Understanding the Performance of Task Offloading for Wearables in a Two-Tier Edge Architecture

Understanding the Performance of Task Offloading for Wearables in a Two-Tier Edge Architecture

Waleed Bin Qaim[1,2], Aleksandr Ometov[1], Claudia Campolo[2], Antonella Molinaro[2], Elena Simona Lohan[1], Jari Nurmi[1]

[1] Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland
[2] DIIES Department, Mediterranea University of Reggio Calabria/CNIT, Reggio Calabria, Italy
Corresponding author: Waleed Bin Qaim (waleed.binqaim@tuni.fi)

***Abstract*—The development of small form-factor handheld electronics is pacing the personal devices market, followed by the increasing number of various applications. Some of those applications also cover computation-hungry use-cases, such as image or video processing and compression, among others. Historically, wearable and handheld devices were not designed to execute computationally intensive operations for reasons ranging from limited battery capacity to radiated heat. Offloading computationally heavy tasks to a comparatively more powerful and less energy-dependent device can help prolong the battery lifetime of a wearable. This paper analyzes different task offloading scenarios from the wearable to a device located at the network edge. Such a device can be a smartphone paired with the wearable or an edge server co-located with the cellular base station. A comprehensive performance evaluation conducted under a wide variety of realistic settings in terms of task demands, processing capabilities, and data rate, unveils the circumstances in which offloading is convenient and when it is not, in terms of meaningful metrics.**

Keywords—Task Offloading, Edge, Computing, Wearables, Internet of Things

I. Introduction

With the technological development and Information and Communication Technology (ICT) evolution, wearable devices such as smartwatches, smart glasses, smart shoes, wrist bands, etc., are getting increasingly popular these days [1, 2]. These devices are capable of connecting to the Internet directly or through a gateway such as a user smartphone, hence they are part of the general Internet of Things (IoT) architecture that aims to connect millions of objects to the Internet for seamless access and control [3]. Wearables are getting more and more powerful and equipped with plenty of sensors capable of providing a plethora of different applications for everyday use [4]. These applications are aimed to improve the overall quality of life [5] and include, for instance, health monitoring, activity recognition, localization, and tracking, as well as various gaming and fun applications [6, 7].

However, wearables still face numerous challenges such as those related to security and privacy aspects, form factor, weight, comfort, etc. [8]. In particular, the major bottleneck lies in the limited computational power and battery lifetime of the devices [9, 10], which restrict their usage. Therefore, as the need for more and more processing power by applications increases, the device's power consumption also gets higher hence downgrading the overall user experience.

Over the years, many solutions have been proposed to conserve the limited resources of wearables [11]. More recently, the concept of Mobile Edge Computing (MEC) has emerged, allowing resource-constrained mobile devices to offload computationally intensive tasks to nearby devices, e.g., edge servers co-located with Base Stations (BSs), Access Points (APs), having high energy and computational capabilities [12].

More in detail, task offloading to the edge refers to the process of transferring some input data over wireless links to a comparatively more powerful edge server, where they are processed, and getting the results back to the requesting device [13, 14].

Besides purpose-built edge servers, today's smartphones are equipped with powerful chipsets employing multicore processors, thus, acting themselves as edge devices and enabling users to enjoy many complex and computationally intensive applications on wearables through the offloading of tasks such as image processing, machine learning algorithms for face, text, and activity recognition, augmented reality applications, etc.; which would quickly deplete the battery of a mobile device otherwise [15, 16].

Several works connected to the area of task offloading for wearables and other resource-constrained devices are proposed in the literature, e.g., [16–21].

In general, task offloading helps increase resource-constrained devices' storage and computing capabilities as added benefits apart from energy conservation. However, the achieved benefits are not always straightforward and need to be analyzed case-by-case to decide *when, what, and where* to offload. Indeed, in some cases, energy spent by the wearable in communication towards the task executor can be much higher than the energy spent for local computation, thus making task offloading not convenient. Additionally, under some circumstances, task offloading may increase the overall task accomplishment time due to multiple reasons, for instance, due to a large amount of input data to be transferred over a low throughput wireless link [22]. Hence, it is crucial to estimate the benefit of task offloading in terms of energy consumption and whether the latency demands of applications are met.

In such a context, complementing the existing works, the main contributions of this paper are as follows:

- We study a two-tier edge architecture for task offloading from wearables to the edge, which encompasses a smartphone and an edge server as candidate task executors.
- We analyze the performance of task execution at wearable devices by identifying the limits.
- We explain to which extent and under which circumstances task offloading to the edge may bring improvements. To this aim, we analyze two meaningful metrics, i.e., task accomplishment time and energy consumption due to mobile devices' computing and communication procedures involved in the task offloading (i.e., wearable and smartphone). Such analysis is provided through a comprehensive and flexible analytical playground under a large variety of realistic settings regarding computing task requirements, device capabilities, and the distance between involved devices.

The rest of the paper is organized as follows. In Section II, we present the system model. Section III provides the numerical results and performance evaluation of the proposed model, and Section IV concludes the work with our major findings and hints on future works.

II. System Model

This section presents the reference architecture followed by the main assumptions and mathematical formulation for deriving the performance metrics of interest.

A. Reference architecture and main assumptions

As an example, we consider an architecture involving a wearable device such as Google Glass [23], paired with the user's smartphone, through a short-range wireless link, acting as a gateway towards the Internet, as shown in Figure 1. The smartphone, in its turn, is connected through a BS that hosts an edge server. Without loss of generality, we refer to the glass as the wearable device which needs to run a processing-hungry task, such as an Augmented Reality (AR) glass's image and video processing/streaming. The user can use the AR glass to capture image stream/video on the go for several applications, such as face recognition, automatic license plate recognition, etc.

Such a computing task is expressed in terms of the input data (e.g., the image to be processed), D (in bits), and the number of CPU cycles/bit required to process the task, C. A program profiler monitoring all the program parameters can be used to estimate C [24]. Program profilers utilize information such as acquired memory, execution time, thread CPU time, number and type of instructions, and function calls, for this purpose [25].

As depicted in Figure 1, any image processing is a computationally intensive task for a wearable device if carried out locally, see scenario (1). Therefore, we try to analyze whether offloading the task from the wearable to nearby devices such as the smartphone (2) and edge server (3) can help conserve the

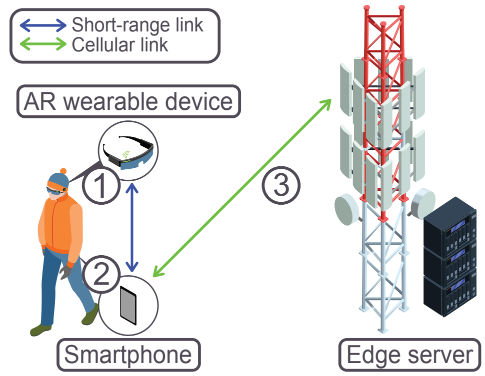

Fig. 1: Reference architecture and scenarios of interest: (1) – Local execution on wearable; (2) – Offloading to the smartphone; (3) – Offloading to the edge server.

energy resources of the wearable device while also satisfying the latency requirements for the application. For this purpose, we analyze the energy consumption and task accomplishment time performance for three different scenarios. The wearable can carry out local processing, resulting in increased task accomplishment time due to the low computational capacity, thus, degrading the overall user experience. Alternatively, the wearable can choose to offload the task to the smartphone or edge server with comparatively much higher computation resources at the cost of extra energy spent by the wearable to transfer data to the task executor and receive the processed results.

The following main assumptions hold for our study:

- In contrast to conventional wearables, many latest wearable devices are equipped with multiple connectivity options such as Bluetooth, Bluetooth Low Energy (BLE), Wi-Fi, millimeter Wave, and/or Long Term Evolution (LTE) communication interfaces [26]. However, low power technologies such as Bluetooth/BLE are not feasible for task offloading due to their low data rates, thus, making task offloading completely infeasible due to huge communication delays [27]. Therefore, for an outdoor scenario, we assume that the wearable device connects to the user's smartphone over Wi-Fi, further accessing the edge server through a cellular network.
- The targeted task is atomic in the sense that it can not be further divided into subtasks.
- The output data size is much smaller than the input data in applications such as face recognition, automatic license plate recognition, etc. Therefore, the time for transferring it from the task executor (smartphone/edge server) to the wearable can be safely neglected [16, 28].

B. Mathematical Formulations

A theoretical background is leveraged to estimate the metrics of interest for all three cases, i.e., (1) local task execution on the wearable, (2) offloading to the smartphone, and (3) offloading to the edge server. In particular, we derive the task accomplishment time and the energy consumption spent for the overall task execution. The main notations used throughout this paper are summarized in Table I.

1) Local task execution on the wearable: A wearable device operates independently and carries out all the execution locally without offloading.

Task Accomplishment Time: The task accomplishment time for executing a task locally on the wearable device T_w can be estimated as follows [20]:

$$T_w = \frac{D \times C}{F_w}, \tag{1}$$

where F_w denotes the processing power available on the wearable device in terms of CPU cycles per second.

Energy Consumption: This metric can be estimated by considering the CPU power consumption that is proportional to the product of CPU frequency F_w and square of supply voltage to the chip, V^2, similar to the work in [29]. Hence, the power consumption can be expressed as:

$$P_w = \alpha(V^2 \times F_w), \tag{2}$$

where α is the effective switched capacitance of each processor, which is related to the chip architecture [30]. Moreover, it has been found that the voltage supply V is approximately linearly proportional to the clock frequency of the CPU [29]. Thus, equation (2) can be rewritten as:

$$P_w = \alpha F_w^3. \tag{3}$$

Therefore, for an input data size of D bits and the computational intensity of the task C cycles/bit, the energy consumption for executing a task locally on the wearable device, E_w, can be estimated as:

$$E_w = P_w \times T_w = \alpha F_w^2 (D \times C). \tag{4}$$

2) Task offloading to smartphone: A wearable is usually coupled with the user's smartphone (see Figure 1, scenario 2), the nearest device available for task offloading with comparatively higher resources than a wearable.

Task Accomplishment Time: The total time consumed in offloading a task for execution at the smartphone T_s can be defined as the sum of the time consumed in input data delivery to the smartphone over the Wi-Fi link, $T_{d,s}$, and the task execution delay at the smartphone, $T_{ex,s}$:

$$T_s = T_{d,s} + T_{ex,s}. \tag{5}$$

The data rate for the wearable device, R_w, to offload a task for execution at the smartphone over Wi-Fi can be estimated as follows [31]:

$$R_w = \frac{DS * M * CR * SS}{SI}, \tag{6}$$

where DS represents the number of data subcarriers that transmit modulated data, M represents the modulation order in terms of the number of bits each data subcarrier can represent, CR represents the coding rate, SS defines the number of spatial streams used, and SI is the symbol interval time. An upper bound of 54Mbps can be achieved for a Wi-Fi (802.11g) link based on the values of the above parameters as mentioned in Table I.

Hence, the transmission time, $T_{d,s}$, for offloading a task from the wearable device to the smartphone over the Wi-Fi interface would be:

$$T_{d,s} = \frac{D}{R_w}. \tag{7}$$

Similar to equation (1), the computation delay for executing a task at the smartphone $T_{ex,s}$ is given as:

$$T_{ex,s} = \frac{D \times C}{F_s}. \tag{8}$$

Energy consumption: The overall energy consumption in offloading a task for execution at the smartphone, E_s can be expressed as:

$$E_s = E_{t,w} + E_{r,s} + E_{ex,s} + E_{w,idle}, \tag{9}$$

where $E_{t,w}$ is the energy consumed by the wearable to transmit input data to the smartphone as:

$$E_{t,w} = \frac{P_{t,w} \times D}{R_w}. \tag{10}$$

The energy consumed by the smartphone to receive input data from the wearable is calculated as:

$$E_{r,s} = \frac{P_{r,s} \times D}{R_w}, \tag{11}$$

where $P_{r,s}$ is the instantaneous power spent during reception over Wi-Fi as per the measured values in [32].

The energy consumed in executing the task on the smartphone is given as:

$$E_{ex,s} = \alpha F_s^2 (D \times C). \tag{12}$$

Finally, the energy spent at the wearable device during idling, while the task gets executed at the smartphone, can be estimated as:

$$E_{w,idle} = P_{w,idle} \times T_{ex,s}. \tag{13}$$

where $P_{w,idle}$ is the power spent by the wearable in idle state.

3) Task offloading to the edge server: Computationally heavy tasks that will consume much higher resources if executed locally can be offloaded to the edge server from the wearable, as in Figure 1, scenario 3. If the task is offloaded to the edge server, the smartphone will act as a relay node and receive the input data from the glass and forward it to the edge server and vice versa for communicating the results back.

Task Accomplishment Time: The total time consumed in offloading the task from wearable to the edge server can be defined as:

$$T_e = T_{d,s} + T_{d,e} + T_{ex,e}, \tag{14}$$

where $T_{d,e}$ is the time taken in offloading the task from the smartphone to the edge server over the cellular network, and $T_{ex,e}$ is the time consumed in executing the task at the edge server. Without loss of generality, we refer to the LTE technology for the cellular network. For the uplink transmission from the smartphone to the edge server, the intracell interference is well mitigated in the LTE network [33]. Therefore, the data rate experienced by the smartphone can be given as [20]:

$$R_s = W_s \log_2 \left(1 + \frac{P_{t,s} H_{s,e}}{N_o}\right), \tag{15}$$

where W_s gives the user bandwidth, and $P_{t,s}$ denotes the transmission power of the smartphone, $H_{s,e}$ denotes the channel gain from the smartphone to the BS, including path loss and fading, and N_o is the Gaussian noise power in the channel. Channel gain $H_{s,e}$ is the reciprocal of path loss. As per the 3GPP standardization [34], for a general non-line-of-sight (NLOS) case, the path loss can be estimated as:

$$PL_{NLOS}(d) = 36.7 \log_{10}(d) + 26 \log_{10}(f_c) + 22.7, \tag{16}$$

where d is the distance between the smartphone and the BS (in meters) and f_c is the carrier frequency (in GHz)[1].

Hence, $T_{d,e}$ can be given as:

$$T_{d,e} = \frac{D}{R_s}, \tag{17}$$

and, similarly to equation (1), the computation delay for executing the task at the edge T_e can be estimated as:

$$T_{ex,e} = \frac{D \times C}{F_e}, \tag{18}$$

where F_e is the computational capacity of the edge server.

Energy consumption: The overall energy consumption in offloading a task for execution at the edge server can be expressed as:

$$E_e = E_{t,w} + E_{r,s} + E_{d,e} + E_{ex,e} + E_{w,idle} + E_{s,idle}, \tag{19}$$

where $E_{d,e}$ is the corresponding energy consumption at the smartphone for delivering the input data to the edge server over cellular network and can be expressed as:

$$E_{d,e} = \frac{P_{t,s} \times D}{R_s}. \tag{20}$$

The smartphone energy consumed while idling, i.e., when the task is executed at the edge server, is calculated as:

$$E_{s,idle} = P_{s,idle} \times T_{ex,e}. \tag{21}$$

The energy consumed at the wearable device (the task is offloaded from the smartphone to the edge server and executed at the edge server) is calculated as:

$$E_{w,idle} = P_{w,idle} \times (T_{d,e} + T_{ex,e}). \tag{22}$$

Finally, the energy consumed in executing the task on the edge server is:

$$E_{ex,e} = \alpha F_e^2 (D \times C). \tag{23}$$

However, being co-located with the BS with no huge energy constraint, compared to the other battery-powered devices in our system model, namely the wearable and the smartphone, the energy consumed in receiving input data from the smartphone and processing it on the edge server, is considered negligible.

[1]The estimation in equation (16) is applicable for the carrier frequency range of 2–6GHz for different antenna heights with the maximum modeling distance range of 2,000 m between the mobile station and the base station, which suits our considered scenario [34].

TABLE I: Main system parameters

Notation	Description	Values [Ref.]
D	Input data size	0.2-2 MB [15]
C	Task computational intensity	10^3 cycles/bit [35, 36]
F_w	Computational capacity of the wearable device	1GHz [23]
F_s	Computational capacity of the smartphone	2.2GHz [37]
F_e	Computational capacity of the edge server	20GHz [20]
DS	Number of data subcarriers over Wi-Fi channel	48 [31]
M	Number of bits each data subcarrier represents	6 [31]
CR	Coding rate	3/4 [31]
SS	Number of spatial streams used	1 [31]
SI	Symbol interval time	$4\mu s$ [31]
$P_{w,idle}$	Idle power consumption at the wearable	22mW [38]
$P_{t,w}$	Wearable transmission power over Wi-Fi	1.28W [32]
α	Effective switched capacitance constant	10^{-28} [30]
f_c	Carrier frequency for communication between smartphone and LTE base station	2.1GHz [39]
W_s	Channel bandwidth for smartphone over cellular network (assuming system bandwidth of 20MHz and 20 users/cell)	1MHz [20]
$P_{s,idle}$	Idle power consumption at the smartphone (with display turned off)	30mW [40–42]
$P_{t,s}$	Transmission power of smartphone over cellular network	0.2W [20]
$P_{r,s}$	Reception power of smartphone over Wi-Fi	0.94W [32]
No	Noise power over cellular channel	−113dBm [36]
d	Distance between the smartphone and the BS	100, 300, 600 m [20]

III. Numerical Results

Different sets of numerical results are derived under settings summarized in Table I, unless separately stated.

A. Local task execution on the wearable

Figure 2 shows the values of the task accomplishment time on the wearable device (no offloading) while increasing

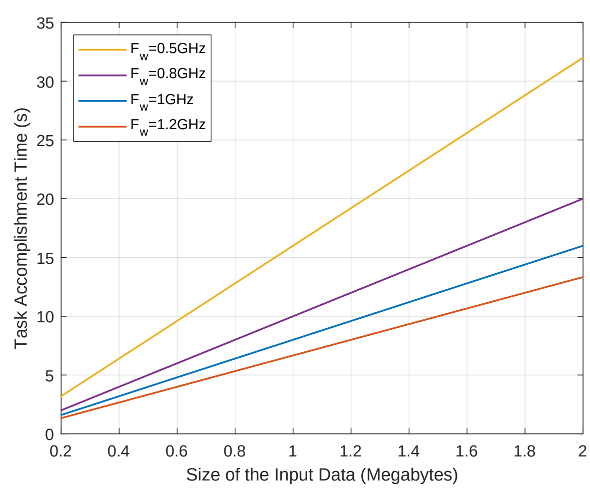

Fig. 2: Task accomplishment time for local task execution on the wearable with varying CPU frequencies and input data sizes.

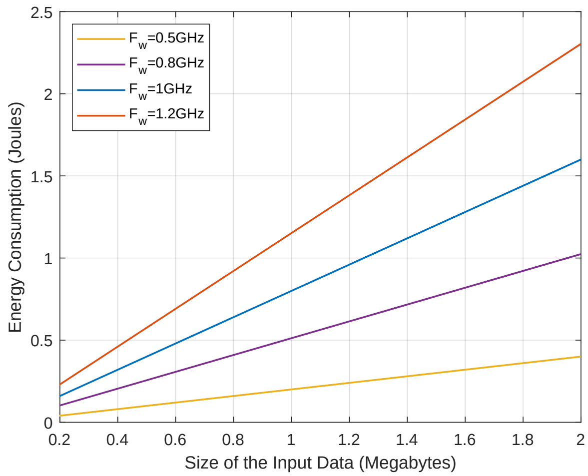

Fig. 3: Energy consumption for local task execution on the wearable with varying CPU frequencies and input data sizes.

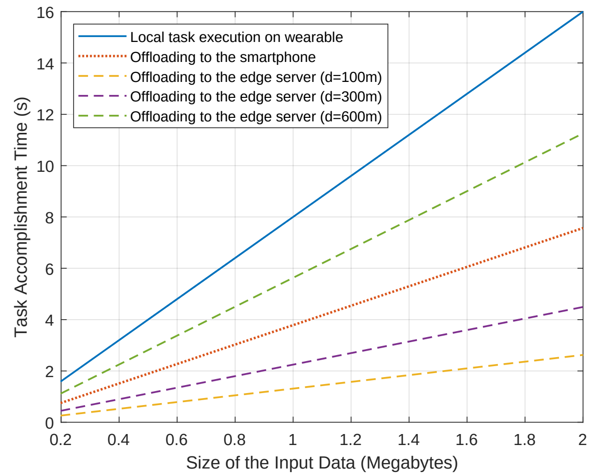

Fig. 4: Task accomplishment time with varying input data sizes for: *(1)* local task execution at the wearable, *(2)* task offloading to the smartphone, *(3)* task offloading to the edge server ($d = 100,\ 300,\ 600$m).

the input data size and for several computational capabilities of the wearable, in terms of CPU frequency, in the range of 0.5GHz to 1.2GHz. Evidently, higher CPU frequencies allow achieving lower task accomplishment time. Hence, task offloading is more beneficial for those devices that have lower computational capacity.

Figure 3 shows the corresponding energy consumption on the wearable device with local computation. A device with a higher CPU frequency can achieve lower task execution time but at the cost of high energy consumption since it is directly proportional to the square of the CPU frequency as follows from equation (4).

B. Local task execution vs. offloading to the edge

In the second set of results, first, we analyze the overall time spent in the execution of a task (when varying input data sizes) for the discussed three scenarios, i.e., when the task is executed locally on the wearable device, when offloading to the smartphone (through a 54Mbps Wi-Fi link), and when offloading to the edge server co-located with a BS that may be located at different distances (100m, 300m, and 600m) from the smartphone (through an LTE link). Different distance settings are considered since, in reality, a user carrying the wearable device and the paired smartphone may be located in different positions within the LTE cell, hence experiencing different radio link performance.

As expected, Figure 4 shows that with an increasing input data size, the task execution time increases for all scenarios. Local task execution on the wearable device performs worst due to the smallest computational resources compared to other scenarios. Whereas, offloading to the edge server when the user is far from the BS shows the second-worst performance. The reason is that when the smartphone gets further from the BS, the link quality degradation causes a significant decrease in the data rate, thus, resulting in prolonged task execution time. Offloading to the edge server when the user is closer to the LTE BS shows the best performance, both due to high data rates achieved in communication and the high computational resources available at the edge server. Offloading to the smartphone performs somewhere in the middle as compared to other scenarios. However, varying data rates over Wi-Fi (54Mbps in our case) certainly will have an impact on the task accomplishment time. Therefore, it might

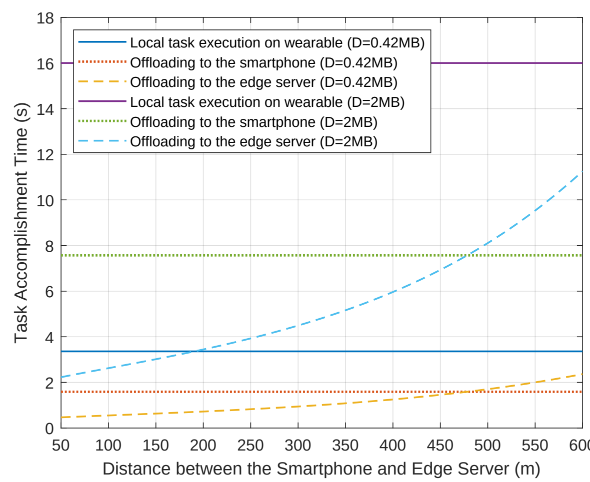

Fig. 5: Task accomplishment time for two different input data sizes ($D = 0.42$MB and $D = 2$MB) when varying distance between the smartphone and edge server for: *(1)* local task execution at the wearable, *(2)* task offloading to the smartphone, *(3)* task offloading to the edge server.

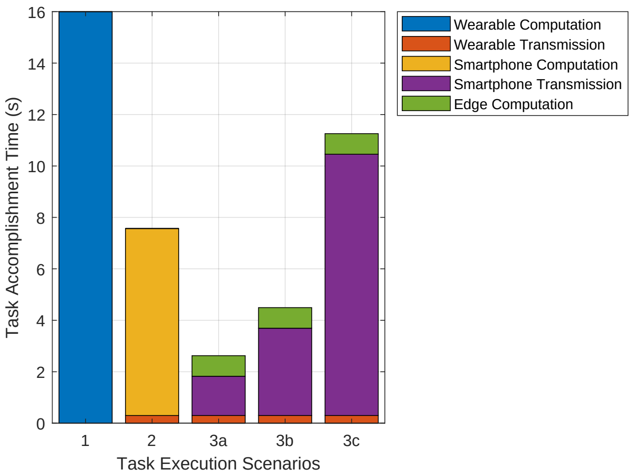

Fig. 6: Task accomplishment time breakdown for an input data size of 2MB for: *(1)* local task execution at the wearable, *(2)* task offloading to the smartphone, *(3a)* task offloading to the edge server (d=100m), *(3b)* task offloading to the edge server (d=300m), *(3c)* task offloading to the edge server (d=600m).

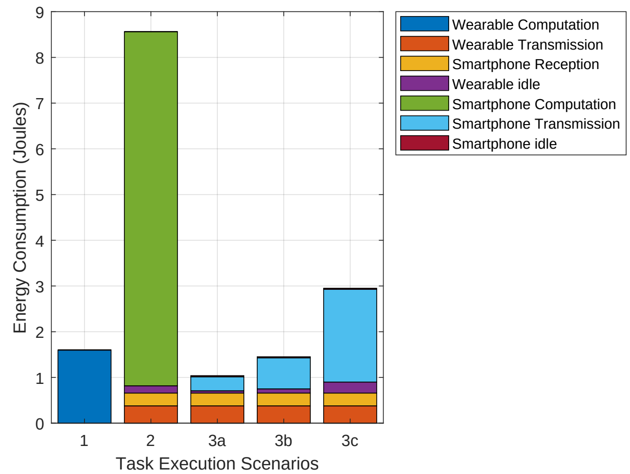

Fig. 7: Energy consumption breakdown for an input data size of 2MB for: *(1)* local task execution at the wearable, *(2)* task offloading to the smartphone, *(3a)* task offloading to the edge server (d=100m), *(3b)* task offloading to the edge server (d=300m), *(3c)* task offloading to the edge server (d=600m).

be beneficial to offload time-critical tasks to meet the desired latency requirements as well as to conserve energy on the wearable device.

Figure 5 depicts the task accomplishment time when varying the distance, d, between the smartphone and the BS, the edge server is co-located with. Here, curves for two different input data sizes, i.e., $D = 0.42$MB and $D = 2$MB, have been shown, corresponding to small and large input data, respectively. Offloading to the edge server becomes significantly costly as the user is far from the server and for large input data since more traffic needs to be exchanged over the wireless short-range and long-range links. Moreover, it can be observed that a task accomplishment time below 1s is achievable for smaller input data sizes when the task is offloaded to a close edge server.

It is also relevant to observe each operational phase's contribution, i.e., communication and computation, to the overall task accomplishment time and energy consumption. Figure 6 shows the corresponding breakdown of the task accomplishment time for all three task execution scenarios. Notably, the smartphone transmission time is comparatively higher as compared to the wearable transmission time due to the reduced data rate over a shared cellular network, which further increases as the user gets farther from the BS. Finally, the edge computation time is much smaller as compared to all other components because the edge server has the highest computational resources in the system model.

Figure 7 shows the energy consumption breakdown for all the three task execution scenarios. In the case of local execution on the wearable device, the total energy consumption originates only from computation at the wearable device. The cumulative system energy consumed in offloading the task to the smartphone comprises energy spent by the wearable in transmitting the input data to the smartphone, energy expended by the smartphone in receiving the input data from the wearable device, and in executing the task. Additionally, while the task gets executed at the smartphone, the wearable operates in idle mode, thus, consuming some energy. Interestingly, the computation part is significantly higher as compared to communication for the second scenario. Finally, offloading the task to the edge server involves energy spent by the wearable in transmitting the input data to the smartphone, energy consumed by the smartphone to receive input data from the wearable over the short-range link, and transmission further towards the edge server over the long-range link. In this case, the wearable shows idle energy consumption until the output of the task is sent back to it from the smartphone. In comparison, the smartphone remains idle when the task gets executed on the edge server, which is comparatively much smaller.

C. Impact of task processing requirements

The nature of a task in terms of processing demands, besides the input data size, also highly affects the overall task accomplishment time and energy consumption.

Figure 8 shows the effect of varying the computational intensity on the overall task accomplishment time for the three task execution scenarios. For the case of local execution on the wearable, it can be observed that task execution time increases significantly with the increase in computational intensity due to the heavier processing load on the resource-limited wearable. Moreover, it can be observed that offloading to the smartphone always seems convenient for the wearable as the task computational intensity increases. However, based on the user's distance from the BS, offloading to the edge server becomes

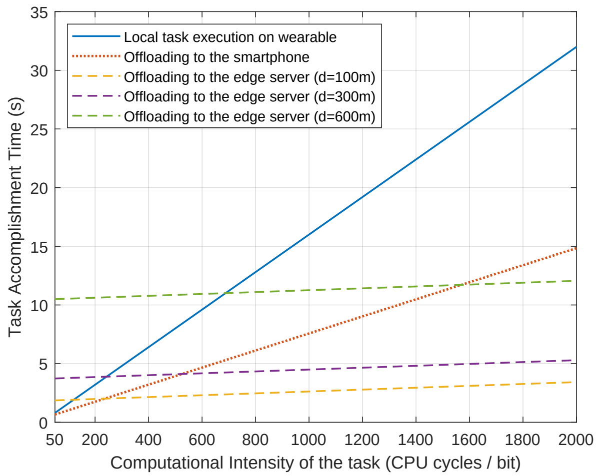

Fig. 8: Task accomplishment time for an input data size of 2MB with varying task computational intensity for: *(1)* local task execution at the wearable, *(2)* task offloading to the smartphone, *(3)* task offloading to the edge server ($d = 100$, 300, 600m).

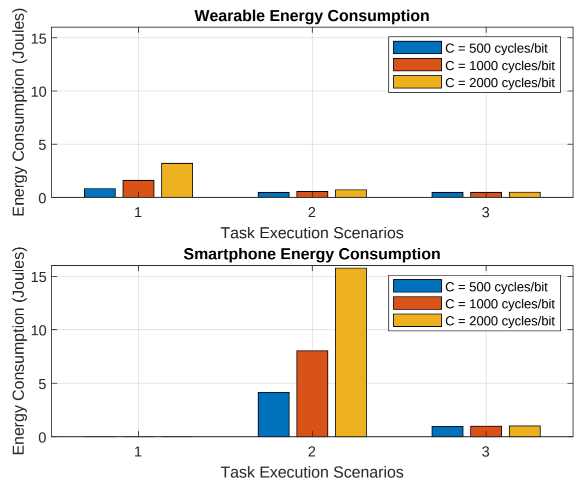

Fig. 9: Energy consumption per device for an input data size of 2MB with varying task computational intensity for: *(1)* local task execution at the wearable, *(2)* task offloading to the smartphone, *(3)* task offloading to the edge server ($d = 300$m).

more beneficial as the computational intensity increases. For instance, even when a user is far from the BS, offloading to the edge server performs better than offloading to the smartphone for very computationally intensive tasks, i.e., for 1600 CPU cycles/bit and beyond. This is because, thanks to the larger processing capabilities of the edge server, the execution time is significantly shorter than the time spent for transmitting over a low-throughput long-range link.

Finally, Figure 9 shows the corresponding variation in the overall energy consumption for the wearable and the smartphone for different computational intensities ($C = 500$, 1000, 2000 CPU cycles/bit) for the three offloading scenarios. Not surprisingly, as the computational intensity increases, the energy consumption gets higher at the devices where the task is executed. Moreover, it is worth observing that the energy spent by the smartphone in case of offloading to it is significantly higher than the energy spent by the wearable when it executes the task locally. However, this is not a big issue since smartphones are typically less energy-constrained than wearables. Energy consumption values are significantly low and almost comparable (with larger values for the smartphone) at both the wearable and the smartphone when the task is offloaded to the edge server.

IV. Conclusion and Future Work

In this paper, we analyzed the performance of task offloading for wearables in terms of task accomplishment time and energy consumption in a two-tier edge architecture involving a smartphone and an edge server as task executors. Our findings reveal that offloading to the smartphone is always more convenient than local execution at the wearable both to conserve the limited energy resources at the wearable and to experience a lower latency in accomplishing the task.

Particularly, offloading to the edge server is almost always better than executing at the smartphone. However, suppose the smartphone is at the cell border (experiencing harsh propagation conditions), it is convenient to execute the task at the smartphone to reduce the task accomplishment time, unless the smartphone is low on battery or the task is significantly computation-heavy. However, we expect that this downside of task offloading, which is due to costly long-range communication, can be minimized by improving cellular connectivity of smartphones, e.g., through Reflective Intelligent Surfaces (RIS) or by leveraging edge capabilities provided by high-density small cells.

Local task execution at the wearable is preferred over offloading for not computation-heavy tasks. The reason behind is that the delay contribution due to the transfer of input data over wireless links may dominate the overall task accomplishment time.

As part of future work, we plan to conduct a simulation-based study to validate the theoretical analysis and jointly optimize the task accomplishment time and energy consumption for both the wearable and smartphone. Additionally, a split computing strategy to execute tasks partially at the edge server and the smartphone/wearable, considered particularly promising within the sixth generation (6G) realm, is another meaningful research direction.

Acknowledgment

The authors gratefully acknowledge funding from European Union's Horizon 2020 Research and Innovation programme under the Marie Sklodowska Curie grant agreement No. 813278 (A-WEAR, http://www.a-wear.eu/).

REFERENCES

[1] "Gartner, Global End-User Spending on Wearable Devices to Total $52 Billion in 2020." [Online] Available: https://www.gartner.com/en/newsroom/press-releases/2019-10-30-gartner-says-global-end-user-spending-on-wearable-dev (Accessed May 30, 2022).

[2] A. Ometov, V. Shubina, L. Klus, J. Skibińska, S. Saafi, P. Pascacio, L. Flueratoru, D. Q. Gaibor, N. Chukhno, O. Chukhno *et al.*, "A Survey on Wearable Technology: History, State-of-the-Art and Current Challenges," *Computer Networks*, vol. 193, p. 108074, 2021.

[3] F. J. Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey," *IEEE Access*, vol. 8, pp. 69 200–69 211, 2020.

[4] L. Piwek, D. A. Ellis, S. Andrews, and A. Joinson, "The Rise of Consumer Health Wearables: Promises and Barriers," *PLoS medicine*, vol. 13, no. 2, p. e1001953, 2016.

[5] J. Lee, D. Kim, H.-Y. Ryoo, and B.-S. Shin, "Sustainable Wearables: Wearable Technology for Enhancing the Quality of Human Life," *Sustainability*, vol. 8, no. 5, p. 466, 2016.

[6] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A Survey of Wearable Devices and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2573–2620, 2017.

[7] D. Kozyrev, A. Ometov, D. Moltchanov, V. Rykov, D. Efrosinin, T. Milovanova, S. Andreev, and Y. Koucheryavy, "Mobility-Centric Analysis of Communication Offloading for Heterogeneous Internet of Things Devices," *Wireless Communications and Mobile Computing*, 2018.

[8] J. Williamson, Q. Liu, F. Lu, W. Mohrman, K. Li, R. Dick, and L. Shang, "Data Sensing and Analysis: Challenges for Wearables," in *Proc. of 20th Asia and South Pacific Design Automation Conference*. IEEE, 2015, pp. 136–141.

[9] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Challenges and Opportunities in Edge Computing," in *Proc. of IEEE International Conference on Smart Cloud*. IEEE, 2016, pp. 20–26.

[10] R. Rawassizadeh, B. A. Price, and M. Petre, "Wearables: Has the Age of Smartwatches Finally Arrived?" *Communications of the ACM*, vol. 58, no. 1, pp. 45–47, 2014.

[11] W. B. Qaim, A. Ometov, A. Molinaro, I. Lener, C. Campolo, E. S. Lohan, and J. Nurmi, "Towards Energy Efficiency in the Internet of Wearable Things: A Systematic Review," *IEEE Access*, 2020.

[12] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[13] M. Chen and Y. Hao, "Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[14] A. Ometov, O. Chukhno, N. Chukhno, J. Nurmi, and E. S. Lohan, "When Wearable Technology Meets Computing in Future Networks: A Road Ahead," in *Proc. of 18th ACM International Conference on Computing Frontiers*, 2021, pp. 185–190.

[15] C. Ragona, F. Granelli, C. Fiandrino, D. Kliazovich, and P. Bouvry, "Energy-Efficient Computation Offloading for Wearable Devices and Smartphones in Mobile Cloud Computing," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[16] Z. Cheng, P. Li, J. Wang, and S. Guo, "Just-in-Time Code Offloading for Wearable Computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 74–83, 2015.

[17] M. Xu, F. Qian, M. Zhu, F. Huang, S. Pushp, and X. Liu, "DeepWear: Adaptive Local Offloading for On-Wearable Deep Learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 2, pp. 314–330, 2019.

[18] M. Golkarifard, J. Yang, Z. Huang, A. Movaghar, and P. Hui, "Dandelion: A Unified Code Offloading System for Wearable Computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 546–559, 2018.

[19] Y. Yang, Y. Geng, L. Qiu, W. Hu, and G. Cao, "Context-Aware Task Offloading for Wearable Devices," in *Proc. of 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2017, pp. 1–9.

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

[21] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4132–4150, 2019.

[22] M. R. Nakhkash, T. N. Gia, I. Azimi, A. Anzanpour, A. M. Rahmani, and P. Liljeberg, "Analysis of Performance and Energy Consumption of Wearable Devices and Mobile Gateways in IoT Applications," in *Proc. of the International Conference on Omni-Layer Intelligent Systems*, 2019, pp. 68–73.

[23] "Google Glass: Tech specs," [Online] Available: https://support.google.com/glass/answer/3064128?hl=en (Accessed May 30, 2022).

[24] X. Lyu and H. Tian, "Adaptive Receding Horizon Offloading Strategy Under Dynamic Environment," *IEEE Communications Letters*, vol. 20, no. 5, pp. 878–881, 2016.

[25] M. Othman, S. A. Madani, S. U. Khan *et al.*, "A Survey of Mobile Cloud Computing Application Models," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 393–413, 2013.

[26] H. Sun, Z. Zhang, R. Q. Hu, and Y. Qian, "Wearable Communications in 5G: Challenges and Enabling Technologies," *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 100–109, 2018.

[27] S. Misra, B. E. Wolfinger, M. Achuthananda, T. Chakraborty, S. N. Das, and S. Das, "Auction-

based Optimal Task Offloading in Mobile Cloud Computing," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2978–2985, 2019.

[28] W. Zhang, Y. Wen, and D. O. Wu, "Energy-Efficient Scheduling Policy for Collaborative Execution in Mobile Cloud Computing," in *Proc. of IEEE INFOCOM*. IEEE, 2013, pp. 190–194.

[29] T. D. Burd and R. W. Brodersen, "Processor Design for Portable Systems," *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 13, no. 2, pp. 203–221, 1996.

[30] Y. Jang, J. Na, S. Jeong, and J. Kang, "Energy-Efficient Task Offloading for Vehicular Edge Computing: Joint Optimization of Offloading and Bit Allocation," in *Proc. of IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[31] "802.11 OFDM Data Rates – The Math Behind The Numbers," [Online] Available: https://dot11.exposed/2018/11/29/802-11-ofdm-data-rates-the-math-behind-the-numbers/ (Accessed May 30, 2022).

[32] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n Power Consumption," in *Proc. of International Conference on Power Aware Computing and Systems*. USENIX Association, 2010, p. 1.

[33] S. Deb and P. Monogioudis, "Learning-based Uplink Interference Management in 4G LTE Cellular Systems," *IEEE/ACM Transactions on Networking*, vol. 23, no. 2, pp. 398–411, 2014.

[34] 3GPP TR 36.814, "Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects (Release 9)," 2017.

[35] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation Rate Maximization in UAV-enabled Wireless-Powered Mobile-Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[36] K. Cheng, Y. Teng, W. Sun, A. Liu, and X. Wang, "Energy-Efficient Joint Offloading and Wireless Resource Allocation Strategy in mMulti-MEC Server Systems," in *Proc. of IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[37] "OnePlus Nord CE 5G Specifications," [Online] Available: https://www.oneplus.com/uk/nord-ce-5g/specs (Accessed May 30, 2022).

[38] R. LiKamWa, Z. Wang, A. Carroll, F. X. Lin, and L. Zhong, "Draining Our Glass: An Energy and Heat Characterization of Google Glass," in *Proc. of 5th Asia-Pacific Workshop on Systems*, 2014, pp. 1–7.

[39] 3GPP TR 36.101, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) Radio Transmission and Reception (Release 8)," 2021.

[40] N. Jain, X. Fan, W. D. Leon-Salas, and A. M. Lucietto, "Extending Battery Life of Smartphones by Overcoming Idle Power Consumption Using Ambient Light Energy Harvesting," in *Proc. of IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2018, pp. 978–983.

[41] D. Ferreira, C. Schuss, C. Luo, J. Goncalves, V. Kostakos, and T. Rahkonen, "Indoor Light Scavenging on Smartphones," in *Proc. of 15th International Conference on Mobile and Ubiquitous Multimedia*, 2016, pp. 369–371.

[42] A. Carroll, G. Heiser *et al.*, "An Analysis of Power Consumption in a Smartphone," in *Proc. of USENIX Annual Technical Conference*, vol. 14. Boston, MA, 2010, p. 21.