Chapter 1

Introduction

1.1 Background

1.1.1 Internet of Things

The Internet of things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks without requiring human-to-human or human-to-computer interaction. In other words, IoT integrate the cyber world with the physical world by sensing and collecting data from the surrounding environment and transmitting it to other devices over the Internet [1]. As shown in Figure 1.1, the IoT system comprises the following components :

- Physical Things: The physical things include sensors, actuators, processing, control, and power. The sensors convert some physical phenomenon into an electrical

Figure 1.1: IoT stack: the system of systems

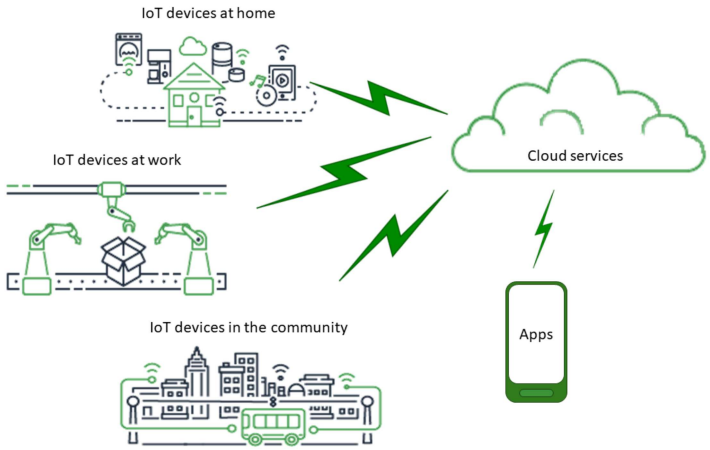

Figure 1.2: IoT devices and environments with cloud computing paradigms.

impulse that can then be interpreted to determine a reading. The processing unit then processes the data, sending the output to the actuators. An actuator converts electrical input into physical action. The control unit provides signals to these units to control them.

- Communication Channels. A communication channel is the medium used to transport information from one network device to another. In IoT system, many combinations of internet protocols and connectivity solutions enable Thing-to-Thing, Thing-to-Server, or Server- to-things data transfer.
- Software. Software provides the ability to ingest, process, store, and analyze data that originates from a Thing. Software also provides application level capabilities for humans to visualize data and interact with the IoT system.
- Operations. The IoT system processes and analyses the data, extracts information from the data, and makes decisions based on the information. Since IoT devices typically have limited computing power, complex tasks can be offloaded to cloud infrastructure.
- Data. IoT devices collect data from their surroundings and send it to other devices via the internet. Data is a byproduct of the IoT system. Without data, the IoT would serve little purpose.

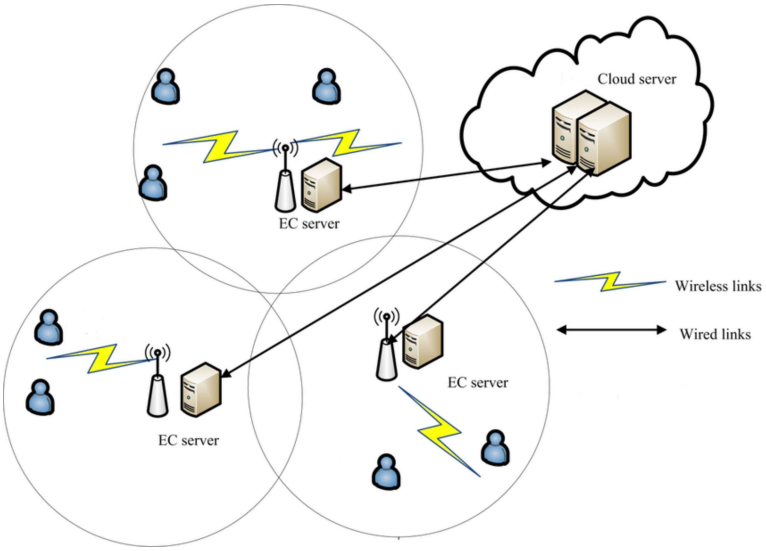

Figure 1.3: IoT devices and environments with edge-cloud computing paradigms.

The world has seen exponential growth in the Internet of Things (IoT) over the past decade, fueling a new set of computing-intensive applications such as virtual/augmented reality (VR/AR), tactile Internet, 4K/8K UHD video, and various other IoT applications [2]. Application (app) vendors typically allow their app users to offload the heavy computation services of such applications to cloud servers to cope with the limited computing capacity of IoT devices [3], as shown in Figure 1.2. However, it frequently suffers from unpredictably high network latency, which causes users to experience unexpected app behavior [4]. Various IoT applications, such as production line operation states in a smart factory, a patient's condition in smart health care, fire detection systems, etc., should be offloaded while ensuring time-bound decision making [5]. Moreover, many IoT applications generate megabytes or gigabytes of data per second, such as security management systems with cameras, self-driving vehicles, traffic control systems, etc. Those require high bandwidth for task offloading and quick response to deal with any real-world situation [6]. Edge computing is proposed to address these issues [7,8]. This technology brings computing power closer to the needed areas to reduce latency and save bandwidth and energy, as shown in Fig. 1.3.

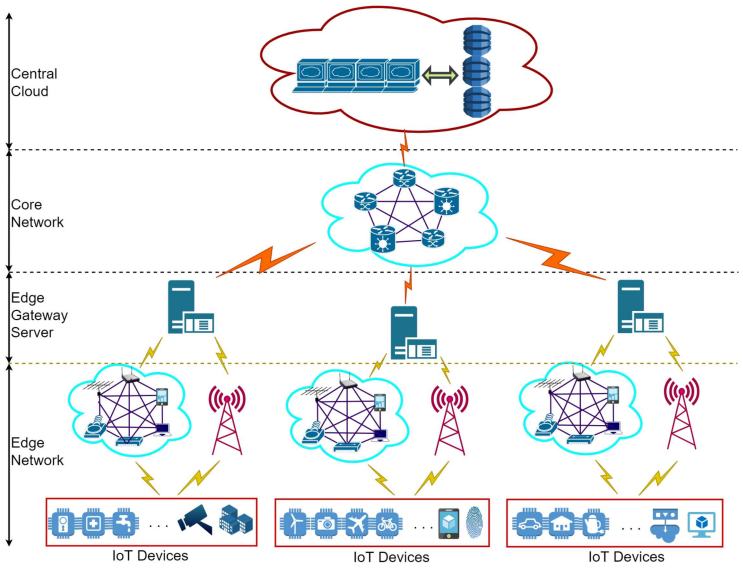

Figure 1.4: Edge computing enabled IoT system.

1.1.2 Edge Computing Enabled IoT

Edge computing is defined as "a part of a distributed computing topology in which information processing is located close to the edge, where things and people produce or consume that information." In edge computing, servers are deployed at various strategic locations by the different service providers [9], e.g., Telstra, AT&T, etc., as shown in Fig. 1.4. Placing servers close to where they are needed reduces network congestion in the core network [10]. Generally, these servers are called edge servers because they are located at the edge of the network. According to the literature, edge servers can be single-tenant or multi-tenant, with each tenant designating a specific app user. They are as follows:

- Single Tenant: A single instance of the software and supporting infrastructure serves a single app user. Essentially, there is no sharing happening in this option.
- Multi-Tenant: Multi-tenancy means that a single instance of the software and its supporting infrastructure serves multiple app users. Each tenant's data is isolated and remains invisible to other tenants.

In this thesis, we study multi-tenant edge servers. These edge servers offer computing resources, e.g., memory, CPU, storage, etc. App vendors hire computing resources

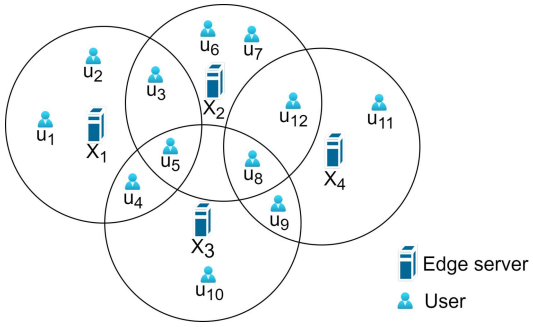

Figure 1.5: The example of geographical distribution of app user and edge servers. The circles represent the coverage area of each edge server.

on the edge servers to serve their app users, deploy app-related services and software on edge servers, and assign app users to edge servers [11–13]. The app vendor pays the edge infrastructure providers for the hired resources. App users then offload tasks from their IoT devices to the allocated edge servers. The edge server executes the offloaded task and provides the required response to the IoT users or different locations [14]. Thus, edge computing is a useful technology that enables IoT app users to offload computational tasks to nearby edge servers via edge networks (WiFi, WSN, 4G/5G, and so on), as shown in Figure 1.4. Task offloading in edge computing has been the subject of extensive research in recent years [15–18].

As illustrated in Fig 1.5, an edge server typically serves a small geographical area. The app users within the coverage area of an edge server can offload their tasks to that edge server. This constraint is termed the ***proximity constraint*** [15]. The area not covered by any edge server is referred to as a non-service area, and app users in this area are unable to use the edge computing services. To avoid the non-service area, the coverage area of edge servers can overlap [19], as shown in Fig 1.5. An app user in the overlap area can connect to any edge server that provides the app-related services. Thus, there are many ways to allocate app users to edge servers. The number of app users connected to an edge server must not exceed the server's capacity; otherwise, Quality of Service (QoS) may suffer. This constraint is termed the ***capacity constraint*** [20]. The ***QoS*** measures the overall performance of a computing service in terms of availability, reliability, software response time, latency, etc.

1.1.3 Game Theory

Game theory is a bag of analytical tools designed to help us understand the phenomena that we observe when decision-makers interact [21]. The basic assumptions that underlie the theory are that decision-makers pursue well-defined exogenous objectives (they are rational) and take into account their knowledge or expectations of other decision-makers' behavior (they reason strategically). In other words, game theory is the study of mathematical models of strategic interactions among rational decision-makers, where ***strategic interactions*** mean that the outcome for each decision-maker depends on the decisions (strategies) of all. Each decision-maker is **rational** in the sense that he is aware of his alternatives, forms expectations about any unknowns, has clear preferences, and chooses his decision deliberately after some process of optimization.

1.1.3.1 Game

A game is a description of strategic interaction that includes the constraints on the actions that the players can take and the players' interests, but does not specify the actions that the players do take. A strategic game consists of following:

- A finite set of decision makers (the set of players)
- A nonempty set contains all the decisions of each decision maker
- A preference relation that defines if a player presented with any pair of decisions, knows which of the pair he prefers, or knows that he regards both decisions as equally desirable. The preferences are consistent in the sense that if the player prefers the action a to the action b, and the action b to the action c, then she prefers the action a to the action c.
- A payoff function for each decision maker that defines the preferences of the player in response to other player.

1.1.3.2 Nash Equilibrium

The most commonly used solution concept in game theory is that of Nash equilibrium [22]. This notion captures a steady state of the play of a strategic game in which each player holds the correct decision in response of the other players' decisions and acts rationally. It does not attempt to examine the process by which a steady state is reached.

1.1.3.3 Best Response Dynamics

A particularly simple theory assumes that in each period after the first, each player believes that the other players will choose their decisions in the previous period [23]. In the first period, each player chooses a best response to an arbitrary deterministic belief about the other players' decisions. In every subsequent period, each player chooses the best response to the other players' decisions in the previous period. This process is known as Best Response (BR) dynamics. A decision profile (tuple of decisions containing one decision of each player) that remains the same from period to period is a pure Nash equilibrium of the game. Further, a pure Nash equilibrium in which each player's decision is her only best response to the other players' decisions is a decision profile that remains the same from period to period.

1.1.3.4 Potential Game

In game theory, a game is said to be a potential game if the payoff of all players to change their decisions can be expressed using a single global function called the potential function [24]. This function monotonically decreases with every update in the decision profile under the best response dynamics [25]. The intuition behind the potential is that it tracks the convergence to Nash equilibrium using BR dynamics.

1.2 Research Motivation

Edge computing lowers the bandwidth requirement while providing low-latency services to IoT device app users. As a result, this technology is especially useful for high-volume streaming applications or critical systems that require real-time decision-making, e.g., health care, autonomous traffic systems, cloud gaming, etc [26]. Edge computing also benefits users of limited storage and computing capabilities devices, e.g., sensors, wearables, smartphones, etc., by offloading intensive computational tasks to the nearby edge servers [27, 28]. In this way, the central cloud is not required to provide all the online services single-handedly.

To use the edge computing services, app vendors hire computing and networking resources on edge servers and deploy their services on them. These resources are allocated to app users or different services in order to offload tasks from IoT devices to edge servers. Allocating edge computing and networking resources is an essential problem as it affects overall system performance in various ways, which motivates us to study it. This study has three main actors, so we investigate this problem from their perspectives. The allocation of edge computing and networking resources impacts different actors in the following ways:

- **From the perspective of app users:** An inappropriate resource allocation to the app users may result in a large number of app users being unallocated due to the proximity and capacity constraints of the edge servers, resulting in the decline of the QoS. Additionally, because there are numerous ways to assign app users to edge servers, the app users' QoS may be compromised if the number of app users assigned to an edge server exceeds the capacity of the edge servers. Therefore, app users aim to improve their QoS while allocating the edge resources.
- **From the perspective of app vendors:** As the app vendor pays the edge infrastructure providers for the hired resources, each app vendor's objective is to lower its costs and maximize the utilization of hired resources on edge servers.

Thus, the benefits to the app vendors depend on how the edge resources are allocated. To achieve the purpose, the app vendor aims to provide services to the maximum number of app users while utilizing the least number of edge servers possible.

- **From the perspective of edge infrastructure providers:** If the number of app users assigned to an edge server is significantly less than the capacity of the edge server, the edge server may be underutilized. However, if the number of app users assigned to an edge server exceeds the capacity of the edge servers, the edge server may become overloaded. As a result, inefficient edge resource allocation reduces the utilization of edge computing infrastructure, reducing the benefits to infrastructure providers. Edge infrastructure providers seek to maximize the overall utilization of the resources they provide.

In the literature, finding an optimal solution for the edge resource allocation problem is always challenging as the geographical density of app users is uneven and the optimal assignment of these app users to edge servers is an NP-complete problem. Thus, the primary challenges are solving the problem in polynomial time and providing an approximate solution much closer to the centralized optimum.

1.3 Research Problems and Objectives

This thesis studies the edge resource allocation problem from the perspectives of app vendors, app users and infrastructure providers. The problem of edge resource allocation is investigated in two stages: 1) the edge computing resource allocation (required for processing the offloaded task) and 2) the networking resource allocation (required for transmitting the data).

1.3.1 Edge Computing Resource Allocation

The computing capacities of edge servers are rented out using a pricing model by the edge infrastructure providers. The pay-as-you-go pricing model used by several infrastructure providers, including Salesforce, Azure, and AWS, is investigated in this thesis. This pricing model defines the cost based on the resource utilization by the app users. Therefore, an app vendor intends to provide app services to the maximum number of users with the least number of edge servers. This optimizes the use of edge resources while lowering overall system costs. However, when an edge server has to serve more app users than its capacity, the Quality of Service (QoS) deteriorates. Thus, establishing a trade-off between cost and QoS is a challenge in the process of allocating the hired edge computing resources to users. Another challenge in solving this problem is that multiple app vendors may compete for the same edge computing resources. In this scenario, it is not easy to allocate these resources to various app vendors so that the overall cost of all app vendors is minimized while maximizing the allocated app users and resource utilization. Moreover, maximizing resource utilization and handling the geographically dynamic user density are other critical challenges in edge resource allocation to the edge servers.

The objectives of the thesis in the process of edge computing resource allocation are as follows:

- Improving the QoS provided to the users.
- Maximizing the number of app users allocated to edge servers.
- Minimizing the required edge servers to reduce app vendors' costs.
- Maximizing resource utilization.
- Balancing the load on the edge servers.

1.3.2 Edge Networking Resource Allocation

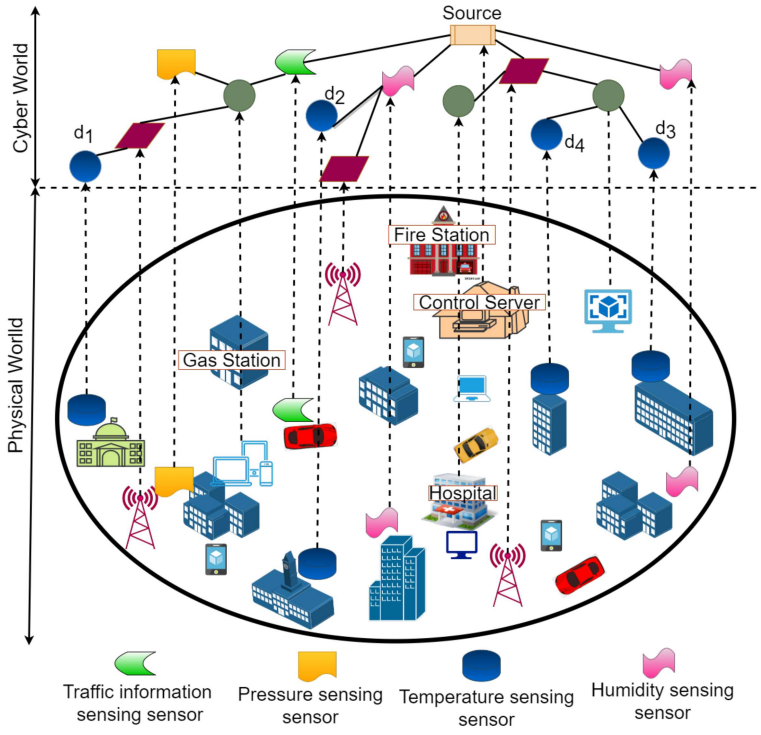

Figure 1.6: An illustration of the data transmission in the smart city.

The multiple IoT devices collectively work in many applications such as disaster management, border security management, smart farming, smart cities, etc. For example, as shown in Fig. 1.6, temperature sensing devices (blue round drum-shaped nodes) can work collaboratively to satisfy the environmental requirements. In this process, any temperature sensing device may need to send its sensing data through different devices to other temperature sensing devices. Thus, IoT devices collectively acquire the data and transmit it to the edge server for further processing [29]. The edge server then takes action based on processed data and provides the required information to various destinations. For an example of a fire scenario (Fig. 1.6), the edge server processes the data transmitted by temperature sensing devices to detect fire. It acts as a source node for a rescue operation, disseminating the information needed to various destinations (fire station, gas station, police station, etc.) to provide services. The other devices (sensors, stations, etc.) help forward the messages for the rescue operation. In such cases, multicast communication is more effective than broadcast or unicast as it

efficiently utilizes edge networking resources [30].

A better QoS with efficient utilization of edge networking resources can be provided by constructing the lowest cost multicast tree for group communication. We consider multiple objectives for building a cost-effective multicast tree, e.g., high throughput, low delay, low energy consumption, etc. These multiple objectives can be efficiently combined in the form of edge costs. Therefore, it is required to construct the lowest cost tree based on the cost of edges, which includes the source node, all destination nodes, and some additional intermediate nodes (required to build the minimum cost tree between source to destinations). This type of lowest cost tree that includes a specific subset of nodes with some additional nodes is called Minimum Steiner Tree (MST) [31]. Thus, one of the approaches to achieve the least cost multicast tree based on the edge costs is to form the MST [32–34]. However, finding an minimum cost multicast tree is an NP-complete problem [35]. Thus, it is challenging to construct a multicast tree that efficiently utilizes the networking resources while providing better QoS to users.

The objectives of the thesis in the process of edge networking resource allocation are as follows:

- Maximizing the throughput for end-to-end data delivery.
- Minimizing the delay in end-to-end data transmission.
- Minimizing energy consumption for end-to-end data transmission.

1.4 Contribution of the Thesis

This thesis proposes game-theoretic approaches to solve the edge resource allocation problem. Game theory is a powerful tool for the design of decentralized mechanisms and has been widely used in the field of distributed computing. There are three primary reasons for using the game-theoretic approach to solve the edge resource allocation problem: 1) App users can pursue their interests based on their specific requirements in

terms of QoS and cost. 2) Empowering each app user with the ability to make decisions addresses the issue of app user allocation in a decentralized manner, easing the burden of finding a centralized solution. 3) It scales with the size of the problem, such as the number of edge servers and app users. The following are the main contributions of this thesis.

1.4.1 Solving the Edge Computing Resource Allocation Problem

The following approaches are proposed to solve the edge computing resource allocation problems:

- In the first study, we investigate the problem of allocating the app user to the hired resources of multi-tenant edge servers by the app vendor while establishing a trade-off between the users' QoS and the vendor's cost. This problem is studied from the perspective of individual app vendors and their app users and is referred to as the App User Allocation (AUA) problem. We proposed a game-theoretic approach to solve the AUA problem that formulates this problem as the User Allocation Game (AUGame), a potential game. This game employs an AUA algorithm to reach the solution faster, establishing a balance between the QoS and the cost.
- In the subsequent study of allocating the edge computing resources, we consider how the different app vendors (services) compete for the same edge computing resources. This problem is referred to as the Edge Resource Allocation (ERA) problem. We propose a game-theoretic approach to solve this problem, which optimizes the cost incurred by the app vendors while maximizing resource utilization. In this approach, the ERA problem is formulated as the Edge Resource Allocation Game (ERAGame), a potential game. This game employs an ERA algorithm to reach the solution faster.
- In the following work, we investigate resource allocation from the perspective of

edge infrastructure and app users to maximize resource utilization while improving the users' QoS. We propose a distributed user allocation approach that finds the bottleneck resource on each multi-tenant edge server and balances the load on them. This approach accommodates the geographically dynamic density of app users by moving the app users from the overloaded edge servers to the underutilized edge servers, which efficiently utilizes the computing resources of the multi-tenant edge servers.

1.4.2 Solving the Edge Networking Resource Allocation Problem

In this study, we allocate the edge networking resources for group communication from an edge server to app users. A cost-effective multicast tree for group communication is required to use the edge networking resources efficiently and effectively. We consider multiple objectives for building a cost-effective multicast tree, e.g., high throughput, low delay, low energy consumption, etc. These multiple objectives can be efficiently combined in the form of edge costs. In this work, a weighted cost-sharing scheme is proposed to divide the cost of network edges among their users. We then proposed a game-theoretic approach that constructs a cost-effective multicast tree using the proposed cost-sharing method. While building the optimal tree, this approach aims to maximize the throughput and reduce the delay and energy consumption for data transmission.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presents the literature review on the edge resource allocation problem. Chapter 3 presents a game-theoretic approach that allocates the app users to the hired resources of the edge servers. Chapter 4 introduces a game-theoretic approach that allocates the edge computing resources to the various app vendors. Chapter 5 presents an app user allocation approach that finds

the bottleneck resource on each edge server and balances the load on the edge servers. Chapter 6 presents a game-theoretic approach that solves the edge networking resource allocation problem for the group communication from the edge server to various app users. Chapter 7 summarizes the thesis work with promising future research directions in the area of edge computing-enabled IoT.