
SLADANA JOŠILO

Doctoral Thesis
Stockholm, Sweden 2018
Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framläggas till offentlig granskning för avläggande av doktorsexamen torsdag den 31 May 2018 klockan 10.00 i sal Q2, KTH, Stockholm.

© Sladana Jošilo, May 2018

Tryck: Universitetsservice US AB
Abstract

The rapid increase in the number of mobile devices has been followed by an increase in the capabilities of mobile devices, such as the computational power, memory and battery capacity. Yet, the computational resources of individual mobile devices are still insufficient for various delay sensitive and computationally intensive applications. These emerging applications could be supported by mobile cloud computing, which allows using external computational resources. Mobile cloud computing does not only improve the users’ perceived performance of mobile applications, but it also may reduce the energy consumption of mobile devices, and thus it may extend their battery life. However, the overall performance of mobile cloud computing systems is determined by the efficiency of allocating communication and computational resources. The work in this thesis proposes decentralized algorithms for allocating these two resources in mobile cloud computing systems.

In the first part of the thesis, we consider the resource allocation problem in a mobile cloud computing system that allows mobile users to use cloud computational resources and the resources of each other. We consider that each mobile device aims at minimizing its perceived response time, and we develop a game theoretical model of the problem. Based on the game theoretical model, we propose an efficient decentralized algorithm that relies on average system parameters, and we show that the proposed algorithm could be a promising solution for coordinating multiple mobile devices.

In the second part of the thesis, we consider the resource allocation problem in a mobile cloud computing system that consists of multiple wireless links and a cloud server. We model the problem as a strategic game, in which each mobile device aims at minimizing a combination of its response time and energy consumption for performing the computation. We prove the existence of equilibrium allocations of mobile cloud resources, and we use game theoretical tools for designing polynomial time decentralized algorithms with a bounded approximation ratio. We then consider the problem of allocating communication and computational resources over time slots, and we show that equilibrium allocations still exist. Furthermore, we analyze the structure of equilibrium allocations, and we show that the proposed decentralized algorithm for computing equilibria achieves good system performance.

By providing constructive equilibrium existence proofs, the results in this thesis provide low complexity decentralized algorithms for allocating mobile cloud resources for various mobile cloud computing architectures.
In loving memory of my mother, Milenija Mihajlović Jošilo.
Acknowledgments

I would like to thank my advisor György Dán for his patience, support and encouragement. His consistent guidance and valuable suggestions at every stage of the research work have helped me to complete the thesis successfully.

I would also like to thank all past and present members of the NSE department for providing a friendly environment. I am thankful to my close friends for their love, enthusiasm and help.

I am deeply thankful to my family for their love and sacrifices that they have made on my behalf.
Contents

1 Introduction 1
   1.1 Background ........................................... 1
   1.2 Challenges ............................................. 2
   1.3 Thesis Structure ....................................... 3

2 Mobile Cloud Computing Resources 5
   2.1 Communication Resources ............................ 5
   2.2 Mobile Cloud Computing Architectures ............... 7

3 Computation Offloading for Mobile Systems 9
   3.1 Computational Tasks ................................... 9
   3.2 Performance Metrics .................................. 10

4 Resource Allocation in Mobile Cloud Computing Systems 13
   4.1 Cost Model ............................................. 13
   4.2 Resource Allocation Problem Formulation .............. 14

5 Summary of Original Work 21

6 Conclusions and Future Work 25


Paper B: A Game Theoretic Analysis of Selfish Mobile Computation Offloading 53
Contents

Paper C: Decentralized Scheduling for Offloading of Periodic Tasks in Mobile Edge Computing 77
1.1 Background

In recent years, the number of mobile devices including smartphones, tablets, sensors, and other portable devices, has been rapidly increasing. According to a recent estimate by Cisco, the number of mobile devices and connections will grow to 11.6 billion by 2021 at a compound annual growth rate (CAGR) of 8% between 2016 and 2021 [1].

Despite such a significant growth in the number of mobile devices, the computational capabilities of individual mobile devices are still limited compared to desktop computers [2]. In addition, mobile devices are mostly battery powered, and thus energy consumption is an important aspect that requires careful attention. Unfortunately, battery technology has still not been able to meet energy consumption requirements without limiting the clock speed of processors; doubling the clock speed approximately octuples the energy consumption [3]. Furthermore, in order to be easily carried, mobile devices must be physically light and small, which puts additional limitations on specific hardware resources, such as computational capabilities, battery capacity, and memory and disk capacity [4].

At the same time, computationally intensive applications, including augmented reality, natural language processing, face, gesture, speech and object recognition are increasingly used on mobile devices; for example, Cisco predicted that only augmented reality traffic will increase sevenfold between 2016 and 2021 [1]. Therefore, it is clear that the computational capabilities of mobile devices are increasing at a slower rate than the computational requirements of the applications.

A widely adopted approach to closing the gap between the limited computational capabilities of mobile devices and high computational requirements of the applications is mobile cloud computing [5, 6]. Mobile cloud computing augments the computational capabilities of mobile devices by allowing them to use external computational resources. By relying on external computational resources, mobile cloud computing may accelerate the execution of applications, may extend the bat-
tery lifetime of mobile devices, and may enable collaboration among mobile devices. All these potential benefits contribute to the growing interest in mobile cloud services; the mobile cloud market is expected to grow from $12.07 billion in 2016 to $74.25 billion by 2023 at a CAGR of 30.1% [7].

1.2 Challenges

The growth in the number of mobile devices over the past years has been followed by a corresponding growth in mobile data traffic. According to a recent estimate by Cisco, the overall mobile data traffic will increase sevenfold between 2016 and 2021 at a CGR of 47% [1]. Strong mobile data traffic growth puts stress on mobile cloud communication and computational infrastructures, and thus it affects users’ perception of mobile cloud computing performance. Therefore, effective management of mobile cloud communication and computational resources is an important part of designing the mobile cloud computing systems.

There are two fundamental challenges facing the design of mobile cloud computing systems. The first is meeting users’ requirements concerning the overall application response time. The application response time is not only affected by the limited computational resources of mobile devices, but it may also be affected by the wireless network constraints in the case of using external computational resources. The other fundamental challenge is meeting users’ requirements concerning the battery lifetime of their mobile devices. The battery lifetime is mostly determined by the energy consumption rate, which may depend on several factors, such as the type of applications and network connection.

In response to these challenges, different mobile cloud architectures have been considered. The traditional architectures make use of the commercial cloud infrastructures, by allowing mobile devices to offload their computational tasks to the remote resourceful clouds, such as Amazon EC2 [8] and Windows Azure [9]. In order to meet the extremely low latency requirements of emerging delay sensitive applications, recently proposed architectures consider the execution of applications in close proximity of the end users.

Mobile edge computing (MEC) [10] proposes bringing computational resources close to the network edge, and thus it is recognized as one of the key emerging technologies for 5G networks [11]. Another architecture that enables execution of applications in close proximity to the end users is fog computing [12]. Fog computing extends MEC services by using heterogeneous devices, such as access points, edge routers and switches as the service nodes, and thus it is considered to be a potential platform for Internet of Things (IoT) applications [13, 14].

In either case, when many mobile devices compete for communication and computational resources, new challenges arise for several reasons. First, the mobile devices are heterogeneous in terms of computational capabilities and in terms of the constraints on the total energy consumption. Second, the applications running on different mobile devices may be different in terms of what computational
tasks they consist of and how often they generate the computational tasks. Third, the mobile devices may be autonomous, and hence they may be interested in improving their own performance. These challenges additionally complicate the design of mobile cloud computing systems, especially in the case of delay sensitive and computationally intensive applications.

1.3 Thesis Structure

The structure of this thesis is as follows. In Chapter 2, we discuss characteristics of different wireless access technologies and we present both traditional centralized and emerging distributed mobile cloud computing architectures. In Chapter 3, we define the computational tasks, and we introduce performance metrics for evaluating mobile cloud computing systems. In Chapter 4, we present different formulations of the resource allocation problem for various mobile cloud computing architectures. In Chapter 5, we provide a summary of the papers included in this thesis, and in Chapter 6 we conclude the work and discuss potential directions for future research.
Mobile Cloud Computing Resources

Mobile cloud computing systems consist mainly of two types of resources, that is, communication and computational resources. Figure 2.1 shows an example of a mobile cloud computing system. As illustrated in the figure, mobile devices can decide whether to perform the computation using local computational resources or to offload the computation to external computational resources through communication networks.

In the case of offloading, mobile devices compete for communication and computational resources, and thus the decision of a mobile device affects both its own performance and the performance of the other mobile devices. In this chapter we present different wireless access technologies and different mobile cloud computing architectures, and we discuss the main factors affecting the performance of mobile cloud computing systems.

2.1 Communication Resources

When offloading their tasks to external computational resources mobile devices rely on wireless networks. Today’s wireless networks are highly heterogeneous, and thus mobile users can usually select among different radio access technologies, such as 2.5G, 3G, 4G and Wi-Fi [15].

The most common problems facing these radio access technologies are intermittent connectivity, variable network conditions, and limited bandwidth [16, 17]. Furthermore, the communication medium is shared among users in the same area, and thus the transmission rate of a user depends on the bandwidth allocation algorithm.

The bandwidth allocation algorithm in the distributed coordination function (DCF) used in the IEEE 802.11 standard uses the CSMA/CA protocol for implementing a fair sharing of the bandwidth [18, 19]. According to the DCF algorithm, the bandwidth of an access point (AP) $a$ is fairly shared among the set $N_a$ of users connected to the AP $a$. Hence, given the set $N_a$ the uplink rate $\omega_{i,a}$ of user $i \in N_a$
can be expressed as
\[ \omega_{i,a} = f_a(N_a), \forall i \in N_a. \]

Other examples of fair bandwidth sharing mechanisms are the ones used in time-fair TDMA and OFDM based medium access protocols in which the uplink rate \( \omega_{i,a} \) of user \( i \) on AP \( a \) does not depend on the specific set \( N_a \) of users sharing the AP, but it may depend on the total number \( |N_a| \) of users sharing the AP [20, 21]. Common to these protocols is that the uplink rate \( \omega_{i,a} \) of user \( i \) on AP \( a \) can be user specific, and given \( |N_a| \) it can be expressed as
\[ \omega_{i,a} = f_{i,a}(|N_a|), \forall i \in N_a. \]

A model similar to the latter uplink rate model can also be used to describe the proportional-fair scheduling (PFS) in 3G networks [22].

Given the overall growth in the number of mobile devices and the latency requirements of emerging delay sensitive applications, it is clear that communication resources in mobile cloud computing systems have to be managed appropriately. There have been a few recent approaches with a strong focus on the communication related problems in mobile cloud computing systems [23, 24, 25, 26, 27]. Approaches considered in [23, 24] propose mechanisms for predicting network connectivity based on the users’ movement and a database of network connectivity over geographical zones. Approaches considered in [25, 26, 27] propose collaboration among mobile devices. The latter approach is especially interesting for two
reasons. First, it can improve bandwidth utilization and can make use of device-
to-device (D2D) communication, which is considered to be a promising technology
for future 5G cellular networks [28, 29, 30]. Second, the concept of collaboration
provides a basis for moving towards highly distributed mobile cloud computing
architectures [31, 32, 33].

2.2 Mobile Cloud Computing Architectures

As illustrated in Figure 2.1, the sources of computational resources may be differ-
ent, from remote commercial clouds and the clouds located at the network edge,
to the user carried mobile devices and mobile devices attached to the vehicles. In
the following we discuss the factors affecting the performance of mobile cloud com-
puting systems for traditional centralized and emerging distributed mobile cloud
computing architectures.

Centralized Clouds

According to a recent data from Synergy Research Group, the four leading com-
cerical cloud providers are Amazon, Azure, IBM and Google [34]. These cloud
providers usually offer three types of cloud computing services, Infrastructure as a
Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [35].
IaaS and PaaS offer a high level of control, flexibility, and management, and thus
they are suitable for application owners that provide the end users with a final
product through SaaS [36]. Therefore, from the perspective of a mobile user the
most relevant cloud computing service is SaaS.

A number of recent works have investigated the performance of commercial cloud
computing services [37, 38, 39, 40]. According to the reported measurements, com-
cercial clouds may have very high end-to-end transmission delays, which mostly
occur because commercial cloud infrastructures offer no guarantees on proximity of
their servers to the end users. As a consequence, computation offloading to remote
commercial clouds may not always be a good solution, especially in the case of
delay sensitive applications.

Mobile Edge Clouds

Emerging distributed mobile edge clouds are widely accepted as a promising alter-
native to the centralized clouds [41]. The key idea of MEC is to install the com-
putational and storage resources together with the existing infrastructures, such as
mobile BSs [10]. This idea is especially attractive for the network operators, since
it gives them an opportunity to profit from the MEC market.

By bringing computational resources close to the network edge, MEC has a po-
tential to improve the performance perceived by the users. However, when many
mobile users use the MEC resources simultaneously, they may experience a de-
graded performance for two reasons. First, the amount of bandwidth that a BS can
assign to a mobile user may not be sufficient for providing satisfactory transmission
times, especially in the case of applications that require offloading huge volumes
of data. Second, MEC provides only limited computational and storage resources
compared to commercial cloud infrastructures. As a consequence, the amount of
computational resources that an edge cloud can assign to a mobile device may not
be sufficient for providing satisfactory execution times, especially in the case of
computationally intensive applications. Therefore, using MEC resources requires
optimizing not only the allocation of communication, but also the allocation of
computational resources, which makes the problem inherently challenging.

Fog resources

Fog computing is about shifting away from the centralized cloud architectures to-
wards highly distributed architectures. Compared to MEC, fog computing proposes
bringing computational resources even closer to the end users [12]. The key idea of
fog computing is to extend the existing centralized cloud computing architecture
by allowing collaboration among distributed edge cloud resources and nearby het-
erogeneous devices. Todays’ devices may have at disposal limited computational
resources when considered alone, but by pooling their resources they can form a
computationally rich distributed computing platform.

The reason for shifting towards distributed architectures is that emerging appli-
cations, such as various IoT applications have requirements that can not be always
addressed by relying only on computational resources of individual devices and
centralized clouds [13, 14]. On the one hand, this is due to that many individual
devices are still not enough powerful to support computationally intensive applica-
tions. On the other hand, some devices may have difficulty connecting to the
centralized cloud due to network bandwidth constraints.

Fog computing is not only beneficial from the perspective of computational
resources, but it is also beneficial from the perspective of communication resources,
because collaboration among nearby devices can make use of D2D communication,
and thus it may improve bandwidth utilization [42]. However, the two biggest
challenges facing fog computing are how to integrate heterogeneous devices into a
common computing platform, and how to efficiently distribute the computational
tasks among numerous devices [43].
Computation Offloading for Mobile Systems

The applications running on the mobile devices may be partitioned into computational tasks at different granularity levels [44]. According to application partitioning requirements, there are two main classes of computation offloading frameworks. The first class considered in [45, 46, 47] does not require partitioning, and the entire application can be offloaded. The second class considered in [5, 48, 49, 50] requires application partitioning into computational tasks that can be offloaded for remote execution and computational tasks that have to be executed on the mobile device. In this chapter, we define the parameters that characterize the computational tasks and we introduce the main performance metrics that can be used to measure the efficiency of resource allocation in mobile cloud computing systems.

3.1 Computational Tasks

In what follows, we consider only the computational tasks that can be offloaded, and we characterize the mobile user i’s computational task by two parameters. The first parameter is the size $D_i$ of the input data that must be offloaded in the case of remote execution, and the second parameter is the complexity $L_i$, that is, the number of CPU cycles required to perform the computation. The relation between the size $D_i$ of the input data and the task complexity $L_i$ can be expressed as $L_i = D_i X_i$ [51], where $X_i$ is the number of CPU cycles per data bit, which can be approximated by a Gamma distribution [52, 53].

The mobile devices and the cloud servers differ from the perspective how fast they can execute the same computational task $<D_i, L_i>$, which mostly depends on their CPU performance. Although every new generation of mobile devices is more and more powerful, yet the gap between the CPU performance of individual mobile devices and server grade computers remains [54].
Chapter 3. Computation Offloading for Mobile Systems

The CPU performance can be characterized by a wide range of performance metrics, among which the basic metrics are the clock cycle time and the clock frequency [55]. Yet, the frequency $F$ at which the processor executes a computational task $<D_i, L_i>$ is often not the same as the clock frequency. This is mostly due to that the processor performs many different tasks simultaneously (e.g., the management of the underlying hardware, operating system activities, and input/output (I/O) operations).

Given that the frequency $F$ is expressed in CPU cycles per second, where the notion of an instruction is different between different instruction set architectures (e.g. RISC, CISC and VLIW), the time $T_{i}^{exe}$ needed to execute a computational task $<D_i, L_i>$ can be expressed as

$$T_{i}^{exe} = \frac{L_i}{F_i}.$$

The frequency $F$ at which the processor executes a computational task $<D_i, L_i>$ also influences the energy consumption. According to the measurements reported in [6, 51], the energy consumption per CPU cycle is linearly proportional to the square of the frequency $F$. Thus, the energy $E_{i}^{exe}$ needed to execute a computational task $<D_i, L_i>$ can be expressed as

$$E_{i}^{exe} = cF^2 L_i,$$

where the constant $c \sim 10^{-11}$ according to the reported measurements.

The above models for the execution time and the energy consumption capture the main characteristics of executing the computational tasks. The models are especially suitable for tasks that are characterized by a large complexity $L_i$, and thus that are not very sensitive to the possible interruptions that may occur during the startup, execution, and termination.

### 3.2 Performance Metrics

The two main performance metrics for assessing the performance of mobile cloud computing systems are the task completion time and the corresponding energy consumption of mobile devices. The factors that affect these performance metrics depend on which computational resources are used to execute a task. In the following we define the task completion time and the energy consumption both in the case of local and remote execution.

#### Task Completion Time

When a task $<D_i, L_i>$ is executed locally, the time $T_{i}^{cl}$ needed to complete the task is the time needed to execute the task using local computational resources at frequency $F_i$

$$T_{i}^{cl} = \frac{L_i}{F_i}.$$
3.2. Performance Metrics

On the contrary, when a mobile user offloads its task \( <D_i, L_i> \) to external computational resources through an AP \( a \), the task completion time consists of three parts. The first part is the time \( T_{t,a} \) needed to transmit the amount of \( D_i \) input data through AP \( a \) at the rate \( \omega_{i,a} \), and it can be expressed as

\[
T_{t,a} = \frac{D_i}{\omega_{i,a}}.
\]

The second part is the time \( T_{exe}^{e} \) needed to execute the task using external computational resources at frequency \( F_i^{e} \)

\[
T_{exe}^{e} = \frac{L_i}{F_i^{e}}.
\]

The third part is the time needed to transmit the result of the computation from external computational resources to the mobile device. For many applications, such as tracking, object, face and speech recognition, the size of the result is much smaller than the size \( D_i \) of the input data, and thus the third part can be neglected \[56, 57, 58\]. Therefore, in the case of computation offloading through an AP \( a \), a simple linear model can be used to model the task completion time \( T_{c,e}^{c,e} \)

\[
T_{c,e}^{c,e} = T_{t,a} + T_{exe}^{e}.
\]

Energy Consumption

When a task \( <D_i, L_i> \) is executed locally, the energy consumption \( E_i^{l} \) of a mobile device is the energy needed to execute the task using local computational resources at frequency \( F_i \)

\[
E_i^{l} = cF_i^{2}L_i.
\]

On the contrary, when a mobile user offloads its task \( <D_i, L_i> \) to external computational resources through an AP \( a \), the energy consumption \( E_{i,a} \) is the energy spent to upload the amount \( D_i \) of the input data. According to measurements reported in \[59\], the energy spent to upload the data over the cellular network consists of three parts. The first part is the energy spent to scan available wireless connections, the second part is the energy spent to transmit data, and the third part is the energy spent to keep the interface up during the transmission period.

When a task is characterized by a large size \( D_i \) of the input data, it is reasonable to consider that the energy spent to transmit data dominates the energy spent to scan available wireless connections and the energy spent to keep the interface up during the transmission period. Consequently, given that a mobile user transmits the data through AP \( a \) at rate \( \omega_{i,a} \) using transmit power \( P_{i,a} \), the energy consumption \( E_{i,a} \) can be expressed as

\[
E_{i,a} = \frac{D_i P_{i,a}}{\omega_{i,a}}.
\]
Resource Allocation in Mobile Cloud Computing Systems

When designing mobile cloud computing systems, one of the main objectives is providing a convenient solution for mobile devices to perform their applications efficiently, in terms of both the application completion time and the energy consumption. Due to increasing use of mobile devices, efficient management of mobile cloud computing resources is crucial to achieve this objective. In the following, we model the cost associated with mobile users, and we discuss different formulations of the resource allocation problem.

4.1 Cost Model

Since mobile devices are heterogeneous in terms of computational capabilities, battery states, and in terms of what type of computational tasks they have to execute, it is reasonable to introduce the notion of preferences over the performance metrics. The heterogeneity among mobile devices can be modeled using two parameters, $0 \leq \gamma^T_i \leq 1$ and $0 \leq \gamma^E_i \leq 1$, which characterize user $i$’s preferences regarding the completion time and the energy consumption, respectively. Given these parameters, the user $i$’s cost can be formulated as a function of the weighted completion time and the weighted energy consumption,

\[
C^l_i = f(\gamma^T_i T^l_i, \gamma^E_i E^l_i),
\]

\[
C^e_{i,a} = f(\gamma^T_i T^e_{i,a}, \gamma^E_i E^e_{i,a}).
\]

The above cost models allow a mobile user to dynamically adjust its objective to the specific application requirements, and to its current battery state by changing the values of the parameters $\gamma^T_i$ and $\gamma^E_i$. 
4.2 Resource Allocation Problem Formulation

In order to provide a general formulation of the resource allocation problem, in the following we consider a mobile cloud computing system that consists of a set $\mathcal{N}$ of mobile users, $|\mathcal{N}| = N$, a set $\mathcal{A}$ of communication resources, $|\mathcal{A}| = A$, and a set $\mathcal{S}$ of computational resources, $|\mathcal{S}| = S$. We use $X \in \{0, 1\}^{N \times A}$ and $Y \in \{0, 1\}^{N \times S}$ to denote communication and computational resource assignment matrices, respectively.

One potential goal could be to minimize the system cost $C$, which is defined as the sum over all users’ costs. Since the resources are shared among the users, the system cost $C$ is a function of $X$ and $Y$, that is, $C(X, Y) = C(X, Y)$. Using the above notation, the problem of resource allocation in a mobile cloud computing system can be formulated as the following $0-1$ nonlinear optimization problem,

$$\min_{X, Y} f_0(X, Y, C(X, Y))$$

s.t.  

$$g_i(X, Y) \leq a_i, \forall i \in \mathcal{N}, \quad (4.2)$$

$$h(X) \leq b_a, \forall a \in \mathcal{A}, \quad (4.3)$$

$$q(Y) \leq c_s, \forall s \in \mathcal{S}, \quad (4.4)$$

$$\sum_{a \in \mathcal{A}, s \in \mathcal{S} \setminus \{i\}} x_{i,a} y_{i,s} + y_{i,i} = 1, \forall i \in \mathcal{N}, \quad (4.5)$$

$$X \in \{0, 1\}^{N \times A}, \quad (4.6)$$

$$Y \in \{0, 1\}^{N \times S}. \quad (4.7)$$

The function $g_i(X, Y)$ takes into account sharing both communication and computational resources. For example, constraint (4.2) may be used to ensure that the task completion time or the energy consumption of each user $i \in \mathcal{N}$ is lower than the threshold specified by $a_i$. The functions $h(X)$ and $q(Y)$ take into account sharing only one type of resources, that is, the communication and computational resources, respectively. The constraints (4.3) and (4.4) can be used to enforce a limitation on the amount of communication and computational resources that can be provided to each user, respectively. The constraint (4.5) ensures that each user $i \in \mathcal{N}$ either performs the computation locally ($x_{i,a} = 0, y_{i,i} = 1, \forall a \in \mathcal{A}, \forall s \in \mathcal{S} \setminus \{i\}$) or it offloads the task to computational resource $s$ using communication resource $a$ ($x_{i,a} = 1, y_{i,s} = 1, x_{i,a'} = 0, y_{i,s'} = 0, \forall a' \in \mathcal{A} \setminus \{a\}, \forall s' \in \mathcal{S} \setminus \{s\}$).

Observe that the above optimization problem can be easily reduced to the completion time minimization problem by setting the completion time parameter $\gamma^T_i = 1$, and the energy consumption parameter $\gamma^E_i = 0$, for all mobile users $i \in \mathcal{N}$. Similarly, we can define the energy consumption minimization problem by setting the energy consumption parameter $\gamma^E_i = 1$, and the completion time parameter $\gamma^T_i = 0$, for all mobile users $i \in \mathcal{N}$. However, solving the problem (4.1) – (4.7) may be impractical in realistic mobile cloud computing systems, because it involves searching a large solution space. With this in mind, in the following we distinguish
between the three different formulations of the resource allocation problem in mobile cloud computing systems, and we discuss the most important results from the literature.

Completion Time Minimization

Completion time minimization is usually considered in the case of delay sensitive applications such as mobile augmented reality, real-time voice and video. Most of the works that aim at minimizing the application completion time consider the joint computation partitioning and resource allocation problem, while meeting various constraints that can arise in mobile cloud computing systems. In the following we discuss two classes of works that consider independent tasks and dependent tasks, respectively.

Independent task scheduling problem: The works presented in [60, 61] studied the completion time minimization problem assuming that there is no dependency among the computational tasks. The authors in [60] considered a mobile cloud computing system that consists of a set of processors with known processing times and, a set of processors with unknown processing times. Given the set of independent tasks in the system, the authors aim at minimizing the time when the processing of the last task is completed, that is, the makespan of the given tasks. For the case when the processing time is unknown for only one of the processors, the authors proposed a constant-factor approximation algorithm for scheduling the tasks. They extended the analysis to the case of multiple processors with unknown processing times, and in this case they proposed a heuristic algorithm. The authors in [61] considered a system with stochastic task arrivals, and they defined the user’s cost as the expected number of its tasks in the system, that is, the product of the user’s expected task completion time and the rate at which its device generates tasks. In the considered system, mobile devices may offload their tasks either to an edge cloud or to a centralized cloud with the objective to minimize their costs under energy consumption constraints. The authors used game theoretical tools to show existence of a mixed strategy equilibrium task allocation and they developed a distributed algorithm for computing it.

Our work presented in Paper A falls into this class of completion time minimization problems. We considered a fog computing system that consists of a centralized cloud and multiple heterogeneous devices, which may process the tasks of each other. We considered stochastic task arrivals, and we modeled the task arrival process of each device as a Poisson process. We used a queuing model to capture the contention for both communication and computational resources, and we denoted by $T_{i,j}(p_{i,j})$ the mean time that is needed to complete device $i$’s task using node $j$’s computational resources. Given the task assignment matrix $P \in [0,1]^{N \times (N+1)}$, the system cost $\bar{C}(P)$ can be defined as the average system delay

$$\bar{C}(P) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \cup \{0\}} p_{i,j} T_{i,j}(p_{i,j}).$$
Chapter 4. Resource Allocation in Mobile Cloud Computing Systems

where $p_{i,i}$, $p_{i,0}$ and $p_{i,j}$ indicate the probability that user $i$ executes its task locally, offloads the task to the cloud, and offloads the task to device $j \in N \setminus \{i\}$, respectively. Given the set $\mathcal{P}$ of all task assignment matrices $\mathbf{P}$ that ensure the stability of the queuing system, the problem can be formulated as the following convex optimization problem

\[
\min_{\mathbf{P} \in \mathcal{P}} \tilde{C}(\mathbf{P}) \tag{4.8}
\]

\[
s.t. \sum_{j \in N \setminus \{0\}} p_{i,j} = 1, \forall i \in N. \tag{4.9}
\]

Since devices in fog computing systems are expected to be autonomous [62], in Paper A instead of solving (4.8) – (4.9) we defined the problem as a strategic game, in which each device plays a mixed strategy and aims at minimizing its own cost. We used game theoretical tools to prove the existence of an equilibrium task allocation in static mixed strategies. We proposed a decentralized algorithm that allocates tasks according to the computed mixed strategy profile, and thus it relies on average system parameters only. By performing simulations, we compared the performance of the proposed algorithm with the performance of an algorithm that allocates tasks according to the optimal static mixed strategy, that is, according to the solution of (4.8) – (4.9), and with the performance of a greedy algorithm. We showed that the proposed algorithm achieves close to optimal system performance, and it performs good even compared with a greedy algorithm that is based on global knowledge of the system state.

**Dependent task scheduling problem:** There are several works that considered the computation offloading problem, while assuming dependency between the tasks [63, 64, 65, 66, 67]. The authors in [63, 64] addressed the problem of the feature-based visual analysis in a sensor network that consists of a single camera node, and multiple camera nodes, respectively. In [63] the authors proposed an approximate optimal solution for minimizing the completion time of distributed feature extraction. In [64] the authors showed that the multi-sensor completion time minimization problem is NP-hard, and they proposed a distributed solution for allocating the computing tasks that is periodically supported by the centralized coordinator, which provides the limited amount of shared information to the sensor nodes. The authors in [65, 66] considered the problem of joint computation partitioning and resource allocation for delay sensitive applications in a mobile edge cloud computing system that serves multiple mobile users. In [66] the authors extended the model from [65] to consider not only the allocation of computational, but also the allocation of communication resources. They proposed a heuristic for minimizing the average application completion time, under the constraints on the dependency of the partitioned tasks and the constraints on the amount of available resources. The authors in [67] considered the problem of computation offloading in a mobile cloud computing system where the cloud resources are shared among many users. Motivated by the observation that the completion time of the application can be reduced by maximizing the parallelism between the mobile device and the cloud
server, the authors proposed two computation offloading algorithms for two different types of the computational tasks. For sequential tasks, they proposed an algorithm that finds the optimal solution, while for concurrent tasks they propose a load-balancing heuristic.

### Energy Consumption Minimization

The need for minimizing the energy consumption of mobile devices is especially prominent in the case of computationally intensive applications such as augmented reality, face and object recognition. There is a significant body of works that analyze the energy consumption minimization problem under delay constraints, and a few works that do not take the delay constraints into account. Common to these classes of works is that they both aim to extend the battery lifetime of mobile devices through energy consumption minimization. However, if the applications are delay sensitive at the same time, then the first class is more relevant, otherwise the latter class can be considered.

**Energy consumption minimization subject to delay constraints:** Among many works [5, 68, 69, 70, 71] that fall into this class, the work presented in [5] is considered to be one of the pioneering works in the mobile cloud computing area. The authors in [5] considered the case of a single user, and they formulated the offloading problem as a 0–1 integer linear program. The solution of the program dictates how to partition the application so that the energy consumption of the device is minimized, while meeting the task completion time constraint.

The authors in [68] considered a mobile cloud computing architecture where each mobile device can decide whether to execute the application locally, or on its clone that runs on a virtual machine in a nearby cloud. They considered two scheduling problems, that is, the scheduling problem in the case of local execution and the scheduling problem in the case of execution in the cloud clone. They solved the corresponding convex optimization problems analytically, and showed that the energy consumption can be minimized by optimally configuring the CPU clock frequency of mobile devices in the case of local execution, and by optimally scheduling the data transmission in the case of execution in the cloud clone. The authors in [69] considered a fog computing system that consists of a set of cloud servers and a set of fog devices that are located close to the end users. They proposed an approximate approach to solve the problem of minimizing the energy consumption of the fog computing system while meeting the end users’ delay constraints.

The works presented in [70, 71] considered a mobile cloud computing system that consists of multiple mobile users and one cloud server. The authors in [70] proposed a method to optimize the allocation of communication and computational resources by solving the optimization problem that minimizes the transmit power of the mobile devices, under the constraint on the maximum delay. The authors in [71] formulated the problem as a competitive game where the users aim at minimizing their energy consumption while meeting the task completion time constraints. They
showed that the game always has a pure Nash equilibrium, and that the equilibrium can be computed efficiently.

Energy consumption minimization without considering delay constraints: The works presented in [72, 73] used game theoretical tools to model and analyze the interaction among multiple devices in a mobile cloud computing system. The authors in [72] considered that each device can decide whether to perform the computation locally or to offload the computation to one of the multiple cloud servers. They modeled the problem as a congestion game with the objective to reduce the overall energy consumption of a mobile cloud computing system, including the energy consumed by mobile devices and the energy consumed by cloud servers. When formulating the game, the authors considered sharing only computational resources, and they proved the existence of a Nash equilibrium that can be computed in a polynomial time. The authors in [73] considered a mobile cloud computing architecture where multiple mobile users collaborate. By assuming that there is a centralized entity in the cloud that provides the required information to the users, the authors formulated the problem as a $0-1$ integer quadratic program and they used a heuristic to find the optimal solution. Since the average running time of the proposed heuristic increases exponentially with the number of users, they modeled the collaboration among mobile devices as a coalition game, and proposed a distributed coalition formation algorithm that does not require the assistance of the centralized entity.

Completion Time and Energy Consumption Minimization

When minimizing the completion time and the energy consumption together, the objective could be to explore the trade off of the overall application response time versus the consumed energy. The corresponding problems can be divided into two categories depending on whether they involve optimizing one objective function, or multiple objective functions simultaneously.

Single-objective computation offloading problem: There is a significant body of works that consider a single-objective offloading problem, where the system cost is defined as a linear combination of the system delay and the energy consumption [74, 75, 76, 77, 56, 78]. The authors in [74] considered the case of a single user, and based on a stochastic model of the dynamic offloading problem, they proposed a dynamic offloading policy. The authors in [75] integrated dynamic offloading with resource scheduling and they proposed two policies for minimizing the application completion time and the energy consumption of mobile devices. The first policy is based on optimally adjusting the CPU clock frequency of the mobile devices and it is relevant in the case of local execution. The second policy is based on optimally adjusting the transmission power, and it is relevant in the case of offloading. The authors in [76] considered a mobile cloud computing system that consists of multiple mobile devices, an edge cloud, and a centralized cloud. They proposed a heuristic that minimizes the weighted sum of the energy consumption and the corresponding
maximum transmission and execution times among all users, and by performing simulations they showed that their algorithm gives close to optimal performance.

The works presented in [77, 56, 78] used game theoretical tools in order to model and analyze the problem of allocating the resources in mobile cloud computing systems that serve multiple mobile users. The authors in [77] considered a centralized mobile cloud computing system where the users are served by one remote cloud. They provided a two stage game-based formulation of the problem, with the objective to minimize the users’ energy consumption and the task completion time in the first stage of the game, and to maximize the cloud service providers’ profit in the second stage of the game. The authors in [56, 78] formulated the problem as a strategic game in order to model the sharing of communication resources in a mobile cloud computing system that consists of multiple mobile users and a cloud server. In [56] the authors considered that devices may offload their tasks to the cloud through a single wireless link, and they provided a decentralized algorithm for computing a pure strategy Nash equilibrium of the game. In [78] they showed that the same algorithm can be used in the case of multiple identical wireless links.

Our work presented in Paper B [79] and Paper C [80] falls into this category of computation offloading problems. For each user $i \in \mathcal{N}$ we defined its cost as a linear combination of the task completion time and the energy consumption,

$\text{local execution: } C_l^i = \gamma_i^T T_{c,l}^i + \gamma_i^E E_{l}^i,$

$\text{offloading through AP } a: \quad C_{e,a}^i = \gamma_i^T T_{c,e}^i + \gamma_i^E E_{i,a}.$

We considered a mobile cloud computing system that consists of a set of APs, one mobile edge cloud and multiple mobile devices that compete for both communication and computational resources. In Paper B we considered the allocation of communication and computational resources, and in Paper C we integrated the resource allocation problem into the scheduling problem by allowing mobile devices not only to choose where to perform their tasks, but also in which time slot. We used $\mathcal{A}, |\mathcal{A}| = A$ to denote the set of APs, and $\mathcal{T}, |\mathcal{T}| = T$ to denote the set of time slots. We considered that each device $i \in \mathcal{N}$ can decide in which time slot it wants to perform the computation, and in the chosen time slot it can decide whether to perform the computation locally or to offload the computation to the cloud server through one of the APs. The problem can be formulated as the following 0−1 integer program

$$\begin{aligned}
\min_{X,Y} & \sum_{t \in T} \sum_{i \in \mathcal{N}} (y_{i,i,t} C_{i,l}^{d,t} + \sum_{a \in \mathcal{A}} (1 - y_{i,i,t}) x_{i,a,t} C_{i,a}^{e,t}(X,Y)) \\
\text{s.t.} & \sum_{t \in T} (y_{i,i,t} + \sum_{a \in \mathcal{A}} (1 - y_{i,i,t}) x_{i,a,t}) = 1, \forall i \in \mathcal{N}, \\
& X \in \{0,1\}^{N \times A \times T}, \\
& Y \in \{0,1\}^{N \times 2 \times T},
\end{aligned}$$

(4.10)
where the constraint (4.11) enforces that each device either performs the computation locally or it offloads the task through an AP in one of the time slots.

In Paper B and Paper C we considered that mobile devices are selfish, and we used game theoretical tools to analyze the problem in the case of a single time slot, and in the case of multiple time slots, respectively. We defined the problem as a player-specific network congestion game, for which the existence of equilibria is not known in general. We proved the existence of equilibrium allocations, and based on our constructive proofs we provided polynomial time decentralized algorithms for computing an equilibrium allocation. By providing constructive equilibrium existence proofs, and by characterizing the structure of an equilibrium allocation, our work presented in Paper B and Paper C is also important from a game theoretical perspective.

Multi-objective computation offloading problem: An interesting approach to investigate the Pareto optimal offloading decisions has been considered in [81, 82], where the authors used multi-objective optimization technique that involves two objective functions. The authors in [81] considered a sequence of tasks generated by a single mobile device that can decide which tasks to perform locally, and which tasks to offload to nearby cloudlets and remote centralized clouds. They proposed a multi-objective dynamic programming approach for making Pareto optimal offloading decisions for each of the tasks such that the energy consumption and the application completion time are minimized. The authors in [82] considered a mobile cloud computing system that consists of neighboring mobile devices that act not only as clients, but also as computational service providers. By considering neighboring mobile devices as service providers, they proposed a two-stage algorithm to select service providers such that the tasks’ completion time is minimized along minimizing the energy consumption.
Summary of Original Work

Paper A: Decentralized Algorithm for Randomized Task Allocation in Fog Computing Systems
Sladana Jošilo and György Dán
submitted to IEEE/ACM Transactions on Networking (ToN).

Summary: In this paper we consider a mobile cloud computing system where multiple devices may offload their computational tasks to each other or to a cloud server with the objective to improve their performance. We consider that devices are interested in minimizing the completion time of their own tasks, and we formulate the problem as a strategic game where each device plays a mixed strategy. We use variational inequality theory to prove the existence of an equilibrium task allocation in static mixed strategies, which we use to design an efficient algorithm for allocating the computational tasks in a decentralized way. The algorithm is based on average system parameters only, and thus it requires low signaling overhead. We perform simulations to evaluate the proposed algorithm, and we show that it achieves good system performance close to that of the greedy algorithm, which requires the full information about the system state, and close to that of an algorithm that allocates the tasks based on the socially optimal static mixed strategy.

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 analytical results concerning the existence of static mixed strategy equilibrium, and carried out the simulations. The analysis of the resulting data was carried out in collaboration with the second author of the paper. The paper was written in collaboration with the second author.
Paper B: A Game Theoretic Analysis of Selfish Mobile Computation Offloading

Sladana Jošilo and Győrgy Dán


**Summary:** In this paper we investigate the problem of resource allocation in a mobile cloud computing system that consists of multiple mobile devices, multiple APs and an edge cloud. We consider that mobile users are selfish, and thus they aim at minimizing their own cost, which we define as a linear combination of the time it takes to complete the computation and the corresponding energy consumption. In order to analyze interactions among mobile devices, we formulate the problem as a player-specific congestion game where users compete for communication and computational resources. We prove that a pure Nash equilibrium of the game exists, and we provide a polynomial complexity algorithm for computing it. We establish a bound on the price of anarchy of the game, and thus we show that the proposed algorithm has a bounded approximation ratio. We use extensive simulations to provide insight into the cost performance and the computational time of the proposed algorithm. We show that the proposed algorithm achieves the cost performance close to optimal cost performance, and the convergence time of the algorithm scales approximately linearly with the number of mobile devices.

**Contribution:** The author of this thesis developed the analytical model in collaboration with the second author of the paper, proved the analytical results for the case of both the elastic and non-elastic cloud models. The implementation of the simulations was carried out by the author of this thesis, and analysis of the resulting data was carried out in collaboration with the second author of the paper. The paper was written in collaboration with the second author.
Paper C: Decentralized Scheduling for Offloading of Periodic Tasks in Mobile Edge Computing

Sladana Jošilo and György Dán
in Proc. of IFIP Networking (NETWORKING), 2018.

Summary: In this paper we consider periodic computation offloading problem in a mobile cloud computing system that serves multiple wireless devices. Each device can choose in which of multiple time slots to perform the computation, and within the time slot it can choose to perform its task locally or to offload the task to a cloud server via one of multiple APs. The objective of each device is to minimize a linear combination of the time it takes to complete the computation and the corresponding energy consumption. We formulate the problem as a player specific congestion game, and based on a game theoretical treatment of the problem we prove the existence of a pure strategy Nash equilibria. Based on the constructive equilibrium existence proof, we characterize the structure of computed equilibria, and we develop an efficient decentralized algorithm for computing it. Our numerical results show that the proposed algorithm can be used to compute an efficient resource allocation at polynomial computational complexity despite combinatorial nature of the problem. Finally, the results show that the algorithm computes equilibria with good cost performance for various scenarios of a mobile cloud computing system.

Contribution: The author of this thesis developed the analytical model in collaboration with the second author of the paper. The second author proved the analytical results concerning the case of a single time slot, and the author of this thesis proved the analytical results concerning the case of multiple time slots. The implementation of the simulations and the analysis of the resulting data were carried out by the author of this thesis. The paper was written in collaboration with the second author.
Publications not included in the thesis


Conclusions and Future Work

In this thesis, we considered the computation offloading problem in mobile cloud computing systems. We analyzed the problem from the perspective of mobile cloud users and with this in mind we focused on the problem of allocating resources for various mobile cloud computing architectures. By using game theoretical tools we analyzed the interactions among mobile users, and we proposed efficient and scalable algorithms for allocating mobile cloud communication and computational resources.

In the first part of this thesis, we considered a highly distributed mobile cloud computing architecture, which allows users to use cloud resources and the resources of each other. We modeled the transmission and the execution of computational tasks using queuing theory, and provided a game theoretical formulation of the problem. We used variational inequality theory to address the question whether the users can compute an efficient equilibrium task allocation in static mixed strategies in a decentralized manner, under the assumption that every user knows only the average statistics on task arrival intensities, transmission rates, and task parameters.

In the second part of the thesis, we considered mobile edge cloud computing system where users can offload their computational tasks to an edge cloud. By assuming that users are selfish, we formulated the problem as a strategic game. We investigated whether there is an efficient decentralized algorithm for computing a pure Nash equilibrium of the game. Furthermore, we extended our model to consider offloading of periodic tasks in the case of homogeneous task periodicities. We addressed the question whether a pure Nash equilibrium exists if devices choose not only where to perform their tasks, but also in which time slot.

There are many open questions concerning the problem of resource allocation in mobile cloud computing system. The first interesting question is whether our results from the second part of the thesis can be extended to the case of heterogeneous tasks periodicities. The second interesting question is whether efficient decentralized algorithms exist for allocating mobile cloud resources in a system where the
Chapter 6. Conclusions and Future Work

actual number of users is not known, which would allow for less signaling between the mobile devices and the cloud/mobile network. Finally, the resource allocation problem could be analyzed not only from the perspective of users, but also from the perspective of mobile cloud service providers. Related to the last question, one could consider a mobile cloud computing system where mobile cloud service providers cooperate in serving users computational requests in order to improve their performance benefits.
Bibliography


