Resilient Resource Allocation for Service Placement in Mobile Edge Clouds

PEIYUE ZHAO

Doctoral Thesis
Stockholm, Sweden 2021

KTH School of Electrical Engineering and Computer Science
TRITA-EECS-AVL-2021:20 SE-100 44 Stockholm
ISBN 978-91-7873-816-8 Sweden

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av doktorsexamen torsdag den 15 April 2021 klockan 13.00 i Kollegiesalen, KTH, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Mobile edge computing makes available distributed computation and storage resources in close proximity to end users and allows to provide low-latency and high-capacity services within mobile networks. Therefore, mobile edge computing is emerging as a promising architecture for hosting critical services with stringent latency and performance requirements, which otherwise are challenging to be addressed in conventional cloud computing architectures. Notable use cases of mobile edge computing include real-time data analytic services, industrial process control, and computation offloading for Internet of things devices. However, those services rely on efficient resource management, including resource dimensioning and service placement, and require to be resilient to cyber-attacks, to faulty components and to operation mistakes. The work in this thesis proposes models of resilient resource management that support rapid response to incidents in mobile edge computing and develops efficient algorithms for the resulting resource management problems.

In the first part of the thesis, we consider resilient resource management for edge computing systems in which failover is realized by restoring additional service instances in different mobile edge computing nodes in case of failures. We first develop a placement algorithm based on Benders decomposition and linear relaxation to determine the mobile edge computing nodes to be opened and to compute the placement of the service instances with respect to a set of considered failure scenarios, with the objective of minimizing operation costs. Upon the occurrence of a failure scenario, service migration is to be triggered to migrate the service instances from one placement to another placement, for which we further develop service migration algorithms to schedule migration under time constraints, so as to minimize service interruptions.

In the second part of the thesis, we consider resilient resource management in mobile edge computing for services with different levels of resilience requirements. Resilience is achieved by synchronizing states of the services to two types of standby instances that maintain the trade-off between energy consumption and activation time such that the standby instances can take over the service seamlessly as an instantaneous failure response. We formulate the joint problem of resource dimensioning and service placement for minimizing energy consumption and prove that it is NP-hard. We propose an efficient approximation algorithm based on Lagrangian relaxation to decide the type, amount, and locations of the computation resources and to compute the placement of service instances and their standby instances. We then consider the same resilience model but for hosting periodic services in mobile edge computing systems with resources portioned into availability zones, under schedulability constraints. We formulate the corresponding resilient resource management problem as a non-linear programming problem and prove that it is NP-hard. We propose efficient solutions based on approximation programming and primal-dual approaches for resilient service placement.

By considering different models of resilient service placement in mobile edge computing, the results in this thesis provide effective, efficient, and scalable resource management algorithms for emerging mobile edge computing systems.

Sammanfattning

Databehandling i det mobila nätverkets utkant (mobile edge computing) innebär att distribuerade beräknings- och lagringsresurser tillgängliggörs direkt i infrastrukturen för det mobila nätverket. Den fysiska närheten till slutanvändarna gör det möjligt att tillhandahålla tjänster med liten fördröjning och hög kapacitet. Därför växer nu teknologin fram som ett lovande alternativ till molnberäkning, särskilt för att driva kritiska tjänster med strikta krav på fördröjning och prestanda som är svåra att uppfylla i traditionella molnberäkningssystem. Några exempel på viktiga tillämpningsområden är tjänster för dataanalys som måste köras i realtid, styrning av industriprocesser, samt avlastning av beräkningsintensiva uppgifter från sakernas-internet-enheter. Denna typ av system ställer dock höga krav på effektiv hantering av resurser, vilket innefattar att tilldela rätt mängd resurser och att avgöra var i nätverket en tjänst ska köras. Dessutom ställs höga krav på resiliens mot cyberattacker, felande komponenter, samt driftfel. Arbetet i denna avhandling lägger fram modeller för resilient resurshantering för databehandling i nätverkets utkant, som kan ställa om snabbt i händelse av fel, samt utvecklar effektiva algoritmer för att optimera hanteringen av resurser i dessa modeller.

I avhandlingens första del studeras en mekanism för resilient resurshantering i system för databearbetning i nätverkets utkant, som i händelse av fel initierar nya instanser av den felande tjänsten i andra beräkningsnoder och återställer tjänsten. En algoritm utvecklas, baserad på Benders dekomposition och relaxation, som tar hänsyn till flera olika felscenarier för att avgöra vilka beräkningsnoder som skall användas och var de nya tjänsteinstanserna skall köras, med målet att minimera driftskostnaderna. Därutöver utvecklas algoritmer för schemaläggning av migration av tjänster från en beräkningsnod till en annan, i den händelse att ett felscenario inträffar, som tar hänsyn till tidsbegränsningar och har som mål att minimera avbrott i aktiva tjänster.

I avhandlingens andra del studeras resilient resurshantering då olika tjänster har olika krav på resiliensnivå. Resiliens uppnås genom att synkronisera tillstånden hos de tjänster som är i drift med två olika typer av ”standby”-instanser som är i viloläge och redo att ögonblickligen och sömlöst ta över efter en felande tjänsteinstans. Ett problem formuleras, samt bevisas vara NP-svårt, för att samtidigt optimera både tilldelning av resurser och av beräkningsnoder till tjänsterna, med målet att minimera energiåtgång. En effektiv algoritm, baserad på Lagrangerelaxation, ges för att hitta en approximativ lösning till problemet, det vill säga för att avgöra typ, mängd och placering av de nödvändiga beräkningsresurserna, samt placering av tjänsteinstanser och deras respektive ”standby”-instanser. Dessutom utvidgas modellen till tjänster som måste köras periodiskt och för att inkludera resurser som är tillgängliga endast i vissa zoner, med begränsningar för schemaläggning av tjänster. Ett icke-linjärt optimeringsproblem formuleras för att lösa detta schemaläggningsproblem och bevisas vara NP-svårt. En effektiv lösningsmetod utvecklas, med hjälp av approximationsmetoder och primal-dual-metoder, för resilient nodplacering av tjänster.

Avhandlingen överväger flera olika modeler för resilient resurshantering, det vill säga för att besluta hur tjänster ska delas upp mellan olika noder i nätverkets utkant, samt lägger fram effektiva och skalbara algoritmer för att optimera resurshantering i framtidens databearbetningssystem i det mobila nätverkets utkant.

Acknowledgments

First of all, I would like to express my deepest appreciation to my supervisor György Dán for providing me the opportunity to pursue my PhD study. He supported me with his expertise, dedication, patience, and motivation. It would not be possible for me to complete my study without his guidance and encouragement. He always stood next to me whenever I faced difficulties and crossroads and he supported me to develop as a researcher and as a person. I treasure this remarkable journey that we worked together.

Also, I am extremely grateful to my co-supervisor Gábor Fodor. He is the person who first introduced me to the world of research. I appreciate his continuous support and exceptional kindness throughout my master and PhD studies. He devoted his time and effort to share knowledge and experience with me and encouraged me to explore new potentials. I also wish to thank Miklós Telek, and I appreciate his valuable inputs and help.

I would like to extend my sincere thanks to everyone in the CERCES project, especially project manager Henrik Sandberg. I am also grateful for the stimulating research environment of the TECoSA competence center at KTH. I always feel being enlightened and inspired through the activities in these projects, and many insights from discussions in these projects boosted my research. In particular, I want to thank my PhD and postdoctoral colleagues Andreas, Henrik Forssell, Jezdimir and Serkan. I enjoyed all the exciting discussions, collaborations and fikas.

I must also thank all the past and present members of the NSE division. Particularly, I very much appreciate Viktoria Fodor for vibrant talks and for providing valuable feedbacks to my thesis. I am thankful to Gunnar Karlsson for accepting me as a member of NSE, and for involving me in teaching activities. Thanks should also go to Emil and Valentino for helping me during my onboarding stage, and to Slađana for being a stimulating and cheerful study buddy. Many thanks to Ezzeldin and Lamia for being creative and bringing new ideas, to Qing for sharing her experience and for having fruitful discussions, and to Ming and Feridun for joining me in lab sessions. Special thanks to Thomas for inspiring discussions and for helping me translating the Swedish abstract of the thesis.

I also cherish all my friends in China and in Sweden for their unwavering accompany and encouragement. I especially thank Dashun who has a broad mindset and inspired me in numerous ways. I would also like to thank Shuo for caring and for being virtuous and joyful.

Finally, yet importantly, I am deeply indebted to my family for their relentless support and profound belief in me. Especially, I thank my parents for their unconditional and immeasurable love. I also gratefully appreciate my uncle-in-law John Liu and my uncle Eric Li for giving me advice on studying in Sweden.

Contents

Chapter 1

Introduction

1.1 Background

The era of 5G is foreseeing a tremendous increase in the number of mobile network subscriptions and in mobile network data traffic. According to a recent forecast by Ericsson for the period of 2020-2026, the global number of mobile subscriptions is expected to increase from 7.9 billion to 8.8 billion, and the amount of global mobile data traffic is expected to more than double [Eri20]. The increase in data traffic is driven by the proliferation of mobile network coverage, the massive deployment of IoT devices, mobile broadband subscriptions, and softwarization in industry. Consequently, the demand for delivering conventional cloud computing based services in mobile networks is also increasing dramatically, for example, video streaming services, social network services and online retail services. At the same time, new types of cloud based services are highly demanded, for example, mobile cloud gaming, remote control services for aerial and ground vehicles, and control services for industrial processes [Abb+17; Pop+21; PM17].

To accommodate the increased demand for cloud services in mobile networks, conventional cloud computing platforms (e.g., based on data centers) are continuously expanding their service capacity and thereby improving the quality of service (QoS) they provide. However, new computing platforms and architectures will be required that scale better with the explosive increase of service demands, for a variety of reasons. First, for mobile users to access conventional cloud services, mobile traffic needs to traverse the mobile backhaul and possibly the mobile core, causing extra communication latency that may exceed the latency requirement of critical services. Furthermore, for conventional cloud computing platforms to improve QoS of the delivered services in mobile networks, intensive investment in computational and communication resources are required, which further increases the cost for accessing the services [Sin17]. Finally, network and context awareness can help to provide localized services to end users, for improving their user experience. The information necessary for implementing such services is, however, usually

only available within the mobile network [Kek+18].

To cater for the increasing demand, mobile edge computing is emerging as a promising architecture to provide cloud computing capability at the edge of mobile networks. Mobile edge computing allows a seamless integration of mobile network operators and services providers, for improving the experience of services of mobile users, and for enabling new types of services.

1.2 Challenges

Mobile edge computing features distributed computational resources and the promise to serve a large number of services and end users. Nonetheless, meeting this promise imposes challenges to be addressed for mobile edge computing to provide services in mobile networks. The first challenge is the need of efficient resource dimensioning schemes for deploying mobile edge computing resources to set the foundations for hosting services [Zen+19; HD19]. Second, there is a need for efficient service placement schemes adapted to the heterogeneous components of mobile edge computing systems and the diverse requirements of the services [Pou+20]. Furthermore, there is a need for effective resilience mechanisms to fit services with various availability requirements [Dob+19; HS18]. Nonetheless, due to the interdependence of these challenges, a joint treatment is desirable for maximizing the quality of service in mobile edge computing.

1.3 Thesis Structure

The remaining chapters of this thesis are organized as follows. In Chapter 2, we formulate a general model of a mobile edge computing infrastructure, discuss the computational and communication resources, and propose an abstraction of failure domains in mobile edge computing. In Chapter 3, we discuss different types of virtualized services in mobile edge clouds, and discuss their placement requirements. In Chapter 4, we discuss related works on resilient resource allocation in mobile edge clouds, including resource dimensioning, service placement, and resilience mechanisms. In Chapter 5, we summarize the contributions of the papers included in this thesis and in Chapter 6 we conclude our work and discuss several interesting avenues for future work.

Chapter 2

Mobile Edge Computing Infrastructures

Mobile Edge Computing (MEC) is a paradigm that aims at providing cloud service capability at the edge of mobile networks, in close proximity to end users [Hu+15; HYW19; Abb+17]. According to the Industry Specification Group (ISG) of the European Telecommunication Standards Institute (ETSI), MEC is characterized by low end-to-end communication latency, by on-premises computational and storage resources, and by awareness of user locations and network context information, which allows to improve QoS and to provide new services in mobile networks [Pat+14]. To fulfill these objectives, MEC requires seamless integration of service providers and mobile networks in terms of architecture design and resource management [Pat+14]. In this chapter, we discuss the architecture of MEC and discuss the abstractions of failure domains for MEC from the view of resilient resource management.

2.1 Edge Computing Architecture

MEC provides services to end users in the form of computational and storage resources. MEC resources are expected to be deployed in mobile networks so as to be close to end users, potentially on-premise. Specifically, MEC resources can be deployed next to base stations (BSs) and access points (APs), both in indoor and in outdoor environments, and within the radio access networks (RANs), which connects user equipments (UEs) and the core network of mobile operators [Hu+15; Dem+13]. The exact deployment of MEC resources depends on physical constraints (e.g., power supply, available spaces and deployment budget), performance requirements, and preference of network operators [Hu+15]. Keeping in mind the various MEC deployment options, in what follows we present MEC components in three categories based on their functions and connections, and illustrate them in Figure 2.1 [ZDa; TKH19; Lu+20; Men+19; CZL18; HNR18; Mai+19]. The first

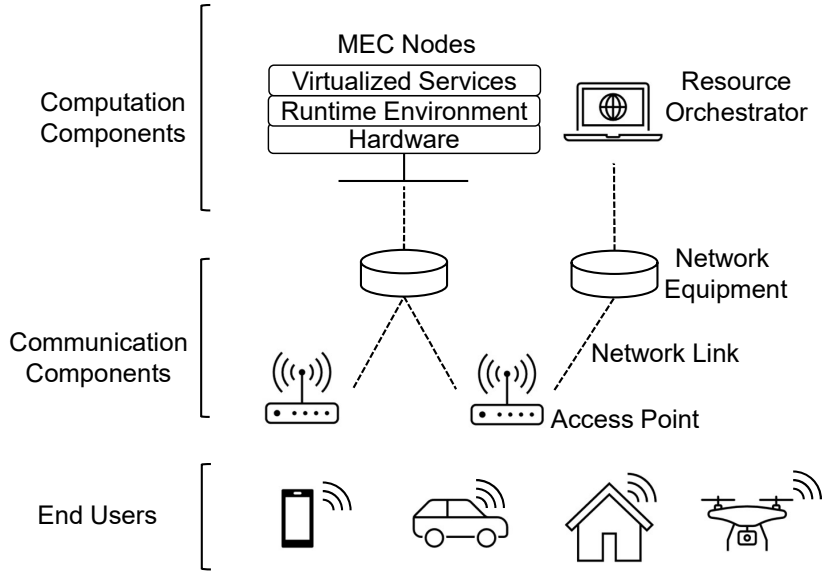

Figure 2.1: An illustration of MEC architecture

category is the end users of MEC, for example, mobile users, connected vehicles, and Internet of Things (IoT) devices. The second category is the computation components, which host services to handle the requests from end users. Finally, the end users and the computation components are connected by the communication components, which is the third category of MEC components.

Communication Components

The communication components consist of communication networks.

Wireless networks are considered the primary means of providing connectivity to the end users in MEC. End users can be connected to wireless networks that utilize unlicensed radio spectrum as a first hop, for example, WiFi and Zigbee networks, and can also be directly connected to BSs and APs of mobile networks that operate on licensed radio spectrum [LSH08; Fod+21]. As an example, 5G networks are supposed to provide enhanced support for MEC with high communication capacity for bandwidth intensive applications [Hai+20]. Furthermore, 5G networks will support Ultra-Reliable Low-Latency Communication (URLLC), which is expected to achieve a latency of 1 millisecond to provide connectivity for use cases with stringent latency requirements (e.g., control services for process control and unmanned aerial vehicles) [Gan+18]. Besides connecting end users, wireless communications has been considered to be used in the mobile backhaul as well, as

an alternative to a wireline backhaul, thus wireless communication also has the potential to be used to connect MEC resources [Kuo+10].

Wired networks are widely used in the mobile backhaul to connect various components of the RAN and the mobile core, establishing connectivity between MEC components and BSs or APs, and thus ultimately the end users. The physical infrastructure consists of cabling (e.g., fiber) and various forwarding and routing equipment. Since the propagation time in the wired media is relatively low, the processing time of the packets in routers and switches accounts for a significant part of the latency.

In communication networks, network functions allow to monitor and manage communication resources. Typical network functions include resource provisioning, performance analytics, fault detection and isolation, firewalls, and load balancing. In the telecommunications industry network function virtualization (NFV) is becoming increasingly popular, as NFV facilitates improving the efficiency and agility of network management by virtualizing network functions that have been traditionally provided by dedicated hardware [Mij+15; Ngu+18]. Importantly, MEC-related network functions can also be virtualized [Ant+20].

Computation Components

The computation components provide computational and storage resources for serving end users of MEC.

MEC nodes can be heterogeneous, depending on performance requirements and deployment scenarios. MEC nodes can consist of commercial off-the-shelf servers, small form factor servers, and purpose-built servers alike (e.g., with onboard artificial intelligence capability, and with high volume storage capability). Conceptually, a MEC node consists of three kinds of components: the virtualized services (VSs) corresponding to user applications, the runtime environment that provides software support for executing the VSs, and the underlying hardware components.

Due to the close integration of MEC and mobile networks, mobile network operators have a natural advantage to become the owners and operators of MEC nodes. However, third-party service providers, for example, owners of real estate facilities, cell tower owners, and vehicle fleet management companies could also own MEC nodes in their own interest, depending on the cost and complexity of deployment [Kek+18].

Resource orchestrators are the entities that monitor and manage the MEC nodes for efficient resource utilization and for providing QoS guarantees. Typical functionalities of resource orchestrators include resource allocation, task scheduling, and software updates. In addition, the resource orchestrators may also be responsible for fault detection in MEC, and for activating failover schemes [Mao+17].

2.2 Failures and Failure Domains in MEC

As MEC systems consist of a variety of components, the operation of MEC is subject to the availability of its constituent resources [Shi+17; Gan+a]. In this section we discuss the potential threats (e.g., failures) of MEC systems and then provide a general abstraction of failure domains in MEC.

Failures in MEC

In general, a failure refers to a condition in which a system is not able to function or to provide services as intended [Vac12]. In MEC failures can happen in all components. Typical failures in the communication components include physical link failures, network protocol failures, network function failures (e.g., routing and forwarding), and degraded performance (e.g., increased latency) [HZL18]. Notable failures in the computation components include outages of computational and storage resources, performance degradation of MEC nodes, and failures in resource management.

Due to the complexity and heterogeneity of MEC components, the causes of failures can be of different kinds. A common cause of failure is faulty components. For example, studies show that hard disks account for 78% of total faults (replacements) on cloud computing servers, and in computer networks load balancing (LB) is the least reliable component [VN20]. The other causes of failures in MEC include software failures, power outage, cyber attacks, and operation mistakes [Sha+16; Gou+16; TDE18].

Failure Domains in Cloud Computing

To capture the impact of failures, the concept of failure domain is often used to refer to the components that are affected by a single failure [Ber+15; Dut20]. Therefore, the scope of a failure domain depends on the type of failures considered. Despite the large variety of failures, in conventional cloud computing systems that consists of data centers, failure domains can be categorized into three levels based on geographical scope [Jak20].

- At the lowest level, a failure domain is usually defined as a rack inside a data center, considering failures that only affect servers within the same rack, for example, the failure of a local network switch, and update failures that affect the servers in a single update group. Failure domains on this level are often referred to as fault domains in cloud computing platforms.

- A medium level failure domain consists of several data centers that are located in a subarea of a geographic region (i.e., a region consists of several such failure domains). Failure domains at this level are also referred to as availability domain or availability zones, subject to area-wide failures (e.g., data centers

connected to a common power source can be affected by the same power outage).

- A high-level failure domain usually consists of all data centers in a single region, referred to as an availability region. For instance, data centers in a single region can be affected by a wide area network (WAN) failure.

Failure Domains in MEC

As MEC has attracted a significant interest from industry and academia, a variety of research works have focused on enhancing the service availability and resilience of MEC [Cui+20; Yin+16; Mou+20; Dev+13; CF20; ZD18; ZDc; Zha+18a]. Nonetheless, consensus on the categorization of failure domains in MEC has not been formed. Motivated by the abstractions of failure domains in cloud computing, the failure domains in MEC can be possibility considered at the granularity of MEC nodes for two reasons. First, MEC nodes serve as basic units for providing computation and storage resources in MEC, and MEC nodes can be considered as a corresponding abstraction of the data centers in MEC. Furthermore, the computation capacity of a MEC node can vary between that of a rack and that of a data center, depending on the deployment configuration. Therefore, it is reasonable to consider failure domains of single MEC nodes and failure domains of multiple MEC nodes.

Failure domain of a single MEC node: Failure domains of single MEC nodes can capture the impact of small scale failures (e,g., local hardware failures and network failures, or operation mistakes). To describe the reliability and failure frequency of a MEC node i, a failure probability p_i can be estimated based on historical data [Zha+14],

$$p_i = \frac{I_i^{\text{failure}}}{I_i^{\text{failure}} + I_i^{\text{recovery}}}, \tag{2.1}$$

where I_i^{failure} and I_i^{recovery} are the mean time between failures and the mean recovery time of MEC node i, respectively.

Failure domains of multiple MEC nodes: This abstraction allows to capture failures that affect multiple MEC nodes simultaneously. For example, a power outage in a local area and communication link failures in the mobile backhaul. Similar to the case of a single MEC node failure, a failure probability can also be calculated for a failure domain consisting of multiple MEC nodes.

Chapter 3

Virtualized Services in Mobile Edge Computing

Mobile Edge Computing (MEC) provides computational resources to end users with low latency and high bandwidth. This allows MEC to be an alternative host for existing services in cloud computing, and allows it to host new types of services [Wan+18; Mei+19]. We refer to the software instances that correspond to those services as VSs, and in this chapter we discuss models of VSs and the requirements to be fulfilled for hosting VSs in MEC.

3.1 Modeling of Virtualized Services

In general, a VS receives and processes data from end users, after which it sends back the computed results or required contents to the end user or to some other end user. Due to the diversity of use cases, VSs can be modelled and categorized with respect to several aspects.

Independent Services and Service Chains

In MEC the service requests from end users may need to be processed by only one service instance, which we refer to as independent services. On the contrary, a service request may need to be processed by multiple interconnected service instances, which constitute a service chain. An independent service can also be considered a service chain that consists of only one service. Each service chain has a source s, where service requests are originated (e.g., a mobile user or a process control plant), a destination d, to where the results of the service chain are delivered, and a set of services. In practice d and s can coincide, for example, d and s can be a process control plant that sends and receives data from a control service.

Depending on the direction of data flow among the services, service chains can be classified as unidirectional, bidirectional and hybrid [Gu+; QN15]. In unidirectional

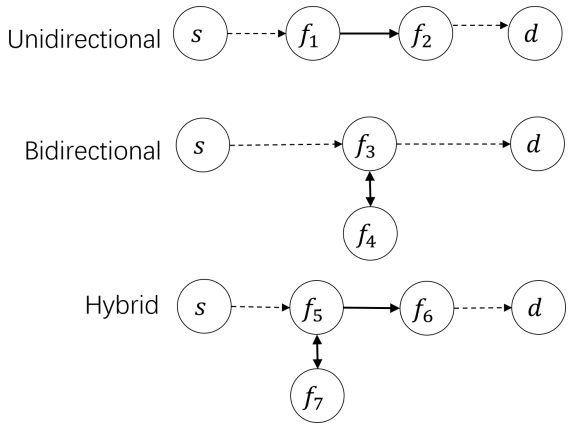

Figure 3.1: An illustration of types of service chains

service chains data pass through the constituent services in one direction, while in bidirectional service chains data are forwarded in both directions. Finally, a service chain is hybrid if data are forwarded both unidirectional and bidirectional. These three types of service chains are illustrated in Figure 3.1, with the solid lines indicating the data flows within the constituent services of a service chain.

Stateful and Stateless Services

From the perspective of whether a VS maintains internal state, the services can be classified based on whether or not they require state to be maintained [Bai18].

Stateful services require state related to the services to be initialized and maintained during execution. The state can be stored locally on the serving MEC node of the service, but it can also be stored in a database. Typical examples of stateful services include proportional integral derivative (PID) controllers and model predictive control (MPC) for processes control, and session-based services (e.g., FTP). Stateful services can be very heterogeneous in terms of the frequency of state updates. For example, services for MPC may have a relatively higher frequency of state updates due to the stringent performance requirement of process control, while session-based services may have a lower frequency of state updates due to interaction with human users.

Stateless services provide services based on the requests from end users, and do not require state to be maintained. Examples of stateless services include image and video analytic services (e.g., object recognition), search engines and RESTful services. By resolving the need of maintaining state, stateless services can reduce the complexity of implementation and that of failure recovery.

3.2 Placement Requirements of Virtualized Services

As MEC provides distributed and heterogenous resources in mobile networks, it allows the freedom to choose on which MEC nodes the VSs are placed. However, it is important for the hosting MEC node to satisfy the placement requirements of the VSs. In this subsection, we discuss the most commonly considered placement requirements of VSs in MEC.

Computational Resources

Arguably the most widely considered requirement of VSs is computational resources. To provide a unified view of the requirements in terms of computational resources, let us consider a general model as follows. Let us denote by $s_{f,i}$ the resource requirement of hosting an instance of VS f on MEC node i, and denote by ω_i the amount of available resources at MEC node i. Then, for each MEC node the following constraint should be enforced,

$$\sum_{f} s_{f,i} \leq \omega_i. \tag{3.1}$$

Depending on the types of services and the granularity of resources, the values and ranges of $s_{f,i}$ and ω_i can be interpreted differently.

- In a general and abstract way, if we consider $s_{f,i}$ as the complexity and workload (e.g., the CPU cycles required per unit time, or the amount of data to be processed) of a service f, then ω_i can model the computing capability (e.g., the CPU frequency) of MEC node i. Therefore, it is reasonable to consider $s_{f,i}$ and ω_i as real numbers.
- For VSs that require isolation for the sake of security and performance guarantees, the computational resource requirement can be allocated based on the execution environment (e.g., virtual machines and containers) [Wai+19]. Therefore, it is customary to set $s_{f,i} = 1$, indicating that each VS requires a dedicated execution environment, while ω_i can be an integer, indicating the number of services that MEC node i can host.
- If we consider a CPU to be the unit of computational resources, then $s_{f,i}$ and ω_i can model the number of CPUs that a VS f requires and the number of CPUs available at MEC node i, where both $s_{f,i}$ and ω_i are integers. For example, bare-metal partitioning hypervisors, such as Jailhouse, allocate a number of virtual CPUs to services [Sin15].
- For VSs that share access to computational resources (e.g., virtual machines (VMs) and CPUs), it is reasonable to consider computational time as the granularity of resource allocation, based on the workload of the services. Let us consider a set of VSs $\mathcal{F}$ that share a MEC node with a single processor, and

each VS f has a periodicity of T_f. Let us denote by $t_{f,i}$ the computational time required by f in each period if f is placed on MEC node i. We can then model the resource utilization of f on i as $s_{f,i} = \frac{t_{f,i}}{T_f}$. In this case ω_i can model the upper bound of the sum utilization of the VSs, such that there exists a schedule that allows each VS to access the required computation time per period. A dynamic bound is proposed in [LL73] as a sufficient condition,

$$\omega_i = U(|\mathcal{F}|) = |\mathcal{F}| \cdot (2^{\frac{1}{|\mathcal{F}|}} - 1). \tag{3.2}$$

The models above mainly focus on systems that have a single type of computational resource. In the case of allocating multiple types of computational resources, for example, CPUs, GPUs and memory [JWG16], a proper model can be chosen for each type of resource independently.

Communication Resources

Communication resources are needed for data transmission between a VS instance and its end users and between services in a service chain. The communication resource requirement of a VS can be characterized in terms of latency and data rate. The latency consists of the latency between an end user and the BS in the mobile network, and the latency in the mobile backhaul. For a MEC node to host a VS f, the latency requirement of the VS must be satisfied, which depends on the use case and the type of service [Gan+b]. As an example, VSs for factory automation (e.g., machine tools, packing machines) have a relatively stringent latency requirement between 0.25ms to 10ms, while VSs for process automation can tolerate a latency up to 100ms [Sch+17; PM17; Abb+17].

VSs for data intensive applications also pose requirements on data rate. Notable applications in this category include augmented reality (AR), video analytic services, and content distribution. The data rate of an individual user depends on the network condition and the allocation of network resources [Vu+19; ZF19].

Redundancy Requirement

As failures occur in MEC, to guarantee the service availability of the VSs, a VS may require more than one instance to be hosted in MEC, which we refer to as the redundancy requirement [Bir13]. The redundancy requirement of a VS can be organized in three categories based on the failover time and the resource consumption [BA12].

Cold redundancy considers VSs with a lose requirement on the failover time. On the occurrence of MEC node failures, an affected VS can wait for the MEC under outage to be repaired, or wait for new instances to be started on a different MEC node.

Warm redundancy considers VSs that can tolerate a short failover period. A VS requiring warm redundancy can have inactive standby instances initialized in

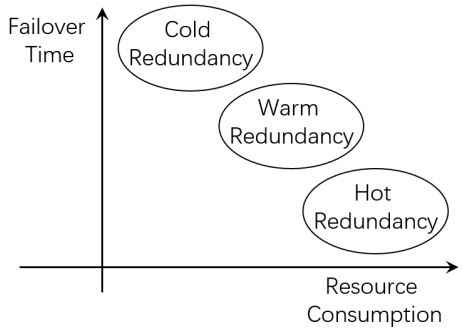

Figure 3.2: Comparison between redundancy schemes

different failure domains, and then the standby instances can be activated upon a failure. For stateful VSs, the state of the VS can be synchronized to the standby instance in real-time or periodically, such that the standby instance can resume from the latest state when activated. When a standby instance for warm redundancy is inactive, the main need for computational resources is due to state synchronization. Therefore, an inactive standby instance tends to consume less computational resources than when it is active. Therefore, warm redundancy allows the possibility to share computational resources for failover of multiple VSs.

Hot redundancy considers VSs with real-time availability requirements. Hot redundancy can be realized by simultaneously running active standby instances of a VS in different failure domains. The state of stateful VSs is synchronized to the standby instances continuously, and thus upon the occurrence of a failure, the standby instances can take over the services immediately. The standby instances for hot redundancy can provide a seamless response to failures, while each of them requires more resources than the standby instances for warm redundancy.

Figure 3.2 compares the redundancy requirements above in terms of resource consumption and failover time.

Chapter 4

Resilient Resource Allocation in Mobile Edge Computing

Works related to the resilient resource allocation in Mobile Edge Computing (MEC) can be categorized in four topical areas.

1. ***Resource dimensioning*** is a prerequisite for hosting virtualized services (VSs) in MEC. Resource dimensioning determines the amount and locations of MEC resources to be deployed, under the constraint of expected workloads and deployment options.

2. ***Service placement*** addresses the problem of allocating MEC resources to VSs subject to their placement requirements. Typical service placement problems focus on optimizing the operational costs, energy consumption, QoS (e.g., service latency), system capacity and resource efficiency. In practice, service placement is constrained by the available MEC resources, and thus imposes requirements on resource dimensioning.

3. ***Fault detection*** involves monitoring the behavior of the MEC system, and identifying potentially faulty components or services. Fault detection schemes in MEC can rely on observing pre-defined system metrics and may infer the state of the system, combined with state estimation techniques and machine learning [Wan+15; SD19; Afz+20]. Fault detection schemes depend on the architecture of MEC systems, on the service placement and on the implementation of the services.

4. ***Incident response*** addresses the problem of failover to guarantee continuous operation of the system, and is usually triggered by fault detection. In works that focus on the service availability of VSs, incident response may pose requirements on resource dimensioning and on service placement.

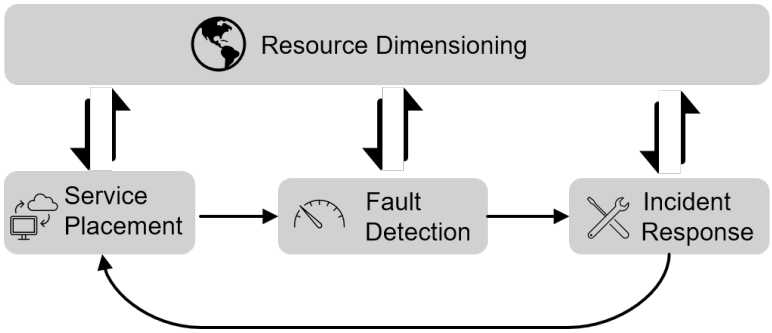

Figure 4.1: Dimensioning, placement, detection and response for resilient resource allocation in mobile edge computing.

Figure 4.1 shows the relationship between the four areas. This thesis considers the areas of resource dimensioning, service placement and incident response, and in the rest of the chapter we discuss the related works in those areas.

4.1 Resource Dimensioning

Resource dimensioning is an essential aspect of cloud computing and MEC, and has thus attracted significant attention. The majority of works in the area consider the dimensioning of computational and communication resources at a set of predefined sites to satisfy the workload of the system without explicitly considering resilience [LW18; Zen+19; Wan+14; Ta+08; Kas+20; CPS17]. Resilience as a system property is addressed in a few works [Yin+16; Dev+13; Cui+20], either as part of the objective function or as a constraint.

Non-Resilient Resource Dimensioning

We first discuss the works on resource dimensioning that do not address the requirement of resilience. Works in this area consider to fulfill the resource demand in the system, which are often modelled as the number of MEC nodes to be deployed or the workload of user requests.

Number of MEC nodes as resource demand: The authors in [Wan+19b] and [TKH19] model the resource demand as the number of MEC nodes to be deployed, as an input to the problem formulation. The authors in [Wan+19b] formulate a resource dimensioning problem to compute the locations of MEC nodes for balancing the workload of and the communication latency of end users. The locations of MEC nodes are constrained to the locations of BSs. The resulting problem is NP-hard, and is solved by an optimization solver. The authors in [TKH19] address the problem of placing a fixed number of MEC nodes in mobile networks

with a set of BS, each of which is further associated with a MEC node to upload computation tasks. The paper formulates four resource dimensioning problems with various objectives to trade off among different performance metrics, including maximal load of a single network link, maximal workload of a single MEC node, and the maximal network traffic.

Workload as resource demand: A line of works investigate to model the resource demand based on the workload of the system. Workloads can provide information on locations and QoS requirements of the resource demands, and thus allow to improve the efficiency of resource dimensioning.

A rather high level approach for resource dimensioning is to consider that each BS aggregates the workload from its users, and the workload is not tied to any specific type of computational resource; this approach is followed in [LW18]. The authors in this paper formulate the problem of resource dimensioning for placing MEC nodes with the objective of minimizing the energy consumption of the system. The energy consumption of the system originates from two sources; one is the base energy consumption that is incurred even when a server is idle and the other is the energy consumption due to hosting user applications. As the base energy consumption can be significant for certain MEC nodes, the paper suggests to maximize the resource utilization of MEC nodes as doing so would minimize the energy consumption. Finally, the paper proposes an algorithm based on the approach of particle swarm optimization to compute a solution to the problem.

The authors in [Zen+19] and [Wan+14] provide fine-grained models of the system workload based on the traffic of base stations and user requests, respectively. The authors in [Zen+19] assume that the computational workload from each BS is proportional to the amount of its network traffic, which can be obtained through historical data. They investigate the problem of minimizing the number of MEC nodes to be deployed, while enforcing constraints on the communication latency and on the number of BSs that a MEC node can serve. The resulting problem is a variant of the minimum dominating set problem on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which is to find a subset of vertices $\mathcal{D} \in \mathcal{V}$ that is adjacent to all other vertices in $\mathcal{V} \setminus \mathcal{D}$. Finally, a greedy algorithm is developed for solving the problem. Authors in [Wan+14] model each BS with the arrival rate of the workload, and characterize each workload as the amount of requested resources in terms of CPU cycles, memory size and hard disk space. The problem is solved by a three-stage algorithm. In the first stage the algorithm selects the locations of MEC nodes for minimizing the number of MEC nodes to be deployed, then the algorithm computes the amount of resources to be placed on each MEC node, by either equalizing the resources on MEC nodes or by adjusting the amount of resources based on the actual workload. Finally, the algorithm determines the link capacity of the network.

The authors in [Ta+08; Kas+20] explore resource dimensioning problems subject to network latency requirements. The authors in [Ta+08] address the problem of server placement for serving a set of users. Each user is assumed to have a delay requirement, and the objective of server placement is to maximize the number of users whose delay requirements are satisfied. The authors propose a strategy

that places servers at locations with high user density, and develop two heuristic solutions. Authors in [Kas+20] consider that each BS aggregates the computation workload of its serving area, and each BS is assigned to a MEC node to obtain computation services. The paper formulates a resource dimensioning problem for minimizing the latency between MEC nodes and BSs (e.g., latency in backhaul networks), and then proposes three heuristic algorithms based on a genetic algorithm and using local search heuristics.

Resource dimensioning for users with mobility is considered in [CPS17]. The authors in this paper jointly dimension computational and network resources for minimizing the deployment cost, subject to MEC node capacity and the aggregated workload at each BS. Furthermore, for each pair of MEC nodes i and j, the authors assume that the mobility of the end users leads to a certain amount of workload shifting from the serving area of i to that of j, and that the amount of the shifted workload is known. Consequently, a number of service instances (i.e., VMs) need to be migrated from MEC node i to MEC node j. To satisfy the resource demands of service migration, the authors compute an upper bound on the maximum number of VMs that can be migrated from MEC node i to MEC node j based on the available network resources (e.g., bandwidth), and then use this upper bound as a constraint of resource dimensioning.

Resilient Resource Dimensioning

Motivated by the observation that resilience requirements are often addressed at the cost of extra MEC resources, the problem of resilient resource dimensioning is considered in [Cui+20; Yin+16; Dev+13].

Authors in [Cui+20; Yin+16] consider resilient resource dimensioning with respect to computational resources. The authors in [Cui+20] investigate the problem of deploying a fixed number of MEC nodes to serve users in a local area. They propose to use the number of MEC nodes that each user can access as a metric of resilience, and then formulate a resource dimensioning problem as a multi-objective Integer Programming (IP) problem. The resulting problem is NP-hard, and an approximation algorithm is proposed. However, in the proposed solution resilience is only considered as part of the objective, and thus the resilience requirement of each user may not be guaranteed. Also, in the proposed resilience model the amount of resources needed for failover is not considered. A slightly different approach to resilience is taken in [Yin+16], where the authors explore MEC resource dimensioning for handling computational workloads with the objective of maximizing the number of users that can be served. The authors consider achieving resilience by reserving extra computational resources on each MEC node, such that under the failure of a single MEC node the affected workload can be served by other MEC nodes. In the proposed solution, the constraints of physical locations are relaxed by allowing an arbitrary placement of computational resources. Users are grouped into several clusters such that the maximal distance between the users in each cluster is bounded by a threshold. Furthermore, the paper considers to place one MEC node

for each cluster, minimizing the distance between each MEC node and its users. Finally, the authors map the obtained MEC node locations to real world locations, and compute the amount of computational resources to be allocated to each MEC node. Results show that the proposed approach can improve the QoS of end users by up to 45% compared to baseline approaches.

The joint problem of dimensioning computational and communication resources is addressed in [Dev+13] for resilience, in the context of cloud computing. The paper proposes a two-step algorithm that first computes the location of servers, and then computes the amount of computational resources to be allocated to each server together with the allocation of network resources. Resilience is achieved by placing standby instances and by computing two independent network paths for each instance.

4.2 Service Placement in MEC

Works on service placement for MEC focus on placing VSs on available MEC nodes. Each VS is a service instance that processes the workload from end users. In what follows we present works categorized based on three criteria.

1. ***Design objective:*** Works on service placement usually consider optimizing the operation cost, the energy consumption, the service capacity, the end-to-end latency, or resource efficiency.

2. ***Placement requirements:*** Depending on the types of services and the use cases, works in this area differ in terms of the placement requirements of the VSs. The majority of the works consider service placement subject to available computational resources, while a line of works also takes into account the available communication resources for data intensive services. Furthermore, satisfying the end-to-end latency of services has also attracted much attention. A number of works consider the placement of service chains, which require to model the communications between VSs and potential bottlenecks in processing, together with other constraints. In case that services are shared by multiple users, the joint problem of service placement and request assignment arises. Finally, in the case of VSs for critical services, resilience requirements have to be included as constraints so as to guarantee the continuity of the services.

3. ***Solution approaches:*** Service placement problems are often found to be related to classical NP-hard problems (e.g., facility location problem, generalized assignment problem and set cover problem), and are thus proven to be NP-hard [Gar+18; Pas+19; BFK20; HNR18]. Commonly used solution approaches include approximation algorithms with a performance bound, heuristic solutions of low computation complexity, and heuristic solutions that can scale to large systems. Furthermore, recent works also apply game theoretical and machine learning approaches to service placement.

A summary of works is shown in Table 4.1.

Service Placement Subject to Computational Resources

Many of the related works mainly focus on satisfying the demand of computational resources when placing VSs [Yu+18; Kir+20; Pas+19; CZL18; Gar+18].

In a general model the capacity of MEC nodes and resource demands of services can be modeled as real numbers, representing the size of tasks or the number of CPU cycles. This approach is used in [Pas+19; Kir+20]. The authors in [Pas+19] formulate a service placement problem with the objective of maximizing the reward of serving the users. By relating to the set cover problem, the authors show that the proposed service placement problem is NP-hard, and then propose an approximation algorithm. Service placement for minimizing the total cost of computational and communication resources is investigated in [Kir+20], and the authors propose a heuristic based on a genetic algorithm.

As MEC is closely coupled with mobile networks, it is attractive to investigate whether jointly considering service placement and RAN design can facilitate the integration of mobile networks and MEC systems. This problem is studied in [Gar+18], with the objective of computing a Pareto optimal solution in terms of network costs and service delay. The considered problem is NP-hard, and the authors propose an algorithm based on the Benders decomposition for computing approximate solutions. The algorithm decomposes the joint problem into a master problem for computing the service placement, and a slave problem for computing the RAN configuration. The solutions of the master problem and the slave problem give lower and upper bounds of the solution, and the algorithm stops when the upper and lower bounds coincide. The results show that the joint approach can reduce the total cost significantly, compared to a baseline.

As an alternative approach to centralized schemes, decentralized service placement schemes are considered in [CZL18; JD18; JD20], where the end users are allowed to express their preference over the set of available MEC nodes based on their own interests. The authors in [CZL18] propose an approach in which the placement decision is made jointly by end users and MEC service providers. They assume that each user has access to both MEC nodes and to a local server (e.g., owned by users). In the first stage of service placement, an algorithm is proposed for each user to decide whether to compute tasks on the local server or on MEC nodes. If it is the latter case, the algorithm also computes a list of candidate MEC nodes for each user service, referred to as user preference. Then a service admission algorithm is executed by the MEC service provider to decide the order of the services to be admitted, based on the workload and the computation costs. An alternative approach is followed in [JD18], where authors consider a MEC system where end users share the communication and computational resources. In this system each user makes a decision about on which MEC to offload its task for minimizing the weighted sum of its energy consumption and response time. The authors provide a game theoretical treatment of the problem, and propose a decentralized task of-

Table 4.1: Classifications of works on service placement in MEC

Paper	Design objective							Placement Constraints								Solution				
	Total cost	Energy	Service capacity	Resource utilization	Latency	Load balancing	Others	Comput. resources	Comm. resources	Resilience	Latency tolerance	Mobility	Traffic routing	Service chain	Request assignment	Approximation	Approximate	Heuristic	Game theory	Machine learning
[Pas+19]			✓					✓								✓				
[Kir+20]				✓				✓					✓					✓		
[Gar+18]	✓				✓			✓			✓		✓				✓			
[CZL18]	✓				✓			✓											✓	
[JD18]		✓			✓			✓	✓										✓	
[JD20]		✓			✓			✓	✓										✓	
[Yu+18]							✓	✓	✓									✓		
[JD19]					✓			✓	✓										✓	
[Add+15]	✓			✓				✓			✓		✓					✓		
[Mai+19]					✓			✓			✓							✓		
[ZL18]					✓			✓										✓		
[HNR18]			✓					✓			✓					✓				
[BFK20]						✓		✓			✓							✓		
[Gao+19]	✓				✓			✓				✓				✓				
[Bad+19]	✓				✓			✓				✓						✓		
[Ouy+19]					✓			✓	✓			✓								✓
[Zha+19a]	✓							✓	✓		✓	✓				✓				
[BG17]	✓													✓				✓		
[WZL17]						✓								✓				✓		
[Wan+19a]				✓				✓	✓					✓				✓		
[Kho+19]	✓							✓						✓				✓		
[Jan+17]		✓	✓					✓	✓				✓	✓		✓				
[Shi+20]		✓			✓			✓							✓				✓	✓
[He+18]			✓					✓							✓			✓		
[Far+19]			✓					✓	✓						✓	✓				
[Pou+19]			✓					✓	✓						✓	✓				
[YFK18]							✓	✓		✓								✓		
[ZH17]				✓				✓	✓	✓								✓		
[Mou+20]				✓	✓			✓		✓	✓							✓		
[CF20]	✓			✓				✓	✓	✓	✓							✓		
[Hma+16]				✓				✓	✓	✓	✓		✓	✓			✓			
Paper A [ZD18]	✓							✓		✓							✓			

floading algorithm with bounded approximation ratio. The work [JD20] considers a similar MEC system with users that generate periodic tasks. The authors propose a decentralized algorithm that gradually involves new users in making decisions for task offloading, allocating tasks among resources and over time slots, and show that the algorithm computes a Nash equilibrium.

Service Placement Subject to Communication Resources

Thanks to the development of new wireless technologies (e.g., millimeter wave communications and multiple antenna techniques), mobile networks can provide relatively high data rates to end users, and thus the data rate may not be a major concern for the placement of services with light traffic [Sha+17; FFT21]. However, for services with high data rates, the allocation of communication resources needs to be considered with respect to the wireless networks and the wired networks (e.g., mobile backhaul).

Resources in wireless networks: In wireless networks (e.g., mobile networks), the communication resources can be allocated in the form of time windows of transmission, or in the form of radio spectrum, depending on the underlying transmission techniques. Considering the allocation of wireless communication resources allows to adapt the achievable data rate of the individual end users to their demands. This approach is used in [Yu+18; JD19]. The authors in this paper consider to place a set of services in MEC to satisfy the need of mobile users subject to the computation and communication capacities of MEC nodes, with the objective of reducing the backhaul traffic. Specifically, the communication capacity of a MEC node is considered as the number of available resource blocks (RBs), which are shared among the users to satisfy the individual data rate requirements. In addition, the authors consider that the computational resource needs of hosting a specific service have two sources: the first part is the energy consumption of basic activities of the service and the second part is proportional to the number of users served. The paper first proposes a greedy algorithm for the case of a single MEC node; in this algorithm a MEC node chooses the services to host based on their efficiency of resource utilization (i.e., the communication resources needed for serving a unit of computation workload). Furthermore, the paper proposes a decentralized scheme based on matching and based on the mutual preferences between the MEC nodes and services. The results show that the joint approach of service placement and radio resource assignment can reduce the backhaul traffic and improve the efficiency of computational resources significantly.

The allocation of wireless communication resources is also considered in [JD19] for task offloading in MEC. In this work, the MEC operator allocates uplink data rate and computational resources among the users, while each user decides whether to process computational tasks locally or on one of the available MEC nodes, with the objective of minimizing its own completion time. The authors model the interaction between the BSs and end users as a Stackelberg game, and propose de-

centralized approximation algorithms with respect to different resource allocation policies of the operator.

Resources in wired networks: In wired networks (e.g., mobile backhaul) the data rate depends on both the forwarding path of the user data and link capacity allocated to each traffic flow. A joint consideration of service placement and communication resource allocation facilitate satisfying the demand of end users. This approach is used in [Add+15] for the placement of virtual network functions (VNFs). The authors in this paper formulate a multi-objective problem for minimizing the deployment cost and balancing the load of network links, subject to latency requirement of individual services and the available computational resources. The authors formulate the resulting problem as a mixed integer and linear programming (MILP) problem, and proposes an efficient heuristic algorithm.

Latency Sensitive Services

In MEC the end-to-end latency of each service varies with the locations of the end users and the locations of the hosting MEC node of each service. Therefore, the impact of latency needs to be considered for latency sensitive services. This challenge is addressed in [BFK20; HNR18; Mai+19; ZL18].

Latency minimization as objective: One approach of handling latency sensitive services is to prioritize the latency performance among all the performance metrics, which can be achieved by considering latency minimization as the main focus of service placement [ZL18; Mai+19]. Specifically, the end-to-end latency of a service can be modeled as the sum of communication latency and the processing latency on MEC nodes. This model is used by authors in [Mai+19], and they consider to place a number of service instances to serve a set of users, subject to the capacity of MEC nodes. In this paper each service instance is modeled as an $M/M/1$ queuing system, and the processing delay is computed as the average service time of the queue. The paper considers a soft latency requirement for each service, allowing to violate the latency requirement at the cost of penalty, and formulates the placement problem for minimizing the sum penalty. The resulting problem is an integer non-linear programming problem, and the authors propose a solution based on a genetic algorithm. Furthermore, the authors in [ZL18] model a MEC system with access to central clouds. The authors assume that service requests that are too resource intensive to be hosted in MEC can be further placed on a central cloud, and thus the end-to-end latency also includes the communication latency to the central cloud. The authors formulate a placement problem for minimizing the average latency of the service requests, and prove that the problem is NP-hard by relating it to the partition problem. To compute an optimal solution, the authors propose a brute-force algorithm that enumerates all the possible placement, and they describe a greedy heuristic for better scalability.

Latency as constraints: Considering latency minimization as the main objective reduces the sum latency of the system, but this approach may not guarantee the latency performance of individual services. Therefore, an alternative approach

is to use the latency requirement of individual services as constraints of service placement. This approach is followed in [HNR18], with the objective of maximizing the number of hosted services. The authors first consider the case that the MEC nodes have no capacity limits and propose an algorithm for computing an optimal solution. Then the authors prove that for the case of capacitated MEC nodes, the problem is NP-hard, and an approximation algorithm is proposed. Service placement under constraints of individual latency tolerance is also addressed in [BFK20]. The authors in this paper consider service placement for balancing the workload among MEC nodes, subject to MEC node capacity constraints. The problem is proven to be NP-hard, and the authors propose a solution based on tabu search, which starts with an initial feasible solution, and improves the solution gradually by swapping the placement of service instances. The algorithm terminates when a maximal number of iterations has been reached.

Services for End Users with Mobility

Due to the mobility of MEC users, adapting service placement based on the real-time locations of users may allow to improve user experience [Urg+15].

A common treatment to user mobility is to divide time into consecutive time slots, and to assume that within each slot the associations between users and BSs can be considered static (i.e., in each time slot a user only moves within the coverage of the same BS). Therefore, a service placement can be computed for each time slot, and the services can be migrated between different time slots to follow the users. This approach is used in [Gao+19; Zha+19a; Bad+19].

Authors in [Gao+19] address the problem of service placement with user mobility for minimizing the average response time of the system. The resulting problem is formulated as an integer non-linear programming problem, and the authors prove that the problem NP-hard. As a first step, the paper proposes an online algorithm for deciding whether a service migration is necessary or not, based on the current system latency (e.g., communication latency and computation latency) and the latency caused by service migration. If a migration is needed, a sub-problem is formulated to compute a new placement. This sub-problem is also NP-hard, and is solved by optimization solvers. However, the paper does not consider the individual latency requirement of services, and thus its applicability for delay sensitive services may be limited. The paper [Zha+19a] considers hosting virtual reality (VR) games for groups of users with mobility, with the objective of minimizing the communication and computation cost. The paper models the capacity and communication requirement of each service, and also takes into consideration the communication latency among the users within each group. Each service is considered to have an individual latency tolerance. The authors consider a general model of user mobility in a time-slotted system, and propose to use MPC to predict the number and locations of users for each time slot, and then place the services for each time slot based on the predictions, which is proven to be NP-hard. To solve the placement problem, an efficient algorithm is proposed based on solving a series of α-expansion

problems. Under the assumption of perfect prediction, the algorithm admits an approximation bound for the placement of each single time slot.

An alternative approach to capture the impact of user mobility is to model the cost of service migration for following the movement of the users. The cost of migration is proportional to the amount of network traffic and the computational resources. This approach is used in [Bad+19] for a time-slotted system, where the authors make the general assumption that the migration cost is proportional to the workload of users, and inversely proportional to the distance between users and MEC nodes. Then the authors formulate the joint problem for maximizing the sum QoS experience of end users while minimizing the service migration cost. Finally, a solution is proposed based on sample average approximation.

Instead of considering a time-slotted system with a fixed time span, it is of practical interest to consider systems with an infinite time span. An online approach for addressing this problem is proposed in [Ouy+19]. The authors in [Ouy+19] investigate service placement in MEC for satisfying the service demand of end users, with the objective of minimizing the weighted sum of the computation latency, the communication latency and the migration latency. The authors first formulate a multi-armed bandit problem, and then propose to use Thompson-sampling to estimate the expected performance of the users for different placement decisions. In this model, the communication latency is considered as a parameter to the system, but the allocation of bandwidth is not considered.

Service Chains

In MEC a service request from an end user may need to be processed by several services, which gives rise to the placement problem of service chains. The dependency among the services adds a new dimension to handling the placement requirements of service chains. For example, in service chains the end-to-end latency of a user request needs to include the communication latency between the constituent services of a service chain and the traffic flow among them.

To deal with the complexity of service chain placement, placement constraints can be relaxed to ease the development of solutions. Service chain placement without considering MEC node capacity is addressed in [BG17]. The authors in [BG17] consider the placement of service chains in MEC and propose a heuristic based on local search and the Hungarian algorithm for minimizing the total cost of computational and communication resources. Furthermore, the placement of service chains is also addressed by [WZL17] with the objective of balancing the workloads of the MEC nodes. The authors model each service chain as a graph, and each vertex and edge on this graph correspond to a service and a communication path, respectively. The resulting placement problem is NP-hard, and the paper proposes two solutions. The first solution focuses on the optimal placement of a single service chain, then the second solution takes an online approach to place multiple service chains, assuming each of them is a tree on a graph.

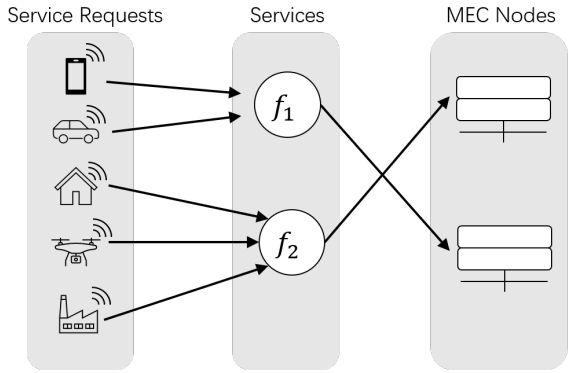

Figure 4.2: Illustration of joint service placement and request assignment

Service chain placement can also be addressed under the constraint of MEC node capacity, including available CPU cycles, memory size and hard disk spaces. This model of MEC node capacity is considered in [Kho+19; Wan+19a]. The authors in [Wan+19a] explore service chain placement for maximizing the efficiency of resource usage. The authors propose a polynomial solution based on graph matching and the Hungarian algorithm. Furthermore, the authors in [Kho+19] investigate the service chain placement problem with the objective of minimizing the deployment cost. The authors propose a heuristic solution based on a genetic algorithm. The authors propose two strategies to apply the proposed solution. One is to compute the placement of the service chains sequentially, and the other is to compute the placement of the service chains jointly. Simulation results show that the latter strategy can reduce total cost while it requires more execution time.

For service chains that handle traffic intensive requests, the forwarding paths of network traffic need to be computed under constraints on the link capacity, traffic load, and the dependency among services (i.e., each traffic flow needs to go through a set of services in a specific order). This problem is investigated in [Jan+17], where the authors consider the joint problem of service chain placement and flow distribution with the objective of jointly optimizing the acceptable flow rate and the energy consumption. The resulting problem is formulated as a MILP problem, and is proven to be NP-Hard. The authors propose an approximation algorithm based on linear relaxation and rounding techniques. The proposed solution first computes the placement of services, and then computes the optimal flow distribution. Numerical results show that a joint allocation of computational and communication resources is essential for optimizing the efficiency and service capacity of the system.

Request Assignment for Services with Multiple Users

In MEC a VS can serve more than one user, for example, in the case of content distribution and database services. To illustrate this scenario, Figure 4.2 shows a system that consists of two services and serves service requests generated by five users. As the figure suggests, two layers of assignment are needed for placing services shared by multiple users. The first layer is the assignment between service instances and MEC nodes and the other is the assignment between service requests and service instances. In general the service requests can have an associated workload and latency requirements, while each service instance has limited service capacity. Therefore, the service request assignment affects the number and locations of service instances to be placed, which makes the joint problem of service placement and request assignment non-trivial [He+18; Far+19; Shi+20; Pou+19].

The number of service requests that a service can handle is constrained by the capacity of each service and the capacity of each MEC node [Shi+20; He+18]. The authors in [Shi+20] consider to place a set of services to handle the service requests from mobile users, with the objective of minimizing the energy consumption and service latency. They propose a solution based on game theory and reinforcement learning, and their results show that the latency requirement of the applications, the capacity of the MEC nodes, and the workload of user requests are crucial to the energy consumption of the system. Furthermore, the authors in [He+18] consider the joint problem of service placement and user request assignment for maximizing the service capacity of the system. By relating to the three partition problem and the maximal cover problem, the authors prove that this joint problem is NP-hard. The authors first propose a greedy algorithm to compute the placement of the services and then they assign the user requests to services by constructing a maximal flow problem on a graph $\mathcal{G}$, where the edges of $\mathcal{G}$ represent users and MEC nodes. The solution is effective, however, only it supports the case that MEC capacities and user requests are homogeneous.

The request assignment and service placement problem can also be coupled with other use cases and placement constraints, for example, the mobility of end users and the communication capacity of MEC nodes, as considered in [Far+19] and [Pou+19], respectively. The authors in [Far+19] focus on a time-slotted MEC system, where a set of service requests arrive at the beginning of each time slot. The paper formulates a placement algorithm for maximizing the number of requests admitted under constraints of computation and communication resources. The authors first formulate a subproblem that places the services for a single time slot, and then prove that the objective function of this subproblem is a monotone submodular function. The authors then propose a polynomial time greedy approach with a bounded approximation ratio. For the case of multiple time slots, a joint problem that considers the correlation across frames is formulated as a mixed integer non-linear programming problem.

The authors in [Pou+19] focus on a system with MEC nodes co-located with BSs, and consider to place services under the constraints on the upload and down-

load capacity of each MEC node, with the objective of maximizing the service capacity of the system (e.g., number of accepted user requests). The formulated joint problem is a generalization of the classical knapsack problem and is thus NP-hard. To solve the problem, the authors propose an approximation algorithm based on linear relaxation and randomized rounding, which utilizes fractional solutions as the probability of placing a service on a MEC node, and of using a network link.

Resilient Service Placement

For services that have requirements in terms of availability, service placement needs to be resilient to potential failures in MEC systems [YFK18; ZH17; Mou+20; Hma+16; CF20]. Common approaches for addressing the challenge of resilience include considering resilience as objective and enforcing resilience constraints.

Resilience as objective: One approach to address resilience is to quantify resilience as a performance metric, and to use it as the objective of service placement. For example, the level of resilience can be reflected by the probability that a minimum of services is available. This resilience metric is used by [YFK18]. The authors in this paper consider to host VNFs in MEC for minimizing the total placement cost while maximizing the resilience metric of the system. The system consists of a centralized cloud computing platform and a MEC system with distributed MEC nodes. The paper proposes two kinds of failures; the first kind of failure concerns a single VNF and the other is the failure of a single MEC node or the centralized cloud. The paper divides the VNF instances into several service groups, and the services in each service group can be affected by a single failure. The authors propose a heuristic based on a genetic algorithm, which gradually improves the solution through crossover and mutation operations.

Resilience as constraint: The challenge of resilience can also be addressed as a constraint of the service placement problem, allowing the system to optimize other performance metrics (e.g., service capacity, energy consumption). A number of abstractions of resilience constraints have been considered by related works in this area.

At the level of system performance, the resilience constraint can be formed as the minimum number of services that needs to be made available. This approach is followed in [ZH17]. The authors in this paper consider to provide VMs as services in MEC for minimizing the cost of computational and communication resources. The authors consider both the failure of a single VM and the failure of a single MEC node, which would make a set of VMs unavailable. The authors further derive the failure probability for each VMs, and then enforce a constraint on the probability of making a minimum number of VMs to be available. Under this model, resilience can be achieved by placing redundant VMs at the cost of extra resource consumption. A heuristic algorithm is proposed to concatenate the placement of VMs and to place VMs on MEC nodes with high capacity. However, the method in this paper aims to provide resilience at a system level, and the resilience requirement of a single service is not considered.

Alternatively, resilience requirements can be specified for each individual service, allowing each user to choose its own level of resilience. Besides the probability of failure, the resilience requirement of a user can be specified as the number and types of instances (i.e., primary and standby instances) to be placed. The primary instances provide service in nominal scenarios, while the standby instances can take over upon the failure of the primary instances. This abstraction of resilience requirement is considered in [Mou+20; CF20]. The authors in [Mou+20] propose a resilience model that allows each service to choose the level of resilience. This work focuses on resilient placement of vehicle-to-everything (V2X) applications in MEC for minimizing the latency of the services, subject to delay requirement and the MEC node capacity. The individual level of resilience is specified as the number of hot standby instances of each service, and the hot standby instances of the same service are placed on different MEC nodes. The paper proposes a greedy solution where each MEC node iteratively chooses the services with the lowest latency to host. Due to the high resource consumption of hot standby instances, resilience is achieved at the cost of increasing the demand for computational resources. Authors in [CF20] propose a resilience scheme that supports shared redundancy. The service placement problem in [CF20] is formulated as a multi-objective optimization problem for optimizing the consumption of computational resources and for minimizing the response times of services. The paper considers two schemes of resilience. In the first scheme, each service is assigned a primary instance and a secondary instance (e.g., warm standby). In the second scheme, backup resources are reserved for multiple users, allowing to provide failover for a limited number of users simultaneously. The second scheme reduces the consumption of computational resources, and is suitable for services with a high tolerance of outage, or for systems where components have high reliability.

Failures in MEC systems can concern both computational and communication resources, while the papers above only address failures related to computational resources. Failover for communication resources can be achieved by planning alternative paths of traffic forwarding, as done in [Hma+16]. This work studies the problem of resilient placement of VNF service chains, which is highly relevant to service placement in MEC. The paper considers single link and single node failures, and proposes three schemes to improve resilience. The first scheme concerns failures of computational resources, and implements two service instances for each VNF at different locations subject to single server failure. The second scheme considers network link failures and then allocates two disjoint physical paths for each virtual link. The third scheme is to implement two instances of each service chain, with disjoint paths to protect the services against both single server and single link failures. The authors formulate the optimization problem as an IP with the objective of minimizing the number of VNF servers, under constraints in terms of latency requirement, resilience requirement, link capacity, and VNF node capacity. Due to the complexity of the formulated problem, the authors use optimization solvers to solve the problem. Numerical results show that both latency requirements and the capacity of VNF servers have impact on the number of VNF servers needed.

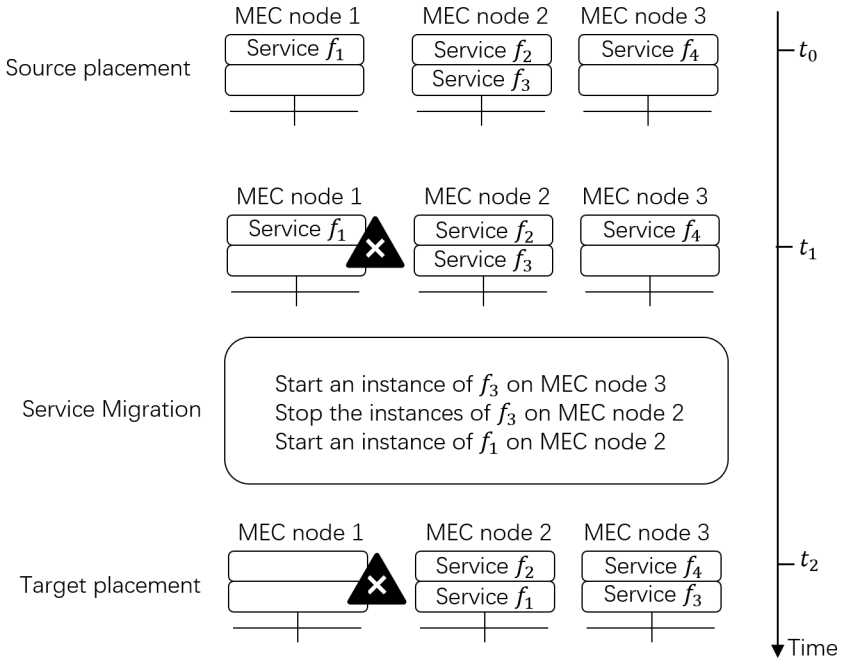

Figure 4.3: An illustration of failover schemes in Papers A, B, and C

Our Paper A also takes the approach of considering resilience as a constraint of service placement, while it allows to consider the failure of an arbitrary set of MEC nodes [ZD18]. We model a set of failure scenarios, each of which includes a set of MEC node failures. Each scenario is associated with a failure probability, which can be obtained through historical data from the system. We formulate an IP problem to compute the MEC node to be opened, and to compute the VS placement for each scenario. When a failure scenario occurs, the VS instances can be placed according to the computed placement to guarantee the availability of all services. We propose an efficient solution based on the Benders decomposition for solving the problem, with the objective of minimizing the placement cost of the system. When the services are stateful, switching between service placements would require service migration to preserve the state of the service instances. Papers B and C in the thesis address this subsequent problem with respect to VSs with homogeneous and heterogeneous resource requirements [ZDd]. We proposed efficient algorithms based on a graph theoretic model of the problem for scheduling the sequences of migration actions, subject to MEC node capacity constraints and migration deadline.

These three papers together constitute a framework for resilient service placement and failure recovery. Figure 4.3 illustrates this framework for a system with

three MEC nodes and four services. At time $t = t_0$ the services are placed according to the placement computed by paper A with all the MEC nodes in normal state, referred to as the source placement. Assume now that at $t = t_1$ MEC node 1 fails due to a faulty component, and thus the availability of service f_1 is affected. As a failover, the services need to be migrated to a new placement, which is also computed by the placement algorithm in paper A and is referred to as the target placement. Note that for the purpose of optimizing placement cost, the placement of service f_3 also needs to be updated. To migrate the services from the source placement to the target placement, a migration schedule can be computed by the migration scheduling algorithms in paper B or in paper C, depending on the model of computational resources.

4.3 Joint Resource Dimensioning and Service Placement

In Section 4.1 the related works on resource dimensioning consider the resource demand of the services as input. On the contrary, the related works in Section 4.2 typically consider MEC node configurations (e.g., locations and capacities) as constraints for service placement. Nonetheless, due to the interdependence between resource dimensioning and service placement, it is appealing to address these two problems jointly to optimize the service placement and resource utilization of the system.

Joint Resource Dimensioning and Service Placement without Resilience

Joint resource dimensioning and service placement without addressing resilience requirement has been addressed in [Zha+18a; Khe+19; Men+19].

Services shared by multiple users are considered in [Khe+19; Men+19]. The authors in [Khe+19] study this joint problem with the objective of minimizing the deployment cost of the system. The authors consider to place at most K MEC nodes in an area to host a set of applications, which further handle the computational tasks from mobile users. The authors prove that this joint problem is NP-hard, and then propose an efficient solution based on decomposing the problem into two sub-problems. The first sub-problem is aimed to assign the mobile users to the applications to be deployed, and then to determine the number of applications to be hosted, subject to the capacity of applications, the individual delay requirements and workloads. In the second sub-problem, the placement of the applications and the MEC nodes are computed. The amount of MEC resources to be deployed is constrained by a predefined upper bound, and thus the proposed algorithm may not satisfy the placement requirement of all users. The authors in [Men+19] consider to place a set of MEC nodes in a mobile network to host a set of services, for handling service requests from mobile users. A service can either be hosted on a MEC node, or can be hosted on a central cloud platform at the cost of increased

service latency due to communication. To solve this joint problem for minimizing the total cost of communication and computational resources, the authors propose a local search based approximation algorithm with two nested loops. The outer loop of the algorithm computes the placement of MEC nodes, and the inner loop computes the placement of the services. Finally, the algorithm constructs instances of the minimum cost maximum network flow problem to assign user requests to the service instances.

Closely related to the area of MEC is the area of cloudlets. The work in [Zha+18a] considers resource dimensioning of cloudlets to handle service requests from mobile users, with the objective of minimizing the service latency of end users. The resource demand of the system is abstracted as the number of servers needed, and is considered as an input to the problem. The paper first considers the case that the users are static, and proposes a two-step algorithm to solve the problem. In the first step the algorithm places cloudlets on BSs with the heaviest incoming service requests. Then each cloudlet is assigned to serve the users within its coverage areas, under constraints in terms of workload balancing. Then the paper proposes an algorithm for the scenario with user mobility. The algorithm relies on historical data to find a list of candidate locations of cloudlets, and then the algorithm finds a cloudlet placement that can satisfy the user demand for all scenarios by solving an instance of the K-median problem. Finally, the algorithm assigns users to the cloudlet as in the case of static users.

Joint Resource Dimensioning and Service Placement with Resilience

The joint problem of resource dimensioning and service placement with resilience is rarely addressed as it involves a number of correlated constraints. The authors in [Lu+20] address this problem by considering server placement and task assignment in MEC with the objective of maximizing the service capacity (i.e., number of accepted user requests), subject to communication and computation capacities of MEC nodes. In terms of resilience, this paper considers simultaneous failures of a maximum number of MEC nodes, and the impact of failures is captured by the loss of service capacity. The authors prove that the resulting problem is NP-hard. An approximation algorithm is then proposed based on finding the subset of MEC nodes with the highest service capacity in the worst case scenario.

The work [Lu+20] above does not allow to address the resilience requirement of individual services, and only considers to minimize the impact of failures passively without providing failover schemes. To close this gap, our paper D considers a general model for resource dimensioning that computes the number and types of MEC nodes to be placed at each location, and computes the placement of service instances subject to individual availability requirements of the services [ZDa]. We formulate the joint resource dimensioning and service placement problem for optimizing the energy consumption of hosting VSs. We address resilient resource allocation for stateful and stateless VSs by considering inactive standbys for reduc-

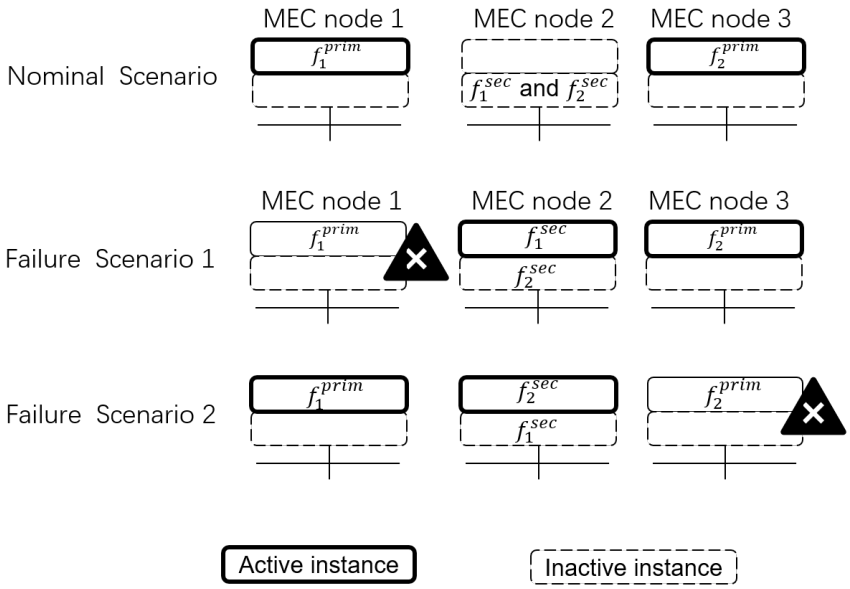

Figure 4.4: An illustration of failover schemes in Papers D and E

ing energy consumption and by considering active standbys for immediate incident response. Results in our paper show that the proposed solution outperforms a baseline that solves the resource dimensioning and service placement problem in a non-joint manner. A more refined model of resilience is considered in our paper E, where we group the MEC nodes into availability zones. Computational resources are shared among the VSs based on their workload, subject to schedulability constraints.

The failover scheme of papers D and E is illustrated in Figure 4.4, for a system with two services. The primary instances of these two services (i,e., f_1^{prim} and f_2^{prim}) are placed on MEC node 1 and MEC node 3, respectively. The secondary instances of the services (i,e., f_1^{sec} and f_2^{sec}) are placed on MEC node 2. In the nominal scenario, only the primary instances are active, and the secondary instances are inactive (e.g., the secondary instances only receive state updates from the primary instances, and can share computing resources with the operating system). On the occurrence of failure scenario 1 (failure scenario 2), the secondary instance of service 1 (service 2) can be activated on MEC node 2 to provide the service, without requiring service migration.

Chapter 5

Summary of Original Work

Paper A: A Benders Decomposition Approach for Resilient Placement of Virtual Process Control Functions in Mobile Edge Clouds

Peiyue Zhao and György Dán

IEEE Transactions on Network and Service Management, vol. 15, no. 4, pp. 1460-1472, 2018.

A shorter version of the paper appeared in Proc. of IFIP Networking 2017.

Summary: In this paper we consider placing virtualized services in mobile edge clouds with the objective of minimizing operational costs, subject to MEC nodes outages caused by cyber attackers, component failures and operational mistakes. We propose a failover scheme with shared redundancy in which common computational resources are reserved and shared by the services to restore additional instances in case of a failure. The resulting resource allocation problem is formulated as an integer programming problem, and we propose an efficient algorithm to compute the set of MEC nodes to be opened and the placement of the services in each failure scenario. The algorithm is based on generalized Benders decomposition that decouples the formulated large scale integer programming problem into two parts, and based on linear relaxation to reduce the number of integer variables. We performed realistic simulations to evaluate the proposed algorithm, and the results show that the proposed solution outperforms a greedy approach and a local search baseline in terms of operational costs and resource efficiency.

Contribution: The author of this thesis developed the system model in collaboration with the second author of the paper. The author of this thesis proved the analytical results concerning the convergence of the proposed algorithm. The author of this thesis carried out the simulations, and analyzed the resulting data in collaboration with the second author. The paper was written in collaboration with the second author.

Paper B: Service Migration under Time Constraints for Mobile Edge Computing

Peiyue Zhao and György Dán

under submission
A shorter version of the paper appeared in Proc. of International Teletraffic Congress 2018

Summary: In this paper we consider a mobile edge cloud in which services need to be migrated between different placements for incident response, and for optimizing the energy efficiency and resource usage of the system. We propose a three-stage migration model of the services, and the model prioritizes the services by their importance. We formulate the migration scheduling problem as a binary programming problem with the objective of minimizing the service interruptions due to migration, subject to time constraints and resource requirements of migration. We propose an efficient algorithm to compute the order and actions of the service migration. The proposed algorithm relies on analyzing the dependency among the services in terms of resource usage, and is built on several problems on graphs. Analytical results characterize the problem instances for which the proposed algorithm can compute an optimal solution, and show that the algorithm is computationally lightweight by providing the worst case complexity. We evaluate the performance of the proposed algorithm in extensive simulations. The numerical results show that the proposed algorithm outperforms a general heuristic for solving binary programming problems, and performs close to the optimal solution.

Contribution: The author of this thesis developed the system model and problem formulation in collaboration with the second author of the paper. The author of this thesis proved the analytical results for the proposed algorithm. The simulations were carried out by the author of this thesis, and analysis of the numerical results was carried out in collaboration with the second author of the paper. The paper was written in collaboration with the second author.

Paper C: Scheduling Parallel Migration of Virtualized Services under Time Constraints in Mobile Edge Clouds

Peiyue Zhao and György Dán

in Proc. of International Teletraffic Congress 2019.

Summary: In this paper we address the problem of scheduling the migration of virtualized services with heterogeneous resource requirements in mobile edge clouds, subject to time constraints of migration. We consider that the services have strict requirements on service continuity, and we formulate the migration scheduling problem as an integer programming problem for minimizing service interruptions during migration. We analyze the complexity of the problem and provide analytical results for reducing the number of variables. Furthermore, we formulate a graph representation of the problem, and then propose an algorithm based on constructing hypergraphs that correspond to migration schedules. The proposed algorithm is computational lightweight, and analytical results show that the algorithm terminates in a finite number of iterations. Extensive simulation results show that the proposed algorithms perform close to the optimal solution, and outperform a baseline algorithm motivated by our own previous works, implying that the solution proposed in this paper is efficient, effective and scalable for mobile edge clouds. Results for various scenarios further provide insights on the trade-offs between the migration time constraint and the service continuity.

Contribution: The analytical model of the paper was developed in collaboration with the second author of the paper. The analytical results for reducing the number of variables were carried out in collaboration with the second author, and the author of this thesis proved analytical results concerning the proposed algorithm. The implementation of the simulations was carried out by the author of this thesis, and the analysis of the simulation results was carried out in collaboration with the second author. The paper was written in collaboration with the second author.

Paper D: Joint Resource Dimensioning and Placement for Dependable Virtualized Services in Mobile Edge Clouds

Peiyue Zhao and György Dán

accepted for publication in IEEE Transactions on Mobile Computing.

Summary: In this paper we address the joint problem of resource dimensioning and placement of dependable virtualized services in mobile edge clouds. We consider virtualized services with hot standby redundancy and shared redundancy requirements, which are further satisfied by activating standby instances with mirrored state to replace the services affected by MEC node outages. We propose an abstraction to encapsulate the reliability, latency, and resource requirements of the services in service level agreements so as to the service provider can address these requirements while optimizing the resource usage. The resulting resource allocation problem is formulated as an integer programming problem for minimizing the energy consumption, and we prove that it is NP-hard. We propose an approximation algorithm based on Lagrangian relaxation that solves the problem in two steps. Analytical results show that the proposed algorithm has a bounded approximation ratio, and terminates in a finite number of iterations. Results from realistic simulations show that the proposed algorithm benefits the system significantly in terms of energy consumption and resource efficiency, compared to two greedy approaches and to a non-joint approach for the dimensioning and placement problems.

Contribution: The author of this thesis developed the analytical model in collaboration with the second author of the paper. The author of this thesis proved the NP-hardness of the problem and proved the analytical results concerning the proposed algorithm in collaboration with the second author of the paper. The simulations were implemented by the author of this thesis and the analysis of the resulting data is carried out in collaboration with the second author of the paper. The paper is written in collaboration of the second author of the paper.

Paper E: Energy-aware Placement of Virtualized Services in Mobile Edge Clouds under Availability and Real-time Schedulability Constraints

Peiyue Zhao and György Dán

under submission

Summary: This paper considers placing periodical virtualized services in mobile edge clouds with MEC resources partitioned into several availability zones, each of which is considered as a failure domain. We consider a shared resource allocation model that allows the virtualized services to utilize the computational resources efficiently based on their workloads, under schedulability constraints. We formulate the resource allocation problem as a non-linear integer programming problem for deciding the MEC nodes to be opened, and for placing the instances of the services to satisfy their redundancy and resource requirements, with the objective of minimizing energy consumption. We prove that the problem is NP-hard, and propose three efficient algorithms based on primal-dual approach and based on matching problems on bipartite graphs. Analytical results show that the proposed solutions all terminate in a finite number of iterations, and numerical results show that the proposed algorithms outperform approaches that do not dynamically address the schedulability constraints. The proposed algorithms further allow to explore trade-offs between energy performance and the success rate of computing feasible solutions for problem instances with different distributions of workloads and redundancy requirements.

Contribution: The author of this thesis developed the analytical model in collaboration with the second author of the paper. The analytical results were carried out by the author of this thesis. The author of this thesis implemented the simulations, and analyzed the resulting data in collaboration with the second author of the paper. The thesis is written in collaboration with the second author of the paper.

Publications not included in the thesis

1. Peiyue Zhao and György Dán. "Resilient Placement of Virtual Process Control Functions in Mobile Edge Clouds". In: ***Proc. of IFIP Networking 2017***, pp. 1–9

2. Peiyue Zhao et al. "A Game Theoretic Approach To Setting The Pilot Power Ratio in Multi-User MIMO Systems". In: ***IEEE Transactions on Communications*** 66.3 (2018), pp. 999–1012

3. Peiyue Zhao and György Dán. "Time Constrained Service-Aware Migration of Virtualized Services for Mobile Edge Computing". In: ***Proc. of International Teletraffic Congress 2018***, pp. 64–72

4. Peiyue Zhao et al. "A Game Theoretic Approach To Uplink Pilot and Data Power Control in Multi-Cell Multi-User MIMO Systems". In: ***IEEE Transactions on Vehicular Technology*** 68.9 (2019), pp. 8707–8720

Chapter 6

Conclusion and Future Work

In this thesis, we considered resilient resource allocation for virtualized services in mobile edge computing systems. We developed efficient resource management solutions combined with incident response schemes.

In the first part of this thesis, we focused on a mobile edge computing system in the presence of a set of potential failure scenarios, each of which consists of failures of a set of mobile edge computing nodes. We showed that resilient service placement is feasible in mobile edge computing through our framework of resilient resource management, which migrates services between different placements for incident response, under constraints in terms of failover time tolerance.

- As a first step, we proposed a resilient service placement algorithm that opens a set of mobile edge computing nodes, and then computes the placement of services for each failure scenario such that the services are always hosted by mobile edge computing nodes that are in nominal state. This algorithm is built based on the generalized Benders decomposition, with the objective of minimizing the operational cost of the system.

- Subsequently, to migrate services between different placements we proposed migration scheduling algorithms that compute the sequence of migration actions under migration time constraints, and the proposed algorithms allow to preserve the state of services. Nonetheless, the proposed migration algorithms are versatile, and can also serve use cases of service migration for optimizing resource utilization, and of service migration for adapting to changing workload.

In the second part of this thesis, we considered energy-aware resilient resource allocation for mobile edge computing systems with failover schemes that switch between primary and standby instances. We showed that resilient resource allocation is possible without the need of exposing the infrastructure of mobile edge computing systems. Specifically, we proposed an abstraction that embeds the availability

requirements of end users into service level agreements. This abstraction allows mobile edge cloud operators the freedom to optimize the system performance while addressing various reliability requirements of the hosted services.

- First, we applied the abstraction above to a mobile edge computing system concerning the outage of single mobile edge computing node, and addressed the problem of joint resource dimensioning and resilient service placement. The formulated problem is NP-hard, and we proposed an approximation algorithm based on Lagrangian relaxation to minimize the energy consumption of the system. Numerical results show that this joint approach is efficient and can reduce the energy consumption significantly, compared to a baseline that solves resource dimensioning and service placement problems separately.
- Furthermore, we considered the proposed abstraction in a system subject to the failures of a set of correlated mobile edge computing nodes, which constitutes an availability zone. In this system, services share computational resources based on their workloads, under the constraint of schedulability. The resulting resource management problem admits non-linear constraints, and is proven to be NP-hard. We proposed three efficient algorithms that allow to trade off the cost performance for the opportunities of finding feasible solutions, and these algorithms can jointly provide solutions to systems with various workload distribution and service constitutions.

This thesis explored the possibility of achieving resilience in resource allocation, and proposed efficient and effective solutions to the resulting resource management problems. Based on the results of this thesis, many interesting branches could be explored as future work.

- The first question is whether decentralized schemes with low communication overhead can be applied to resilient resource management in mobile edge computing such that multiple platforms can be federated to provide services to end users, without the coordination of a central entity.
- The second interesting question is whether the abstraction used in the second part of this thesis can be considered for addressing failures of multiple availability zones, which would further enhance the level of resilience of the system.
- The third interesting question is whether our works can be extended to scenarios where only partial knowledge of the service demand is known, potentially combined with machine learning approaches for estimating and forecasting the service demand.

Bibliography

[Abb+17] Nasir Abbas et al. "Mobile Edge Computing: A Survey". In: ***IEEE Internet of Things Journal*** 5.1 (2017), pp. 450–465.

[Add+15] Bernardetta Addis et al. "Virtual Network Functions Placement and Routing Optimization". In: ***Proc. of IEEE International Conference on Cloud Networking***. 2015, pp. 171–177.

[Afz+20] Zeeshan Afzal et al. "Using Features of Encrypted Network Traffic To Detect Malware". In: ***Proc. of Nordic Conference on Secure IT Systems November***. 2020, p. 37.

[Ant+20] Kiril Antevski et al. "On The Integration of NFV and MEC Technologies: Architecture Analysis and Benefits for Edge Robotics". In: ***Computer Networks*** 175 (2020), p. 107274.

[BA12] Eric Bauer and Randee Adams. ***Reliability and Availability of Cloud Computing***. John Wiley & Sons, 2012.

[Bad+19] Hossein Badri et al. "Energy-Aware Application Placement in Mobile Edge Computing: A Stochastic Optimization Approach". In: ***IEEE Transactions on Parallel and Distributed Systems*** 31.4 (2019), pp. 909–922.

[Bai18] Haishi Bai. ***Programming Microsoft Azure Service Fabric***. Microsoft Press, 2018.

[Ber+15] Angelo Bernasconi et al. ***IBM Spectrum Virtualize and IBM Spectrum Scale in An Enhanced Stretched Cluster Implementation***. IBM Redbooks, 2015.

[BFK20] Bouziane Brik, Pantelis A Frangoudis, and Adlen Ksentini. "Service-Oriented MEC Applications Placement in A Federated Edge Cloud Architecture". In: ***Proc. of IEEE International Conference on Communications***. 2020, pp. 1–6.

[BG17] Tayebeh Bahreini and Daniel Grosu. "Efficient Placement of Multi-Component Applications in Edge Computing Systems". In: ***Proc. of ACM/IEEE Symposium on Edge Computing***. 2017, pp. 1–11.

[Bir13] Alessandro Birolini. ***Reliability Engineering: Theory and Practice***. Springer Science & Business Media, 2013.

[CF20] Hernani D Chantre and Nelson Luis Saldanha da Fonseca. "The Location Problem for The Provisioning of Protected Slices in NFV-Based MEC Infrastructure". In: ***IEEE Journal on Selected Areas in Communications*** 38.7 (2020), pp. 1505–1514.

[CPS17] Alberto Ceselli, Marco Premoli, and Stefano Secci. "Mobile Edge Cloud Network Design Optimization". In: ***IEEE/ACM Transactions on Networking*** 25.3 (2017), pp. 1818–1831.

[Cui+20] Guangming Cui et al. "Trading off Between User Coverage and Network Robustness for Edge Server Placement". In: ***IEEE Transactions on Cloud Computing*** (2020).

[CZL18] Zhi Cao, Honggang Zhang, and Benyuan Liu. "Performance and Stability of Application Placement in Mobile Edge Computing System". In: ***Proc. of IEEE International Performance Computing and Communications Conference***. 2018, pp. 1–8.

[Dem+13] Panagiotis Demestichas et al. "5G on The Horizon: Key Challenges for The Radio-Access Network". In: ***IEEE Vehicular Technology Magazine*** 8.3 (2013), pp. 47–53.

[Dev+13] Chris Develder et al. "Joint Dimensioning of Server and Network Infrastructure for Resilient Optical Grids/Clouds". In: ***IEEE/ACM Transactions on Networking*** 22.5 (2013), pp. 1591–1606.

[Dob+19] Simon Dobson et al. "Self-Organization and Resilience for Networked Systems: Design Principles and Open Research Issues". In: ***Proceedings of the IEEE*** 107.4 (2019), pp. 819–834.

[Dut20] Dinesh G Dutt. ***Cloud Native Data Center Networking: Architecture, Protocols, and Tools***. O'Reilly Media, 2020.

[Eri20] Ericsson. ***Ericsson Mobility Report***. Nov. 2020. URL: `https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf`.

[Far+19] Vajiheh Farhadi et al. "Service Placement and Request Scheduling for Data-Intensive Applications in Edge Clouds". In: ***Proc. of IEEE INFOCOM***. 2019, pp. 1279–1287.

[FFT21] Gábor Fodor, Sebastian Fodor, and Miklós Telek. "Performance Analysis of A Linear MMSE Receiver in Time-Variant Rayleigh Fading Channels". In: ***IEEE Transactions on Communications*** (2021).

[Fod+21] Gabor Fodor et al. "5G New Radio for Automotive, Rail, and Air Transport". In: ***arXiv:2101.08874*** (2021).

[Gan+a] Milad Ganjalizadeh et al. "Impact of Correlated Failures in 5G Dual Connectivity Architectures for URLLC Applications". In: ***Proc. of IEEE Globecom Workshops***, pp. 1–6.

[Gan+b] Milad Ganjalizadeh et al. "Translating Cyber-Physical Control Application Requirements To Network Level Parameters". In: ***Proc. of IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications***, pp. 1–7.

[Gan+18] Sandip Gangakhedkar et al. "Use Cases, Requirements and Challenges of 5G Communication for Industrial Automation". In: ***Proc. of IEEE International Conference on Communications Workshops***. 2018, pp. 1–6.

[Gao+19] Bin Gao et al. "Winning At The Starting Line: Joint Network Selection and Service Placement for Mobile Edge Computing". In: ***Proc. of IEEE INFOCOM***. 2019, pp. 1459–1467.

[Gar+18] Andres Garcia-Saavedra et al. "Joint Optimization of Edge Computing Architectures and Radio Access Networks". In: ***IEEE Journal on Selected Areas in Communications*** 36.11 (2018), pp. 2433–2443.

[Gou+16] Antonios Gouglidis et al. "Threat Awareness for Critical Infrastructures Resilience". In: ***International Workshop on Resilient Networks Design and Modeling***. 2016, pp. 196–202.

[Gu+] Sijia Gu et al. "An Efficient Auction Mechanism for Service Chains in The NFV Market". In: ***Proc. of IEEE INFOCOM***, pp. 1–9.

[Hai+20] Habtegebreil Haile et al. "End-To-End Congestion Control Approaches for High Throughput and Low Delay in 4G/5G Cellular Networks". In: ***Computer Networks*** (2020), pp. 1–22.

[HD19] Kamal Hakimzadeh and Jim Dowling. "Karamel: A System for Timely Provisioning Large-Scale Software Across IaaS Clouds". In: ***Proc. of IEEE International Conference on Cloud Computing***. 2019, pp. 391–395.

[He+18] Ting He et al. "It'S Hard To Share: Joint Service Placement and Request Scheduling in Edge Clouds With Sharable and Non-Sharable Resources". In: ***Proc. of IEEE International Conference on Distributed Computing Systems***. 2018, pp. 365–375.

[Hma+16] Ali Hmaity et al. "Virtual Network Function Placement for Resilient Service Chain Provisioning". In: ***Proc. of IEEE International Workshop on Resilient Networks Design and Modeling***. 2016, pp. 245–252.

[HNR18] Dor Harris, Joseph Naor, and Danny Raz. "Latency Aware Placement in Multi-Access Edge Computing". In: ***Proc. of IEEE Conference on Network Softwarization and Workshops***. 2018, pp. 132–140.

[HS18] David Hutchison and James PG Sterbenz. "Architecture and Design for Resilient Networked Systems". In: ***Computer Communications*** 131 (2018), pp. 13–21.

[Hu+15] Yun Chao Hu et al. "Mobile Edge Computing: A Key Technology Towards 5G". In: ***ETSI White Paper*** (2015).

[HYW19] Najmul Hassan, Kok-Lim Alvin Yau, and Celimuge Wu. "Edge Computing in 5G: A Review". In: ***IEEE Access*** 7 (2019), pp. 127276–127289.

[HZL18] Yunzhou Han, Xianglin Zhao, and Jianbin Li. "Computer Network Failure and Solution". In: ***Journal of Computer Hardware Engineering*** 1.1 (2018).

[Jak20] Michał Tomasz Jakóbczyk. ***Practical Oracle Cloud Infrastructure***. Springer, 2020.

[Jan+17] Insun Jang et al. "Joint Optimization of Service Function Placement and Flow Distribution for Service Function Chaining". In: ***IEEE Journal on Selected Areas in Communications*** 35.11 (2017), pp. 2532–2541.

[JD18] Slađana Jošilo and György Dán. "Selfish Decentralized Computation Offloading for Mobile Cloud Computing in Dense Wireless Networks". In: ***IEEE Transactions on Mobile Computing*** 18.1 (2018), pp. 207–220.

[JD19] Slađana Jošilo and Gyorgy Dan. "Joint Management of Wireless and Computing Resources for Computation Offloading in Mobile Edge Clouds". In: ***IEEE Transactions on Cloud Computing*** (2019).

[JD20] Slađana Jošilo and György Dán. "Computation Offloading Scheduling for Periodic Tasks in Mobile Edge Computing". In: ***IEEE/ACM Transactions on Networking*** 28.2 (2020), pp. 667–680.

[JWG16] Yichao Jin, Yonggang Wen, and Kyle Guan. "Toward Cost-Efficient Content Placement in Media Cloud: Modeling and Analysis". In: ***IEEE Transactions on Multimedia*** 18.5 (2016), pp. 807–819.

[Kas+20] Shahrukh Khan Kasi et al. "Heuristic Edge Server Placement in Industrial Internet of Things and Cellular Networks". In: ***IEEE Internet of Things Journal*** (2020).

[Kek+18] Sami Kekki et al. "MEC in 5G Networks". In: ***ETSI White Paper*** (2018).

[Khe+19] Nouha Kherraf et al. "Optimized Provisioning of Edge Computing Resources With Heterogeneous Workload in IoT Networks". In: ***IEEE Transactions on Network and Service Management*** 16.2 (2019), pp. 459–474.

[Kho+19] Mohammad Ali Khoshkholghi et al. "Optimized Service Chain Placement Using Genetic Algorithm". In: ***Proc. of IEEE Conference on Network Softwarization***. 2019, pp. 472–479.

[Kir+20] Nahida Kiran et al. "VNF Placement and Resource Allocation in SND/NFV-Enabled MEC Networks". In: ***Proc. of IEEE Wireless Communications and Networking Conference Workshops***. 2020, pp. 1–6.

[Kuo+10] Fang-Chun Kuo et al. "Cost-Efficient Wireless Mobile Backhaul Topologies: An Analytical Study". In: ***Proc. of IEEE Global Telecommunications Conference***. 2010, pp. 1–5.

[LL73] Chung Laung Liu and James W Layland. "Scheduling Algorithms for Multiprogramming in A Hard-Real-Time Environment". In: ***Journal of the ACM (JACM)*** 20.1 (1973), pp. 46–61.

[LSH08] Tomas Lennvall, Stefan Svensson, and Fredrik Hekland. "A Comparison of Wirelesshart and Zigbee for Industrial Applications". In: ***Proc. of IEEE International Workshop on Factory Communication Systems***. 2008, pp. 85–88.

[Lu+20] Dongyu Lu et al. "Robust Server Placement for Edge Computing". In: ***Proc. of IEEE International Parallel and Distributed Processing Symposium***. 2020, pp. 285–294.

[LW18] Yuanzhe Li and Shangguang Wang. "An Energy-Aware Edge Server Placement Algorithm in Mobile Edge Computing". In: ***Proc. of IEEE International Conference on Edge Computing***. 2018, pp. 66–73.

[Mai+19] Adyson M Maia et al. "Optimized Placement of Scalable IoT Services in Edge Computing". In: ***Proc. of IFIP/IEEE Symposium on Integrated Network and Service Management***. 2019, pp. 189–197.

[Mao+17] Yuyi Mao et al. "A Survey on Mobile Edge Computing: The Communication Perspective". In: ***IEEE Communications Surveys & Tutorials*** 19.4 (2017), pp. 2322–2358.

[Mei+19] Sebastian Meixner et al. "Automatic Application Placement and Adaptation in Cloud-Edge Environments". In: ***Proc. of IEEE International Conference On Emerging Technologies and Factory Automation***. 2019, pp. 1001–1008.

[Men+19] Jiaying Meng et al. “Joint Heterogeneous Server Placement and Application Configuration in Edge Computing”. In: ***Proc. of IEEE International Conference on Parallel and Distributed Systems***. 2019, pp. 488–497.

[Mij+15] Rashid Mijumbi et al. “Network Function Virtualization: State-of-The-Art and Research Challenges”. In: ***IEEE Communications surveys & tutorials*** 18.1 (2015), pp. 236–262.

[Mou+20] Abdallah Moubayed et al. “Edge-Enabled V2X Service Placement for Intelligent Transportation Systems”. In: ***IEEE Transactions on Mobile Computing*** (2020).

[Ngu+18] Van-Giang Nguyen et al. “SND Helps Velocity in Big Data”. In: (2018).

[Ouy+19] Tao Ouyang et al. “Adaptive User-Managed Service Placement for Mobile Edge Computing: An Online Learning Approach”. In: ***IEEE INFOCOM***. 2019, pp. 1468–1476.

[Pas+19] Stephen Pasteris et al. “Service Placement With Provable Guarantees in Heterogeneous Edge Computing Systems”. In: ***IEEE INFOCOM***. 2019, pp. 514–522.

[Pat+14] Milan Patel et al. “Mobile-Edge Computing Introductory Technical White Paper”. In: ***White paper, mobile-edge computing (MEC) industry initiative*** 29 (2014), pp. 854–864.

[PM17] Jianli Pan and James McElhannon. “Future Edge Cloud and Edge Computing for Internet of Things Applications”. In: ***IEEE Internet of Things Journal*** 5.1 (2017), pp. 439–449.

[Pop+21] Paul Pop et al. “The FORA Fog Computing Platform for Industrial IoT”. In: ***Information Systems*** 98 (2021), p. 101727.

[Pou+19] Konstantinos Poularakis et al. “Joint Service Placement and Request Routing in Multi-Cell Mobile Edge Computing Networks”. In: ***Proc. of IEEE INFOCOM***. 2019, pp. 10–18.

[Pou+20] Konstantinos Poularakis et al. “Service Placement and Request Routing in MEC Networks With Storage, Computation, and Communication ConstraInts”. In: ***IEEE/ACM Transactions on Networking*** 28.3 (2020), pp. 1047–1060.

[QN15] Paul Quinn and Tom Nadeau. “Problem Statement for Service Function Chaining”. In: ***Internet Requests for Comments, RFC Editor, RFC*** 7498 (2015).

[Sch+17] Philipp Schulz et al. “Latency Critical IoT Applications in 5G: Perspective on The Design of Radio Interface and Network Architecture”. In: ***IEEE Communications Magazine*** 55.2 (2017), pp. 70–78.

[SD19] Ezzeldin Shereen and György Dán. "Model-Based and Data-Driven Detectors for Time Synchronization Attacks Against Pmus". In: ***IEEE Journal on Selected Areas in Communications*** 38.1 (2019), pp. 169–179.

[Sha+16] Yogesh Sharma et al. "Reliability and Energy Efficiency in Cloud Computing Systems: Survey and Taxonomy". In: ***Journal of Network and Computer Applications*** 74 (2016), pp. 66–85.

[Sha+17] Mansoor Shafi et al. "5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice". In: ***IEEE journal on selected areas in communications*** 35.6 (2017), pp. 1201–1221.

[Shi+17] Syed Noorulhassan Shirazi et al. "The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog From A Security and Resilience Perspective". In: ***IEEE Journal on Selected Areas in Communications*** 35.11 (2017), pp. 2586–2595.

[Shi+20] Dian Shi et al. "Mean Field Game Guided Deep Reinforcement Learning for Task Placement in Cooperative Multi-access Edge Computing". In: ***IEEE Internet of Things Journal*** 7.10 (2020), pp. 9330–9340.

[Sin15] Valentine Sinitsyn. June 2015. URL: `https://www.linuxjournal.com/content/jailhouse`.

[Sin17] Sachchidanand Singh. "Optimize Cloud Computations Using Edge Computing". In: ***Proc. of International Conference on Big Data, IoT and Data Science***. 2017, pp. 49–53.

[Ta+08] Duong Ta et al. "Network-Aware Server Placement for Highly Interactive Distributed Virtual Environments". In: ***Proc. of IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications***. 2008, pp. 95–102.

[TDE18] Roberta Terruggia, Giovanna Dondossola, and Mathias Ekstedt. "Cyber Security Analysis of Web-of-Cells Energy Architectures". In: ***Proc. of International Symposium for ICS & SCADA Cyber Security Research***. 2018, pp. 41–50.

[TKH19] Ayaka Takeda, Tomotaka Kimura, and Kouji Hirata. "Evaluation of Edge Cloud Server Placement for Edge Computing Environments". In: ***Proc. of IEEE International Conference on Consumer Electronics***. 2019, pp. 1–2.

[Urg+15] Rahul Urgaonkar et al. "Dynamic Service Migration and Workload Scheduling in Edge-Clouds". In: ***Performance Evaluation*** 91 (2015), pp. 205–228.

[Vac12] John R Vacca. ***Computer and Information Security Handbook***. Newnes, 2012.

[VN20] Kashi Venkatesh Vishwanath and Nachiappan Nagappan. "Characterizing Cloud Computing Hardware Reliability". In: ***Proc. of ACM Symposium on Cloud Computing***. 2020, pp. 193–204.

[Vu+19] Trung Kien Vu et al. "Joint Path Selection and Rate Allocation Framework for 5G Self-Backhauled Mm-Wave Networks". In: ***IEEE Transactions on Wireless Communications*** 18.4 (2019), pp. 2431–2445.

[Wai+19] Philipp Waibel et al. "Viepep-C: A Container-Based Elastic Process Platform". In: ***IEEE Transactions on Cloud Computing*** (2019).

[Wan+14] Xue Wang et al. "Planning and Online Resource Allocation for The Multi-Resource Cloud Infrastructure". In: ***Proc. of IEEE International Conference on Communications***. 2014, pp. 2938–2943.

[Wan+15] Tao Wang et al. "Fault Detection for Cloud Computing Systems With Correlation Analysis". In: ***Proc. of IFIP/IEEE International Symposium on Integrated Network Management***. 2015, pp. 652–658.

[Wan+18] Shiqiang Wang et al. "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning". In: ***Proc. of IEEE INFOCOM***. 2018, pp. 63–71.

[Wan+19a] Meng Wang et al. "An Efficient Service Function Chain Placement Algorithm in A MEC-NFV Environment". In: ***Proc. of IEEE Global Communications Conference***. 2019, pp. 1–6.

[Wan+19b] Shangguang Wang et al. "Edge Server Placement in Mobile Edge Computing". In: ***Journal of Parallel and Distributed Computing*** 127 (2019), pp. 160–168.

[WZL17] Shiqiang Wang, Murtaza Zafer, and Kin K Leung. "Online Placement of Multi-Component Applications in Edge Computing Environments". In: ***IEEE Access*** 5 (2017), pp. 2514–2533.

[YFK18] Louiza Yala, Pantelis A Frangoudis, and Adlen Ksentini. "Latency and Availability Driven VNF Placement in A MEC-NFV Environment". In: ***IEEE Global Communications Conference***. 2018, pp. 1–7.

[Yin+16] Hao Yin et al. "Edge Provisioning With Flexible Server Placement". In: ***IEEE Transactions on Parallel and Distributed Systems*** 28.4 (2016), pp. 1031–1045.

[Yu+18] Nuo Yu et al. "Collaborative Service Placement for Mobile Edge Computing Applications". In: ***Proc. of IEEE Global Communications Conference***. 2018, pp. 1–6.

[ZDa] Peiyue Zhao and György Dán. "Joint Resource Dimensioning and Placement for Dependable Virtualized Services in Mobile Edge Clouds". accepted for publication in ***IEEE Transactions on Mobile Computing***.

[ZDb] Peiyue Zhao and György Dán. "Resilient Placement of Virtual Process Control Functions in Mobile Edge Clouds". In: ***Proc. of IFIP Networking 2017***, pp. 1–9.

[ZDc] Peiyue Zhao and György Dán. "Scheduling Parallel Migration of Virtualized Services Under Time ConstraInts in Mobile Edge Clouds". In: ***Proc. of International Teletraffic Congress 2019***, pp. 28–36.

[ZDd] Peiyue Zhao and György Dán. "Time Constrained Service-Aware Migration of Virtualized Services for Mobile Edge Computing". In: ***Proc. of International Teletraffic Congress 2018***, pp. 64–72.

[ZD18] Peiyue Zhao and György Dán. "A Benders Decomposition Approach for Resilient Placement of Virtual Process Control Functions in Mobile Edge Clouds". In: ***IEEE Transactions on Network and Service Management*** 15.4 (2018), pp. 1460–1472.

[Zen+19] Feng Zeng et al. "Cost-Effective Edge Server Placement in Wireless Metropolitan Area Networks". In: ***Sensors*** 19.1 (2019), p. 32.

[ZF19] Ming Zeng and Viktoria Fodor. "Dynamic Spectrum Sharing for Load Balancing in Multi-Cell Mobile Edge Computing". In: ***IEEE Wireless Communications Letters*** 9.2 (2019), pp. 189–193.

[ZH17] He Zhu and Changcheng Huang. "Availability-Aware Mobile Edge Application Placement in 5G Networks". In: ***Proc. of IEEE Global Communications Conference***. 2017, pp. 1–6.

[Zha+14] Qi Zhang et al. "Venice: Reliable Virtual Data Center Embedding in Clouds". In: ***Proc. of IEEE INFOCOM***. 2014, pp. 289–297.

[Zha+18a] Lei Zhao et al. "Optimal Placement of Cloudlets for Access Delay Minimization in SND-Based Internet of Things Networks". In: ***IEEE Internet of Things Journal*** 5.2 (2018), pp. 1334–1344.

[Zha+18b] Peiyue Zhao et al. "A Game Theoretic Approach To Setting The Pilot Power Ratio in Multi-User MIMO Systems". In: ***IEEE Transactions on Communications*** 66.3 (2018), pp. 999–1012.

[Zha+19a] Yuan Zhang et al. "Dynamic Service Placement for Virtual Reality Group Gaming On Mobile Edge Cloudlets". In: ***IEEE Journal on Selected Areas in Communications*** 37.8 (2019), pp. 1881–1897.

[Zha+19b] Peiyue Zhao et al. "A Game Theoretic Approach To Uplink Pilot and Data Power Control in Multi-Cell Multi-User MIMO Systems". In: ***IEEE Transactions on Vehicular Technology*** 68.9 (2019), pp. 8707–8720.

[ZL18] Lei Zhao and Jiajia Liu. "Optimal Placement of Virtual Machines for Supporting Multiple Applications in Mobile Edge Networks". In: ***IEEE Transactions on Vehicular Technology*** 67.7 (2018), pp. 6533–6545.