Chapter 2

Literature Review

Edge computing is a natural extension of cloud computing in terms of infrastructure deployment and network topology, with a more geographically distributed architecture than cloud computing. This architecture brings the benefits and capabilities of cloud computing to IoT devices by deploying edge servers in various locations closer to them. In the edge computing environment, IoT app vendors install required applications and services on edge servers to provide the computation capacities to their IoT app users [36]. IoT app users offload their computation tasks to these edge servers [37]. The edge server processes the tasks that have been offloaded and sends the results to app users. In this process, communication may take place from IoT device to IoT device, IoT device to server, server to server, and server to IoT device.

Over the last decade, extensive research has been conducted on computational task offloading from IoT devices to edge servers [15–18]. These studies show that an appropriate allocation of edge resources to app users is required for efficient use, as resource allocation affects system performance across all paradigms. Thus, it entices researchers to work on assigning edge computing and networking resources to users for task offloading, processing, and responding. We group the edge resource allocations into two categories: edge computing resource allocation and edge networking resource allocation.

2.1 Literature Review on Computing Resource Allocation

In the past years, researchers have studied the edge computing resource allocation to the IoT devices from different perspectives, and they present surveys about edge computing enabled IoT challenges and issues [38–40] that must be addressed, such as cost optimization [2, 41, 42], Quality of Service (QoS) [43–45], optimum resource allocation [19], energy consumption [46], architectural issue [39], security and privacy [47, 48], etc. They emphasizes the importance of efficiently allocating edge computing resources to IoT users.

[26] demonstrates that allocating app users to edge servers with proximity and capacity constraints is equivalent to the variable-sized vector bin packing problem. The studies in [41, 42] verify that the edge computing resource problem is an NP-Complete problem. Finding a solution to the edge computing resource problem becomes more difficult because each app user requires a different amount of various types of resources on edge servers [49–51]. Therefore, researchers explored various algorithms based on heuristic and meta-heuristic approaches for solving the resource allocation problem [19, 42]. However, a centralized solution may not meet the needs of each independent app user motivated by self-interest (who has own QoS and resource requirements). Consequently, the system performance may not be effective [52]. Moreover, heuristic approaches do not effectively solve this resource allocation problems if the number of app users is large [53]. Furthermore, meta-heuristic approach-based algorithms take a higher time to converge at the solution [53]. The studies [2, 41] proposed the game-theoretic approach for finding the solution to the resource allocation problem. Researchers have also recently investigated various resource allocation strategies using different methodologies, such as game theory [54, 55], matching based [56, 57], distributed [58, 59], reinforcement learning [60, 61], machine learning [62], etc., for optimizing the different objectives. The game-theoretic and learning-based approaches can give the freedom to rational app users to optimize their objective functions and fulfill

their requirements. However, these approaches require a significant amount of time. In this thesis, we present the state-of-the-art works from three aspects: 1) QoS-driven resource allocation, 2) cost-effective resource allocation, and 3) optimization based resource allocation.

2.1.1 QoS-Driven Resource Allocation

Given the exponential growth of IoT devices, the literature introduced various mechanisms to enhance the QoS to the users while increasing the network performance via network resource allocation. These mechanisms are artificial intelligence [63], blockchain [64], game theory [65], and optimization theory [66], etc. The study [67, 68] allocates computing and transmission resources to users in a QoS-aware manner that optimizes offloading decisions and minimizes network overhead. The papers formulated the transmission resources allocation problem as an Integer Linear Program (ILP) to achieve the QoS requirements while reducing the IoT devices' cost for using transmission resources [69, 70]. These studies use a heuristic approach to give an approximate solution. Zhang et al. [71] considered the users' uncertain resource demands and presented an approach to transform the problem into deterministic convex programming. They used a method with adjustable constraint control coefficients to solve this problem. Alrawahi et al. [72] investigated the problem of managing QoS that optimizes energy consumption, response time, and cost. Tan et al. [73] investigated the user resource allocation problem, aiming to maximize QoS-preferred users while shielding WiFi users with the LAA-LTE mode. The logistic function is used by Lai et al. [19] to formulate the relationship between QoE and QoS measures, and a greedy-like methodology is adapted to approximate overall QoE maximization. Ma et al. [68] introduced a task scheduling and cooperative service caching technique based on Gibbs sampling and water filling for minimizing the service response time. [74] investigate edge computing support for network slicing through offloading and QoS-based resource allocation.

2.1.2 Cost-Effective Resource Allocation

In this area, many research studies have been proposed that allocates the edge computing resources to users in order to minimizes the system cost [41, 70, 75–77]. Researchers investigated various game-theoretic and heuristic approaches to address the cost-effective resource allocation problem. The game-theoretic approach determines the Pure Nash Equilibrium (PNE) solution, which allocates optimal resources to services based on their budget and maximizes resource utilization [2]. The study [41] looked into the problem of allocating edge computing resources to minimize app vendor costs and proposed a game-theoretic-based decentralized algorithm to solve it. Both proposed game-theoretic approaches are highly complex. The studies [42,78] used heuristic-based methods to allocate computing resources to app users for fast convergence. However, the effect of cost optimization on app user QoS was not investigated. The research study [79] objective is to help mobile app vendors maximize their benefits by allocating maximum users to edge servers in a specific area at the lowest computing resource and transmitting power costs. This study proposed a game theoretic approach to solve the edge computing resource allocation problem. In [80], authors proposes the optimal data distribution strategy that minimize the cost incurred, which includes two major components, the cost of data transmission between the cloud to edge servers and the cost of data transmission between edge servers. In [81] a parametric Bayesian optimizer is implemented for hardware resource allocation to increase the overall computational capacity of a 5G-based MEC system. In [82, 83], the authors assumed that the edge servers' coverage areas do not intersect while solving the resource allocation problem. However, edge server coverage areas overlap to avoid non-service areas [19].

2.1.3 Optimal Resource Allocation

Many types of research have been done to attain optimality in the area of edge resource allocation to users in terms of energy consumption [84–86], load optimization [69],

resource utilization [87], maximizing the served users [88], etc. These studies show that allocating resources in a way that maximizes the use of limited resources at edge devices significantly impacts system performance.

Authors in [89] proposed a new bio-inspired hybrid algorithm (NBIHA) for managing edge computing resources and task scheduling. The primary purposes of their proposed work are to maximize resource utilization by efficiently scheduling task offloading. The study [13] proposed a distributed alternating-direction multiplier (ADM) algorithm for resource allocation that efficiently uses limited resources and improves the content caching strategy. [90] proposed a window-based Rate Control Algorithm (w-RCA) that allocates optimal resources to users while maximizing resource utilization. The authors in [91] investigate the resource allocation and IoT task scheduling problems and propose a decentralized approach (DoSRA) that maximizes resource utilization while improving the user experience. [92] proposes an Adaptive Priority-based Resource Allocation (A-PBRA) mechanism for maximizing the user request for task offloading by efficiently utilizing edge resources. Lai et al. [42] proposed a heuristic approach for allocating edge resources to application users that minimizes the number of edge servers required, lowering application users' costs. Nguyen et al. [2] proposed a price-based resource allocation approach, a game-theoretic approach for allocating the optimal bundle of edge computing resources to each service's users to maximize their utilization. The authors in [14] suggested a combined edge resource allocation and IoT task scheduling solution through the use of a linear programming approach that maximizes the revenue.

Li et al. [85] developed the dynamic resource allocation method for the edge computing enabled IoT environment to meet the users' diverse performance needs while achieving energy efficiency. By dynamically modulating the CPU-frequency scaling of edge servers, Chen et al. [86] minimize the energy consumption of the resource allocation scheme in edge computing. In [93], the authors propose a heuristic solution to allocate the users to edge servers from the perspective of edge server infrastructure

providers. The study [94] propose the dynamic service placement framework that gives an approximate solution handling end-user mobility for cost-efficient edge computing. The edge infrastructure deals automatically with end-user to edge server allocation. The researchers in paper [95] examine the edge resource allocation scenario with users' mobility, requiring the reallocation of users among edge servers. Their proposed approach to user allocation seeks to reduce the number of reallocations while maximizing users. These studies do not optimize the requisite servers for app vendors, which is one of our objectives. Qiang et al. [41] and Phu et al. [42] propose game-theoretic based algorithms and heuristic approaches, respectively, to solve the ERA problem from the app vendor's perspective. However, they see the problem in respect of single app vendors and single infrastructure providers. The other dimension to view the app user allocation problem is to allocate the edge servers' resources to different apps (services). In [2], the authors investigate the resource allocation for edge computing, where multiple app vendors compete for computing resources. One of the major drawbacks of studies [41,42] is that these proposed approaches take exponential iteration to reach Nash equilibrium. [81] propose a new approach to optimizing hardware resource allocation for edge nodes in a multitier MEC hierarchy

2.2 Literature Review on Networking Resource Allocation

Trillions of things are deployed at the sensing layer in various scenarios to integrate the physical world with the cyber [30]. In many applications, sensor-enabled IoT devices are grouped such as disaster management mechanisms [53], multimedia communications [96], smart cities [97], etc. In grouped-oriented applications, multicast is used to transmit the data to various destinations [53, 96, 97]. In last decade, various research investigated multicast communication in different dimensions such as authentication mechanism [30], cost minimization [98], delay reduction [99], energy efficient transmission [100], etc.

Finding an minimum cost multicast tree is an NP-complete problem [35]. To tackle this issue, the researchers explored the heuristic approaches to find approximate solution [96, 99, 101, 102]. Nevertheless, the heuristic-based solutions give undesirable outcomes for dense networks [103]. The researchers used the meta-heuristic approach to find an efficient solution for group communication to tackle this issue [104–107]. Following this, Wang et al. [31] and Haghighat et al. [108] proposed approaches to construct the multicast routing tree based on Particle Swarm Optimization and Genetic Algorithm, respectively. However, these techniques suffer from high complexity because of the large multidimensional search space [53, 103]. In addition, the destination users can have their individual requirements in terms of throughput, energy, delay, jitter, etc. Such solutions are not implementable as rational users may not follow centralized protocol due to individuals' interests [52].

Various studies proposed the game-theoretic approach to reduce the cost of multicast data transmission where the multicast tree is a PNE [52, 109–114]. Game theory is a powerful tool to design complex problems for distributed systems in a decentralized manner. It gave freedom to each user to construct their path to get feed. In these studies, the multicast tree construction process is modeled as a cost-sharing network game with destination users as players where each destination has an individual objective to minimize its cost. In a cost-sharing network game, the PNE depend on the cost-sharing mechanism of edges among their users [110]. The research works in [109,111] formulated the problem of multicast tree construction with optimum cost as a network cost-sharing game. These studies used a fair-division cost-sharing scheme derived from the Shapley value to divide the cost of network edges. In [52, 113], the authors evaluated the quality of PNE with the uniform cost-sharing schemes for the network design to multicast transmission. In some scenarios, nodes have their weight according to their priorities. The research studies [112,115,116] proposed network cost-sharing games with weighted cost-sharing games derived from the Shapley value to deal with these cases.

2.3 Research Gap

Certain limitations have been observed in the approaches of edge computing and networking resource allocation, which are as follows:

- Both the allocation of edge computing resources for task offloading and the allocation of edge networking resources for group communication are NP-Complete problems [42]. As a result, no approach provides the global optimum solution while leaving room for improving the existing solution's quality.
- Researchers explored various algorithms based on heuristic approaches for solving these problems [19, 42]. Nevertheless, the heuristic-based solutions give undesirable outcomes for dense networks [103]. To get the more accurate results, the meta-heuristic approaches are used to find the efficient solutions for these problems. However, meta-heuristic-based algorithms take longer to converge to the solution because of the large multidimensional search space [53]. Moreover, a centralized solution may not meet the needs of each independent app user motivated by self-interest (who has their own QoS and resource requirements) [52]. Consequently, the system performance may not be effective. To solve the problems in the environment of self-interested users, [2, 41] proposed the game-theoretic approach. However, these approaches also suffer from the high time complexity.
- Existing studies do not consider the geographically dynamic and uneven user density to solve the edge resource allocation problem. Additionally, they do not take into account the bottleneck resources on the multi-tenant edge servers while solving the problem.
- To the best of our knowledge, only a few researchers have studied the resource allocation problem from the perspective of app vendors, and no study has investigated the trade-off between the app vendors' benefits and QoS for their app users. A few studies examine how to improve the resource utilization of multi-tenant edge servers.