Task Allocation in Extreme Edge Computing for Complex IoT Services

Rawan F. El-Khatib[1], Sara A. Elsayed[2], Nizar Zorba[3], Hossam S. Hassanein[3]
[1]Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada
[2]School of Computing, Queen's University, Kingston, ON, Canada
[3]Electrical Engineering Department, Qatar University, Doha, Qatar
Email: rawan.elkhatib@queensu.ca, selsayed@cs.queensu.ca, nizarz@qu.edu.qa, hossam@cs.queensu.ca

***Abstract*—The rise of 6G technology will enable various innovative applications to deliver transformative experiences and services with unprecedented speed, reliability, and interactivity. On one hand, the realization of such innovative applications relies on the processing of large volumes of data generated at the Extreme Edge of the network, requiring time-critical and resource-intensive processing. On the other hand, these applications require handling multi-modal data or inputs from several sensing data sources, and as a result, the resulting computing tasks encompass multiple subtasks that are crucial to service delivery. Conventional offloading schemes overlook the complexity of these applications, jeopardizing the task success rate and application QoS. In this work, we highlight the dire need for a computational offloading scheme that addresses the intricate nature of such applications and their computing tasks, and present a preliminary problem formulation to tackle these needs.**

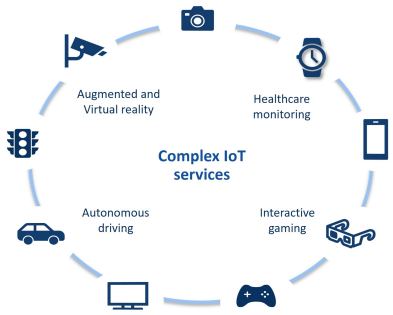

Fig. 1. Examples of IoT devices that produce multi-modal data to enable complex IoT services.

I. Introduction

Recent technological advancements for mobile communications that go Beyond the Fifth Generation (B5G) and the Sixth Generation (6G) signal the beginning of a new era of innovation. Built on the foundations laid by its predecessors, 6G promises faster data speeds, lower latencies, and greater reliability to enable transformative applications and services across various domains. From ultra-high definition video streaming and interactive gaming to real-time remote surgery and autonomous transportation systems, personalized services will enable unprecedented levels of immersion, engagement, and productivity [1].

The successful delivery of these innovative applications requires handling extensive volumes of data generated at the extreme edge of the network, some of which are time-sensitive and require resource-intensive processing. In many cases, an application will combine multi-sensory modalities or inputs from several distributed data producers that contribute crucial insights for service delivery, as shown in Fig. 1. For instance, an Unmanned Aerial Vehicle (UAV) supplies visual, range, speed, and acceleration data to enable a multi-modal localization service. In other cases, applications may need preprocessing of the distributed data streams that are inherently susceptible to failures due to the presence of measurement noises, potential sensor malfunctioning, etc. For example, in an industrial application intended to automate equipment operations, many industrial machines produce massive amounts of data in separate data flows that must be analyzed to extract potentially valuable information.

The common characteristic among such applications is the intricate nature of the processing required to extract meaningful insights. Typically, the data processing task encompasses a multitude of subtasks, where each subtask targets at a singular data stream. This often requires the execution of multiple subtasks in parallel to meet stringent performance and time requirements [2]. Consequently, the success of these applications is contingent not only on efficiently processing the data at the edge to meet their real-time demands, but also on effectively coordinating and managing the execution of numerous subtasks to ensure seamless service provision.

Driven by the growth of these data-intensive applications and the proliferation of IoT devices, Extreme Edge Computing (EEC) has emerged as an efficient computing paradigm offering the opportunity to utilize the idle computational resources of Extreme Edge Devices (EEDs). The average EED is equipped with modern powerful computing units and an array of wireless communication capabilities (e.g., Bluetooth, WiFi, etc.), enabling real-time data exchange and processing. EEC offers several advantages, including proximity to the data source, reduced latency, fault tolerance, scalability, and enhanced cost-effectiveness [3], [4].

We argue that for the subtask-wise partitioned application pattern considered in this work, EEC is a well-fitted computing paradigm. Unlike the conventional method of offloading a large

computing task to a single Edge Computing (EC) server, EEC allows the separate offloading of each computational subtask to an EED. This strategy offers several benefits: a) eases the strain of rising demands on EC servers, b) increases the utilization of EED resources, and most importantly, c) mitigates delays caused by transferring distributed input data required for task completion to a single EC server.

A myriad of literature explored computational offloading schemes for EEC. The primary optimization objectives are a) minimizing energy consumption (e.g., [5]), b) minimizing latency (e.g., [6]), or trade-offs between these objectives. However, these works cannot meet the requirements of the proliferating applications exemplified above, for which, service requesters' satisfaction hinges not only on fulfilling individual subtask requirements, but also on full task execution to produce meaningful insights. Hence, with the explosive growth of these complex IoT applications aggravating the pressure on EC servers, it becomes imperative to devise solutions tailored to the intricate demands of these computing tasks. In the sequel, we present a preliminary problem formulation that aims to address these gaps in the literature.

II. System Model and Proposed Solution

Consider a system with a set of service requesters and a set of subtasks denoted by $\mathcal{K}$ and $\mathcal{I}$, respectively. Subtasks are members of nonoverlapping sets $\mathcal{B}_k$, where each set represents a "parent" task produced by service requester k. Let u_k denote an arbitrary amount of reward that is earned when service requester k is satisfied with its task completion. Each subtask i is restricted by a deadline $t_i^{deadline}$. Moreover, there exists a set of workers $\mathcal{J}$, where each worker has a maximum available computing capacity denoted by f_j^{max}. Let $D_{i,j}$ denote the total communication and execution delays for subtask i at worker j, and $c_{i,j}$ the amount of computational capacity dedicated from worker j to subtask i.

Our objective is to maximize the number of completed tasks, hence satisfying the highest number of service requesters. Consequently, we formulate the problem as a Binary Linear Program (BLP), with two decision variables: the binary variable $x_{i,j}$ denotes whether task i is assigned to worker j, and the binary decision variable y_k denotes whether task k is fully assigned. The mathematical problem formulation is written as follows:

$$\underset{x_{i,j},\, y_k}{\text{maximize}} \quad \sum_{k \in \mathcal{K}} u_k y_k$$

subject to

$$\text{C1:} \sum_{i \in \mathcal{I}} c_{i,j} x_{i,j} \leq f_j^{max} \quad , \quad \forall j \in \mathcal{J},$$

$$\text{C2:} \sum_{j \in \mathcal{J}} D_{i,j} x_{i,j} < t_i^{deadline}, \quad \forall i \in \mathcal{I},$$

$$\text{C3:} \sum_{j \in \mathcal{J}} x_{i,j} = y_k \quad , \quad \forall i \in \mathcal{I}, \forall k | i \in \mathcal{B}_k$$

In the above formulation, constraint C1 ensures that the cumulative workload of subtask assignments at worker j does not exceed its available computational capacity. Constraint C2 ensures that any subtask i is completed within its designated deadline. Finally, constraint C3 ensures that if task k is selected for execution, all subtasks belonging to this task (that is, batch $\mathcal{B}_k$) are assigned. In this way, we ensure that the requester is satisfied because all the subtasks needed for the delivery of the service are assigned. This all-or-nothing approach also extends to the achievement of the reward u_k, as it is associated with the full execution of the task.

We anticipate that the proposed solution will outperform conventional offloading schemes in several dimensions. Firstly, conventional schemes fail to guarantee that all subtasks belonging to a single service requester will be executed, thus failing to ensure high task success. Hence, our proposed solution is expected to significantly increase the task success rate, resulting in higher service requester satisfaction. Secondly, since our proposed scheme assigns resources exclusively to batches of subtasks, the amount of resource waste due to subtask assignments that do not contribute to full service delivery will be eliminated, improving the resource utilization rates. Finally, an often overlooked aspect in the literature is the effects of service requesters' contention for limited resources. The full task completion utility can be designed in a way that differentiates among different tasks in terms of computational demands, hence increasing fairness levels among service requesters. In future work, we will perform an extensive performance evaluation of the proposed scheme to validate the anticipated performance enhancements.

Acknowledgement

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant ALLRP 549919-20; in part by the Distributive, Ltd. This work was also supported in part by Qatar University under Grant IRCC-2024-494.

References

[1] M. Banafaa et. al., "6G Mobile Communication Technology: Requirements, Targets, Applications, Challenges, Advantages, and Opportunities", in *Alexandria Engineering Journal,* vol. 64, pp. 245-274, 2023, doi:10.1016/j.aej.2022.08.017.

[2] R. F. ElKhatib, S. A. ElSayed, N. Zorba, H. S. Hassanein, "Optimal Proactive Resource Allocation at the Extreme Edge," in *Proc. IEEE International Communications Conference (ICC),* 2022, pp. 1-6.

[3] R. Olaniyan, O. Fadahunsi, M. Maheswaran, and M. F. Zhani, "Opportunistic Edge Computing: Concepts, Opportunities and Research Challenges," in *Future Generation Computing Systems,* vol. 89, pp. 633–645, 2018, doi:10.1016/j.future.2018.07.040.

[4] Y. Sahni, J. Cao, S. Zhang and L. Yang, "Edge Mesh: A New Paradigm to Enable Distributed Intelligence in Internet of Things," in IEEE Access, vol. 5, pp. 16441-16458, 2017, doi:10.1109/ACCESS.2017.2739804.

[5] U. Saleem, Y. Liu, S. Jangsher, X. Tao, and Y. Li, "Latency Minimization for D2D-enabled partial Computation Offloading in Mobile Edge Computing," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4472–4486, Apr. 2020, doi:10.1109/TVT.2020.2978027.

[6] Q. Lin, F. Wang, and J. Xu,"Optimal Task Offloading Scheduling for Energy Efficient D2D Cooperative Computing," in *IEEE Communications Letters,* vol. 23, no. 10, pp. 1816–1820, Oct. 2019, doi:10.1109/LCOMM.2019.2931719.