

Energy Efficient Healthcare Monitoring System using 5G Task Offloading

T Sigwele^{1*}, A Naveed², M Susanto³, M Ali² and Y F Hu²

¹Department of Computing and Informatics, Faculty of Science, BIUST University, Private Bag 16, Palapye, Botswana

²Faculty of Engineering and Informatics, University of Bradford, BD7 1DP, United Kingdom,

³Department of Electrical Engineering, Faculty of Engineering, University of Lampung, Jl. Prof. Sumantri Brojonegoro No. 1, Bandar Lampung 35145, Indonesia

*Email: sigwelet@biust.ac.bw

Article Information

Received:
30 September 2019

Received in revised form:
16 October 2019

Accepted:
02 November 2019

Volume 1, Issue 2, December 2019
pp. 46 – 52

<http://dx.doi.org/10.23960/jesr.v1i2.12>

Abstract

Healthcare expenses are significantly reduced, and lives are saved by enabling the continuous monitoring of patient health remotely using Wireless Body Sensor Network (WBSN). An energy-efficient mobile gateway (e.g., 5G smartphone) is required, which moves with the patient in real time to process the data from the biosensors without depleting the battery. Therefore, this research aimed to propose a 5G-based healthcare cardiovascular disease Remote Monitoring system called 5GREM using an Electrocardiogram (ECG) biosensor as a BSN device. The objective was to monitor and analyze the heart rhythms of patients and send emergency alerts during irregularities to the nearest caregivers, ambulance, or physician in minimizing heart attacks and heart failures while saving energy. Since ECG signal processing was computationally intensive, the sensor requests were executed locally on the gateway, offloaded to nearby mobile devices, or transmitted to the 5G edge for execution. This was done by considering the battery level, CPU level, transmission power, delays, and task failure rate.

Keywords: 5G, Remote Healthcare Monitoring, Wireless Body Sensor Network, Energy Efficiency, Mobile Edge Computing.

I. INTRODUCTION

Experts predict that in the next decade, healthcare delivery will experience a major shift. Rather than being focused mainly in hospitals, care will increasingly move toward a balance between hospital and home, a trend that started around 2020. By 2030, this transformation is expected to move even further, with most healthcare becoming remote and centered in the homes of people [1], which is made possible through Wireless Body Sensor Network (WBSN). The model continuously and remotely monitors vital parameters, consisting of body temperature, blood pressure, heart rate, and breathing rate of patients suffering from chronic diseases, including diabetes, asthma, and heart attacks. As the system monitors the patient, it sends alerts to a caregiver, physician, or emergency department when abnormalities are detected, potentially saving lives and reducing medical costs. However, when a patient moves beyond the range of a fixed healthcare gateway, such as leaving a smart

home for a stadium or city center, the smart devices require a mobile healthcare gateway that travels with the patient, an area where research remains limited.

Through the introduction of 5G [2], smartphones are becoming energy efficient, with low transmission delays (1ms), high data-rates up to 20Gbps, and improved storage. For example, Apple Inc. has recently introduced iPhone XS Max [3] with up to 512GB of storage and an A12 Bionic chip with improved processing capability. These smartphones are the perfect choice for the healthcare mobile gateway for relaying WBSN data to the remote medical personnel or to a remote cloud for storage and analysis. However, using a smartphone as a healthcare gateway is computer-intensive, draining energy in excess due to computation and transmission power consumption.

This research is extended from previous investigations in [4] and [5] called IEE5GG. During the research, a 5G and Mobile Edge Computing (MEC) based mobile gateway connected to body sensors was proposed for minimizing energy consumption, task

failure rate, as well as health service time. The extension in this research is the introduction of mobile-to-mobile task offloading using device-to-device (D2D) communication to argument the gateway processing power and reduce the energy consumption. In addition, the introduction of emergency alerts to ambulances and physicians during irregularities detected by the biosensors is another extension. The contributions of this research are extended in the following paragraphs.

i) A 5G Cloud Radio Access Network (C-RAN) based energy-efficient cardiovascular remote and mobile Monitoring system called 5GREM is proposed for sending alerts to caregivers, physicians, and emergency personnel. This is performed during unusual changes in the Electrocardiogram (ECG) signals connected to the patient suffering from chronic heart attacks.

ii) Improving heart attack alert response time and energy efficiency using the MEC offloading scheme in 5GREM. This method helps save energy in 5G smartphone healthcare gateways, where ECG signal execution is divided into modules/task. Decisions are then made to execute certain task locally, while others are offloaded for execution on nearby mobile devices via D2D communication or on Virtual Machines (VMs) in the MEC cloud, considering transmission cost and delay.

iii) A 5G-based architecture for 5GREM is proposed, leveraging cloud computing technology for resource pooling. The 5G gNB-DU small cells are used to improve capacity, while the macro gNB-DU cells are applied for maintaining coverage.

II. MATERIALS AND METHODS

A. Related Work

Extensive research had been conducted on healthcare monitoring system using WBSN, and the reader was directed to [6]. The main focus of this research was on saving energy in the mobile gateways to avoid interruption during monitoring. The MEC paradigm was a new area and an attractive option for augmenting the processing of computer-intensive healthcare applications, such as augmented reality or ECG [7], by offloading application task to the MEC [8]. During this process, only energy-efficient smartphone gateways using MEC and D2D offloading would be addressed. [9] proposed an energy-efficient system that offloads mobile task from a mobile gateway device to another mobile device called a cloudlet to reduce energy consumption and improve performance. However, the system was based on an outdated radio access network architecture with delays and high bandwidth penalties. [10] recommended a dynamic mobile cloudlet cluster policy to use a group of mobile devices as a supplement for the fog server for offloading task from a mobile

gateway device. Zhang K. et al. in [11] proposed an Energy Efficient Computation Offloading (EECO) mechanism for MEC in 5G heterogeneous network suitable for mobile gateways. Consequently, the author considered MEC servers with limited capacity that were not consolidated in a centralized fashion as in 5G C-RAN. [12] investigated a green smart gateway MEC system and developed an effective computation offloading strategy. The execution cost, comprising both execution latency and task failure, was used as the performance metric. On the other hand, the author ignored the execution delay introduced by the MEC server and assumed an unrealistically battery capacity, making the method impractical.

All the schemes mentioned were based on MEC offloading or D2D offloading, but not combining both. The schemes were also according to random architectures, of which some were labeled as 5G. This research addressed the gateway framework from a standardized 5G architecture [2] with accurate 5G requirements considered.

B. Proposed 5GREM Framework System Model

The proposed 5GREM framework in Fig. 1 considered a 5G system with modification where the MEC servers were centralized to form a MEC cloud located in the gNB-CU, as the acronyms were shown in Fig. 1 in the Next Generation RAN (NG-RAN). 5G base-stations (BS) were called 5G Base-station Distributed Unit (gNB-DU). In 5GREM, macro 5G BS (gNB-DUm) were overlaid by small cell (gNB-DUs). The gNB-DUm provided coverage while the gNB-DUs were to increase capacity in hotspot areas. The gNB-DU was connected to the gNB-CU by high-bandwidth low low-latency fiber using the F1 interface [2].

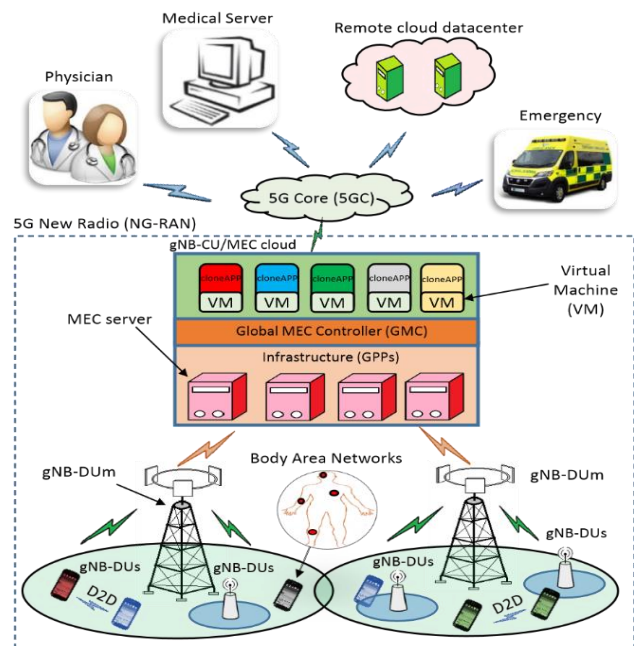


Fig. 1. Proposed 5G REM architecture. gNB-DUM = 5G Macro Distributed Unit BS, gNB-DUs = 5G Small Cell Distributed Unit BS, gNB-CU = 5G Centralised Unit Base Station.

For simplicity, the gNB-DUM was termed as Macro Remote Radio Head (MRRH) and overlaid by small cell gNB-DUs, representing small cell/pico RRH (PRRH). The set of RRH was defined as $R = \{n : n = 1, 2, \dots, N_{rrh}\}$, where N_{rrh} signified the total number of PRRHs. The term smartphone was used interchangeably with user equipment (UE) and mobile device. A set of UEs was signified as $K = \{k : k = 1, 2, \dots, N_{UE}\}$. Each UE k was associated with a serving RRH by the criterion of Cell Range Expansion (CRE) [13] to maximize transmission rate. In the MEC cloud/gNB-DU, there were multi-core high-processing MEC General Purpose Processors (GPPs) with VMs instantiated according to the traffic demand of offloaded task requests from smartphone gateways. Following the discussion, the Global MEC Controller (GMC) in the gNB-CU was responsible for receiving task requests from the smartphone gateway and scheduling the information to VMs. The machines for processing application task were abstracted from the GPP using the GMC. The gNB-CU was originally for baseband processing, which was the same as VMs. This research solely focused on VMs for processing task for healthcare applications offloading.

C. Communication Model

The 5G system bandwidth, B , was divided into N_{ch} channels, each of bandwidth w . A set of channels in the system was represented as $C = \{c : c = 1, 2, \dots, N_{ch}\}$. The analysis assumed that smartphones in the same cell transmitted over orthogonal channels, while UEs of different cells might interfere with each other. The process ensured each smartphone used a 5G healthcare application, which was split into several task. Each task T_k of UE k could be executed locally on the smartphone and remotely on the MEC cloud or nearby devices by computation offloading. The analysis considered that k could offload T_k via the MRRH or PRRH. $a_{k,m,c} = \{0, 1\}$ was the offloading decision profile of k when the user device selected modes, which were computing locally, transmitting via the MRRH or PRRH, and D2D offloading. The variable $a_{k,m,c} = 1$ represented device k used mode m to offload task T_k through channel c or when $a_{k,m,c} = 0$. The item c was meaningless when $m = 1$ as there were no channels in local computing. Therefore, $a_{k,1,1} = 1$ was taken as the indicator that device k selected local computation. When the smartphone k offloaded the task T_k via the MRRH on channel c , the accurate uplink data rate of the UE was computed as follows.

$$r_{k,c}^M = w \log_2 \left(1 + \frac{P_k^M H_k^M}{\sum_{j=1, j \neq k}^{N_{UE}} a_{k,3,c} P_j^S H_j^M + \sigma^2} \right) \tag{1}$$

Where w was the channel bandwidth, P_k^M represented the transmission power from UE k to the MRRH, H_k^M signified the channel gain between UE k and the MRRH. The denominator was the interference caused by other UEs using the same channel for transmission. Following the process of the computation, the variable σ^2 signified the background noise power. The total uplink data rate of UE k to the MRRH was calculated as follows.

$$r_k^M = \sum_{c=1}^{N_{ch}} a_{k,2,c} r_{k,c}^M \tag{2}$$

When the UE offloaded a task via the PRRH through channel c , replacing M for macro to S for small cell in equation (1) gave the uplink data rate as $r_{k,c}^S$ and the total uplink data rate became r_k^S from (2). When the UE offloaded a task to a nearby UE, the data rate was assumed to be constant 13.5Mbps as adapted in [14] for D2D communication.

D. Computation Model

Each task of UE k was represented as $T_k = (B_k, D_k, t_k^{max})$. Where B_k was the size of computation input data in bytes (e.g., the program codes and input parameters) included in the computation task T_k and D_k signified the processing prerequisite in million instructions per second (MIPS) required to accomplish the computation task T_k . The variable t_k^{max} represented the maximum latency required by the computation task T_k or the execution deadline in milliseconds (ms).

1. Local Computation: Local computation was when the smartphone k executed its computation task T_k locally. During the process, F_k^l was represented as the computation capability of the smartphone in MIPS. The analysis was essential to assume that k could have various computation capabilities.

The execution time (t_k^l) and energy consumed (e_k^l) for executing task T_k for k were expressed as follows.

$$t_k^l = \frac{D_k}{F_k^l}, \text{ and } e_k^l = t_k^l P_a \tag{3}$$

where P_a is the power consumed by k when active.

2. MEC Computation: When k selected to compute its task by the MEC server, the input data was transmitted to the VM through the MRRH or the PRRH. This signified that k would incur the extra overhead in terms of time and energy for transmitting the computation

input data via 5G wireless access. As k offloaded T_k via MRRH, the total time duration (t_k^M) was calculated as transmission time plus time during MEC cloud execution of task T_k

$$t_k^l = \frac{D_k}{F_k^l}, \text{ and } e_k^l = t_k^l P_a \quad (4)$$

Where F_k^{mec} was the computation ability of the MEC server VM. Several VMs in the MEC cloud could process task of an application in parallel. The total energy consumed by k via offloading through the MRRH was calculated as follows.

$$e_k^M = \frac{B_k}{r_k^M} P_k^M + \frac{D_k}{F_k^{mec}} P_{idle} \quad (5)$$

For offloading via the PRRH,

$$t_k^S = \frac{B_k}{r_k^S} + \frac{D_k}{F_k^{mec}} \quad (6)$$

$$e_k^S = \frac{B_k}{r_k^S} P_k^S + \frac{D_k}{F_k^{mec}} P_{idle} \quad (7)$$

3. *Nearby D2D offloading:* In case of UE offloading a task to a nearby device, the time of execution which signified t_k^{nd} was equivalent to the local computation execution time in (5) plus transmission time ($t_k^{nd} = t_k^l + B_k/13.5Mbps$) where 13.5Mbps was D2D uplink data rate. Similarly, the energy consumption during the process was denoted as $e_k^{nd} = t_k^{nd} * P_a$.

E. Problem Formulation

The aim was to minimize energy consumption in the smartphone by offloading some application task to the 5G MEC cloud, where transmission delay, propagation delay, task processing time, as well as energy consumption in both MEC server and smartphone are minimized while transmission data rate is maximized. The optimization problem during the analysis was formulated as follows in (11).

$$\min_{\{a_{k,m,c}\}} \sum_{k=1}^{N_{UE}} (a_{k,1,1} e_k^l + \alpha_{k,2} (P_k^M \frac{B_k}{r_k^M} \sum_{c=1}^{N_{ch}} a_{k,2,c} + P_{idle} \frac{D_k}{F_k^{mec}}) + \alpha_{k,3} (P_k^S \frac{B_k}{r_k^S} \sum_{c=1}^{N_{ch}} a_{k,3,c} + P_{idle} \frac{D_k}{F_k^{mec}}) + e_k^{nd}) \quad (8)$$

such that, C1: $a_{k,1,1} \cdot t_k^l \leq t_k^{\max}, k \in K$

$$C2: \sum_{k=1}^{N_{UE}} \sum_{c=1}^{N_{ch}} a_{k,m,c} \leq N_{ch}, m = \{2, 3\} \quad (9)$$

Where, $s_{k,m} = 1(\sum_c^{N_{UE}} a_{k,m,c} > 0, j = \{2, 3\})$

The function $1(x)$ was an indicator function that was equal to 1 when x was true and zero otherwise. The first constraint $C1$ ensured that the delay constraints were met. The second constraint ensured that the total number of channels allocated to smartphones did not exceed the entire sum of channels in the system. In addition, constraint $C3$ explained that only one channel could be allocated to one smartphone.

F. 5GREM Offloading Framework

The section described the 5GREM offloading framework in detail, as shown in Fig. 3. The smartphone comprised an elastic application and other components that enabled partitioning as well as offloading application task. The system components used during this research included the following. i) Device profiler: Collected smartphone hardware context at runtime and passed the information to the offloading agent. The hardware context included the battery State of Charge (SoC), average CPU usage, and memory usage. ii) Resource monitor: Resided in both the smartphone gateway and the allocated VM. It collected network-related context at runtime and passed the information to relevant modules such as the offloading agent. Additionally, the network context included network connection state, bandwidth, and signal strength.

Program profiler: The program profiler tracked the execution of the program and collected program context information such as total instructions executed, execution time, and memory allocated. The profile was updated at every invocation and stored in the smartphone database.

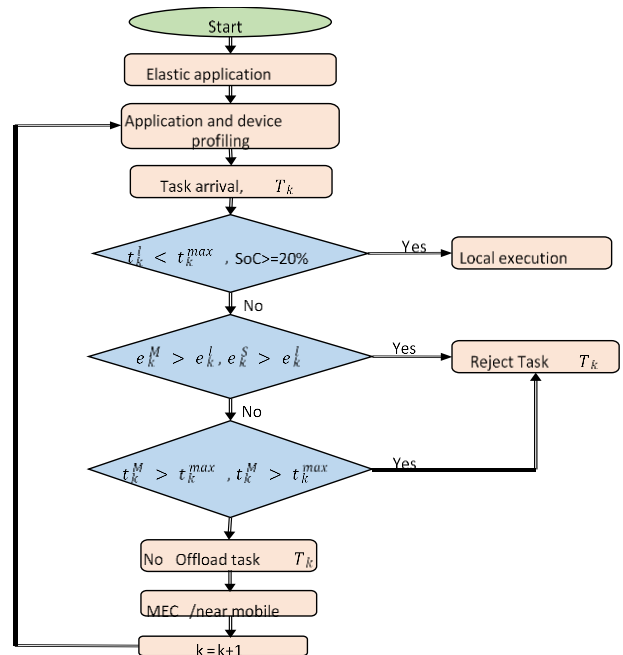


Fig. 2. The 5GREM offloading agent flow chart

Offloading Agent: The process consisted of a set of cost estimation models, such as the delay or execution

time, and the energy models. Based on the received context earlier, the offloading agent decided on when, where, and how to offload the task. When MEC VMs with adequate processing power were available, the data would be registered in the directory services to execute the task. The offloadable class codes task was then uploaded to the class loader of the MEC server VM, which executed the class code in the recipient OS. After execution, the results were loaded back to the offloading agent in the client device. On the other hand, when a nearby smartphone device with enough battery power and computing requirements was available, the task class codes were loaded to the client device and

executed in the class loader. The flow chart in Fig. 2 showed the operation of the offloading agent.

The offloading agent in this research started with the arrival of a task, T_k . When local execution time was less than the maximum delay tolerable and the battery $SoC \geq 20\%$, the task was executed locally. Consequently, when the offloading delay deadlines were met and energy was saved using offloading, the task was offloaded to the VM in the MEC server or in the nearby mobile device. The next task during the process followed the same order in the flow chart.

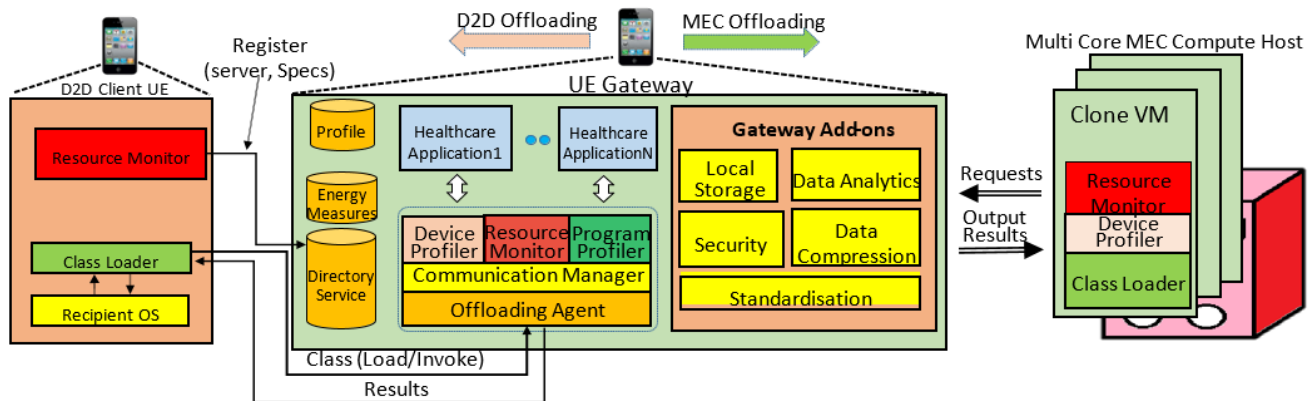


Fig. 3. Proposed 5GREM offloading framework.

Communication manager: The process created as well as maintained connection between the client and the server side. It serialized the code on the client side and the request from the client on the cloud side. The process also kept the client and the server VM in synchronization. The communication manager checks when the required files and programs existed on the server side, it contacted the client device to obtain the files and related libraries for remote execution.

Healthcare gateway add-ons: The healthcare gateway could be incorporated with some add-ons,

namely local temporary storage in case the network is down, data analytics, security, standardization of the smart device data using healthcare standards, including HL7, HL7 2, or HL7 3, and data compression before uploading to a remote cloud storage. This could be at the hospital to reduce data size and maximize 5G cellular bandwidth usage. All these add-ons could have personal task processed in the MEC server to save energy in the smartphone.

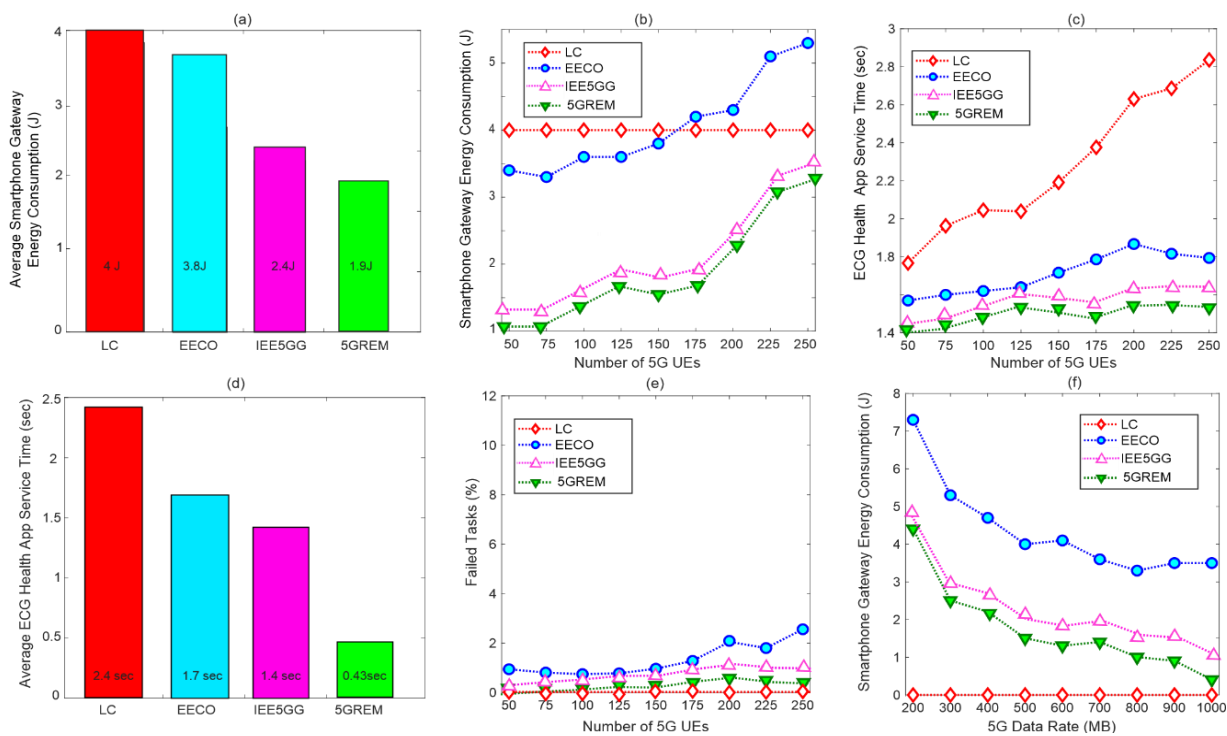


Fig. 4. Simulation results of (a) Average smartphone gateway energy consumption, (b) Effects of increased network UEs on smartphone gateway energy consumption, (c) Health app task service time, (d) Average service time of task, (e) Failed task due to mobility or VM overload, (f) Effects of 5G data rate on smartphone gateway energy consumption.

III. RESULTS AND DISCUSSIONS

A. Results Simulation Tools

MRRHs overlaid by 12 PRRHs per MRRH were considered to analyze the performance of the proposed 5GREM framework. The proposed 5GREM framework was compared with the following during the process. 1) Local Computation scheme signified LC in the results, where all health task were processed locally in the gateway. 2) Previous research in [5] introduced in Section I) the EECO scheme, which was also presented in Section II in [11]. In EECO, the MEC servers were of limited processing capacity and were alone. Table 1 showed the simulation parameters used during the process of the analysis. The considered system was a heavy computation health application that used an ECG device connected to the smartphone of the patient to monitor heartbeat rhythms. The ECG application was divided into two task (Task1 and 2) with the parameters shown in Table 1. During the analysis, the simulation was performed using an open-source Edgecloudsim simulation tool [15] for modeling as well as simulation of resource management methods in IoT and MEC by considering the communication aspects.

Table 1. Simulation Parameter Settings

Parameters	MRRH	PRRH
Cellular Layout	1 macro	12 per MRRH
BS transmission power	46 dBm	30 dBm
Number of smartphones	50 to 250	

MEC Server processing	20000 MIPS
Smartphone processing	1000MIPS
Smartphone active power	0.9 Watts
Smartphone idle power	0.3 Watts
Mobile transmission power	0.3 Watts
Health App Task1	3500MIPS, 500bytes
Health App Task2	1000MIPS, 1000bytes
Probability of offloading	0.1
Poisson interarrival time	3 sec

B. Discussion

Fig. 4 showed the average energy consumed in the smartphone for all the schemes. The IEE5GG and the proposed 5GREM scheme saved 40% and 52.5% on average, compared to the local computation. This was because both schemes included computation offloading with 5GREM extending the offloading to nearby mobile devices. As the number of network UEs increased, energy consumption in the smartphone gateway improved because more UEs shared the available bandwidth, causing a lower uplink data rate for the EECO, IEE5GG and 5GREM schemes, as shown in Fig. 4(b).

Fig. 4(c) showed that as the number of smartphones increases, the service time improved compared to the local computation scheme due to the sharing of bandwidth. Fig. 4(d) showed that the EECO, IEE5GG, as well as the proposed 5GREM schemes saved 33%, 41% and 82% on average, with 5GREM 49% better due

to high processing speed in the centralized MEC servers augmented by D2D offloading. As more UEs occupied the network, the failure dropping probability of task increased, but for local computation, the failure rate was not affected by the increase in network UEs, as shown in Fig. 4(e). The user mobility in D2D had not been included as devices were assumed to be static, where the request drop in D2D was omitted, and consideration of user mobility in the future. Fig. 4(f) showed that as the gateway transmission rate increased, the energy consumed slightly decreased since the transmission time was reduced. As the processing power of the device became higher, the energy consumed reduced since less time was used when executing some task locally.

IV. CONCLUSIONS

In conclusion, this research proposed an energy-efficient cardiovascular remote monitoring system called 5GREM for sending alerts to caregivers, physicians, and emergency personnel during unusual changes in ECG signals connected to the patient suffering from chronic heart attacks. The 5GREM framework adopted the 5G technology, MEC, and D2D task offloading to reduce energy consumption in the mobile gateway. Additionally, the model showed that deaths from heart attacks could be significantly reduced according to the results of the analysis.

REFERENCES

- [1] C. E. Koop, R. Mosher, L. Kun, *et al.* (2008). Future delivery of health care: Cybercare, *IEEE Engineering Medicine and Biology Magazine*. Vol. 27, no. 6, pp. 29-38.
- [2] NEC. (2018, August). Making 5g a reality. [online]. Available: https://www.nec.com/en/global/solutions/ns_p/5g_vision/doc/w_p2018ar.pdf
- [3] Apple-Inc. (2018, August). Compare iphone models. [Online]. Available: <https://www.apple.com>
- [4] T. Sigwele, Y. F. Hu, M. Ali, J. Hou, M. Susanto, and H. Fitriawan. (2018). An intelligent edge computing based semantic gateway for healthcare system interoperability and collaboration. 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 370-376.
- [5] T. Sigwele, Y. Hu, M. Ali, J. Hou, M. Susanto, and H. Fitriawan. (2018). Intelligent and energy efficient mobile smartphone gateway for healthcare smart devices based on 5g. 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1-7.
- [6] S. Pirbhulal, H. Zhang, W. Wu, S. C. Mukhopadhyay, and Y.-T. Zhang. (2018). Heart-beats based biometric random binary sequences generation to secure wireless body sensor network. *IEEE Transactions on Biomedical Engineering*.
- [7] H. T. Dinh, C. Lee, D. Niyato, and P. Wang. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611.
- [8] A. Ahmed and E. Ahmed. (2016). A survey on mobile edge computing. *Intelligent System and Control (ISCO)*, 2016 10th International Conference. pp. 1–8.
- [9] Y. Zhang, D. Niyato, and P. Wang. (2015). Offloading in mobile cloudlet system with intermittent connectivity,” *IEEE Transactions on Mobile Computing*. Vol. 14, no. 12, pp. 2516–2529.
- [10] Y. Li, N. T. Anh, A. S. Nooh, K. Ra, and M. Jo. (2018). Dynamic mobile cloudlet clustering for fog computing. 2018 International Conference on Electronics, Information, and Communication (ICEIC). pp. 1-4.
- [11] K. Zhang, Y. Mao, S. Leng, Q. Zhao *et al.* (2016). Energy-efficient offloading for mobile edge computing in 5g heterogeneous network. *IEEE Access*. Vol. 4, pp. 5896–5907.
- [12] Y. Mao, J. Zhang, and K. B. Letaief. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*. Vol. 34, no. 12, pp. 3590–3605.
- [13] H. Jiang. (2016). System utility optimization of cell range expansion in heterogeneous cellular network. In *Communication Software and Network (ICCSN)*, 2016 8th IEEE International Conference on. IEEE, pp. 412–417.
- [14] U. N. Kar and D. K. Sanyal. (2017). An overview of device-to-device communication in cellular network. *ICT Express*. Vol. 4 (4), pp. 203-208.
- [15] C. Sonmez, A. Ozgovde, and C. Ersoy. (2017). Edgecloudsim: An environment for performance evaluation of edge computing system. In *Fog and Mobile Edge Computing (FMEC)*, 2017 Second International Conference on. IEEE, pp. 39–44.