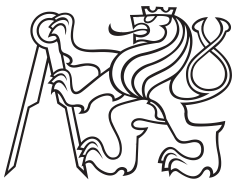
Bachelor Project



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Radioelectronics

Computation Offloading to Network's Edge in Environment with Dynamic Channel Quality Changes

Adam Jeník

Supervisor: Ing. Pavel Mach, Ph.D.
Field of study: Telecommunication engineering
Subfield: Mobile networks
January 2024

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Jeník Adam** Personal ID number: **503181**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Radioelectronics**

Study program: **Open Electronic Systems**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Computation Offloading to Network's Edge in Environment with Dynamic Channel Quality Changes

Bachelor’s thesis title in Czech:

Přenášení výpočtů na hranu sítě v prostředí s dynamickými změnami kvality rádiového kanálu

Guidelines:

This thesis focuses on Edge Computing paradigm enabling to offload computationally intensive tasks to a server located at the edge of the mobile network. First, study existing literature on the offloading decision problem determining where the tasks should be computed and the means of delivering tasks to the cloud. The conventional offloading decision algorithms assume that the channel quality of links used for offloading are always static. Nevertheless, in mobile networks the channel quality is never static and can change very quickly during the offloading process, e.g., if offloading user and/or relaying user helping with offloading change their location. Such a change in the quality of the connection during the tasks’ offloading can make existing offloading decision algorithms ineffective and performing poorly. Thus, design an algorithm for offloading decision that will cope with the channel quality changes during the offloading process. The main objective is to find a trade-off between the sum energy consumption over all users and the number of tasks completed within a required deadline. Demonstrate the gain of the proposed algorithm in terms of energy consumption and/or number of tasks completed within the deadline depending on the number and speed of devices, or other relevant parameters.

Bibliography / sources:

[1] P. Mach and Z. Becvar, „Mobile Edge Computing: A Survey on Architecture and Computation Offloading,“ IEEE Communications Surveys & Tutorials, vol. 19, no. 3, 2017.
[2] P. Mach and Z. Becvar, „Device-to-Device Relaying: Optimization, Performance Perspectives, and Open Challenges Towards 6G Networks,“ IEEE Communications Surveys & Tutorials, vol. 24, no. 3, 2022.

Name and workplace of bachelor’s thesis supervisor:

Ing. Pavel Mach, Ph.D. Department of Telecommunications Engineering FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **04.10.2023** Deadline for bachelor thesis submission: _____

Assignment valid until: **21.09.2025**

Ing. Pavel Mach, Ph.D.	doc. Ing. Stanislav Vítek, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

Acknowledgements

Chtěl bych tímto nesmírně poděkovat vedoucímu této práce panu Ing. Pavlu Machovi, Ph.D. za jeho cenné rady, za všechen čas, který věnoval konzultacím, a za jeho pozitivní a rovnocenný přístup, který mi po celou dobu byl velkou motivací. Nemalý dík patří také mé rodině a přítelkyni za to, že mě v (občasných) krušných chvílích vždy podrželi nad vodou.

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 9. January 2024

Abstract

In this thesis, we focus on offloading a computing task from user equipment (UE) to a multi-access edge (MEC) computing server. We assume an offloading case where the UEs are energy constrained devices, such as smartphones, unmanned aerial vehicles (UAVs) or IoT devices. We also assume the channel between the UEs and the BS is changing dynamically during the offloading. We first analyze a current state-of-the-art offloading optimization framework in an environment with dynamic channel quality changes and show that it is not usable, as only 50% of the UEs finish the offloading successfully. To this end, we propose a framework for the optimization of offloading the computing tasks to the MEC server in an environment with dynamically changing channel quality, which on average yields 99% of tasks successfully offloaded with only 16% growth in energy consumption compared to the current solution.

Keywords: multi-access edge computing, dynamic channel changes, mobility, energy constrained devices

Supervisor: Ing. Pavel Mach, Ph.D.
České vysoké učení technické v Praze - Fakulta elektrotechnická,
Technická 2,
Praha

Abstrakt

Tato práce je zaměřena na problém přenášení výpočetně náročných úloh z uživatelských zařízení na server umístěný na hraně sítě. Předpokládáme scénář, kdy mají uživatelská zařízení omezený přísun energie (například mobilní telefony nebo IoT zařízení). Předpokládáme také, že se během přenosu dat může dynamicky měnit kvalita spojení (například v důsledku pohybu uživatelských zařízení). Nejprve analyzujeme současné řešení problému a ukážeme, že v prostředí s dynamickými změnami v kvalitě spojení je toto řešení nepoužitelné, jelikož se stihne včas zpracovat pouze 50% přenášených úloh. Následně navrhneme řešení, které v průměru dosahuje 99% včas zpracovaných úloh, přičemž spotřeba energie oproti současnému řešení vzroste pouze o 16%.

Klíčová slova: přenášení výpočtů na hranu sítě, dynamické změny v kvalitě spojení, snížení spotřeby energie, mobilita

Překlad názvu: Přenášení výpočtů na hranu sítě v prostředí s dynamickými změnami kvality rádiového kanálu

Contents

Figures

Tables

Chapter 1

Introduction

Multi access-edge computing (MEC)([1][2][3][4]) is a relatively new concept in mobile networks. The idea is that a server is placed at the edge of the network which can receive computational tasks from other devices connected to the network (usually called user equipment or UEs). After the MEC server is done computing a task, it sends results back to the UE which requested the task. This approach brings many benefits, one of which is energy saving at the devices which utilize the MEC capability of the network. Offloading of demanding computation tasks to the MEC instead of processing them locally can be an effective way to save energy. This is especially relevant for energy constrained devices (f.e. smartphones, unmanned aerial vehicles (UAVs) or IoT devices).

The computational tasks are also often constrained by a deadline, after which the results need to be available at the UE. If we want to guarantee, that the results will be available within the given time constraint while also minimizing energy consumption at the UEs, making the decision of whether to offload or compute locally can be quite challenging. The key challenges being [1]: decision on the computation offloading ([5][6][7][8][9][10]), allocation of the computing resources ([11][12]) and guaranteeing service continuity if the UEs exploiting the MEC roam throughout the network ([13][14][15][16]).

A lot of energy constrained devices which could potentially benefit from utilizing MEC capabilities are devices, which are not static but moving. Movement of the UEs adds another layer of complexity to the offloading problem, because now the channel quality between the UE and the base station (BS) or relay can change rapidly during the offloading process due to shadowing or simply due to UE heading away from the BS.

While there exist many works on the topic of the optimization of the offloading to the MEC, the topic of mobility aware optimization is often omitted or left for future works. We have been able to find some mobility-aware approaches to the problem (i.e. [17], [18]). Nevertheless, in our research we have not discovered any general mobility aware solution for setting the optimal transmission power of an UE to finish the offloading within given time constraint. For example, even though the work in [5] minimizes energy consumption of the UEs in case they are all static, it would not work properly for dynamic scenarios with moving users. In such dynamic scenario, it would result in many tasks not offloaded in specified deadline, as we demonstrate later in this thesis.

Thus, the objective of this thesis is to find a solution that is able to manage offloading of tasks in a dynamic scenario with moving UEs. In particular, the goal is to maximize the number of tasks offloaded in time to MEC while still achieving low energy consumption as in [5].

The remainder of this paper is organized as follows: In chapter 2 we define the system model used in this work. Chapter 3 contains a precise formulation of the problem this work is aiming to solve. In chapter 4 we introduce a current state-of-the-art optimization framework and analyze its performance in an environment with dynamic channel changes. We then further examine the trade-off between the number of successfully offloaded tasks and the energy consumption in chapter 5. We propose a sub-optimal solution to the problem in 6. In the same chapter follows also the performance analysis of the proposed solution. Conclusion of this work is contained in 7 jointly with suggestions for future work.

Chapter 2

System model

This chapter contains an overview of the system model used in this thesis, including network model, communication and computing models, and channel model. Individual models are described in the following subsections.

2.1 Network model

We consider a scenario with one powerful MEC server located at the BS. We further assume N UEs distributed over an area without any obstacles obstructing the line-of-sight (LOS) to the BS. The UEs generate computationally demanding tasks. Each task has a time constraint, after which the computation results need to be available at the UE. Each task can be either offloaded to the MEC server or computed locally at the UE. In the current model, the offloading to the MEC server is done directly without help of any relays. The extension to the relays is relegated to potential future works.

2.2 Communication model

This section first describes communication model for static scenario with users not moving. Further, communication model for dynamic scenario is tackled as well.

2.2.1 Static communication model

We compute channel capacity C for any UE offloading its task according to Shannon ([5]) as:

$$C = b \cdot \log_2 (1 + \frac{p \cdot g}{b \cdot (I_b + \sigma)}) \tag{2.1}$$

where b, p and g are the allocated bandwidth, transmission power and channel gain between the UE and the BS respectively, σ is the noise spectral density and I_b is the background interference.

The size of the data $D(t)$ offloaded after transmitting over the duration of time t can be expressed as:

$$D(t) = C \cdot t \tag{2.2}$$

The transmission delay t_d for offloading a task to the MEC server can therefore be expressed as:

$$t_d = \frac{D_t}{C} \tag{2.3}$$

where D_t is the size of the offloaded task.

2.2.2 Dynamic communication model

For dynamic communication model, we make an assumption, that the only time dependent quantity affecting the channel capacity is the gain between the BS and the UE (allocated power and bandwith, background interference and the noise spectral density remain constant over the duration of the offloading). The time-dependent channel capacity $C(\tau)$ can be expressed as:

$$C(\tau) = b \cdot \log_2 (1 + \frac{p \cdot g(\tau)}{b \cdot (I_b + \sigma)}) \tag{2.4}$$

where $g(\tau)$ is the time dependent channel gain between the UE and the BS. The size of the data $D(t)$ offloaded after transmitting over the duration of time t can be expressed as:

$$D(t) = \int_0^t C(\tau) \cdot d\tau \tag{2.5}$$

2.3 Computing model

This section describes the individual parameters of each task. In particular, any offloading task has the following parameters:

- D_t ... size of the task
- T_{max} ... time constraint placed on the task
- c ... average number of CPU cycles needed to process one bit of the task

Then, the task can be either offloaded to the MEC server or computed locally. In case of the offloading, the time t_{MEC} needed by the MEC server to compute the task is obtained as:

$$T_{MEC} = \frac{D_t \cdot c}{F_{MEC}} \tag{2.6}$$

where F_{MEC} is the number of CPU cycles the MEC server is able to process per second. The time T_{local} to compute the task locally is obtained as:

$$T_{local} = \frac{D_t \cdot c}{F_{UE}} \tag{2.7}$$

where F_{UE} is the number of cycles the CPU of the UE is able to process per second.

The energy E_{off} consumed by an UE offloading its task to the MEC server is expressed as:

$$E_{off} = (T_{max} - T_{MEC}) \cdot p \tag{2.8}$$

where p is the UE's transmission power. Then, the energy E_{local} consumed by an UE computing its task locally is expressed as [19]:

$$E_{local} = E_c \cdot D_t \cdot c \tag{2.9}$$

where E_c is the energy consumption at the UE to run one CPU cycle. In this work, we do not focus on the energy consumed by processing the tasks at the MEC server.

2.4 Channel model

We base our channel model on the following widely used formula [20]. According to this channel model, the received power P_{rx} at the BS can be expressed as:

$$P_{rx} = P_{tx} \cdot g(\mathbf{c}) \cdot \Psi(\mathbf{c}) \cdot \|h(\mathbf{c})\|^2 \tag{2.10}$$

where P_{tx} is the transmission power of the UE, $g(\mathbf{c})$ is the channel gain between the UE and the BS dependent on UE coordinates $\mathbf{c}$, $\Psi(\mathbf{c})$ is the location dependent shadowing (also known as slow fading) and $h(\mathbf{c})$ is the small scale fading (also known as fast fading). We assume the small scale fading is going to be average out over the offloading time period [20][21]. We also assume the UEs have LOS to the BS at any location (there are no obstacles in the area), therefore no location dependent shadowing is present. We can simplify the formula 2.10 as:

$$P_{rx} = P_{tx} \cdot g(\mathbf{c}) \qquad \text{[W]} \tag{2.11}$$

or in alternative notation:

$$P_{rx} = 30 + 10 \cdot \log_{10}(P_{tx}) - PL(\mathbf{c}) \qquad \text{[dBm]} \tag{2.12}$$

where $PL(\mathbf{c})$ is the location dependent path-loss.

The path-loss and gain are two different ways to describe the same quantity and the conversion between them is done according to the formula:

$$PL(\mathbf{c}) = -10 \cdot \log_{10}\left(g(\mathbf{c})\right) \qquad [\text{dB}] \quad (2.13)$$

The channel gain and path-loss represent deterministic decay of the transmitted signal over a distance. They depend solely on the distance between the UE and the BS and can be expressed by deterministic mathematical models.

Chapter 3

Problem formulation

We formulate a resource allocation problem to minimize the sum energy consumption over UEs consumed by offloading of the tasks from the UEs directly to the BS, while ensuring that the probability P_{fail} that each task fails to finish the offloading within its time constraint is not going to exceed an arbitrary upper limit $P_L \in [0, 1]$. This is achieved by joint optimization of the transmission time and power at all UEs. The problem can in general formuted as:

$$\begin{aligned} \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{P}} = \min & \sum_n E_n \\ \text{s.t.} \quad & \text{(b) } P_{fail} = P(T_{off} > (T_{max} - T_{MEC})) \leq P_L \\ & \text{(b) } T_{off} > 0 \\ & \text{(c) } p_n \leq p_{max} \end{aligned} \tag{3.1}$$

where (a) ensures that the probability each task is not processed within T_{max} is lower or equal to an arbitrary upper limit P_L, (b) ensures that the offloading time T_{off} is going to be a positive number and (c) limits the transmission power to p_{max}.

Similarly as in many works (e.g. [5], [22]), we don't take the time nor the energy consumed to transmit the results from the BS back to the UE into account. That is because firstly, the amount of the data is usually much smaller than the task itself and secondly because it would have no impact on the results as it can be counted in to the time for processing the task at the MEC server.

Chapter 4

Performance analysis of current solution

In this chapter, we are going to analyze first the performance of current state-of-the-art offloading solution, initially tailored for static environment, in an environment with dynamic channel quality changes.

4.1 Offloading model

We base the offloading model on the current state-of-the-art solution for optimizing the offloading energy consumption proposed in paper [5]. Without loss of generality, we assume only one hop between the UE and the BS (the data is going be transmitted directly without help of any relays – direct offloading). According to [5], we do not take the changes of channel quality during the offloading process into account. Therefore, we make our decision based on initial channel gain g_i between UE and BS at the moment of making the offloading decision.

Time available for the offloading of the task to the MEC server is obtained by subtracting the time needed to process the task at the MEC server from the tasks time constraint:

$$T_{off} = T_{max} - T_{MEC} \tag{4.1}$$

where T_{off} is the time available for the offloading, T_{max} is the time constraint for the task and T_{MEC} is the time to process the task at the MEC server (computed according to the formula 2.6).

For static channel model we can substitute the constant capacity according to equation 2.3. Thus, we set the transmission power p of an UE as:

$$p = \frac{b \cdot (I_b + \sigma)}{g} \cdot \left(2^{\frac{D}{T_{off} \cdot b}} - 1\right) \tag{4.2}$$

The power remains constant during the transmission, therefore the energy consumed by offloading E can be obtained from the formula:

$$E = T_{off} \cdot p = \frac{T_{off} \cdot b \cdot (I_b + \sigma)}{g} \cdot \left(2^{\frac{D}{T_{off} \cdot b}} - 1\right) \tag{4.3}$$

In the following section, we analyze what happens if the optimization for static scenario is exploited for dynamic scenario, which is inherent to mobile networks.

4.2 Performance metrics

In this section, we introduce the performance metrics used to evaluate the performance of the current offloading solution.

4.2.1 Offloading success rate

Offloading success rate demonstrates the probability with which we can offload the task within its time constraint if we choose a given offloading model. *The tasks which are impossible to offload within the time constraint without exceeding the maximum limit for transmission power p_{max}, even for the static case, are not counted into this metric.*

Offloading success rate R_s in percent is computed as:

$$R_s = 100 \cdot \frac{N_{success}}{N_{total} - N_{impossible}} \tag{4.4}$$

where $N_{success}$ is the number of tasks offloaded within the time constraint, N_{total} is the total number of all tasks and $N_{impossible}$ is the number of the tasks which which are impossible to offload within the time constraint without exceeding the maximum limit for transmission power, even if the UEs would be static.

4.2.2 Average energy consumed by offloading

Average energy consumed by offloading demonstrates the average energy consumed per UE if we decide to offload all the tasks to the MEC server. *The tasks which are impossible to offload within the time constraint without exceeding the maximum limit for transmission power p_{max} are not counted into this metric.*

This metric is useful for analyzing the energy consumption comparisons between different optimization solutions without being distorted by specific setting of maximum transmission power p_{max} and the cost of the local computing.

Average energy E_{off} consumed by the offloading is computed according to the following formula 4.5 as a sum of energy consumed at all tasks divided by the total number of the tasks.

$$E_{off} = \frac{E_{all}}{N_{total} - N_{impossible}} \tag{4.5}$$

where E_{all} is the sum energy consumed at all UEs which do not exceed p_{max} (computed according to the formula 4.3), N_{total} is the total number of all tasks and $N_{impossible}$ is the number of task which which are impossible to offload within the time constraint without exceeding the maximum limit for transmission power.

4.2.3 Average energy consumed in total

Average energy consumed in total demonstrates the average energy consumption per UE if we make a realistic decision whether to compute locally or to offload the task to the MEC server. This energy includes both the energy consumed by local computing and the energy consumed by offloading (computed according to the formulas 2.9 and 4.3 respectively).

This metric is useful for obtaining a more real-world applicable energy consumption of an optimization solution than the previous metric.

The algorithm for making the offloading decision is demonstrated in Algorithm 1. The input arguments p, p_{max}, T_{max}, T_{local}, E_{local} are in this respective order: offloading transmission power, maximum transmission power of the UEs, time constraint for the task, time needed to compute the task locally and energy consumed by computing the task locally. The outputs of the algorithm $[offload, compute\_locally]$ are boolean values which decide whether to either i) offload the task to the MEC server, ii) compute the task locally, or iii) do neither of the two. The last option is chosen only when the task is both impossible to offload within the time constraint without exceeding p_{max} and simultaneously impossible to compute locally within the time constraint.

The average energy consumed in total E_{total} is computed according to the formula 4.6 as a sum of the energy consumed by local computing and by offloading divided by the total number of possible tasks:

$$E_{total} = \frac{E_{local} + E_{off}}{N_{total} - N_{impossible}} \tag{4.6}$$

Algorithm 1 Algorithm for offloading decision

function MAKE_OFFLOADING_DECISSION(p, p_{max}, T_{max}, T_{local}, E_{local})→ $[offload, compute\_locally]$
  $offload = false$
  $compute\_localy = false$
  if $p \leq p_{max}$ **then**
    $E_{offload} = p \cdot t$
    if $E_{local} > E_{offload}$ **then**
      $offload = true$
    else
      if $T_{local} \leq T_{max}$ **then**
        $compute\_localy = true$
      end if
    end if
  else
    if $T_{local} \leq T_{max}$ **then**
      $compute\_localy = true$
    end if
  end if
end function

where E_{local} and E_{off} are the energies consumed by the local computing and the offloading of the tasks to the MEC server respectively, N_{total} is the total number of all tasks and $N_{impossible}$ is the number of the tasks which are both impossible to offload and compute locally within the time constraint.

4.3 Simulation setup

In this section, we describe the simulation scenario, UE movement model, and the settings we have used for the simulations in Matlab.

4.3.1 UE mobility model

The UEs move according to the wrap-around random direction movement model [23][24] with a following slight modification: the movement speed is not chosen randomly at each UE, but remains constant across all UEs. That is because we want to analyze how different parameters and solutions depend on the movement speed of the UEs. Now follows a brief description of the

model:

- At the beginning of the simulation, each UE randomly chooses a movement direction $\mathbf{d} \in \mathbb{R}^2, \|\mathbf{d}\| = 1$ and a duration in seconds. The UEs then start moving in the directions they have chosen for the chosen duration.
- After an UE finishes moving, we go back to step 1 (it chooses new movement direction and duration and starts a new movement).
- If an UE hits the boundary of the simulation area, it instantly reappears at the opposite side.

All important movement model parameters can be found in table 4.1.

Parameter	Value
New direction time	uniform distribution, $U(0, 20)$ seconds
New direction distribution	uniform

Table 4.1: UE movement model parameters.

4.3.2 Simulation scenario and settings

In this section we outline the simulation scenario and settings. For simulations in Matlab we adopt a rural scenario with no buildings. The BS is located at the center of the area at coordinates $[250; 250]$. The initial positions of the UEs are randomly generated uniformly over the area. The BS splits the available bandwith between all the UEs equally.

The offloading decision is made at the beginning of the simulation. The UEs then start moving at a constant speed according to movement model described in previous chapter 4.3.1, whilst simultaneously offloading their tasks to the BS. The channels between the UEs and the BS are obtained according to the model described in 2.2.1.

The simulation moves forward in time by time-steps. The size of the time-step T_{step} is equal to the total offloading time interval T_{off} uniformly divided into N equal steps. The channel stays constant for the duration of

each time-step, thus the size of the data offloaded by j-th UE in i-th time step is equal to:

$$D_{ji} = C_j(T_{step,j} \cdot i) \cdot T_{step,j} \tag{4.7}$$

where $T_{step,j}$ is length of j-th UE's time-step and $C_j(t)$ is the channel capacity between the j-th UE and the BS dependent on time.

All important movement model parameters can be found in table 4.2.

Parameter	Value	Parameter	Value
Area size	500x500m	B	100MHz
BS coord.	[250,250,35]	p_{max}	23dBm
Carrier freq.	2Ghz	D	[0.2 5]Mbits
$\sigma + I_b$	-150 dB	c	$[1.5, 2]$x10^3 cyc./bit
F_{UE}	$[1.5, 2]$x10^9 cyc./s	F_{BS}	40x10^9 cyc./s
E_c	[0 20]x10^{-11}	Time steps	100
Numb. of UEs	100	Numb. of drops	30

Table 4.2: Simulation parameters.

4.3.3 Channel path-loss model

For the specific path-loss model, we adopt general modified COST 231 Hata path-loss model at 2GHz.

4.4 Success rate of the current solution

This section analyzes the succes rate of the current state-of-the-art solution. We can see the resulting success rate dependent on UE speed and offloading time constraint in figure 4.1. Following the intuition only 50% of the tasks were offloaded within the time constraint (channel quality in our model depends solely on the distance from the BS – half of the UEs are going to move towards the BS and the other half is going to move away).

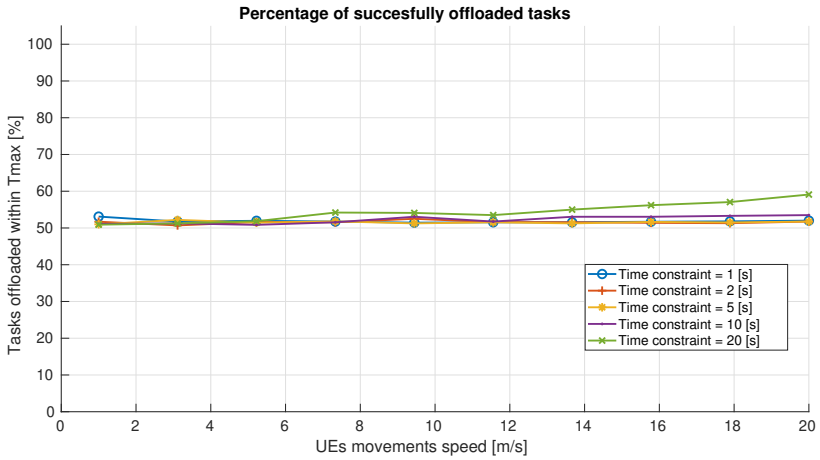


Figure 4.1: Success rate of static channel offloading model depending on UE movement speed and time constraint for the offloaded tasks.

We can also see, that for 20 second time constraint and movement speeds over 5 meters per seconds, the success percentage is starting to grow. We conclude that with growing UE trajectory lengths (higher speeds and time constraints), the improvements in channel capacity start to slowly outweigh the worsening. However, for realistic movement parameters and time constraints, the observed improvements in offloading success rate are too small to be useful.

4.5 Energy consumption of the current solution

We can see the average energy consumed by offloading in figure 4.2 and the average energy consumed (total) in figure 4.3. The energy grows with decreasing offloading time according to the formula 4.3.

We see that the slight improvements in success rate for long trajectory lengths (observed in previous section 4.4) also impact the energy, as it is slightly decreasing for the UEs which travel for longer distances. That is because energy follows the formula $E = p \cdot t$ and the number of UEs who finish the offloading before the time constraint grows. Thus the average offloading time decreases as well as average energy consumed.

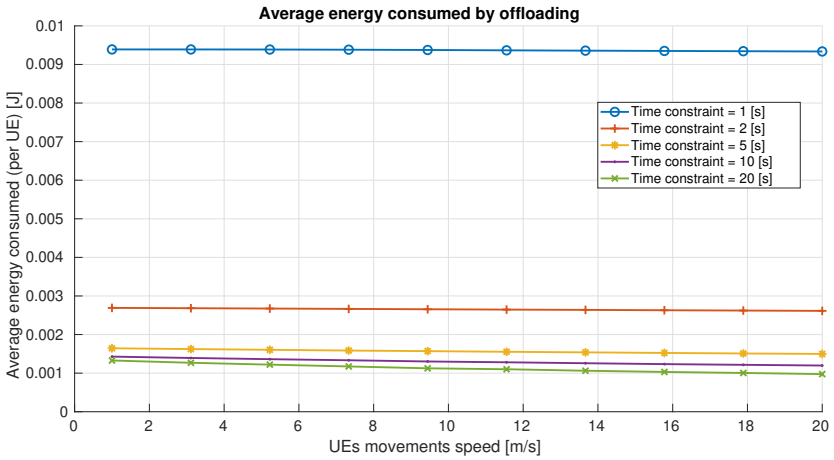


Figure 4.2: Average energy consumed per UE by offloading with static channel offloading model depending on UE movement speed and time constraint for the offloaded tasks.

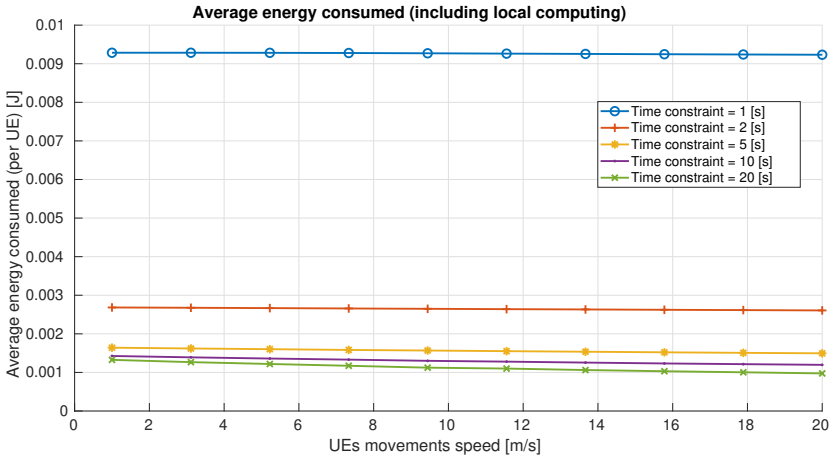


Figure 4.3: Average energy consumed per UE with static channel offloading model (including local computing) depending on UE movement speed and time constraint for the offloaded tasks.

4.6 Conclusion

We found out that when the current solution is used for dynamically changing channel, only half of the offloaded tasks are completed within their respective time constraints. This solution is therefore not usable for dynamically changing channel.

Chapter 5

Closer inspection of the trade-off

In this chapter, we propose a simple method addressing the problem formulated in the previous section. This method should allow us to closely examine the trade-off between the success rate and energy consumption.

The basic principle of the proposed method is shortening of the offloading time to have some safety margin if the UEs are moving. Then, even if the UEs are moving, the transmission power is set in such way that the task is offloading before its current deadline and even if the channel gain is decreasing during offloading itself, we are able to increase the number of offloaded tasks in the required time deadline accordingly.

We propose to lower the original offloading time as follows:

$$T_{lowered} = T_{max} \cdot (1 - \Delta_t) \tag{5.1}$$

where Δ_t represents the margin by which T_{max} is shortened. Please note that the offloading decision is identical to decision described in chapter 4, only difference being that we substitute T_{max} for $T_{lowered}$.

Of course, the proposed shortening of T_{max} also leads to some increase in the energy consumption during the offloading, since longer offloading time results in lower energy consumption at the side of the UEs, and vice versa. Thus, the setting of Δ_t should be carefully optimized to obtain a reasonable trade-off between the number of the tasks successfully offloaded and energy consumption of the offloading. The analysis is delivered in the next subsections.

5.1 Performance analysis of proposal

Note that for the performance analysis of Δ_t setting, we adopt the same performance metrics and simulation setup as in chapter 4. Now let's analyze again the success rate and energy consumption.

5.1.1 Success rate

Figures 5.1-5.3 demonstrate the percentage of the tasks offloaded within T_{max} depending on Δ_t if T_{max} is set as 1 second, 5 second, and 20 seconds, respectively. Notice that if $\Delta_t = 0$, the situation copy the current state of

the art as offloading time is equal to T_{max}. We can see the resulting success rates dependent on relative time reserve for each particular T_{max}.

With high enough time reserve, we are able to compensate for the channel changes and eventually get to 100 percent of the tasks offloaded within the time constraint. It is visible, that with growing time constraints and UE movement speeds, we need to increase the time reserve to compensate for the channel changes. For example, for UE moving at 1 meter per second and the time constraint 5s, mere 5 percent of time reserve will result in almost 100 percent success rate. On the other hand, for movement speed of 20 meters per second and the time constraint 20 seconds, we need 80 percent of time reserve to achieve a comparable success rate.

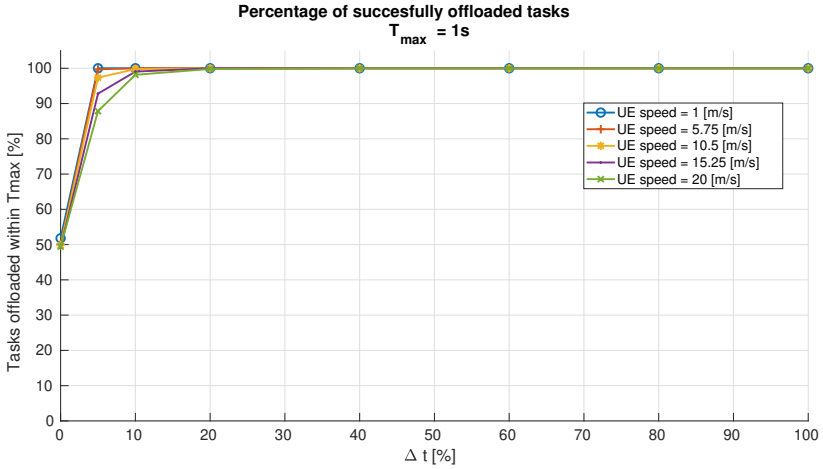


Figure 5.1: Success rate of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 1 second.

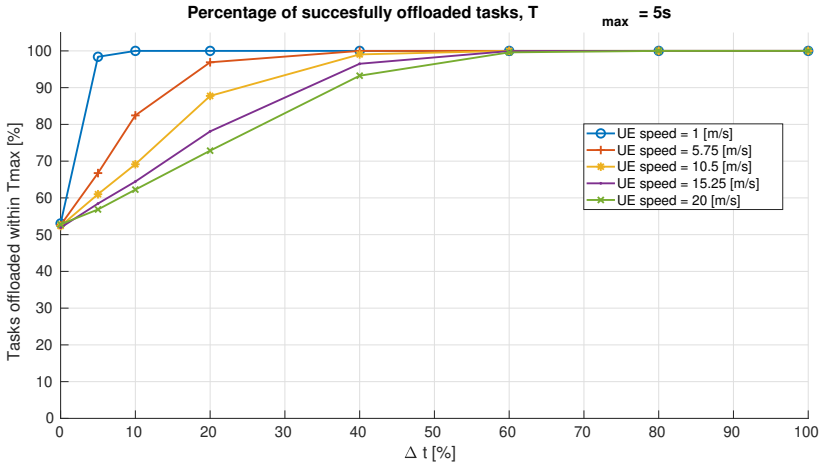


Figure 5.2: Success rate of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 5 seconds.

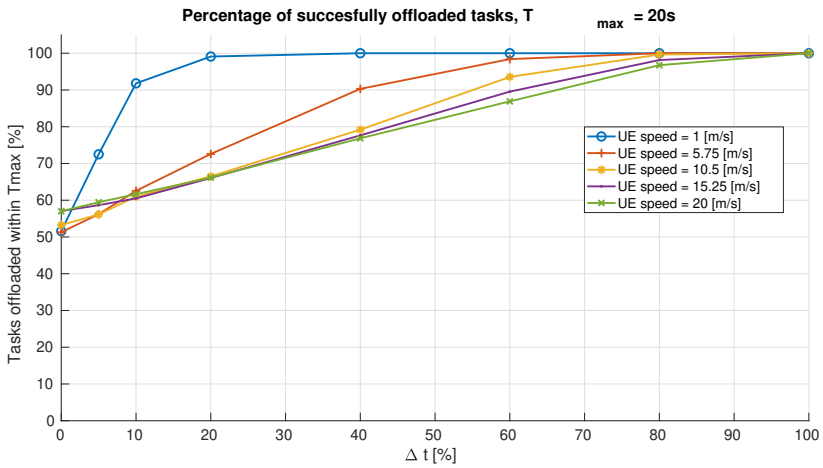


Figure 5.3: Success rate of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 20 seconds.

This result is consistent with our intuition - the channel between the UE end the BS changes more rapidly when the UE travels across longer trajectory during the offloading. Thus, with growing movement speeds and time constraints, we need to increase the time reserve to compensate for the dynamic changes of the channel.

5.1.2 Energy consumed

In previous section, we have demonstrated that even simple setting of Δ_t to a reasonable value can result in almost 100% success of offloaded tasks. This section now answers what is the cost in terms of energy consumption. We can see the energy consumed by offloading dependent on relative time reserve for the time constraints of 1 second, 5 seconds and 20 seconds in figures 5.4, 5.5 and 5.6 respectively. Furhter, the total energy consumed, considering also local computing, and dependent on relative time reserve for the time constraints of 1 second, 5 seconds and 20 seconds in figures 5.7, 5.8 and 5.9 respectively. The energy again grows with decreasing offloading time according to the formula 4.3.

For a better illustration of the success rate - energy trade-off, we include a comparison table of "optimal" trade-off for different time constraints (table 5.1). For each time constraint we have chosen the smallest value of Δ_t so that the resulting success rate is over 90 percent for any UE movement speed. We can observe that even though for 20 second time constraint we need much greater Δ_t than for 1 second (80% versus 10%), the relative increase

in consumed energy is quite comparable. Furthermore, the 20 second time constraint has much lower overall energy consumption than 1 second, so the 40% energy increase is much less "punishing". In overall, we can say that the increase of successfully offloaded tasks easilly outweights the incereased energy consumption (please note that in case state-of-the-art solution, only roughly 50% of tasks are offloading in time).

T_{max} [s]	Δ_t [%]	**Min. success rate** [%]	**Max. energy increase** [%]
1	10	98	35
5	40	93	26
20	80	97	40

Table 5.1: Comparison of the "optimal" trade-off for each time constraint versus the current state-of-the-art solution.

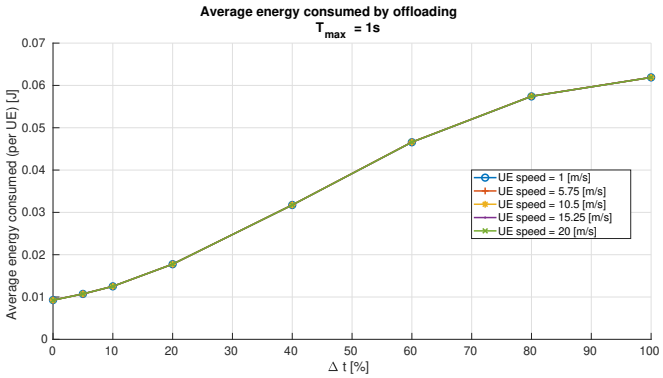


Figure 5.4: Average energy consumption (per UE) of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 1 second. Only the energy consumed by the offloading.

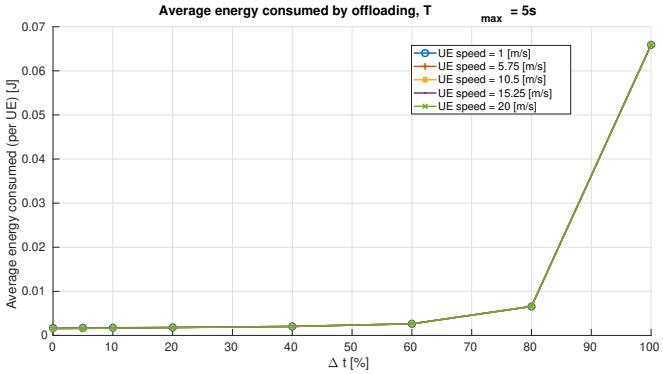


Figure 5.5: Average energy consumption (per UE) of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 5 seconds. Only the energy consumed by the offloading.

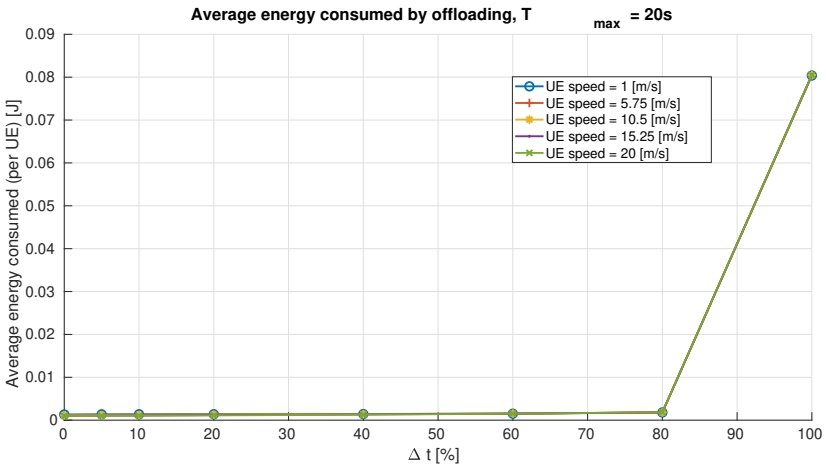


Figure 5.6: Average energy consumption (per UE) of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 20 seconds. Only the energy consumed by the offloading.

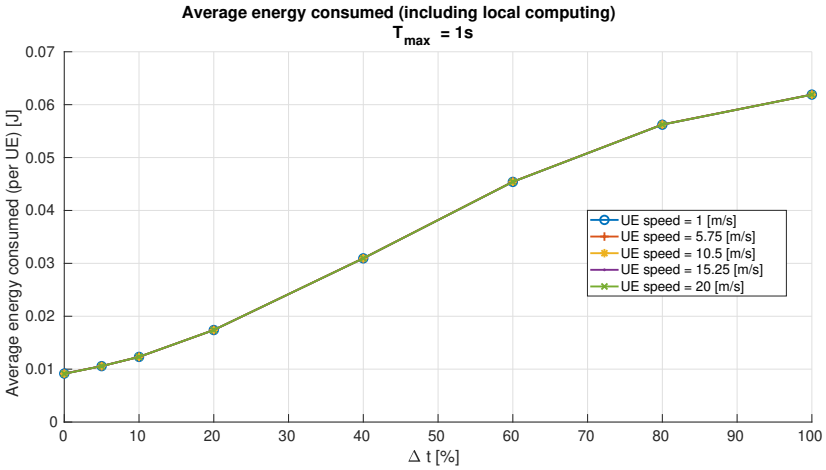


Figure 5.7: Average energy consumption (per UE) of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 1 second. Total energy including local computing at the UEs.

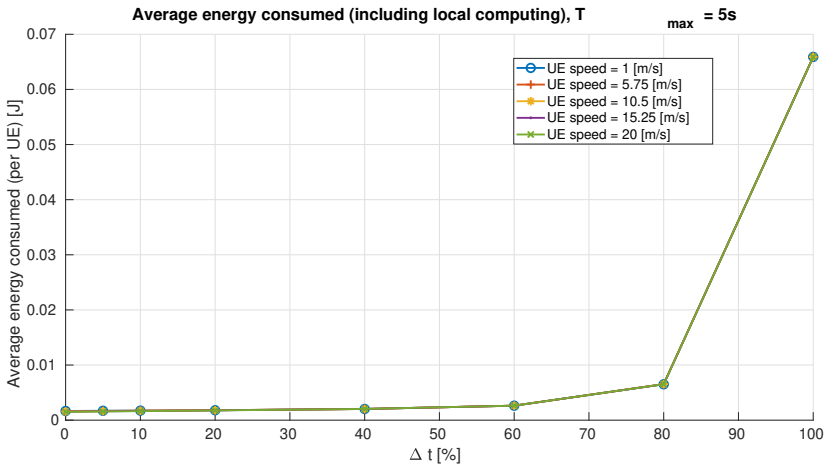


Figure 5.8: Average energy consumption (per UE) of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 5 seconds. Total energy including local computing at the UEs.

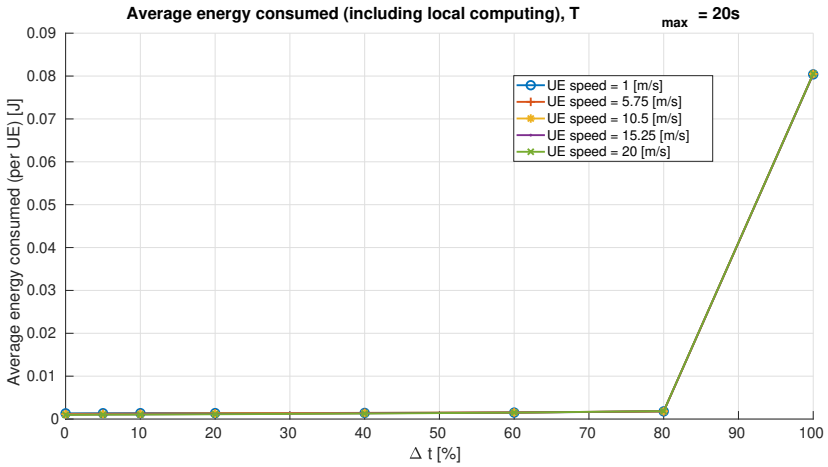


Figure 5.9: Average energy consumption (per UE) of the static channel offloading model improved by time reserve, depending on the relative time reserve Δ_t and displayed for various UE movement speeds and a fixed time constraint of 20 seconds. Total energy including local computing at the UEs.

5.2 Conclusion

There is a visible trade-off between success rate and consumed energy. Both the success rate and the energy consumed are growing with increasing time reserve. But there is no "universally optimal" time reserve, as the trade-off between the success rate and consumed energy differs for every combination of UE movement speed and time constraint.

An important lesson can be taken from this chapter: the trade-off between the success rate and the consumed energy differs for different movement speeds and time constraints. We can take this conclusion one level further: The channel changes in our model depend solely on UE movement (see 2.4, 2.2.2) - therefore, the solution to the problem of finding the optimal trade-off between the consumed energy and offloading success rate has to be built upon the parameters of the UE movement.

Chapter 6

Offloading optimization using prediction of future movement

We have demonstrated in the previous chapter that even simple shortening of offloading time can achieve significant improvements in terms of successfully offloaded tasks at the cost of only marginally increased energy consumption. Still, the proposed method sets the Δ_t for all UEs disregarding their current moving direction or potential changes of moving direction during the offloading. Thus, in this chapter, our aim is to further improve the proposed solution to get even more promising results.

To this end, we propose a sub-optimal solution to the problem (defined in 3). First, we start by deriving a formula for making the offloading decision based on predicted future values of channel gain (across the UE's future trajectory). Then we introduce a very simple UE channel gain prediction framework. This allows us to make to make an offloading decision, which takes dynamicity of the channel into account.

We derive the proposed solution in following steps:

1. We propose a method for setting the optimal offloading transmission power at the UE supposing we had the information about the future values of channel gain to the BS across the whole offloading time period (6.1).
2. We propose a sub-optimal solution for the prediction of the UE's future channel gain values to the BS across the offloading time period (6.2).

6.1 Optimal transmission power

We are now going to derive a formula, which takes the future (predicted) values of the channel gain between the UE and the BS as input parameters and which yields us the optimal transmission power needed to finish the offloading within the time constraint.

According to the dynamic channel model described in 2.2.2 we know, that size of offloaded data can be obtained from the following equation:

$$D = \int_0^t C(\tau) \cdot d\tau = \int_0^t b_n \cdot \log_2 (1 + \frac{p \cdot g(\tau)}{b \cdot (I_b + \sigma)}) \cdot d\tau \tag{6.1}$$

To derive an analytic formula, we adopt a widely used ([25], [26]) approximation that $log_2(1 + \gamma) \approx log_2(\gamma)$, where $\gamma = (p \cdot g(\tau))/(b \cdot (I + \sigma))$ is the

signal-to-interference-plus-noise ratio (SINR). This approximation is generally valid under the condition that $\gamma >> 1$, thus it is often called "high SINR regime". In realistic offloading scenarios, we transmit relatively large amounts of data under short time constraints (often in real time) and the SINR is therefore high enough to make this approximation valid.

We can now rewrite the integral as:

$$D \approx \int_0^t b_n \cdot \log_2 \left(\frac{p \cdot g(\tau)}{b \cdot (I_b + \sigma)}\right) \cdot d\tau \tag{6.2}$$

And derive the formula for the transmission power (as in chapter 4.1).

$$p = b \cdot (I_b + \sigma) \cdot 2^{\frac{D}{t \cdot b} - \frac{1}{t} \cdot \int_0^t log_2(g(\tau)) \cdot d\tau} \tag{6.3}$$

Further we split the exponential for $\frac{D}{t \cdot b}$ and $-\frac{1}{t} \cdot \int_0^t log_2(g(\tau)) \cdot d\tau$:

$$p = \frac{b \cdot (I_b + \sigma)}{2^{\frac{1}{t} \cdot \int_0^t log_2(g(\tau)) \cdot d\tau}} \cdot 2^{\frac{D}{t \cdot b}} \tag{6.4}$$

If we compare this result with the formula for optimal offloading power derived under the approximation $log_2(1 + \gamma) \approx log_2(\gamma)$ (assuming static channel capacity $C = D/t$), which is:

$$p = \frac{b \cdot (I_b + \sigma)}{g} \cdot 2^{\frac{D}{t \cdot b}} \tag{6.5}$$

We see that in the divisor of the fraction in formula 6.4 we have obtained an equivalent of the static channel gain. And the channel gain in this case is equal to geometric mean of the future gain values.

$$g_m = 2^{\frac{1}{t} \cdot \int_0^t log_2(g(\tau)) \cdot d\tau} = GM(g) \tag{6.6}$$

This result can be also rewritten into summary form for obtaining the prediction of channel gain from future channel gain measurements (taken uniformly over an time window):

$$g_m = 2^{\frac{1}{t} \cdot \sum_{i=1}^{n} log_2(g_i) \cdot t_{step}} = \sqrt[n]{\prod_{i=1}^{n} g_i} = GM[g] \tag{6.7}$$

where g_i is the i-th channel gain measurement and $t_{step} = t/n$ is the time-step between the individual measurements. We now have the tools to make the offloading decision for dynamic channel based on future (predicted) values of gain across the UE's future trajectory.

We then insert the mean channel gain g_m obtained according to the formulas 6.6, 6.7 into the formula for setting the offloading power p without the approximation $\gamma >> 1$ (4.2) described in the previous chapter 4.1:

$$p = \frac{b \cdot (I_b + \sigma)}{g_m} \cdot \left(2^{\frac{D}{T_{off} \cdot b}} - 1\right) \tag{6.8}$$

where g_m is the mean channel gain between the UE and the BS over the offloading period.

6.2 Prediction of future channel gain to BS

We are now going to present a sub-optimal solution to obtain future channel gain values for making the offloading decision. Because this work does not take shadowing into account, the channel gain depends solely on the distance between the UE and the BS and can be expressed by deterministic mathematical models. Thus to obtain the information about the UE's channel gain to the BS at specific time, all we need to know is the UE's location at that time. The channel gain prediction problem is therefore transformed into one of predicting trajectory. We chose to divide the trajectory prediction problem into two sub-problems.

We assume we have access to the information about geographical location, UE's movement speed and direction of movement at the time of making the offloading decision. Note that UE's location can be easily obtained by GPS or even by mobile network itself depending on channel qualities to several BSs. From this information, we are able to make predictions about the UE's future trajectory. We further assume, that the predicted trajectory is going to deviate from the real one more the farther we are from the starting point. In other words, the location prediction error is going to grow with time. If we were to choose an arbitrary lower limit for location prediction error, after which the prediction is not usable for making the offloading decision, there might exist future point in time T_{change} in which the prediction error is going to drop under the chosen limit. It is therefore beneficial to divide the offloading time interval into two parts, and solve the problem for each part individually:

- **Sufficent trajectory prediction**
  - $t = [T_{start}, T_{change}]$
  - We have a "good enough" UE trajectory prediction at hand.
- **Uncertain trajectory**
  - $t = [T_{change}, T_{end}]$,
  - We do not have any sufficient prediction of UE future trajectory.

This approach is illustrated in figure 6.1.

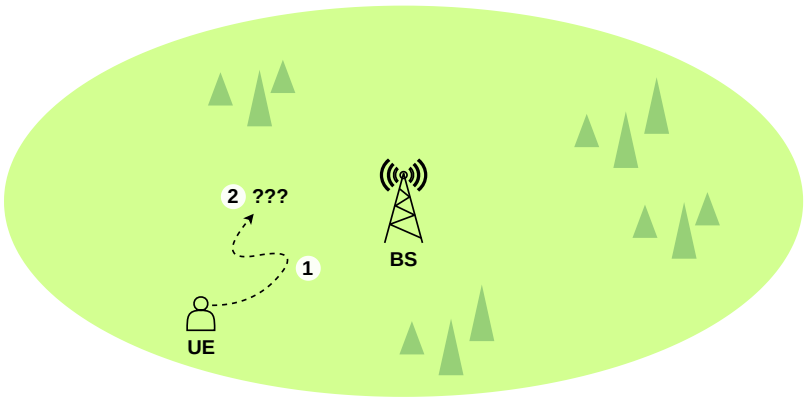


Figure 6.1: Illustration of the problem. For the first part of the offloading interval (1), we have a sufficent prediction of the UEs future trajectory. For the second part (2) we have no relevant information about the trajectory.

6.2.1 Sufficent trajectory prediction

In this part we propose a simple UE trajectory prediction model. We look at the UE's current geographical location, movement speed and movement direction. We then assume that the UE will continue moving in the same direction until T_{change}. The UE's future trajectory can be than obtained from the equation:

$$\mathbf{c_{UE}} = \mathbf{c_{UE,start}} + \alpha \cdot v_{UE} \cdot t_{change} \cdot \mathbf{d_{UE}}, \forall \alpha \in [0; 1] \tag{6.9}$$

where $\mathbf{c_{UE}}$ are UEs coordinates on Cartesian plane, $\mathbf{c_{UE,start}}$ is UEs starting position and $\mathbf{d_{UE}}$ is UEs movement direction ($\|\mathbf{d_{UE}}\| = 1$). The prediction of channel gain can be then computed as line integral of first type:

$$g = 2^{\frac{1}{\|l\|} \cdot \int_l log_2(g(\|\mathbf{c_{UE}} - \mathbf{c_{BS}}\|)) \cdot dl} \tag{6.10}$$

where $g(\|\mathbf{c_{UE}} - \mathbf{c_{BS}}\|) = g(d)$ is channel gain between the UE and the BS dependent on UEs distance from the BS, l is UEs trajectory defined in 6.9 and $\|l\|$ is the trajectory length.

6.2.2 Uncertain trajectory

In case we do not have any sufficient prediction of the UE's trajectory, we do not have many options. Because we need to achieve a 100% success percentage, we decided to assume the worst-case scenario. Thus, we assume the UE will travel straight away from the BS for the whole time. The prediction of channel gain for this case can be derived from 6.6 as following:

$$g = 2^{\frac{1}{l}\cdot\int_{d_{start}}^{d_{start}+l} log_2(g(\sqrt{h_{diff}^2+d_{BS}^2}))\cdot\sqrt{\frac{d_{BS}^2+h_{diff}^2}{d_{BS}^2}}\cdot dd_{BS}} \tag{6.11}$$

where $l = t_{movement}\cdot v_{UE}$ is UEs trajectory length, $d_{start} = \|\mathbf{c_{UE,start}} - \mathbf{c_{BS}}\|$ is UEs starting distance from the BS (on 2-D Cartesian plane), $h_{diff} = h_{BS} - h_{UE}$ is the height difference between the UE and the BS and $d_{BS} = \|\mathbf{c_{UE}} - \mathbf{c_{BS}}\|$ is UEs distance from the BS on 2-D Cartesian plane.

6.2.3 Optimal setting of T_{change}

In this section, we derive a probabilistic formula for the optimal setting of the time T_{change}. T_{change} expresses the time for which we assume the UE is going to travel in its initial direction. We imagine the UE is going to change it's direction in the time T_{change} and is going to travel straight away from the BS afterwards for the remainder of the offloading time.

Lemma 6.1. *Suppose we have an UE moving according to the random direction movement model. Suppose the time before the UE changes its direction t_1 has an uniform probability distribution $t_1 = U(0, t_{max})$ with cumulative distribution function (CDF) $F(t_1)$. We let the UE move for the duration of $t = [0; t_2]$.*
Then the probability $P(t_2)$ that the UE is going to change its direction before the time t_2 can be expressed as:
$P(t_2) = F(t_2)$

Proof. From the definition of the cumulative distribution function. □

Lemma 6.2. *Suppose we have an UE moving according to the random direction movement model. Suppose the time before the UE changes its direction t_1 has an uniform probability distribution $t_1 = U(0, t_{max})$. We let the UE move for the duration of $t = [0; t_2]$, where $t_2 > 0$.*
Suppose the UE has changed its direction before the time t_2. Then the conditional probability $P(t_3|t_2)$ that the UE has changed its direction before

the time $t_3 \in [0, t_2]$ given it has changed its direction before the time t_2 has an uniform probability distribution $t_3 = U(0, t_2)$.

Proof. From the definition of the (conditional) probability: $P(t_3|t_2) = \frac{P(t_2 \cap t_3)}{P(t_2)} = \frac{P(t_3)}{P(t_2)} = \frac{t_3 \cdot t_{max}}{t_2 \cdot t_{max}} = \frac{t_3}{t_2} =$ CDF of $U(0, t_2)$. □

Lemma 6.3. *Suppose we have an UE moving according to the random direction movement model. Suppose the time before the UE changes its direction t_1 has an uniform probability distribution $t_1 = U(0, t_{max})$. We let the UE move for the duration of $t = [0; t_2]$, where $t_2 > 0$.*
Suppose we are given an upper limit P_L on probability $P(t_3 \cap t_2) \leq P_L$. Then the maximum time t_3 which satisfies the limit can be expressed as:
$t_3 = P_L \cdot t_{max}$ *for* $t_2 \in \{t_2|P(t_2) \geq P\}$
$t_3 = t_2$ *for* $t_2 \in \{t_2|P(t_2) < P\}$

Proof. $P \geq P(t_3 \cap t_2) = P(t_3|t_2) \cdot P(t_2)$, therefore $P(t_3|t_2) \leq \frac{P}{P(t_2)}$. Let $F^{-1}(t)$ be the inverse CDF of $U(0, t_2)$. Then according to 6.1, 6.2 and the definition of the inverse CDF of uniform distribution: $t_3 \leq F^{-1}(\frac{P}{P(t_2)}) = \frac{P}{P(t_2)} \cdot t_3 = P \cdot T_{max}$ for $\frac{P}{P(t_2)} \leq 1$ and $t_3 \leq t_2$ for $\frac{P}{P(t_2)} \geq 1$. □

We set the T_{change} according to the lemma 6.3 as:

$$T_{change} = P_L \cdot T_{max} \tag{6.12}$$

For offloading time satisfying $T_{off} \in \{T_{off}|P(T_{off}) \geq P_L\}$, and:

$$T_{change} = T_{off} \tag{6.13}$$

otherwise (for offloading time satisfying $T_{off} \in \{T_{off}|P(T_{off}) < P_L\}$).

P_L is a desired upper limit of the probability that the UE is going to change it's direction before T_{change}, $P(T_{off}) = F(T_{off})$ is a cumulative distribution function of the uniform distribution $U(0, T_{max})$ and T_{max} is the parameter of the uniform distribution.

6.3 Combined solution

Now it is time to put both parts of the solution derived in the previous sections together. We suppose the UE is going to move in its initial direction until T_{change} and afterwards it is going to move straight away from the BS

for the rest of the offloading time. The combined channel gain prediction g_e for the whole UE trajectory is obtained according to formula 6.6 as:

$$g_p = \sqrt{g_1 \cdot g_2} \tag{6.14}$$

where g_1 is the mean channel gain computed over the duration of $[0, T_{change}]$ (according to 6.2.1), g_2 is the mean channel gain computed over the duration $[T_{change}, T_{off}]$ (according to 6.2.2) and the T_{change} is obtained according to 6.2.3.

We then set the offloading transmission power of each UE (identically as in the chapter 4.1) as:

$$p = \frac{b \cdot (I_b + \sigma)}{g_p} \cdot \left(2^{\frac{D}{T_{off} \cdot b}} - 1\right) \tag{6.15}$$

where g_p is the predicted mean channel gain of the UE.

The combined solution is illustrated in figure 6.2.

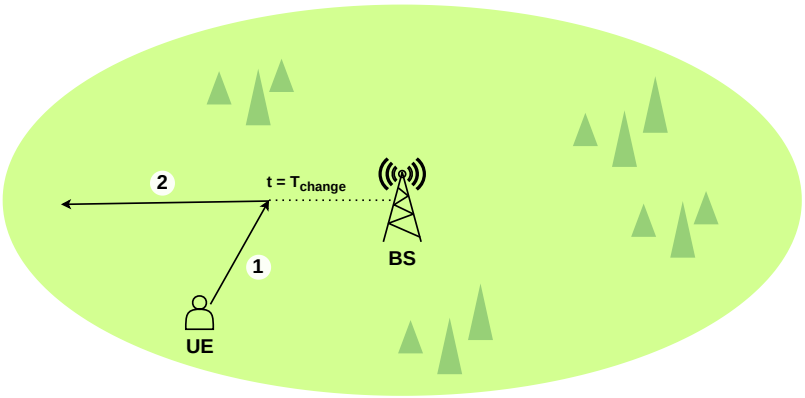


Figure 6.2: Illustration of the proposed solution. We assume (1) that the UE is going to continue moving in its initial direction until T_{change} (2) that it is going to move straight away from the BS for the rest of the offloading time.

6.4 Performance analysis of the proposed solution

In this section, we analyze the solution for dynamically changing channel proposed in former section of this chapter.

We start by analyzing two extreme settings of the parameter P_L: $P_L = 1$ and $P_L = 0$. *While those settings result in poor offloading performance, they help us to analyze and demonstrate some of the properties of the proposed solution.* After the analysis of the both extremes we move to the analysis of the proposed solution for $P_L = 0.96$ which we experimentally determined to result in a good trade-off between the success rate and consumed energy.

6.4.1 Simulation setup

We adopt the same performance metrics and simulation setup as in chapter 4. We also often compare the proposed solution to the ideal, obtained in a following manner. We first let all the UEs move according to their random movement model for the duration of T_{off}. We save the information about their channel gain to the BS and their geographical location in each time-step. From the information about the channel gains we compute the optimal channel gain according to the chapter 6.1.

Then follows the offloading process where the UEs offload their data while moving. The UEs start from their originally generated locations, but instead of moving according to the random movement model, they follow precisely the same path we have saved before. All important simulation parameters are summarized in the table 4.2.

6.4.2 Analysis of the proposed framework for extreme settings of P_L

Performance analysis for $\mathbf{P_L = 1}$

First, we are going to analyze our solution for the setting of the parameter P_L as $P_L = 1$. *This setting of P_L means, that when predicting the trajectory of*

the UEs, we suppose that the UEs will move in the initial movement direction for the whole time of the offloading.

This is not a proposed offloading optimization solution as the setting $P_L = 1$ is an extreme that yields unusable results. But analysis of this extreme helps us to demonstrate some of the properties of our solution.

The offloading success rate, average energy consumption for offloading and average energy consumption including local computing are displayed in figures 6.3, 6.4 and 6.5, respectively.[1]

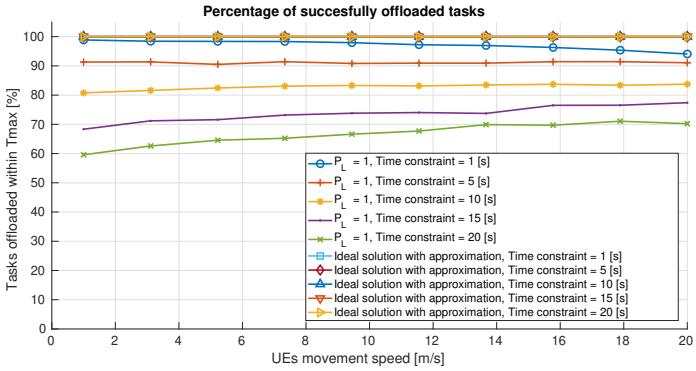


Figure 6.3: Comparison of success rate of the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the extreme setting $P_L = 1$. The success rate depends on UEs movement speed and the results are displayed for different time constraints.

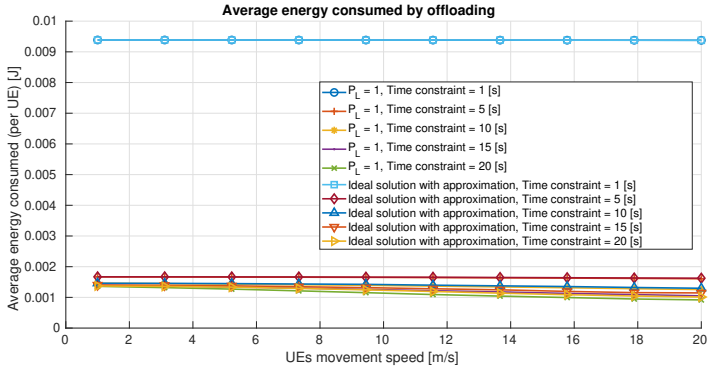


Figure 6.4: Comparison of energy consumed by offloading by the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the extreme setting $P_L = 1$. The energy consumption depends on UEs movement speed and the results are displayed for different time constraints. The energy consumption of both solutions overlap.

[1] We display these specific results for larger number of time constraints than in latter sections, so that the aspects of the success rate results we want to demonstrate here are better visible (in exchange for worse readability of the energy consumption).

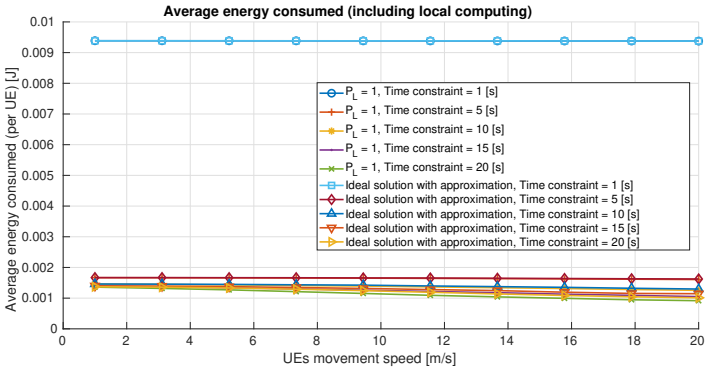


Figure 6.5: Comparison of total energy consumed by the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the extreme setting $P_L = 1$. The energy consumption depends on UEs movement speed and the results are displayed for different time constraints. The energy consumption of both solutions overlap.

The success rates are somewhat evenly distributed between 60 and 100 percent. The probabilities of the UEs changing their movement direction during the offloading in our simulation are: 5 percent for time constraint of 1 second, 25 percent for 5seconds, 50 percent for 10 seconds, 75 percent for 15 seconds and 100 percent for 20 seconds.

The displayed results make sense, because the probability of the UE worsening its channel gain to the BS after changing it's direction is about 50 percent (as shown in the chapter 4.1). Therefore, even if all of the UEs change their direction during the offloading time, only 50 percent are not going to finish the offloading within the time constraint. Thus the real probability P_{real} that the UE finishes offloading can be expressed approximately as:

$$P_{real} \approx P_L + \frac{1 - P_L}{2} \tag{6.16}$$

The additional improvements in success rates for the UEs with longer time constraint and movement speeds are caused by the "high SINR regime" approximation under which we compute the channel gain (see 6.1), as it slowly falls apart for such long time constraints.

Energy-wise is this solution almost identical with the ideal solution. We can see the energy slightly increasing over the ideal solution for the UEs who travel for longer trajectories during the offloading. This goes in hand with the improvements in offloading success rate improvements described in the previous paragraph.

Performance analysis for $P_L = 0$

We are now going to analyze our solution for the setting of the parameter P_L as $P_L = 0$. *This setting of P_L means, that when predicting the UE trajectory, we suppose that the UE will move straight away from the BS for the whole time of the offloading.* In other words, we are trying to be always on the safe side so that tasks are offloaded successfully, but at the cost of higher energy consumption.

We emphasize, that this is not a proposed offloading optimization solution as the setting $P_L = 0$ is an extreme that yields unusable results. But analysis of this extreme helps us to demonstrate some of the properties of our solution.

The offloading success rate, average energy consumption for offloading and average energy consumption including local computing are displayed in figures 6.6, 6.7 and 6.8 respectively.

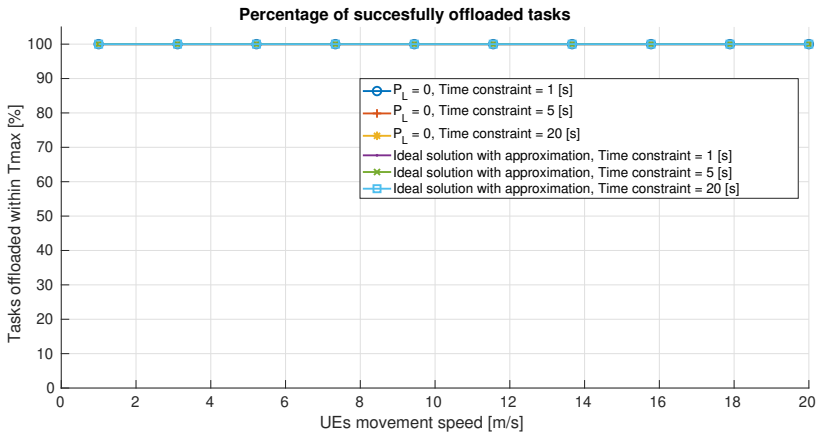


Figure 6.6: Comparison of success rate of the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the extreme setting $P_L = 0$. The success rate depends on UEs movement speed and the results are displayed for different time constraints.

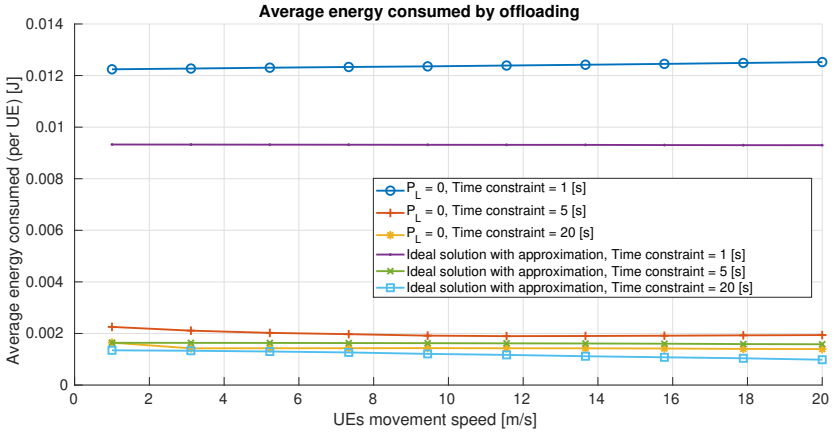


Figure 6.7: Comparison of energy consumed by offloading by the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the extreme setting $P_L = 0$. The energy consumption depends on UEs movement speed and the results are displayed for different time constraints.

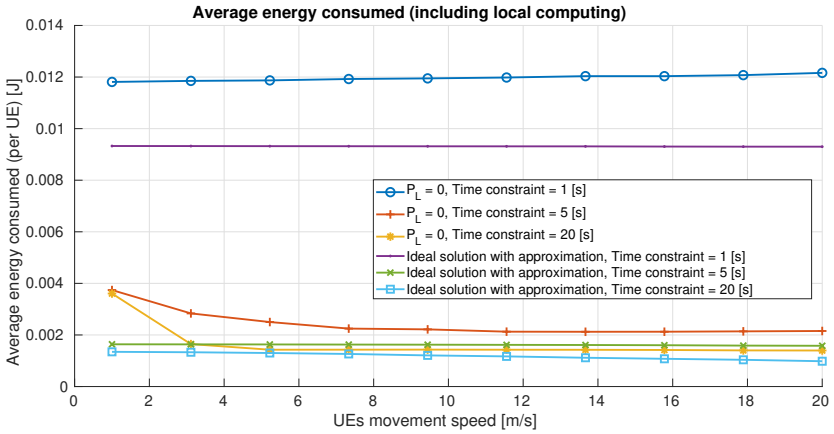


Figure 6.8: Comparison of total energy consumed by the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the extreme setting $P_L = 0$. The energy consumption depends on UEs movement speed and the results are displayed for different time constraints.

According to the intuition, the offloading success rate now stays at 100 percent for every time constraint and every UE movement speed. On the other hand, we can see a large increase of the consumed energy (both consumed by offloading and in total). We are wasting a lot of energy, because about 50 percent of the UEs do not travel away from the BS and almost none of the UEs do really move straight away from the BS. We are assuming the worst possible scenario, but we can see that this assumption is unreasonably pessimistic for the vast majority of the cases.

Interesting is also the energy "spike" for time constraints of 5 and 20 seconds at low movement speeds. This is due to the UEs whose starting position is at the edge of the simulation. Some part of the UEs which have really low initial channel gain to the BS (and on top of that possibly even travel further away from the BS) cross the threshold of the transmission power limit p_{max}. These UEs then decide to compute their tasks locally. But the local computing of the tasks on average costs much more energy than offloading of the tasks to the BS. Further, the energy increase caused by this decision makes a bigger difference for longer offloading time constrains, as these have a lower average energy consumption.

6.4.3 Performance analysis of the proposed solution

We have achieved the best overall trade-off between the success rate and consumed energy by trial and error approach with the setting of $P_L = 0.96$. The offloading success rate, average energy consumption for offloading and average energy consumption including local computing are displayed in figures 6.9, 6.10 and 6.11 respectively. With this setting of P_L, the resulting success rates stay above $\approx 98\%$ (in accordance to formula 6.16).

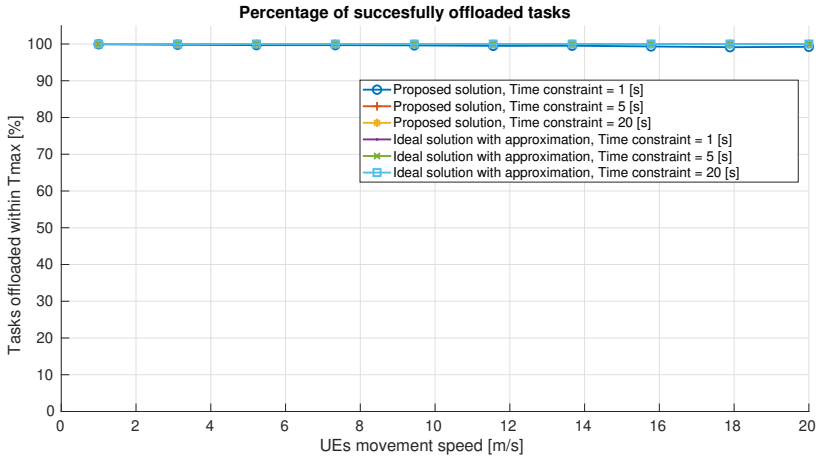


Figure 6.9: Comparison of success rate of the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The success rate depends on UE movement speed and the results are displayed for different time constraints.

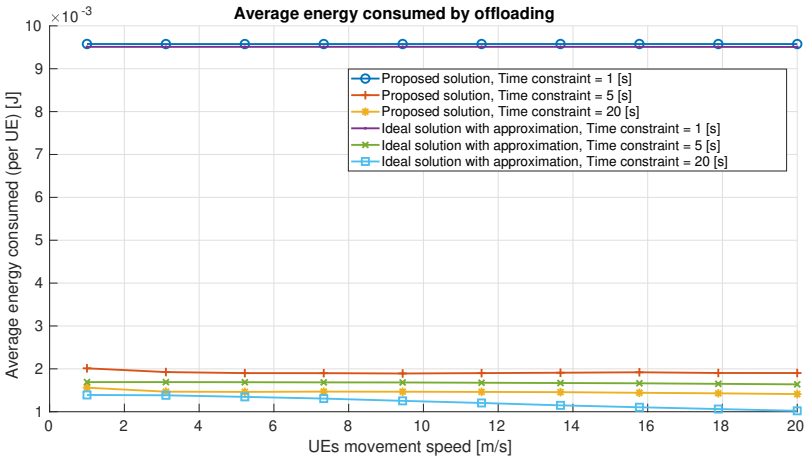


Figure 6.10: Comparison of energy consumed by offloading by the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The energy consumption depends on UE movement speed and the results are displayed for different time constraints.

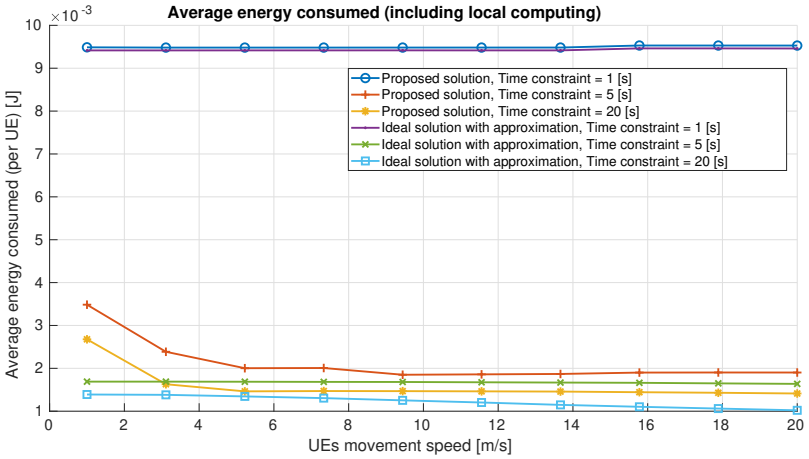


Figure 6.11: Comparison of total energy consumed by the ideal solution (obtained under the approximation SINR>> 1, 6.1) and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The energy consumption depends on UE movement speed and the results are displayed for different time constraints.

Both the energy consumption by offloading and the energy consumed in total stay (relatively) close to the ideal. Much more information can be seen if we make a relative comparison between the proposed solution and the ideal solution. The comparison for energy consumed by offloading and total energy consumption are displayed in figures 6.12 and 6.13, respectively. For short time constraints, the energy increase over the ideal stays around few percent (1 percent for the time constraint of 1s). It gets much worse for longer time constraints - for the time constraint of 20s we can see a peak increase of

$\approx$ 40% in energy consumed by offloading and a colossal peak increase of $\approx$ 200% in total energy consumed. These are again caused by some of the UEs computing locally instead of offloading their task due to being positioned far from the BS in combination with sub-optimal future trajectory prediction (as in 6.4.2).

In real world, the relative energy difference between the proposed and the ideal solution does not matter by far as much as the absolute increases in consumed energy. The greatest absolute increase in energy consumption occurs for short time constraints (the energy consumed grows exponentially with decreasing offloading time, see 4.3). By setting the $P_L = 0.96$, we can achieve a very good trade-off between the success rate and energy consumption for short time constraints, while keeping the success rates for every time constraint and UE movement speed over $\approx$ 98%.

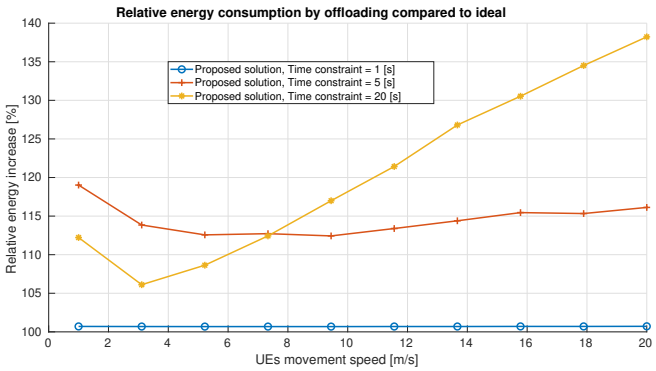


Figure 6.12: Relative comparison of the increase in energy consumed by the offloading by proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$ compared against the ideal solution (obtained under the approximation SINR>> 1, 6.1). The relative energy consumption depends on UE movement speed and the results are displayed for different time constraints.

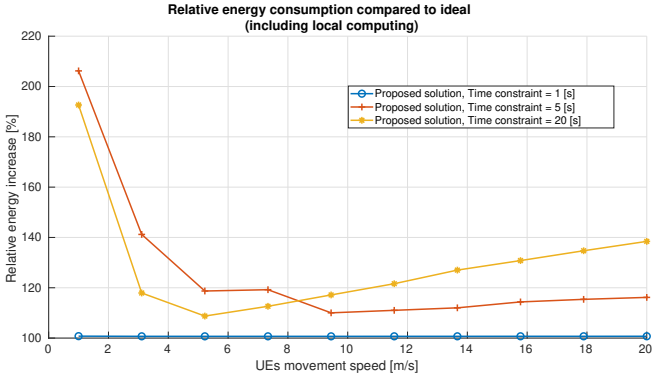


Figure 6.13: Relative comparison of the increase in total energy consumed by proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$ compared against the ideal solution (obtained under the approximation SINR>> 1, 6.1). The relative energy consumption depends on UE movement speed and the results are displayed for different time constraints.

6.4.4 Performance comparison of our solution ($P_L = 0.96$) and the current state-of-the-art

We are now going to compare the performance of our solution to the current state-of-the-art solution for the best setting of P_L, that is for $P_L = 0.96$. The offloading success rate, average energy consumption for offloading and average energy consumption including local computing are displayed in figures 6.14, 6.15 and 6.16 respectively. Success rates of the proposed solution stay above $\approx 98\%$ (in accordance to formula 6.16), while the current solution stays on $\approx 50\%$ (the slight increase for the time constraint of twenty seconds is explained in chapter 4.4).

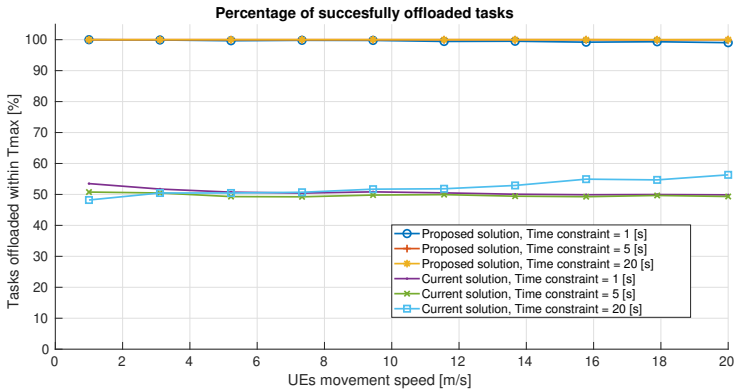


Figure 6.14: Comparison of success rate of the current state-of-the-art (static) offloading solution and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The success rate depends on UE movement speed and the results are displayed for different time constraints.

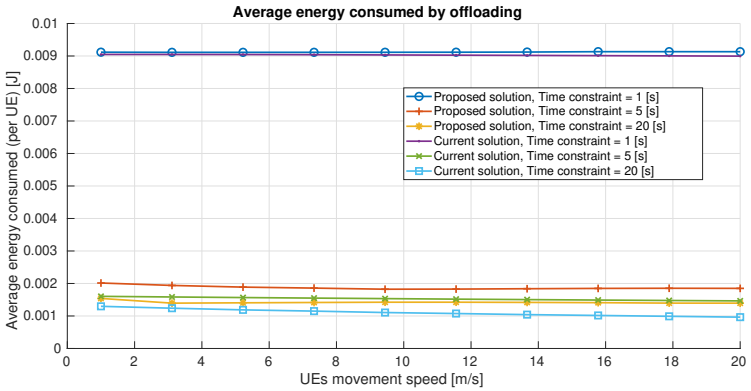


Figure 6.15: Comparison of energy consumed by offloading by the current state-of-the-art (static) offloading solution and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The energy consumption depends on UE movement speed and the results are displayed for different time constraints.

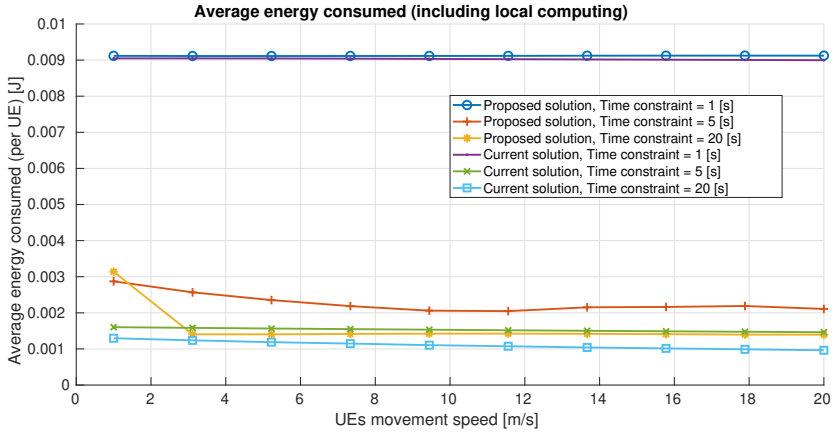


Figure 6.16: Comparison of total energy consumed by the current state-of-the-art (static) offloading solution and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The energy consumption depends on UE movement speed and the results are displayed for different time constraints.

The relative comparison for energy consumed by offloading and total energy consumption are displayed in figures 6.17 and 6.17 respectively. Comparisons of the energy consumption (both absolute and relative to ideal) are comparable with the energy consumption analyzed in the previous section 6.4.3.

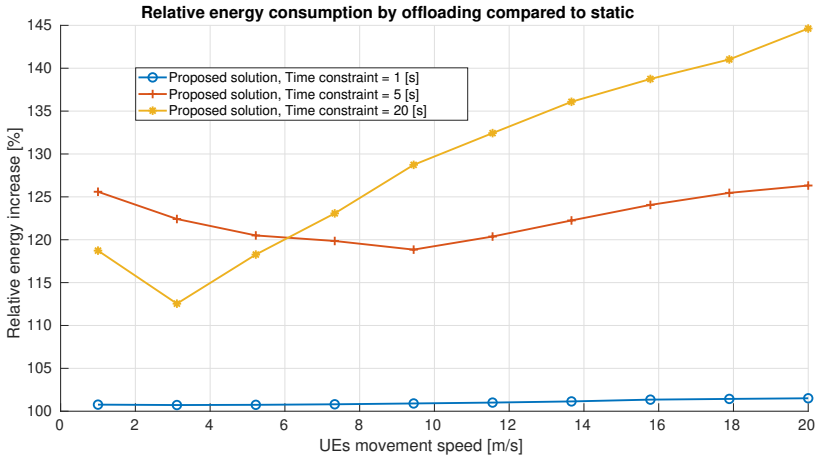


Figure 6.17: Relative comparison of the increase in energy consumed by the offloading by proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$ compared against the current state-of-the-art (static) offloading solution. The relative energy consumption depends on UE movement speed and the results are displayed for different time constraints.

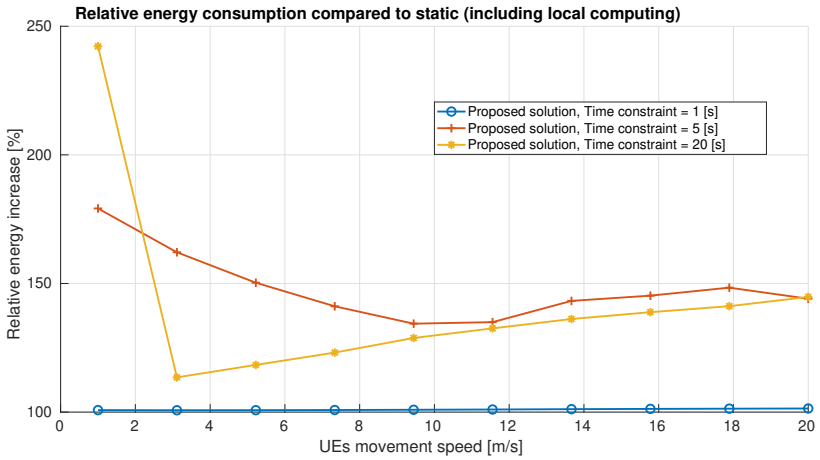


Figure 6.18: Relative comparison of the increase in total energy consumed by proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$ compared against the current state-of-the-art (static) offloading solution. The relative energy consumption depends on UE movement speed and the results are displayed for different time constraints.

We also include a general result comparison of the two solutions averaged over twenty different values of movement speed ($[0, 20]$ meters per second) and five different time constraints (1, 5, 10, 15 and 20 seconds). This comparison is displayed in figures 6.19, 6.20, 6.21 for success rate, average offloading energy and total energy respectively.

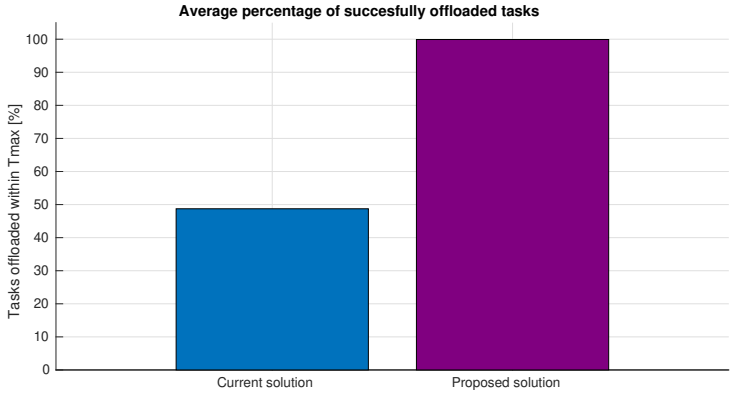


Figure 6.19: Overall comparison of success rate of the current state-of-the-art (static) offloading solution and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The success rate is averaged over twenty different values of movement speed ($[0, 20]$ meters per second) and five different time constraints (1, 5, 10, 15 and 20 seconds).

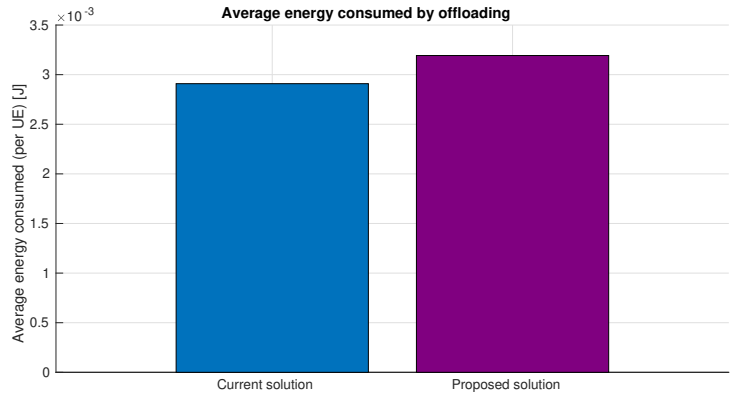


Figure 6.20: Comparison of energy consumed by offloading by the current state-of-the-art (static) offloading solution and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The energy is averaged over twenty different values of movement speed ($[0, 20]$ meters per second) and five different time constraints (1, 5, 10, 15 and 20 seconds).

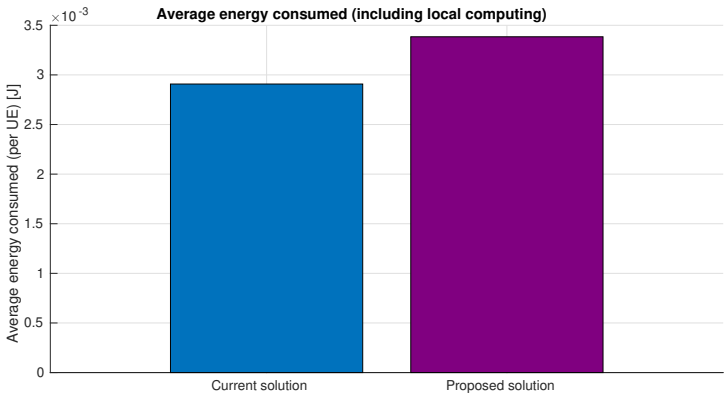


Figure 6.21: Comparison of total energy consumed by the current state-of-the-art (static) offloading solution and the proposed optimization framework under the proposed "optimal" setting $P_L = 0.96$. The energy is averaged over twenty different values of movement speed ($[0, 20]$ meters per second) and five different time constraints (1, 5, 10, 15 and 20 seconds).

6.4.5 Conclusion

We see that for realistic movement speeds and time constraints, we were able to achieve $\approx 99\%$ offloading success rate, which is twice the success rate of the current state-of-the-art solution ($\approx 50\%$). The increase in energy consumption of our solution is on other hand on average only $\approx 10\%$ when we count only offloading and $\approx 16\%$ if we count in also the cost of local computing.

Chapter 7

Conclusion

In this thesis, we have analyzed the current state-of-the-art offloading optimization framework in an environment with dynamic channel quality changes and we have shown that it is not usable. We then analyzed a simple modification of the current framework, where we left a time-reserve for the offloading to have some safety margin if the UEs are moving. This analysis allowed us to better understand the trade-off between the consumed energy and the number of tasks successfully offloaded. With this trade-off in mind, we have proposed a framework for the optimization of offloading computing tasks to the MEC server in an environment with dynamically changing channel quality. We have transformed the channel quality prediction problem into one of predicting the future trajectory of the UEs. We then introduced a simple proof-of-concept method for predicting the UEs trajectories.

We have shown, that the current state-of-the-art offloading solution is unusable when the channel is changing dynamically during the offloading, because only $\approx$ 50% of the tasks finish the offloading within their time constraints. On the other hand, the framework proposed in this thesis is able to guarantee $\geq$ 98% of the tasks offloaded within the time constraint, while the average energy increase over the current state-of-the-art static channel solution is for realistic movement speeds and time constraints on average only $\approx$ 10% and $\approx$ 16% if we do not consider local computing or if the local computing is assumed, respectively.

Even though the thesis presents very encouraging results, there are many open challenges that can be addressed in the future. Some of them are:

- Splitting of the tasks with longer time constraints into smaller parts and making the offloading decision for each part individually. The energy consumption of the proposed solution is growing for longer time constraints, as we are not able to predict the future channel quality as accurately (see 6.4.4). Optimizing the splitting of the tasks into smaller sub-tasks could therefore bring great benefits.
- The UE trajectory prediction framework proposed in this thesis is simple but sub-optimal. There exist more complex and accurate frameworks (as summarized i.e. in [27]). Leveraging a more optimized mobility prediction framework would certainly improve the results and the real-world applicability of our solution.
- In this work, we imagine a scenario, where all of the UEs have direct line-of-sight (LOS) to the BS and the proposed solution is therefore not applicable to scenarios where the line of sight is obstructed (for example urban scenarios with buildings). Without the line of sight to the BS, our solution could be improved by utilizing a framework to predict shadowing based on the UEs geographical location (proposed for example in [28]).

- Extension from direct offloading to offloading in multiple hops over relays (see for example [5], [29]) could prove very beneficial both in environments without the line-of-sight and overall. Here, an interesting direction to investigate could be the optimization of the relay selection based on the movement of both the UE and the relays.

Appendix A

List of Acronyms

Acronym	Meaning
UE	User equipment
BS	Base station
MEC	Multi-access edge computing
UAV	Unmanned aerial vehicle
SINR	Signal-to-interference-plus-noise ratio
IoT	Internet of things
LOS	Line of sight

Appendix B

Bibliography

[1] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[2] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[3] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.

[4] L. Nadeem, M. A. Azam, Y. Amin, M. A. Al-Ghamdi, K. K. Chai, M. F. N. Khan, and M. A. Khan, "Integration of d2d, network slicing, and mec in 5g cellular networks: Survey and challenges," *IEEE Access*, vol. 9, pp. 37590–37612, 2021.

[5] P. Mach, Z. Becvar, and M. Nikooroo, "Multi-hop relaying with mixed half and full duplex relays for offloading to mec," in *2023 IEEE Global Communications Conference (GLOBECOM)*, 2023.

[6] M. Kamoun, W. Labidi, and M. Sarkiss, "Joint resource allocation and offloading strategies in cloud enabled cellular networks," in *2015 IEEE International Conference on Communications (ICC)*, pp. 5529–5534, 2015.

[7] W. Labidi, M. Sarkiss, and M. Kamoun, "Joint multi-user resource scheduling and computation offloading in small cell networks," in *2015*

IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 794–801, 2015.

[8] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 26–30, 2013.

[9] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, 2015.

[10] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2016.

[11] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing," in *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2015.

[12] X. Guo, R. Singh, T. Zhao, and Z. Niu, "An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems," pp. 1–7, 05 2016.

[13] P. Mach and Z. Becvar, "Cloud-aware power control for cloud-enabled small cells," in *2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 1038–1043, 2014.

[14] T. Taleb and A. Ksentini, "An analytical model for follow me cloud," pp. 1291–1296, 12 2013.

[15] A. Ksentini, T. Taleb, and M. Chen, "A markov decision process-based service migration procedure for follow me cloud," in *2014 IEEE International Conference on Communications (ICC)*, pp. 1350–1354, 2014.

[16] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *2015 IFIP Networking Conference (IFIP Networking)*, pp. 1–9, 2015.

[17] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2169–2182, 2023.

[18] C. Li, H. Wang, and R. Song, "Mobility-aware offloading and resource allocation in noma-mec systems via dc," *IEEE Communications Letters*, vol. 26, no. 5, pp. 1091–1095, 2022.

[19] T. Fang, F. Yuan, L. Ao, and J. Chen, "Joint task offloading, d2d pairing, and resource allocation in device-enhanced mec: A potential game approach," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3226–3237, 2022.

[20] L. S. Muppirisetty, T. Svensson, and H. Wymeersch, "Spatial wireless channel prediction under location uncertainty," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1031–1044, 2016.

[21] A. Goldsmith, L. Greenstein, and G. Foschini, "Error statistics of real-time power measurements in cellular channels with multipath and shadowing," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 439–446, 1994.

[22] Y. Deng, Z. Chen, X. Chen, and Y. Fang, "Task offloading in multi-hop relay-aided multi-access edge computing," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1372–1376, 2023.

[23] P. Nain, D. Towsley, B. Liu, and Z. Liu, "Properties of random direction models," vol. 3, pp. 1897 – 1907 vol. 3, 04 2005.

[24] M. Liu, Y. Wan, and F. L. Lewis, "Analysis of the random direction mobility model with a sense-and-avoid protocol," in *2017 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2017.

[25] M. Najla, Z. Becvar, and P. Mach, "Reuse of multiple channels by multiple d2d pairs in dedicated mode: A game theoretic approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4313–4327, 2021.

[26] Y. Huang, A. A. Nasir, S. Durrani, and X. Zhou, "Mode selection, resource allocation, and power control for d2d-enabled two-tier cellular network," *IEEE Transactions on Communications*, vol. 64, p. 3534–3547, Aug. 2016.

[27] N. Rajule, M. Venkatesan, R. Menon, and A. Kulkarni, "Mobility prediction in cellular networks: A survey," in *2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC)*, pp. 1–9, 2023.

[28] L. S. Muppirisetty, T. Svensson, and H. Wymeersch, "Spatial wireless channel prediction under location uncertainty," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1031–1044, 2016.

[29] P. Mach and Z. Becvar, "Device-to-device relaying: Optimization, performance perspectives, and open challenges towards 6g networks," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 3, pp. 1336–1393, 2022.

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Jeník Adam** Personal ID number: **503181**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Radioelectronics**

Study program: **Open Electronic Systems**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Computation Offloading to Network's Edge in Environment with Dynamic Channel Quality Changes

Bachelor’s thesis title in Czech:

Přenášení výpočtů na hranu sítě v prostředí s dynamickými změnami kvality rádiového kanálu

Guidelines:

This thesis focuses on Edge Computing paradigm enabling to offload computationally intensive tasks to a server located at the edge of the mobile network. First, study existing literature on the offloading decision problem determining where the tasks should be computed and the means of delivering tasks to the cloud. The conventional offloading decision algorithms assume that the channel quality of links used for offloading are always static. Nevertheless, in mobile networks the channel quality is never static and can change very quickly during the offloading process, e.g., if offloading user and/or relaying user helping with offloading change their location. Such a change in the quality of the connection during the tasks’ offloading can make existing offloading decision algorithms ineffective and performing poorly. Thus, design an algorithm for offloading decision that will cope with the channel quality changes during the offloading process. The main objective is to find a trade-off between the sum energy consumption over all users and the number of tasks completed within a required deadline. Demonstrate the gain of the proposed algorithm in terms of energy consumption and/or number of tasks completed within the deadline depending on the number and speed of devices, or other relevant parameters.

Bibliography / sources:

[1] P. Mach and Z. Becvar, „Mobile Edge Computing: A Survey on Architecture and Computation Offloading,“ IEEE Communications Surveys & Tutorials, vol. 19, no. 3, 2017.
[2] P. Mach and Z. Becvar, „Device-to-Device Relaying: Optimization, Performance Perspectives, and Open Challenges Towards 6G Networks,“ IEEE Communications Surveys & Tutorials, vol. 24, no. 3, 2022.

Name and workplace of bachelor’s thesis supervisor:

Ing. Pavel Mach, Ph.D. Department of Telecommunications Engineering FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **04.10.2023** Deadline for bachelor thesis submission: _____

Assignment valid until: **21.09.2025**

Ing. Pavel Mach, Ph.D.	doc. Ing. Stanislav Vítek, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature