Analysis of Performance and Energy Consumption of Wearable Devices and Mobile Gateways in IoT Applications

Mohammad R. Nakhkash
monakh@utu.fi
University of Turku
Turku, Finland

Tuan Nguyen Gia
tuan.nguyengia@utu.fi
University of Turku
Turku, Finland

Iman Azimi
imaazi@utu.fi
University of Turku
Turku, Finland

Arman Anzanpour
armanz@utu.fi
University of Turku
Turku, Finland

Amir M. Rahmani
a.rahmani@uci.edu
TU Wien
Vienna, Austria

Pasi Liljeberg
pakrli@utu.fi
University of Turku
Turku, Finland

ABSTRACT

Smartphones and wearable devices, such as smart watches, can act as mobile gateways and sensor nodes in IoT applications, respectively. In conventional IoT systems, wearable devices gather and transmit data to mobile gateways where most of computations are performed. However, the improvement of wearable devices, in recent years, has decreased the gap in terms of computation capability with mobile gateways. For this reason, some recent works present offloading schemes to utilize wearable devices and hence reducing the burden of mobile gateways for specific applications. However, a comprehensive study of offloading methods on wearable devices has not been conducted. In this paper, nine applications from the LOCUS's benchmark have been utilized and tested on different boards having hardware specification close to wearable devices and mobile gateways. The execution time and energy consumption results of running the benchmark on the boards are measured. The results are then used for providing insights for system designers when designing and choosing a suitable computation method for IoT systems to achieve a high quality of service (QoS). The results show that depending on the application, offloading methods can be used for achieving certain improvements in energy efficiency. In addition, the paper compares energy consumption of a mobile gateway when running the applications in both serial and multi-threading fashions.

KEYWORDS

Wearables, Performance and Energy Evaluation, Mobile Gateway, Internet-of-Things

1 INTRODUCTION

Internet-of-Things (IoT) as a promising paradigm increasingly gains attention in many application areas such as healthcare, transportation, and smart spaces [1, 4, 9, 10, 15, 22]. IoT systems leverage various disciplines such as electronics, communication and data science to provide ubiquitous connectivity and shared knowledge of objects for a better service. In IoT context, the objects are equipped with sensing, communication, and computational resources, using which they can locally exchange information or communicate with remote servers.

IoT systems are traditionally partitioned into three main layers as illustrated in Figure 1 [2, 13]. First, the sensor network is responsible for continuous data collection. Second, the gateway layer also known as Fog [18] performs as a bridge between the sensor network and the cloud servers, enabling data transmission and lightweight computing tasks. Third, the cloud servers carry out data storage and data analytics using powerful computers.

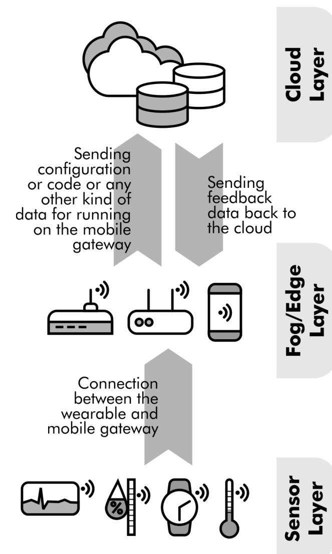

Figure 1: Fog-based IoT applications computing structure

In recent years wearable devices such as fitness trackers, smart watches, augmented reality (AR) glasses have become widely used products [3]. Consequently, these devices have been used as sensor node devices in the IoT paradigm which, as a result, have become a motive for companies to improve the processing capability [6]. Due to the resource constrained nature of wearables in terms of battery power and processing capability, these devices often rely on a mobile device (mobile gateway) such as a smartphone for performing the edge processing of the IoT application. In IoT applications hardware specifications of mobile devices and wearables such as battery lifetime and performance are considered as key QoS requirements of real time IoT applications. Recent works have presented offloading techniques providing an optimal or near optimal offloading technique to improve QoS requirements such as response

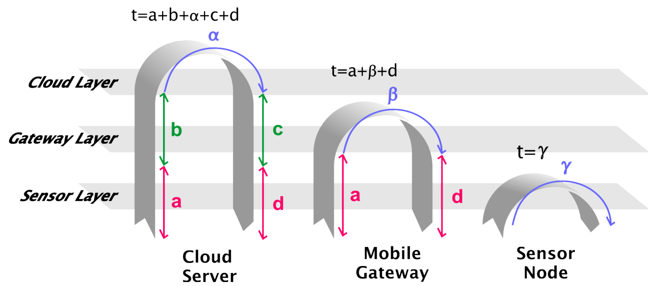

Figure 2: Response time of different IoT computation structures including data transmission latency (a, b, c, d) and execution time (α, β, γ)

time and battery life-time [17, 20, 23]. To deploy these solutions or understand how much computation can be offloaded between mobile gateways and wearable devices, an energy and performance evaluation of these devices is essential. Recent works have reported performance and energy consumption comparison between wearable devices [6] from a hardware perspective, and also compared the performance overhead of wearable operating systems (OS) [3], however they did not provide any comparison against mobile devices. A recent work has proposed an approach for partitioning of deep learning based applications between wearable and mobile devices, and also provided a performance and energy comparison, however it does not cover other types of IoT applications [23].

In this paper, we investigate and analyze the performance and energy consumption of four different wearable and gateway devices while they are running 9 diverse applications, ranging from a simple encryption or decryption algorithm to a complex machine learning program. The wearable devices and the selected mobile gateways are compared when running the LOCUS benchmark which includes a broad range of real-time IoT applications that can be executed on wearable and mobile gateway devices [6]. In addition, we compare the energy consumption of a mobile device while running 9 applications of the LOCUS benchmark in both serial and multi-threading manners. The paper presents practical insights for system designers when choosing a suitable computation method (i.e., offloading method) for IoT systems in order to achieve a high quality of service (QoS).

The structure of the paper is as follows. The background of our study and related works to this paper are presented in Section 2. Section 3 discusses the experimental setup. We evaluate and discuss the results in Section 4. Finally, Section 5 concludes the paper.

2 BACKGROUND AND RELATED WORK

In this section, we first outline energy consumption and execution time of wearable devices. Then, we describe a background on the LOCUS benchmark.

2.1 Energy Consumption

Energy consumption is one of the most important metrics evaluating the quality of IoT-based systems. wearable devices of IoT systems are often small size and lightweight [7]. Correspondingly, their battery, which is light and small, does not include a large capacity. Therefore, the energy consumption of wearable devices in the IoT systems must be carefully considered. When the energy consumption is high, it can cause serious consequences such as interruptions in services and applications, reducing their Quality of Service (QoS).

In traditional IoT-based systems, wearable devices only collect and send data to gateways which forward the data to cloud servers for further processing. The traditional IoT systems' architecture has several limitations such as the inefficiency of energy consumption and bandwidth utilization. Fortunately, these limitations can be solved or legitimated with a Mobile Edge Computing (MEC) architecture which introduces an extra edge layer in between gateways and cloud servers [5]. In the MEC architecture, data collected by sensor nodes are transmitted to smart gateways for further processing as the gateways are often equipped with high computational capability CPU, large memory and a large capacity battery.

By shifting the burden load from sensor nodes to smart gateways, the energy consumption of the sensor nodes can be dramatically saved [21]. Currently, due to the evolution of wearable devices and sensor devices, these devices are equipped with high computational capability microprocessor and large memory. It leads to a question that "Does a method of sending all data to smart gateways and processing at the gateways help to achieve the higher level of energy efficiency than a method of processing partly at wearable sensor devices and partly at smart gateway?". However, it is arduous to answer this question comprehensively as the answer depends on the complexity of the running application and wireless communication protocols between sensor nodes and gateways.

In [23], the authors showed that a method of processing partly at a wearable sensor node and partly at a smart gateway (Nexus 6 - mobile gateway) helps to achieve a high level of energy efficiency in most of the cases when running different deep learning models such as language modeling and document classification (i.e., TextCNN). However, it cannot be concluded that the method is suitable for all applications. Therefore, this paper provides a comprehensive analysis of energy consumption of widely used processors/boards when running different applications i.e., applications in LOCUS benchmark which is described in detail in the below sections.

2.2 Execution Time

The execution time becomes more prominent in determining the response time as the computation is shifted towards the sensor node. While the response time of fog based applications is dependent on the execution time of the gateway (in our case mobile gateway) and the connection delay, the response time of the application will only be dependant on the execution time (γ) if all of the execution was done on the wearable device. In order to judge on how much computation can be offloaded to the sensor node, a comparison of the execution time between the mobile gateway and the wearable device is needed.

Conventionally, the sensor devices such as wearable devices had significant constraints in terms of energy and performance. As a result, it was insufficient to perform the whole application on sensor node devices. However, with the development of wearable devices in recent years, the gap between γ and β has decreased. These changes have opened the window for offloading some computation to wearable devices which contributions like [20] or [23] have enhanced the response time by finding the optimal offloading scheme according to both wearable and mobile gateway performance and battery life time constraints. Samie *et al.* [20] compared the power consumption and bandwidth utilization for different offloading levels and showed that for an optimal configuration both maximum bandwidth utilization and increased energy saving for the gateway and sensor nodes can be achieved. In Xu *et al.* [23], the energy and performance of both a mobile gateway and a wearable device running different amount of deep neural network algorithms was compared. The work indicated that by offloading a suitable amount of the computation energy can be saved in both devices and also the overall response time will be improved.

However, these works only explored one or a specific type of applications and since the connection delay varies in different IoT applications, the comparison will not be valid. In this work an evaluation between wearable devices with different configurations and mobile gateways is conducted in order to give a view on how much computation can be offloaded to wearable devices.

2.3 LOCUS Benchmark

The LOCUS benchmark was proposed for evaluating and comparing a 16 multi-core message passing processor with other wearable processors that are used in IoT applications [6]. At the time the paper was published there was no available benchmark suite for wearable processors so the authors introduced there own benchmark. This benchmark is a set of representative kernels which are widely used applications in wearable devices as IoT applications.

The LOCUS benchmark includes a dynamic time warping (DTW) kernel which is used in several applications such as data mining and speech recognition. Furthermore, other included kernels are A Star which is used as a navigation kernel, electrocardiogram (ECG) R-peaks detection is a widely used application in most smartwatches. For secure data communication, AES encrypt and decrypt is widely used in most IoT applications are a part of the LOCUS benchmark kernels. Since image processing is increasingly applied to wearable and IoT applications, 2D Convolution and Histogram kernels which are extensively used for augmented reality in smart glasses and mobile devices are included in the benchmark. The Support Vector Machine (SVM) that is extensively used for classifying pattern based on sensor data is also included in the benchmark. The last kernel of the LOCUS benchmark used in this work is the Haar Transform variant of Discrete Wavelet Transform which is used for compressing sensor data.

Although the benchmark was meant to run on the proposed processor, in order to compare the processor with nowadays shared memory processors, the authors introduced three versions of the kernels. The first version is a serial version of the kernel, the second version is the paralleled version using POSIX thread which is adjustable according to the target processors specifications and third, there is a Message Passing Interface (MPI) version of the kernels.

In this paper since the target wearable and mobile devices have a shared memory architecture, the serial and pthread version of the benchmark kernels are executed.

Table 1: Hardware specification

Board	Processor	Core(s) & Thread(s) per core	Memory	Operating System
Intel Galileo Gen 2	Intel Quark SoC X1000 (400 Mhz)	1 & 1	256MB DDR3 RAM	Yocto Linux
Onion Omega2+	MT7688 SoC (580 MHz MIPS)	1 & 1	128MB DDR2 RAM	OpenWrt (Onion)
Raspberry Pi Zero	ARM1176JZF-S (1GHz)	1 & 1	512MB RAM	Raspbian v8 (with Gui)
Odroid-Xu4	Cortex-A15 (2GHz) and Cortex-A7 (1.4GHz)	8 & 1	2GB LPDDR3 RAM	Ubuntu Mate

3 EXPERIMENTAL EVALUATION AND COMPARISON

This section explains the specification of the boards used as a wearable and a mobile device then the experimental setup and the conditions considered while extracting the results are explained. Furthermore, the presented results which is the performance and energy consumption of each kernel code is compared and discussed.

3.1 Experimental Setup

The boards that are used in the experiments are shown in Table 1. The Intel Galileo Gen 2 board includes a low power SoC which has an Intel Pentium-class processor and has been considered as a wearable device in several works [8, 14]. The Onion Omega 2+ board is an evaluation board with few components and a small board size which is widely use for developing smart devices [16]. The Raspberry Pi Zero has a low power single core ARM processor and can be considered as a wearable device [19]. The mobile device in our experiments is the Odriod Xu4 board that has an octa-core ARM processor and has near specifications to the Samsung Galaxy S5 mobile phone [11]. The wearable boards have all single core processors because single core processors are the most common processors for a diverse range of wearable devices. Since for the diverse range of IoT applications various features of the OS is needed, the benchmark was executed on the standard OS given for each board which can be seen in Table 1. Accordingly, the Pthread version of the benchmark was compiled by the Pthread library associated with the kernel of each board.

The power consumption of each board was measured and provided by a power monitor but for the Intel Galileo Gen 2 and Odroid Xu4 boards, a step-up voltage regulator was used. Therefore, the power measurements of these boards were multiplied by the regulator's efficiency ratio and also for excluding the OS and board peripherals power consumption, the measured power while executing the process was divided by the power consumption when the target board was in idle mode. The execution time of each kernel of the benchmark was measured by modifying the benchmark kernels and using the *time.h* C library of the Linux kernel. The power and execution time were measured simultaneously and are the average of 10 times running the code. Accordingly, energy consumption is

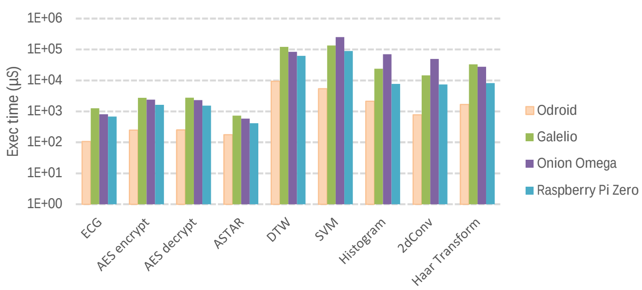

Figure 3: Execution time of the serial version of LOCUS benchmark

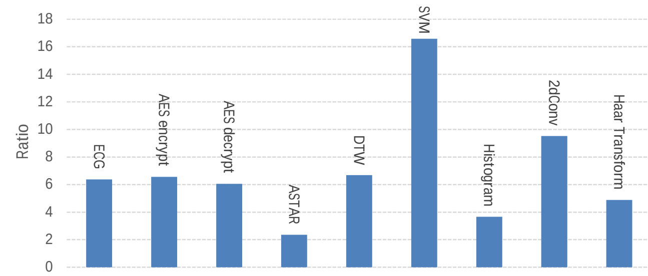

Figure 4: The performance ratio of the Odroid board to the Raspberry Pi Zero when executing the serial version of the Locus benchmark

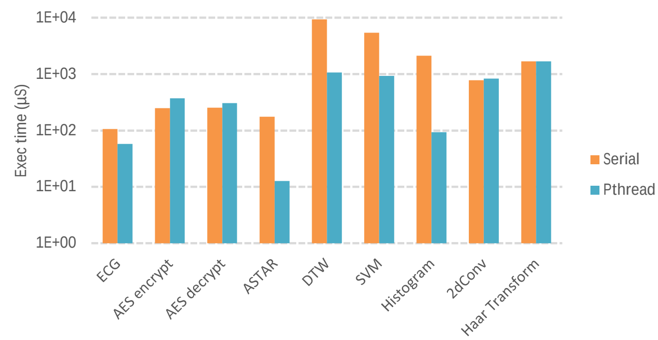

Figure 5: The execution time of the pthread and serial versions of the LOCUS executed on the Odroid board

computed by multiplying the average power consumption and the average execution time. In the following, we present the energy and delay results.

3.2 Execution Time

The execution time measured is the result of the performance of processor and the operating system of each board. In order to compare one core of the Odroid Xu4 board CPU with the single core processor of the wearable devices, we first compare the serial version of the benchmark executed on all devices which can be seen in Figure 3.

The results show that the Raspberry Pi Zero has the best results between the wearable boards but compared to the Odroid Xu4 board there is a performance gap. The performance gap between the Raspberry Pi Zero and the Odroid Xu4 board shown in Figure 4 indicates that for applications that have less computation to perform like ASTAR, Histogram and Haar Transform, there is a lower performance gap. However, for other applications that have intensive and complex computation like ECG which have fast fourier transform computation in some of its processing stages the performance gap is higher. The performance even gets more substantial in kernels like SVM and 2DConv that need to read a huge amount of data and their computation mostly consists of float data, since these kind of applications often need to utilize the floating point unit of the processor which for mobile devices like Odroid Xu4 has higher performance than the floating point unit in wearable devices like Raspberry Pi Zero. The AES encrypt and decrypt [12] and the ECG kernel have complex and also long computation routine that has resulted in a higher performance gap. Overall the result of the serial version of the benchmark shows that a single core of the Odroid Xu4 board has substantial performance gap with the wearable boards but in some applications which have low weight or complex processing, the performance gap is lower.

In order to evaluate the amount of parallelism each application can leverage and how much gain can be achieved, the execution time of the serial and Pthread version of the LOCUS benchmark is illustrated in Figure 5. Although the creators of the benchmark reported an execution speedup for all kernels [6] but in our experiments, the AES encrypt, decrypt and the 2DConv kernels had a decrease in performance. The reason behind the speed down is multifold but one fold is probably because of the I/O read and write operations to the files (2DConv) which is related to the OS performance. Other kernels had considerable speedup in execution time like ASTAR, SVM or Histogram. This speedup is due to intrinsic parallelism these applications have which the same operation is performed on multiple data. For example, the ASTAR kernel computes multiple routes and at the end selects the route with the lowest cost. A similar concept is present in applications like SVM, for classifying multiple nodes compute the input data and the results are combined at the end or in Histogram a comparison is performed on every pixel to extract the tonal distribution of the pixels. The Pthread version of the benchmark shows considerable speedup can be achieved in applications that perform simple operations on multiple data but for applications with complex operations, the performance gain is limited.

3.3 Energy Consumption

Energy consumption of the embedded boards listed in Table 1 when running different applications of the LOCUS benchmark in a sequential manner is shown in Figure 7. The results show that energy consumption of all the boards when running applications such as ECG, AES encrypt, AES decrypt and ASTAR is much lower than the energy consumption when running DTW, SVM, Histogram, 2DConv, and Haar Transform. In addition, the results show that Intel Galileo Gen 2 and Odroid Xu4 consume the most and the least energy consumption among all boards when running all applications of the LOCUS benchmark. In most of the cases, the Raspberry

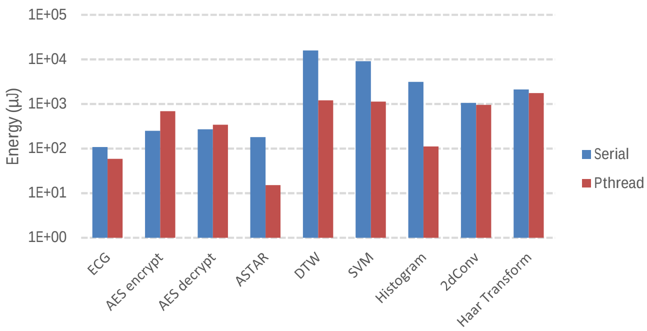

Figure 6: Comparison of the energy consumption of the Pthread and version of the Locus benchmark executed on the Odroid Xu4 board

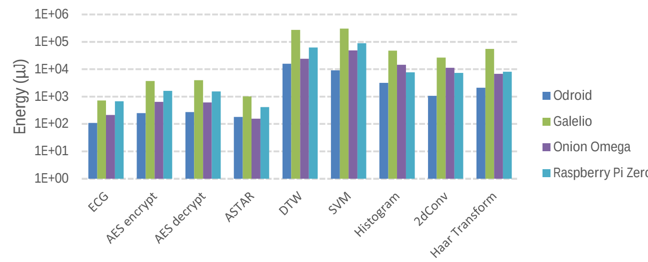

Figure 7: Energy Consumption of the serial version of the benchmark codes

Pi Zero consumes the second most energy consumption except for the cases of Histogram and 2DConv applications.

One of the reasons causing the large difference in energy consumption of these two boards can be microprocessors and their frequency. As can be seen that Odroid Xu4 is equipped with 2 GHz microprocessor whilst Raspberry Pi Zero and Intel Galileo are equipped with 1 GHz and 400 MHz microprocessors, respectively.

In the experiment cases, applications of the benchmark require many computations; therefore, applying microprocessors with a higher frequency can help to achieve some levels of energy efficiency. However, energy consumption of Raspberry Pi Zero is higher than energy consumption of Onion Omega 2+ in most of the cases when running the LOCUS benchmark's applications although Raspberry Pi Zero and Onion Omega 2+ are equipped with 1 GHz and 580 MHz, respectively. One of the reasons causing the low energy consumption of the Onion Omega 2+ is that the board is supplied with 3 V whilst other boards (i.e., Raspberry Pi Zero) require 5 V power supply. In cases of Histogram and 2DConv applications, energy consumption of Onion Omega 2+ is larger than Raspberry Pi Zero because of many "read-from-files" and "write-to-files" instructions in these applications, that cause the overhead of the Onion Omega 2+ operating system performance. Therefore, it can be concluded that depending on the applications (e.g., complexity, computation requirements, and Input/Output access frequency), different microprocessor's frequencies should be applied.

As mentioned above, all applications of the LOCUS benchmark are run in the serial and multi-threading manners on the Odroid Xu4 board. The results show that applying a multi-threading method for running applications of the LOCUS benchmark helps to reduce the energy consumption of the board except for the case of AES encrypt and AES decrypt. One of the reasons causing the high energy consumption of Odroid Xu4 in these applications is that in AES encrypt and decrypt applications, many computation functions require and wait for the results from the other or previous functions run in the past. It can be concluded that depending on the applications, microprocessors with multi-core and multi-threading can help to achieve some levels of energy efficiency.

4 DISCUSSION

The energy consumption and execution time results show that board components like input/output (I/O) ports can increase the energy consumption of wearable devices despite the performance they have. The energy consumption of a board is affected by the operating system and the software running on the board. When an applications uses system calls too much, the operating system becomes a major overhead. Therefore, it is essential to suppose that the relation of energy consumption and execution time may not be consistent when applying offloading techniques.

The execution time results of the Pthread and serial version of the benchmark indicate that for applications like ECG that have multiple stages for processing and have complex computation, it is better to use an offloading scheme which offloads some of the stages onto the wearable device and leave the higher complex computation for the mobile device. Applications that process images like Histogram, 2DConv which perform the same operation on multiple data, is better to use an offloading scheme that partitions the data to process some of the data on the wearable device and the rest of the data on the mobile gateway. Since there is a huge performance gap (compared to the performance gap of other kernels) between the Odroid Xu4 and wearable boards for the SVM which is a classification kernel or other kernels like ECG that are more compute intensive compared to the data, the offloading technique as presented in previous works [23], should significantly be conducted towards breaking the process into multiple stages and offload the stages that have less complex computations onto the wearable device. Although the Pthread version of the benchmark had performance speedup for most applications, in real-world scenarios, the mobile device has multiple applications to process which as a result it cannot use all of its resources for one process which is an important factor to be considered in choosing an offloading technique and conducting it.

According to the execution time results of the serial version of the LOCUS benchmark, there is a considerable gap in terms of performance between the CPU core of a mobile gateway and wearable devices. Due to the limited scope of the paper, the paper does not consider other aspects which also affects the energy consumption of wearable devices such as wireless connection protocol, data rate, and connection delay. It is recommended that a system administrator needs to consider these aspects together with other aspects such as microprocessor's frequency, multi-threading in order to achieve a high level of energy efficiency. Depending on the application and wearable sensor devices, one of the methods such as offloading all computation on wearable devices, a combination of partly offloading on wearable devices and partly processing on mobile gateways, or running completely algorithms on mobile gateways.

5 CONCLUSION

The paper has compared the energy and execution time of various platform boards considered as wearable device and mobile device via the LOCUS benchmark to provide a guide for how and what offloading techniques to use for increasing QoS in IoT applications. The results show that the chosen mobile gateway device has higher performance and is more energy efficient compared to the chosen wearable devices but according to the type of the application, the performance and energy consumption gap differ. The Pthread and serial version of the benchmark indicates that for applications with complex computation, computation offloading can be more effective whereas for applications that the data intensity is more than the computation complexity, offloading techniques that offload the processing of some of the data will be more suitable. The wearable energy consumption results indicate that not always better performance brings better energy efficiency, and factors such as board components that consume power or processor frequency play a critical role in determining wearable energy consumption. We encourage the readers to see the results as a guide to consider the execution time and energy consumption of each application compared to each other to conduct a suitable offloading technique which will increase the QoS in IoT applications such as response time and/or battery lifetime.

ACKNOWLEDGEMENT

This material is based upon work supported partially by the US National Science Foundation (NSF) WiFiUS grant CNS-1702950 and Academy of Finland grants 311764 and 311765.

REFERENCES

[1] A. Anzanpour *et al.* 2015. Internet of things enabled in-home health monitoring system using early warning score. In *Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare*. 174–177.

[2] A. M. Rahmani *et al.* 2018. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. *Future Generation Computer Systems* 78 (2018), 641–658.

[3] Vicente JP Amorim, Saul E Delabrida, and Ricardo AO Oliveira. 2018. A comparative analysis of wearables operating systems based on functional constraints. In *15th Annual Consumer Communications & Networking Conference*. IEEE, 1–6.

[4] L. Atzori et al. 2010. The Internet of Things: A survey. *Computer Networks* 54, 15 (2010), 2787–805.

[5] B.Negash *et al.* . 2018. Leveraging fog computing for healthcare iot. In *Fog Computing in the Internet of Things*. Springer, 145–169.

[6] C. Tan *et al.* 2017. LOCUS: Low-power customizable many-core architecture for wearables. *ACM Transactions on Embedded Computing Systems (TECS)* 17, 1 (2017), 16.

[7] D. Amiri *et al.* 2018. Edge-Assisted Sensor Control in Healthcare IoT. In *Proc. IEEE Globecom*.

[8] D. Azariadi *et al.* 2016. ECG signal analysis and arrhythmia detection on IoT wearable medical devices. In *International Conference on Modern Circuits and Systems Technologies, 2016 5th*. IEEE, 1–4.

[9] F. Firouzi *et al.* 2018. Internet-of-Things and big data for smarter healthcare: from device to architecture, applications and analytics.

[10] J. Gubbi et al. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29, 7 (2013), 1645–60.

[11] Hardkernel co. (accessed 1-1-2019). https://www.hardkernel.com/shop/odroid-xu4/.

[12] I. B. Dhaou *et al.* 2017. Low-latency hardware architecture for cipher-based message authentication code. In *International Symposium on Circuits and Systems*. IEEE, 1–4.

[13] J. Lin et al. 2017. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal* 4, 5 (2017), 1125–42.

[14] M. Ambrosin *et al.* 2016. On the feasibility of attribute-based encryption on internet of things devices. *IEEE Micro* 36, 6 (2016), 25–35.

[15] Riitta Mieronkoski, Iman Azimi, Amir M Rahmani, Riku Aantaa, Virpi Terävä, Pasi Liljeberg, and Sanna Salanterä. 2017. The Internet of Things for basic nursing care - A scoping review. *International journal of nursing studies* 69 (2017), 78–90.

[16] Onion Corporation. (accessed 1-1-2019). https://docs.onion.io/omega2-docs/.

[17] P. Sanabria *et al.* 2018. Code offloading solutions for audio processing in mobile healthcare applications: a case study. In *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*. ACM, 117–121.

[18] A. M. Rahmani et al. 2017. *Fog Computing in the Internet of Things - Intelligence at the Edge*. Springer.

[19] Raspberry Pi Foundation. (accessed 1-1-2019). https://www.raspberrypi.org/.

[20] Farzad Samie, Vasileios Tsoutsouras, Lars Bauer, Sotirios Xydis, Dimitrios Soudris, and Jörg Henkel. 2016. Computation offloading and resource allocation for low-power IoT edge devices. In *3rd World Forum on Internet of Things (WF-IoT)*. IEEE, 7–12.

[21] T. N. Gia *et al.* . 2015. Fog computing in body sensor networks: An energy efficient approach. In *Proc. IEEE Int. Body Sensor Netw. Conf.(BSN)*. 1–7.

[22] T. N. Gia *et al.* . 2017. IoT-based continuous glucose monitoring system: A feasibility study. *Procedia Computer Science* 109 (2017), 327–334.

[23] Mengwei Xu, Feng Qian, and Saumay Pushp. 2017. Enabling Cooperative Inference of Deep Learning on Wearables and Smartphones. *arXiv preprint arXiv:1712.03073* (2017).