Towards the Advanced Data Processing for Medical Applications Using Task Offloading Strategy

Daria Alekseeva, Aleksandr Ometov, and Elena Simona Lohan
Faculty of Information Technology and Communication Sciences, Tampere University
Korkeakoulunkatu 6, Tampere, Finland, FI-33720
Emails: {name.surname}@tuni.fi

***Abstract*—Broad adoption of resource-constrained devices for medical use has additional limitations in terms of execution of delay-sensitive medical applications. As one of the solutions, new ways of computational offloading could be developed and integrated. The recently emerged Mobile Edge Computing (MEC) and Mobile Cloud Computing (MCC) paradigms attempt to address this problem by offloading tasks to a the resource-rich server. In the context of the availability of eHealth services for all patients, independently of the location, the implementation of MEC and MCC could help ensure a high availability of medical services. Remote medical examination, robotic surgery, and cardiac telemetry require efficient computing solutions. This work discusses three alternative computing models: local computing, MEC, and MCC. We have designed a Matlab-based tool to calculate and compare the response time and energy efficiency. We show that local computing demands 48 times more power than MEC/MCC with increasing packet workload. On the other hand, the throughput of MEC/MCC highly depends on the parameters of the communication channel. Finding an optimal trade-off between the response time and energy consumption is an important research question that could not be solved without investigating the system's bottlenecks.**

***Index Terms*—Mobile Cloud Computing (MCC), Mobile Edge Computing (MEC), local computing, 5G mobile communication**

I. Introduction

Robot-assisted surgeries climbed from 1.8% to 15.1% of all general surgeries in several years [1]. Assisted robots allow to carry out minimally invasive surgeons, as they exclude hand tremors. The interest in telemedicine has grown after the pandemic as it could provide an equal level of qualified on-demand medical service even staying at home [2]. Under the building of an intelligent eHealth system stays a complex computing system that allows the processing of an enormous amount of medical and private data. At the same time, the communication requirements have become stricter as it needs a very reliable channel [3].

Recently emerged computing paradigms offer vast computing capacity, which opens new capabilities to the medical applications [4]. Mobile Cloud Computing (MCC) gives enormous computational resources for remote use. Mobile Edge Computing (MEC) shows better performance than MCC in terms of latency, because of the proximate location server to the user's devices [5]. Therefore, the offloading strategy for medical cases is a promising research direction. An intelligent combination of several computing paradigms can develop a system to improve the device energy consumption and satisfy the requirements of latency-sensitive applications.

The implementation of Edge and Cloud computing is a prospective solution for eHealth applications because the predicted growth of Internet of Medical Things (IoMT) needs an enormous computing power, mass storage of medical data, and trust sharing of medical information, which a cloud platform could provide [6]. For example, an intelligent orchestration of computing resources for health monitoring applications improves their energy budget and Quality of Service (QoS) [7].

Several directions in optimizing the computing system could be found in the literature – optimizing the task execution itself, improving the system response time, or joint resources use with task offloading strategy. Some researchers propose a optimization-and-offloading decision with multiple mobile users named Heuristic Offloading Decision Algorithm (HODA) based on prioritizing users with maximum utility [8]. A dynamic environment, caused by the inherent variability of wireless networks, queuing delays in the servers, and user's device parameters, brings extra challenges to the offloading. Deep Reinforcement Learning is a potential solution for the intelligent allocation of computing resources in the dynamic environment [9], [10]. Adoption of digital twin-based architecture allows for real-time monitoring and provides the information for decision-making processes [11]. New strategies in the computing resource allocation of massive IoMT devices could reach the optimization latency and efficient data processing goal, as well as improve the system security [12].

Optimization of the communication part is another research direction in the MEC/MCC computing optimization model. Spectrum sharing benefits throughput maximization, which increases the computational rate of energy-constrained Internet of Things (IoT), or rather IoMT devices [13]. The recently emerged Fifth Generation New Radio Network (5G NR) is a promising technology for future smart healthcare with ultra high-speed transmission and efficient spectrum utilization by new channel multiplexing Non-Orthogonal Multiple-Access (NOMA) [14]. Another solution is the implementation of advanced energy-efficient dynamic decision-based scheduling and orchestration algorithms that improves energy consumption and systems response [15], [16].

The contribution of this work are summarized as:

- formalization of the optimization problem as the minimization function of response time and energy consumption;
- design of Matlab-based tool that helps to get the fast calculation on the response time and energy consumption of

The work was supported by the national DELTA doctoral training network, the Pekka Ahonen Fund, and doctoral grant of the Information Technology and Communications Science Faculty at Tampere University.

the packet (the code is available in the open access: https://github.com/alekseevadaria/iot_edge_cloud_matlab);

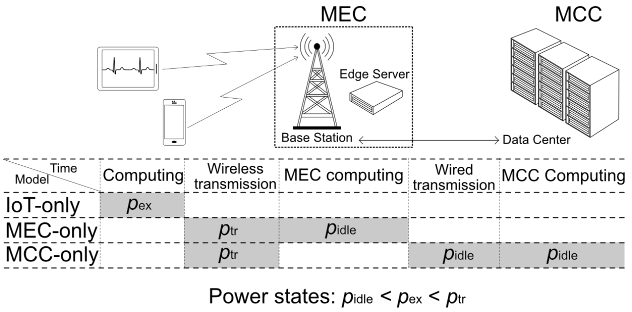
- providing computing suggestions for medical scenarios.

This paper is structured as follows. Section II describes the methodology how to obtain the response time and energy consumption. Section III refers to the optimization problem formulation. Section IV presents the designed tool, simulation parameters and the results. Section V is the conclusion.

II. System Model

Assume that there are three possible computing locations in the system. The first one occupies all resources on the device (i.e., local execution, IoT-only), the second one offloads tasks at the edge of the network (i.e., MEC-only), and the third one sends data to the cloud to proceed (i.e., MCC-only). Each model (IoT-only, MEC-only, MCC-only) has pros and cons. Local execution (IoT-only) benefits in terms of security as it does not send data outside the device, but it has limited computing and energy resources [17]. MEC and MCC have more capabilities to proceed with a large amount of data. Anyway, they face other challenges, e.g., intelligent user allocation [18] or communication reliability [19], that could reduce or increase system latency.

The systems response time and energy consumption are essential metrics to evaluate the system performance. The components of the response time and power states are illustrated in Fig. 1, where you can see that during the remote computation device is using the idle power state, i.e., power saving mode of light sleep to save energy. Obviously, when it is time to send data, the device uses the transmission power state. In the MCC-only model, the device sends data to the first Base Station (BS) in the transmitting power state, then it switches to the idle state, as it does not care about where data is transferring after the BS gets data till it reaches Cloud. Finally, execution time corresponds to the computing power state. The following subsections provide a discussion about response time and energy consumption.



Model \ Time	Computing	Wireless transmission	MEC computing	Wired transmission	MCC Computing
IoT-only	p_{ex}				
MEC-only		p_{tr}	p_{idle}		
MCC-only		p_{tr}		p_{idle}	p_{idle}

Power states: $p_{idle} < p_{ex} < p_{tr}$

Fig. 1: Illustration of the time components in local and remote strategies and the correlation to the device's power states.

A. Response Time

The overall response time consists of time that was spent on computing and time that was spent transferring data (i.e., communication time) $T = T_{comp} + T_{comm}$. The following paragraphs explain how to get their values.

Computing time is spent for the processing data, which mainly depends on Central Processing Unit (CPU) parameters of the node involved in computing, and could be calculated:

$$T_{comp} = \frac{\omega_i \cdot \lambda}{f_{CPU}}, \tag{1}$$

where ω_i – task workload of the task *i* [*Mb*]; f_{CPU} – the CPU processing frequency of the device or server [*Hz*]; λ – computation to data ratio [*cycles/bit*]. The comparison of different processor units is based on their working speed with data. Computation to data ratio varies for different workloads and mainly depends on the physical parameters of the computer itself [20]. For this reason, Floating point operations per second (FLOPS) is a universal measure of CPU performance, which describes the capability of delivering any arithmetic or logic operations per second [21].

Communication time corresponds to the time spent transmitting data, and it depends on the channel parameters such as bandwidth, carrier frequency, bit rate, and propagation conditions, which is presented as:

$$T_{comm} = \frac{\omega_i}{C} + T_d, \tag{2}$$

where ω_i – task workload of the task *i* [*Mb*]; C – bit rate [*Mbps*]; T_d – variable that defines all kinds of delays that exist in the communication path (for example, delay on hops, communication delay, etc.) [*ms*].

There is no communication time in the local-computing mode, because the data is not transferred outside the device. Hence, the response time for local computing stays the same as in eq. 1. The communication time for MEC and MCC depends on the bit rate of the chosen wireless technology. The next paragraphs explain how to calculate the bit rate for 5G NR.

Supported maximum data rate in 5G NR New cellular network 5G NR promise is to improve the Quality of Experience (QoE) for the user and provide new services that were not supported before, with extreme mobile bandwidth, ultra-reliability, and ultra-low latency ($\leq$ 1 ms). The maximum data rate in 5G NR could be defined as follows [22]:

$$C_{5G} = 10^{-6} \cdot \sum_{j=1}^{J} \left(v_l \cdot Q_m \cdot f \cdot R_{max} \cdot \frac{N_{RB} \cdot K}{T_s^{\mu}} \cdot (1 - OH) \right), \tag{3}$$

where 10^{-6} – conversion to Mbps; J – a number of aggregated component carriers in a band; v_l – the maximum number of layers; Q_m – modulation coefficient; f – the scaling factor; R_{max} – maximum code rate; N_{RB} – is the maximum resource blocks allocation in bandwidth; K – the number of subcarriers per resource block; T_s – the average symbol duration in the correlated Orthogonal Frequency Division Multiplexing (OFDM) numerology μ; OH – overhead value.

The above is standardized by 3GPP and ETSI and can be found in [22]–[24]. Here are some parameters in more detail. The entire channel resource is divided into resource blocks. The resource block consists of $K = 12$ subcarriers with a given subcarrier spacing (SCS). The maximum code rate R_{max} shows the proportion of useful data to redundant information and equal constant $948/1024$. Parameter f is the scaling factor, and it depends on a number of Multiple-Input Multiple-Output (MIMO) and modulation order. It could take values 1, 0.8, 0.75, and 0.4. μ is the OFDM numerology which is chosen according to the subcarrier spacing [24]. T_s is the average symbol duration that could be defined as $T_s = \frac{10^{-3}}{14 \cdot 2^{\mu}}$. OH is the overhead, and it takes values according to the frequency range and stream direction [22].

B. Energy Consumption

Battery life is no less critical metric of the device's performance than the system's response time. It influences the working time of the device and depends on battery capacity, which is limited in the IoT [25]. The energy taken for the working process is calculated as the product of the spent time and the device's power. The power spent by the device for task computing and task transmission is not the same. Assume that there are three different power states: p_{ex} is the power that device uses for task execution, p_i is the power of the device in the idle state, p_{tr} is the power that used for task transmission, at that $p_i < p_{ex} < p_{tr}$ [26]. The idle power state refers to the computing time in the remote server, in other words, when the device is waiting when the computing will be done in the edge or in the cloud. The overall energy that was spent on the offloading could be calculated as $P = P_{comp} + P_{comm}$.

Local computing: Based on equation (1) from above, the local execution energy consumption could be found as:

$$P_l = T_{comp} \cdot p_{ex} = \frac{w_i}{f_{CPU}} \cdot p_{ex}, \tag{4}$$

where w_i – task workload of the task *i* [*Mb*]; f_{CPU} – the CPU processing frequency of the device or server [*Hz*]; p_{ex} – the power that was spent for computing [*mW*].

Remote computing corresponds to MEC and MCC models. Based on eqs. 1 and 2, the energy consumption is:

$$P_r = T_{comp} \cdot p_i + T_{comm} \cdot p_{tr} = \frac{w_i}{f_{CPU}} \cdot p_i + \left(\frac{w_i}{C} + T_d\right) \cdot p_{tr}, \tag{5}$$

where w_i – task workload of the task i [*Mb*]; C – bit rate [*Mbps*]; T_d – delays if any [*ms*]; p_i – the idle state power [*mW*]; p_{tr} – the power that spent for task transmission [*mW*].

III. Problem Statement

First, assume that any application might be presented as a sequence of tasks t_i, with $i = \{1, \ldots, M\}$. Each task is assumed to have a workload w_i and it can be computed locally or outside the device. The allocation of the computing outside the device could be one of the followings: i) at the Edge of the network in close proximity to the end-user (MEC, Cloudlets), or ii) in the distant Cloud (MCC). The offloading decision of task t_i is denoted as k_i in the sequence of the computing decisions K and it could be one of the following $k_i \in \{-1, 0, 1\}$, where $k_i = 0$, if the task is processing locally, $k_i = -1$ and $k_i = 1$, if the task is processing in the edge or in the cloud, respectively. Based on eqs. (1) and (2) the system response time for task i is:

$$\begin{gathered} T(k_i, t_i) = \big(1 - |k_i|\big) \cdot T_{comp}(t_i) + \\ |k_i| \cdot \big(T_{comp}(t_i) + T_{comm}(t_i)\big), \\ k_i = \begin{cases} -1, & \text{if MEC} \\ 0, & \text{if local execution} \\ 1, & \text{if MCC} \end{cases} \end{gathered} \tag{6}$$

Thus, the energy consumption system model for task i is:

$$P(k_i, t_i) = \big(1 - |k_i|\big) \cdot P_l(t_i) + |k_i| \cdot P_r(t_i), \tag{7}$$

where k_i denotes to offloading decision for task i ($k_i = 0$ for local computing on the IoT device, $k_i = -1$ for the MEC, $k_i = 1$ for the MCC).

The optimization problem for the offloading decision strategy can be seen as a multi-objective optimization function trying to minimize both of the system's response time and its energy consumption. We formulate the multi-objective optimization problem as follows:

$$\min_{\{k_i \in \{-1,0,1\}\}_{i=1,\ldots,M}} \left(\sum_{i=1}^{M} T(k_i, t_i), \sum_{i=1}^{M} P(k_i, t_i) \right), \tag{8}$$

where $\sum_{i=1}^{M} T(k_i, t_i)$ is the total response time; $\sum_{i=1}^{M} P(k_i, t_i)$ is the total energy consumption.

IV. Simulation And Results

A. The Designed Tool

Using Matlab R2020b 64-bit App Designer software, we have created an application to analyze the local and remote computing systems, see Fig. 2. In fact, this tool is a response time (T) and energy consumption (P) calculator for the MEC and MCC. computingThe app calculates the mentioned parameters by changing the slider with the workload. The application consists of two areas: input and output. The user could change the device parameters in the input area, set the power capacity, and adjust wireless and wired parameters. The output area displays the response time and energy consumption values and draws the bars. This application is useful for students who are beginning to learn computing science to clarify some basic features and differences between local and remote processing.

B. The Workload From the Practical Application Perspective

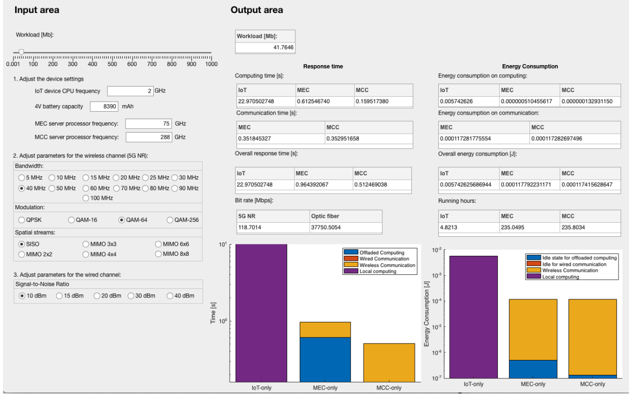
5G NR and edge-cloud computing enable to shift the medical care location from the sorrowful and comfortless hospitals to homes [28]. Also, remote medical expertise and telesurgery play an important role in the emergency because it could provide qualified medical help in any location. Table I contains the performance requirements for the medical use cases from the 3GPP TR 22.826 V17.2.0 [27] and TS 22.104 V18.0.0 [28]). The next paragraphs provide their broad descriptions.

Duplicating videos on additional monitors in the context of robotic surgery, the procedure is complemented by the imaging system for the surgeon and their assistant. This use

TABLE I: Requirements based on 3GPP TR 22.826 V17.2.0 [27] TS 22.104 V18.0.0 [28]) for suggested computing locations.

Use case	Ref.	Latency [ms]	Bit rate	Direction	Message size [kb]	UE speed [km/h]	Number of User Equipment (UE)	Computing location
Duplicating Video on additional monitors	[27]	< 1	120 Gbps	UL	∼12 – ∼72	0	1	Network Edge (i.e. MEC)
AR Assisted Surgery	[27]	$<$ **0.75**	**30; 12 Gbps**	**UL**	**∼12 – ∼72**	**0**	**1**	**Short network distance from the operating room (i.e. MEC)**
Robotic Aided Surgery	[27]	< 2	240 Gbps	UL; DL	∼12 – ∼72	0	1	Short network distance from the operating room (i.e. MEC)
	[28]	$<$ **2**	**2 – 16 Mbps**	**UL; DL**	**2 – 16**	**0**	**1**	**Edge or Cloud (i.e. MEC or MCC)**
Telesurgery	[27]	< 20	2 – 16 Mbps	UL; DL	2 – 16	0	$<$ 2: 1000 km^2	Cloud (i.e. MCC)
	[28]	< 20	2 – 16 Mbps	UL; DL	2 – 16	0	$<$ 2: 1000 km^2	Edge or Cloud (i.e. MEC or MCC)
Robotic Aided Diagnosis	[28]	< 20	2 – 16 Mbps	N/A	0.64	0	20: 100 km^2	Edge or Cloud (i.e. MEC or MCC)
Cardiac telemetry outside the hospital (body-worn IoT device)	[27]	$<$ **100**	**0.5 Mbps**	**N/A**	**≤8**	**≤500**	**10 – 1000: 1 km^2**	**Hospital cloud (i.e. MCC)**

UL – Uplink DL – Downlink N/A – not available in the corresponding document



Input area

Workload [Mb]:

1. Adjust the device settings

IoT device CPU frequency 2 GHz

4V battery capacity 8390 mAh

MEC server processor frequency: 75 GHz

MCC server processor frequency: 288 GHz

2. Adjust parameters for the wireless channel (5G NR):

Bandwidth:

○ 5 MHz ○ 10 MHz ○ 15 MHz ○ 20 MHz ○ 25 MHz ○ 30 MHz
● 40 MHz ○ 50 MHz ○ 60 MHz ○ 70 MHz ○ 80 MHz ○ 90 MHz
○ 100 MHz

Modulation:

○ QPSK ○ QAM-16 ● QAM-64 ○ QAM-256

Spatial streams:

● SISO ○ MIMO 3x3 ○ MIMO 6x6
○ MIMO 2x2 ○ MIMO 4x4 ○ MIMO 8x8

3. Adjust parameters for the wired channel:

Signal-to-Noise Ratio

● 10 dBm ○ 15 dBm ○ 20 dBm ○ 30 dBm ○ 40 dBm

Output area

Workload [Mb]:
41.7646

Response time

Computing time [s]:

IoT	MEC	MCC
22.970502748	0.612546740	0.159517380

Communication time [s]:

MEC	MCC
0.351845327	0.352951658

Overall response time [s]:

IoT	MEC	MCC
22.970502748	0.964392067	0.512469038

Bit rate [Mbps]:

5G NR	Optic fiber
118.7014	37750.5054

Energy Consumption

Energy consumption on computing:

IoT	MEC	MCC
0.005742626	0.000000510455617	0.000000132931150

Energy consumption on communication:

MEC	MCC
0.000117281775554	0.000117282697496

Overall energy consumption [J]:

IoT	MEC	MCC
0.005742625686944	0.000117792231171	0.000117415628647

Running hours:

IoT	MEC	MCC
4.8213	235.0495	235.8034

Fig. 2: The tool interface designed in the Matlab App

case assumes at least two monitors without noticeable delays for each surgery operator in the room, which allow the surgery team to work efficiently during the operation. The image from the laparoscope is going to a medical application instantiated at the network edge and then broadcast to the monitors in the operating room. Any image delay can lead to operators' desynchronization and patient injury. *Augmented Reality (AR) assisted surgery* is a promising use case for minimally invasive operations based on the 3D display of patient anatomy. Two head-mounted displays, i.e., AR glasses, show the real-time video stream from the surgical tools with implanted cameras. This use case needs a lot of computation capacity and ultra low latency (less than 750 μs) because the mistake caused by the operator could lead to the patient's death. *Robotic aided surgery* is an innovative surgical operation that uses tiny instruments to access the problematic body areas when the surgeon and the robot are collocated in the same operating room. The surgical devices and motion controllers are synchronized to achieve an accurate incision and operated at frequencies around 1 GHz, referred to as human tactile sensing. *Telesurgery* is a use case very similar to robotic aided surgery, with the difference that the surgeon console and the robotic set have different geographical locations. It also provides highly accurate invasive procedures with robotic equipment. *Robotic aided diagnosis* is a part of the mobile specialist practice, which describes when a medical expert could provide a high-quality examination to a patient in any location around the world. Due to the distance between the patent and the expert, the medical application is initiated remotely in the Edge or Cloud. *Cardiac telemetry* this use case contains wearable IoT devices to monitor the patient's activity and vital measurements continuously. Since the on-body IoT device must be worn for weeks or even months, it needs to be discreet and lightweight for comfortable use.

To summarize the above, all medical use cases could be divided into three groups: i) use cases with high intensive tasks and strict performance requirements (i.e., AR Assisted Surgery), ii) use cases with moderate intense tasks and moderate performance requirements (i.e., Robotic Aided Surgery), and iii) use cases with low-intensive tasks and low-performance requirements (i.e., Cardiac Telemetry). Further in this work, the high, medium and low intensive tasks are referred to AR Assisted Surgery, Robotic Aided Surgery, and Cardiac Telemetry use cases.

C. Simulation Parameters

For the simulation, we took the following parameters. Assume that the device's CPU works with a 2 GHz frequency, referred to as Qualcomm processors [29]. The battery capacity parameter is 8390 mAh [25]. The Edge server has a 16 core 4.7 GHz CPU (e.g., 1.5U Rackmount AMD Ryzen Server) [30], and the Cloud server has a 96 unit 3 GHz CPU (e.g., Amazon EC2) [31]. The CPU frequencies satisfy the formula $f_{IoT} < f_{MEC} < f_{MCC}$, comprising that the Cloud has more computational power than MEC, and both are stronger than IoT. Also, assume that the device consumes power equal to 0.9 Watts during the processing time, 1.2 Watts for data transmission to the base station, and an idle state – 0.003 Watt [32], [33]. This values meet the condition $p_i < p_{ex} < p_{tr}$ [26], described in Section II-B. Table II contains the used parameters for simulation.

We also assume that the 5G NR is the chosen wireless communication technology. According to eq. (3), the bit rate depends on the bandwidth, modulation, and the number of antennas. Parameters used for calculating the supported maximum bit rate in the wireless channel are summarised in

TABLE II: Simulation parameters

Parameter	Acronym	Value	Ref.
Processor:			
IoT	f_{IoT}	2 GHz	[29]
MEC	f_{MEC}	75 GHz	[30]
MCC	f_{MCC}	288 GHz	[31]
Computation to data ratio	λ	$1.1 \cdot 10^3$ cycles/bit	[20]
Wireless communication:			
Bandwidth	B	40 MHz	
Modulation	Q_m	QAM-64	
Spatial streams	N_{ss}	SISO	
Scaling factor	f	0.75	[22]
Maximum code rate	R_{max}	948/1024	[22]
Number of resource blocks	N_{RB}	106	[23]
Subcarrier spacing	SCS	30 kHz	[24]
OFDM numerology	μ	1	[24]
Average symbol duration	T_s	$0.0357 \cdot 10^{-3}$	[22]
Overhead value	OH	0.2	[22]
Power consumption:			
Local execution	p_{ex}	0.9 W	[26]
Transmission state	p_{tr}	1.2 W	[32]
Idle state	p_i	0.003 W	[33]
Power budget	A	8390 mAh	[25]
Operating voltage	V	3.3 V	[33]

Table II. Shifting the frequency range to mmWave values in 5G NR is advantageous in terms of throughput and latency but disadvantageous in terms of propagation because it increases the path loss [34]. Propagation modeling is one of the key parameters for understanding wireless communication. In this work, we assume that the communication channel is established and the BS knows which modulation is going to be used. In the direct propagation and big amount of antennas, the bit rate could bring the values up to 3260 Mbps. As an example, we use a device, which could not include more sophisticated antenna configurations yet, thus, bandwidth for data transmission is set to 40 MHz with only one spatial stream (Single-Input Single-Output (SISO)), and Quadrature Amplitude Modulation (QAM) – 64. Assume that fiber optic is used for wired communication between the BS and the MCC.

D. Numerical Results

Fig. 3 shows the performance parameters on the small workload (10 kb) that are referred to as the low intensive tasks. The power consumption spent for the local execution is higher than for data transmission. Task execution on the resource-limited device is not the best solution for the high workloads, as it could cause a system failure. Figures 4 and 5 show that with the increasing workload (1 Mb and 10 Mb), the offloaded strategies are winning in terms of consumed power that will increase the device lifetime.

E. System's Performance Bottlenecks

The system has two bottlenecks. The first bottleneck appears with the amount of computing capacity. In this work, we assume that the task could use all the computing capacity of the server. In real, half or more of CPU power could be allocated to other users' tasks. The computing time depends on CPU parameters, which means that if the device has a small computing capacity, e.g., IoT, it will not get more packages until the first one proceeds. So, when the processor is overloaded, it will filter the packets. The high overloading could lead the system failure. Important to mention that the quantity parameter of the CPU is measured in FLOPS. It mainly depends on the processor architecture, so the amount of bits processing in one register simultaneously is correlated with the exact CPU parameters [21].

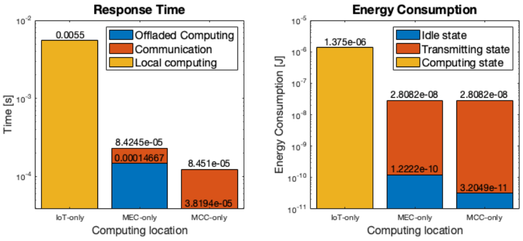


Fig. 3: The response time (T) and power consumption (P) for IoT-only, MEC-only, MCC-only with workload w = 0.01 Mb

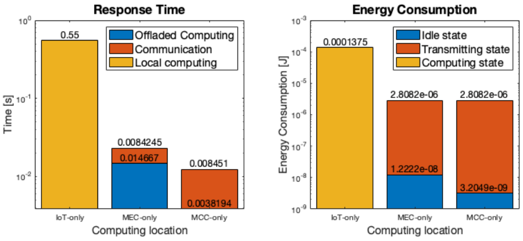


Fig. 4: The response time (T) and power consumption (P) for IoT-only, MEC-only, MCC-only with workload w = 1 Mb

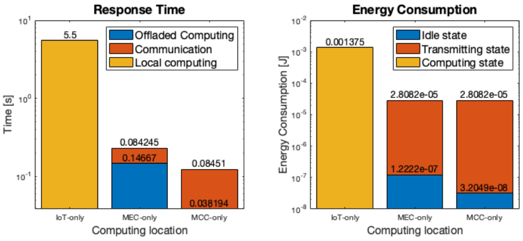


Fig. 5: The response time (T) and power consumption (P) for IoT-only, MEC-only, MCC-only with workload w = 10 Mb

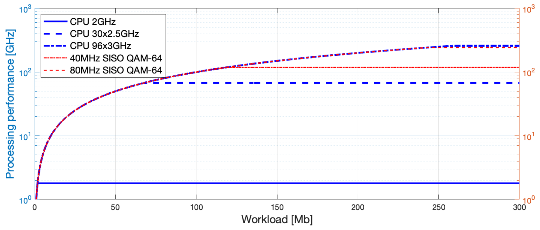


Fig. 6: The local computing and communication time limits

At the same time, MEC and MCC paradigms have incomparably more computational power. In these paradigms, the bottleneck also appears in the channel throughput. The communication time in the mentioned paradigms depends on the communication technology and channel parameters – bandwidth, modulation, and spatial streams. The system will not take more than it is allowed to process at one time or bring the system failure. Thus, the remaining part of the workload is waiting for when the channel will be free.

Fig. 6 shows that the tasks can be waiting in the queue due to processor overloading or channel overloading. In the MEC-only model, referred to as a blue dotted line in the figure, an error is likely to occur on the server since the channel throughput allows to send more data than the server could proceed. In the MCC-only model, referred to as a blue dash-dotted line in the figure, where computational power is much greater, transmission delays occur on the devices with simple communication parameters (e.g., SISO).

V. Conclusion

This work discussed three computing strategies and wireless communication problems in a MEC and MCC systems. Processing locally is the best choice in terms of security and response time as the data is not sent to the third-party, but it has strict limitation in the battery lifetime. MEC and MCC are perspective solutions to satisfy the requirements of medical applications running on energy-constrained IoT devices. Simulation results have revealed that MEC and MCC offloading strategies provide up to 100 times more computational capabilities (depending on the technical characteristics of the server). Latency optimization of the communication in the medical use cases refers to joint resource allocation and intelligent orchestration of the task. This work is a part of the research, so the future direction is to implement multiple users and servers in the system and improve the system by introducing offloading strategies for not-idealistic cases, i.e., partial offloading and constrained communication resources.

References

[1] "Robotic surgeries surge to 15% of all procedures, despite limited evidence." https://www.medtechdive.com/news/robotic-surgeries-surge-to-15-of-all-procedures-despite-limited-evidence/570370/. 2020.

[2] "Telehealth: A quarter-trillion-dollar post-COVID-19 reality?." https://www.mckinsey.com/industries/healthcare-systems-and-services/our-insights/telehealth-a-quarter-trillion-dollar-post-covid-19-reality. 2021.

[3] D. Alekseeva, A. Ometov, O. Arponen, and E. S. Lohan, "The Future of Computing Paradigms for Medical and Emergency Applications," *Computer Science Review*, vol. 45, p. 100494, 2022.

[4] E. Cuervo, A. Balasubramanian, *et al.*, "MAUI: Making Smartphones Last Longer With Code Offload," in *Proc. of 8th International Conference on Mobile Systems, Applications, and Services*, pp. 49–62, 2010.

[5] B. Zhou and R. Buyya, "Augmentation Techniques For Mobile Cloud Computing: A Taxonomy, Survey, And Future Directions," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–38, 2018.

[6] L. Sun, X. Jiang, H. Ren, and Y. Guo, "Edge-Cloud Computing and Artificial Intelligence in Internet of Medical Things: Architecture, Technology and Application," *IEEE Access*, pp. 101079–101092, 2020.

[7] S. Shahhosseini, A. Kanduri, *et al.*, "Towards Smart and Efficient Health Monitoring Using Edge-Enabled Situational-Awareness," in *Proc. of 3rd International Conference on Artificial Intelligence Circuits and Systems*, pp. 1–4, IEEE, 2021.

[8] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds," *IEEE Trans. on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, 2016.

[9] Z. Zhang, "A Computing Allocation Strategy For Internet of Things' Resources Based on Edge Computing," *International Journal of Distributed Sensor Networks*, vol. 17, no. 12, p. 15501477211064800, 2021.

[10] J. Wang, L. Zhao, *et al.*, "Smart Resource Allocation For Mobile Edge Computing: A Deep Reinforcement Learning Approach," *IEEE Trans. on emerging topics in computing*, vol. 9, no. 3, pp. 1529–1541, 2019.

[11] Y. Pan, T. Qu, *et al.*, "Digital Twin Based Real-Time Production Logistics Synchronization System in a Multi-Level Computing Architecture," *Journal of Manufacturing Systems*, vol. 58, pp. 246–260, 2021.

[12] J. Wang and L. Wang, "A Computing Resource Allocation Optimization Strategy for Massive Internet of Health Things Devices Considering Privacy Protection in Cloud Edge Computing Environment," *Journal of Grid Computing*, vol. 19, no. 2, pp. 1–14, 2021.

[13] N. Biswas, H. Mirghasemi, and L. Vandendorpe, "Sharing is Caring: A Mobile Edge Computing Perspective," in *Proc. of 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1298–1303, IEEE, 2021.

[14] Z. Ning, P. Dong, *et al.*, "Mobile Edge Computing Enabled 5G Health Monitoring For Internet of Medical Things: A Decentralized Game Theoretic Approach," *IEEE J. on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2020.

[15] A. Ali, M. M. Iqbal, *et al.*, "An Efficient Dynamic-Decision Based Task Scheduler for Task Offloading Optimization and Energy Management in Mobile Cloud Computing," *Sensors*, vol. 21, no. 13, p. 4527, 2021.

[16] S. Tuli, S. R. Poojara, *et al.*, "COSCO: Container Orchestration Using Co-Simulation and Gradient Based Optimization for Fog Computing Environments," *IEEE Trans. on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 101–116, 2021.

[17] S. Wang, N. Li, *et al.*, "Collaborative Physical Layer Authentication in Internet of Things Based on Federated Learning," in *Proc. of 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 714–719, IEEE, 2021.

[18] Q. He, G. Cui, *et al.*, "A Game-Theoretical Approach for User Allocation in Edge Computing Environment," *IEEE Trans. on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 515–529, 2019.

[19] F. B. Dihn, A. A. Razzac, *et al.*, "Adaptive Data Replication for URLLC in Cooperative 4G/5G Networks," in *Proc. of 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1376–1381, IEEE, 2021.

[20] A. P. Miettinen and J. K. Nurminen, "Energy Efficiency of Mobile Clients in Cloud Computing," in *Proc. of 2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10)*, 2010.

[21] S. Rumley, K. Bergman, M. A. Seyedi, and M. Fiorentino, "Evolving Requirements and Trends of HPC," in *Springer Handbook of Optical Networks*, pp. 725–755, Springer, 2020.

[22] ETSI TS 138 306 V15.3.0, "5G; NR; User Equipment (UE) radio access capabilities (3GPP TS 38.306 v. 15.3.0 Rel. 15)," October 2018.

[23] ETSI TS 138 101-1 V16.4.0, "5G; NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone (3GPP TS 38.101-1 v. 16.4.0 Rel. 16)," July 2020.

[24] ETSI TS 138 211 V16.2.0, "5G; NR; Physical channels and modulation (3GPP TS 38.211 v. 16.2.0 Rel. 16)," July 2020.

[25] M. Lauridsen, R. Krigslund, *et al.*, "An Empirical NB-IoT Power Consumption Model For Battery Lifetime Estimation," in *Proc. of 87th Vehicular Technology Conference (VTC Spring)*, IEEE, 2018.

[26] H. Wu, K. Wolter, *et al.*, "EEDTO: An Energy-Efficient Dynamic Task Offloading Algorithm For Blockchain-Enabled IoT-Edge-Cloud Orchestrated Computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2163–2176, 2020.

[27] 3GPP TR 22.826 V17.2.0, "Study on Communication Services for Critical Medical Applications." Rel. 17, March 2021.

[28] 3GPP TS 22.104 V18.0.0 , "Service Requirements for Cyber-Physical Control Applications in Vertical Domains." Rel. 18, March 2021.

[29] "Qualcomm APQ8053 SoC." https://www.qualcomm.com/products/technology/processors/application-processors/apq8053#Overview. 2022.

[30] "1U Rackmount AMD Ryzen Server." https://www.onlogic.com/eu-en/mk100b-40/. 2022.

[31] "Amazon EC2 Instance Types." https://aws.amazon.com/ec2/instance-types/. 2022.

[32] B. Dusza, C. Ide, *et al.*, "CoPoMo: a Context-Aware Power Consumption Model For LTE User Equipment," *Trans. on Emerging Telecommunications Technologies*, vol. 24, no. 6, pp. 615–632, 2013.

[33] 3GPP TR 45.820 V13.1.0 (2015-11), "Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT) (Rel. 13)," November 2015.

[34] T. S. Rappaport, S. Sun, *et al.*, "Millimeter Wave Mobile Communications For 5G Cellular: It Will Work!," *IEEE Access*, vol. 1, pp. 335–349, 2013.