as the unit price per CPU speed for the kth end-user's task data to process at the fog node, i. Therefore, the computational cost to process the $l_{k,i}$ amount of task data at the fog node, i will be $\mu_{k,i}\, l_{k,i}\, C_{k,i}$. Since we focus on the delay-aware task offloading, the above computational cost is reflected as a delay to the end-user's side. As a result, the total cost per slot including the processing delay (i.e, $t_{k,i}$) and the computational cost at the fog node, i for the kth end-user is defined as

$$U^{\mathsf{user}}_{k,i}(l_{k,i}, \mu_{k,i}) \triangleq t_{k,i} + \mu_{k,i} l_{k,i} C_{k,i}\ . \tag{3}$$

Revenue at individual fog node. Based on the price set for the kth end-user, i.e., $\mu_{k,i}$, the revenue in each time slot limited by the constraint **C1** at the fog node, i is expressed as

$$U^{\mathsf{fog}}_{i}(\boldsymbol{\mu}_i) = \sum_{k=1}^{K_i} \mu_{k,i} l_{k,i} C_{k,i}\ , \tag{4}$$

where $\boldsymbol{\mu}_i = [\mu_{1,i}, \mu_{2,i}, \ldots, \mu_{K_i,i}]$.

III. GAME FORMULATION: OBJECTIVE FUNCTIONS

A fog computing-enabled service provider aims to maximize its revenue by selling its computational resources to the end-users under the constraint **C1**. However, at the same time, due to either the high price set by its primary fog node or the computational resource limitation in **C1**, an end-user is not always able to offload the task data to the primary fog node. Interestingly, assuming different amount of maximum computing resources per fog node by the service provider and uneven end-user distribution per fog node, there is a high possibility that a significant amount of computing resources remains unused in the fog node if we consider only standalone fog to serve the end-users. In our network model, the FM manages the remaining computational resource allocation for the end-users

---

[1] If the end-user is not mobile, it has practical advantage that each end-user connects to only one fog node that has best link quality and it also facilitates the access control.

that were not able to offload their task data due to the above mentioned reasons. Ignoring the transmission delay between the fog nodes among different service providers, the primary fog node, i offloads the task data to the neighboring fog nodes (denote as secondary fog nodes). These neighbor fog nodes get the revenue from the end-users that are directly connected to the fog node, i. This in turn further improves in revenue of underloaded fog nodes. At the same time, such arrangement is advantageous for the end-users connected to the overloaded fog node while minimizing delay and computational cost.

Follower problem. Basically, the kth end-user of the fog node, i aims to minimize its cost (see, (3)), i.e.

$$\mathbf{P1}: \quad \min_{l_{k,i}} \quad U^{\mathsf{user}}_{k,i}(l_{k,i},\mu_{k,i}) \quad \text{s.t. } 0 \le l_{k,i} \le D_k\,. \tag{5}$$

Leader problem. In the similar way, we express the total revenue by maximizing the individual fog node's revenue as

$$\mathbf{P2}: \quad \sum_{i=1}^{N} \max_{\boldsymbol{\mu}_i \succeq 0} U^{\mathsf{fog}}_i(\boldsymbol{\mu}_i) \quad \text{s.t. } (2)\,. \tag{6}$$

Optimum offloaded task data to the primary fog node. When the end-user decides to offload the task data, the optimal value of the offloaded data is obtained such that the offloading time and local processing time at the end-user are equal, i.e., $t^{\mathsf{user}}_{\mathsf{cpu},k,i} = t^{\mathsf{OL}}_{k,i}$. First, we rewrite (1) as $t^{\mathsf{OL}}_{k,i} = \beta_{k,i}\, l_{k,i}$, where $\beta_{k,i} = \big(1/r^{\mathsf{UL}}_{k,i}\big) + \big(C_{k,i}/f^{\mathsf{fog}}_{k,i}\big) + \big(\alpha_{k,i}/r^{\mathsf{DL}}_{k,i}\big)$. Let $m_{k,i}$, $0 \le m_{k,i} \le D_{k,i}$ be the value of offloaded task data for the kth end-user to the fog node, i. Then, by replacing $l_{k,i}$ by $m_{k,i}$ in $t^{\mathsf{user}}_{\mathsf{cpu},k,i} = t^{\mathsf{OL}}_{k,i}$, we obtain $C_{k,i}(D_k - m_{k,i})/f_k = \beta_{k,i}\, m_{k,i}$, afterward $m_{k,i} = C_{k,i}\, D_k/(\beta_{k,i} f_k + C_{k,i})$.

Upper-bound of $\mu_{k,i}$. Using $t^{\mathsf{user}}_{\mathsf{cpu},k,i} = t^{\mathsf{OL}}_{k,i}$, we rewrite (3) as

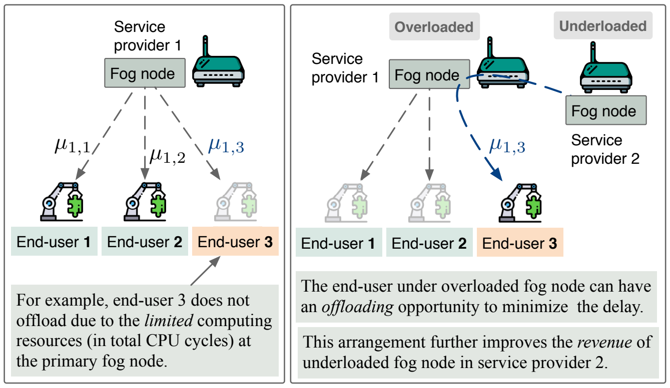
$$\begin{aligned} t^{\mathsf{user}}_{\mathsf{cpu},k,i} + \mu_{k,i} m_{k,i} C_{k,i} &= \frac{(D_{k,i}-m_{k,i})C_{k,i}}{f^{\mathsf{user}}_{k,i}} + \mu_{k,i} m_{k,i} C_{k,i} \\ &= \left(\mu_{k,i} - \frac{1}{f^{\mathsf{user}}_{k,i}}\right) m_{k,i} C_{k,i} + \frac{D_{k,i} C_{k,i}}{f^{\mathsf{user}}_{k,i}}\,. \end{aligned} \tag{7}$$

Then, from the end-user's perspective, $\mu_{k,i} \le 1/f^{\mathsf{user}}_{k,i}$.

IV. PROPOSED COMPUTATIONAL OFFLOADING STRATEGY

A. Stage 1: Pricing Scheme by the Primary Fog

At first stage, the fog node in the network acts as a *standalone* fog node. Note that a uniform pricing [8], [9], where a fog node sets same price for all end-users, does not provide the optimal solution for both revenue maximization and average delay minimization due to the different value of $f^{\mathsf{user}}_{k,i}$. In the following, we apply differential pricing scheme in [8] to maximize the revenue for individual fog node and minimize the average delay for the end-users. Let $\mathbb{1}_{k,i} = \{1, 0\}$ be an offloading indicator, where $\mathbb{1}_{k,i} = 1$ indicates that the kth end-user offloads $m_{k,i}$ amount of task data to the fog node, i and if $\mu_{k,i} > 1/f^{\mathsf{user}}_{k,i}$ (see, (7)), otherwise the kth end-user prefers to locally process the entire task data, i.e $\mathbb{1}_{k,i} = 0$. Therefore, the revenue maximization Problem **P2** for the fog node can be rewritten as



For example, end-user 3 does not offload due to the *limited* computing resources (in total CPU cycles) at the primary fog node.

The end-user under overloaded fog node can have an *offloading* opportunity to minimize the delay.

This arrangement further improves the *revenue* of underloaded fog node in service provider 2.

Fig. 2. Illustration of the proposed pricing and offloading strategy. The end-users that were not able to offload due to computational capacity constraint or optimal offloading decision, can *further* use the resources of other fog nodes that have unused computing resource after serving their own end-users.

$$\mathbf{P3}: \quad \sum_{i=1}^{N} \left(\max_{\mathbb{1}_{k,i} \in \{0,1\}} \sum_{k=1}^{K_i} \frac{\mathbb{1}_{k,i}\, m_{k,i} C_{k,i}}{f^{\mathsf{user}}_{k,i}} \right)$$
$$\text{s.t. } \sum_{k=1}^{K_i} \mathbb{1}_{k,i}\, m_{k,i} C_{k,i} \le \mathbb{V}_i \quad \forall i \in \{1, \ldots, N\}. \tag{8}$$

Basically, Problem **P3** is a binary knapsack problem and is NP-complete, so we apply dynamic programming [10] to solve the problem with the weight $m_{k,i} C_{k,i}$ and the value $m_{k,i} C_{k,i}/f^{\mathsf{user}}_{k,i}$ in pseudo-polynomial time.[2]

B. Stage 2: Pricing Scheme by the Fog Manager

In multi-fog environment, after solving problem **P3** for the individual fog node, there may exist end-users that can not offload due to constraint **C1**. As illustrated in Fig. 2, in the next stage, we leverage the remaining computational resources to serve these end-users thereby increasing the revenue. Accordingly, the FM gathers the unused computing resources as

$$f^{\mathsf{FM}} = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \left(1 - \mathbb{1}_{k,i}\right) \frac{f_i}{K_i} \tag{9}$$

and

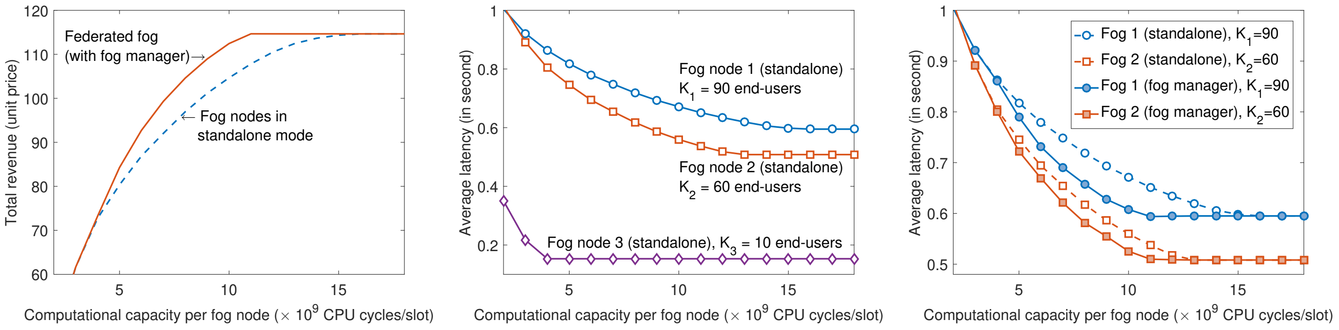
$$\mathbb{V}^{\mathsf{FM}} = \sum_{i=1}^{N} \left(\mathbb{V}_i - \sum_{k=1}^{K_i} \mathbb{1}_{k,i}\, m_{k,i}\, C_{k,i} \right). \tag{10}$$

Denote $\mathcal{K}^{\mathsf{FM}} = \{1, 2, \ldots, K^{\mathsf{FM}}\}$ as the number of remaining end-users that have earlier decided not to offload to the fog node, i will participate in this pricing scheme co-ordinated by the FM, where

$$K^{\mathsf{FM}} = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \left(1 - \mathbb{1}_{k,i}\right). \tag{11}$$

---

[2] We have used kp01(.) software package in MATLAB.



(a) (b) (c)

Fig. 3. $K_1 = 90$ end-users, $K_2 = 60$ end-users, and $K_3 = 10$ end-users. Performance of (a) total revenue and (b)-(c) average latency for the end-users.

Now, the CPU speed allocated to the kth end-user, where $k \in \mathcal{K}^{\mathsf{FM}}$ can be written as $f_k^{\mathsf{FM}} = f^{\mathsf{FM}}/K^{\mathsf{FM}}$. In this way, we find the optimum amount of task data for the kth end-user under the overloaded fog node, i as $m_k^{\mathsf{FM}} = C_{k,i}\, D_k/\big(\beta_k^{\mathsf{FM}}\, f_k^{\mathsf{FM}} + C_{k,i}\big)\,\forall k \in \mathcal{K}^{\mathsf{FM}}$, where $\beta_k^{\mathsf{FM}} = \big(1/r_{k,i}^{\mathsf{UL}}\big) + \big(C_{k,i}/f_k^{\mathsf{FM}}\big) + \big(\alpha_{k,i}/r_{k,i}^{\mathsf{DL}}\big)$. Now, denote $\mathbb{1}_k^{\mathsf{FM}} = 1$ when the kth end-user offloads the task data to avail the computing resources co-ordinated by the FM, and 0, otherwise. Thus, the revenue maximization problem becomes

$$\mathbf{P4}: \quad \max_{\mathbb{1}_k^{\mathsf{FM}} \in \{0,1\}} \sum_{k=1, k\in\mathcal{K}^{\mathsf{FM}}}^{K^{\mathsf{FM}}} \frac{\mathbb{1}_k^{\mathsf{FM}}\, m_k^{\mathsf{FM}}\, C_k}{f_{k,i}^{\mathsf{user}}}$$

$$\text{s.t.} \sum_{k=1, k\in\mathcal{K}^{\mathsf{FM}}}^{K^{\mathsf{FM}}} \mathbb{1}_k^{\mathsf{FM}} m_k^{\mathsf{FM}}\, C_k \le \mathbb{V}^{\mathsf{FM}}\,. \qquad (12)$$

We apply the dynamic programming to solve this NP-complete binary knapsack problem **P4**, similar to **P3**, in pseudo-polynomial time, where corresponding weight is selected as $m_k^{\mathsf{FM}}\, C_{k,i}$ and the value or profit is taken as $m_k^{\mathsf{FM}}\, C_{k,i}/f_{k,i}^{\mathsf{user}}$.

V. SIMULATION RESULTS

In the simulation setup, we consider three service providers with one fog node per service provider, i.e., $N = 3$ fog nodes. Besides, K_i is uniformly distributed over $[10, 100]$. Moreover, the number of CPU cycles per bit and the input data size for kth end-user are uniformly distributed over $[500, 1500]$ cycles/bit and $[100, 500]$ KB, respectively. In addition, $\mathbb{V}_i = 6 \times 10^9$ cycles/slot, $\alpha_{k,i} = 0.2$, $f_i = 100$ GHz, and the local CPU speed of the end-user is uniformly selected from the set $[0.1, 0.2, \ldots, 1]$ GHz. We further assume that $r_{k,i}^{\mathsf{UL}}$ and $r_{k,i}^{\mathsf{DL}}$ are uniformly distributed over $[15, 25]$ Mbps, $[20, 30]$ Mbps, respectively. In simulations[3], we use MATLAB and average the results over $10,000$ different runs.

As shown in Fig. 3(a), at lower computational capacity ($\mathbb{V}_i = 4 \times 10^9$ CPU cycles/slot), the revenue performances for both standalone and fog federation with fog manager are same. The main reason is that due to low CPU resources, the end-users are not able to offload the task data to the fog node. Thus, the fog nodes already reached their computing capacity limit, i.e., $\mathbb{V}_i$ in a standalone mode. Moreover, at the higher CPU resources ($\mathbb{V}_i = 16 \times 10^9$ CPU cycles/slot), most of the end-users offload their optimum value of task data to the primary fog nodes that act in standalone mode. As a result, the fog nodes already reached their maximum revenue without any intervention of fog manager in fog federation. However, at the mid range of the CPU resources (i.e., 4 to 16×10^9 cycles/slot), the proposed two-stage pricing and offloading scheme outperforms the standalone mode of the fog node. This is because the fog manager further maximises the revenue by selling the remaining CPU resources to the end-users that were not able to offload at the standalone mode of the fog nodes.

Fig. 3(b) and 3(c) illustrate the average latency performance for the end-users in three fog nodes. It is clearly observed that the average latency performance for the end-users served by the fog node 3 has better performance than the end-users under two other fog nodes. It is mainly due to the low number of end-users (i.e., $K_3 = 10$ end-users) that easily avail sufficient computational resources. As the number of end-users is high in fog node 1 (also in fog node 2), the competition among the end-users to avail the limited computational resources becomes higher. As a result, the latency performance degrades. However, the proposed offloading strategy with the fog manager provides significant improvement for the end-users under overloaded fog nodes (fog node 1 and 2) by leveraging the total CPU cycles of the underloaded fog node 3 (see Fig 3(c)). From simulations we observe that the end-users under fog node 3 does not suffer any latency performance degradation.

VI. CONCLUSION

In this paper, we studied the computation offloading for the delay-sensitive tasks in fog computing networks with an aim to maximize the service provider-wise revenue. We proposed a two-stage offloading policy, where at the first stage, the individual fog node allocates its finite computational resources to the end-users in its proximity to achieve the revenue. Later, with the help of fog manager, the unused resources of the fog nodes have been exploited to obtain additional revenue. By extensive simulation, we have shown that while maximizing the revenue in fog node side, the end-users gain the advantage of utilizing the remaining computational resources, instead of

---

[3] The source code is available at https://github.com/MithunHub/SourceCodeIEEECLFogFederation

using the standalone fog node. As a part of future work, we plan to consider the inter-fog transmission delay and a pricing agreement scheme among the service providers while giving computational resources to the deadline-aware tasks.

REFERENCES

[1] S. Luo, X. Chen, Z. Zhou, X. Chen, and W. Wu, "Incentive-aware micro computing cluster formation for cooperative fog computing," *IEEE Trans. Wireless Commun.*, pp. 1–1, Jan. 2020.

[2] M. Mukherjee, M. Guo, J. Lloret, R. Iqbal, and Q. Zhang, "Deadline-aware fair scheduling for offloaded tasks in fog computing with inter-fog dependency," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 307–311, Feb. 2020.

[3] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach," *IEEE Trans. Cloud Comput.*, pp. 1–15, June 2018.

[4] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *Proc. IEEE ICC*, May 2017, pp. 1–6.

[5] M. Mukherjee, Y. Liu, J. Lloret, L. Guo, R. Matam, and M. Aazam, "Transmission and latency-aware load balancing for fog radio access networks," in *Proc. IEEE GLOBECOM*, Dec. 2018, pp. 1–6.

[6] M. Nasir, K. Muhammad, J. Lloret, A. K. Sangaiah, and M. Sajjad, "Fog computing enabled cost-effective distributed summarization of surveillance videos for smart cities," *J. Parallel and Distrib. Comput.*, vol. 126, pp. 161–170, Apr. 2019.

[7] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet of Things J.*, vol. 4, no. 6, pp. 2082–2091, Dec. 2017.

[8] M. Liu and Y. Liu, "Price-based distributed offloading for mobile-edge computing with computation capacity constraints," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 420–423, June 2018.

[9] Y. Liu, R. Wang, and Z. Han, "Interference-constrained pricing for D2D networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 475–486, 2017.

[10] S. Martello, D. Pisinger, and P. Toth, "Dynamic programming and strong bounds for the 0-1 knapsack problem," *Management Science*, vol. 45, no. 3, pp. 414–424, 1999.