Energy Efficient Healthcare Monitoring System using 5G Task Offloading

T Sigwele[1*], A Naveed[2], M Susanto[3], M Ali[2] and Y F Hu[2]

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

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

[3]*Department 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

©Universitas Lampung

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

Abstract

Healthcare expenses can be significantly reduced, and lives saved by enabling the continuous monitoring of patient health remotely using Wireless Body Sensor Networks (WBSN). However, 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 bio sensors without depleting the battery. This paper proposes a 5G based healthcare cardiovascular disease Remote Monitoring system called 5GREM using Electrocardiogram (ECG) bio sensor as a BSN device. The aim is to monitor and analyse the patient's heart rhythms and send emergency alerts during irregularities to the nearest caregivers, ambulance or physician to minimize heart attacks and heart failures while saving energy. Since ECG signal execution is computer intensive, requests from the ECG sensor are either executed locally on the gateway, offloaded to nearby mobile devices or to the 5G edge while considering the battery level, CPU level, transmission power, delays and task fail rate.

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

I. INTRODUCTION

It has been estimated that in the next 10 years, the way healthcare is currently provided will be transformed from hospital centred, first to hospital home balanced in 2020, and then ultimately to remote and home-centred in 2030 [1] powered by Wireless Body Sensor Networks (WBSN). WBSNs continuously and remotely monitor patient's vital parameters (body temperature, blood pressure, heart rate and breathing rate) of patients suffering from chronic diseases such as diabetes, asthma and heart attacks and alerts a caregiver, physician or emergency department when there are abnormalities as such significantly saving lives and reducing care medical expenses. However, when a patient moves out of the range of the fixed healthcare gateway, e.g. away from the smart home into a stadium or city centre, the smart devices then requires a healthcare mobile gateway that moves with the patient and research in this area is limited.

With the introduction of 5G [2], smartphones are becoming energy efficient, with low transmission delays (1ms), high data-rates up to 20Gbps and improved storage e.g. Apple Inc. has recently introduced iPhone XS Max [3] with upto 512GB of storage and 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 and will drain a lot of energy due to computation and transmission power consumption.

This paper is an extension of our previous works in [4] and [5] called IEE5GG, where a 5G and Mobile Edge Computing (MEC) based mobile gateway connected to body sensors was proposed for minimising energy consumption, task failure rate and health service time. The extension in this paper is the introduction of mobile to mobile task offloading using device to device (D2D) communication to argument the gateway processing power and reduce the gateway energy consumption. Also, the introduction of emergency alerts to ambulances and physicians during irregularities detected by the bio sensors is another extension. The contributions of this paper 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 during unusual changes in Electrocardiogram (ECG) signals connected to the patient's suffering from chronic heart attacks.

ii) Improving heart attack alert response time and energy efficiency using MEC offloading scheme in 5GREM to save energy in the 5G smartphone healthcare gateways where the ECG signal execution is partitioned into modules/tasks where decisions are made as whether to execute the tasks on the local device while other tasks are offloaded and executed both in nearby mobile devices via Device to Device (D2D) Communication and in 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 used for maintaining coverage.

II. Materials and Methods

A. Related Work

Extensive research has been done on healthcare monitoring systems using WBSN and the reader is directed to [6]. The main focus of this paper is saving energy in the mobile gateways to avoid interruption during monitoring. The MEC paradigm is a new area and an attractive option for augmenting the processing of computer intensive healthcare applications like augmented reality or ECG [7] by offloading application tasks to the MEC [8] as such, only energy efficient smartphone gateways using MEC and D2D offloading will be addressed. The author in [9] proposes an energy efficient system that offloads mobile tasks from a mobile gateway device to another mobile devices called a cloudlet to reduce energy consumption and improve performance, however the system is based on an outdated radio access network architecture with delays and high bandwidth penalties. The author in [10] proposed a dynamic mobile cloudlet cluster policy to use a group of mobile devices as a supplement for the fog server for offloading tasks from a mobile gateway device. Zhang K. et al. in [11] proposed an Energy Efficient Computation Offloading (EECO) mechanisms for MEC in 5G heterogeneous networks suitable for mobile gateways. However, the author considers MEC servers with limited capacity that are not consolidated in a centralised fashion as in 5G C-RAN. The author in [12] investigates a green smart gateway MEC system and develop an effective computation offloading strategy. The execution cost, which addresses both the execution latency and task failure, is adopted as the performance metric. Nevertheless, the author ignores the execution delay caused by the MEC server and assumes that the battery capacity is sufficiently large which is impractical.

All the above schemes were based on MEC offloading or D2D offloading but not combining both. Also, the above schemes are based on random architectures of which some were just labelled as 5G but this paper addresses the gateway framework from a standardised 5G architecture [2] with accurate 5G requirements considered.

B. Proposed 5GREM Framework System Model

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

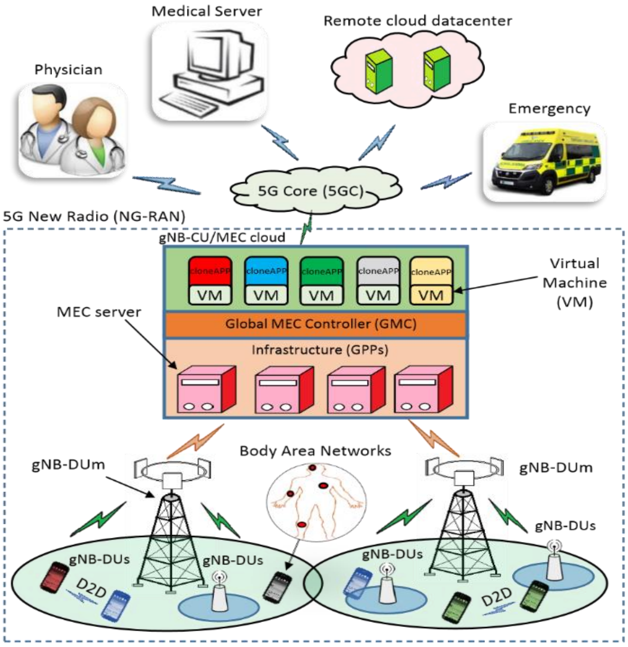

Fig. 1. Proposed 5GREM architecture. gNB-DUm = 5G Macro Distributed Unit BS, gNB-DUs = 5G Small Cell Distributed Unit BS, gNB-CU = 5G Centralised Unit Base Station.

For simplicity, the gNB-DUm is termed as Macro Remote Radio Head (MRRH) and overlaid by small cell gNB-DUs herein termed small cell/pico RRH (PRRH). Define the set of RRH as $R = \{n : n = 1,2,...,N_{rrh}\}$, where N_{rrh} denote the total number of PRRHs. The term smartphone is used interchangeably with user equipment (UE) and mobile device. Consider a set of UEs denoted as $K = \{k : k = 1,2,...,N_{UE}\}$. Each UE k will associate with a serving RRH by the criterion of Cell Range Expansion (CRE) [13] to maximise transmission rate. In the MEC cloud/gNB-DU, there are multi core high processing MEC General Purpose Processors (GPPs) with VM instantiated according to traffic demand of offloaded task requests from smartphone gateways. The Global MEC Controller (GMC) in the gNB-CU is responsible for receiving task requests from the smartphone gateway and scheduling them to Virtual Machines (VMs). The VMs for processing application tasks are abstracted from the GPP using the GMC. The gNB-CU is originally for baseband processing as such some of the VMs are for baseband processing. We only concentrate on VMs for processing tasks for healthcare applications offloading.

C. Communication Model

The 5G system bandwidth, B, is divided into N_{ch} channels each of bandwidth w. Denote a set of channels in the system as $C = \{c : c = 1,2,...,N_{ch}\}$. We assume that smartphones in the same cell transmit over orthogonal channels, whereas UEs of different cells may interfere against each other. We consider that each smartphone runs 5G healthcare application, which can be split into several tasks. Each task T_k of UE k can be executed either locally on the smartphone or remotely on the MEC cloud or nearby devices by computation offloading. Consider that k can offload T_k either via the MRRH or via the PRRH. Denote $a_{k,m,c}$ = {0,1} as the $t_k^l = \frac{D_k}{F_k^l}$, and $e_k^l = t_k^l P_a$ e of k where m = {1,2,3,4} is nodes which are computing locally, transmitting via the MRRH, transmitting via the PRRH and D2D offloading respectively. The variable $a_{k,m,c}$ = 1 means device k uses mode m to offload task T_k through channel c otherwise if $a_{k,m,c}$ = 0, otherwise. The item c is meaningless when m = 1 as there are no channels in local computing, thus $a_{k,1,1}$ = 1 is taken as the indicator that device k select local computation. In case smartphone k offload the task T_k via the MRRH on channel c, the accurate uplink data rate of the UE can be computed as:

$$r_{k,c}^M = wlog_2(1 + \frac{P_k^M H_k^M}{\sum_{j=1,j\neq k}^{N_{UE}} a_{k,3,c} P_j^S H_j^M + \sigma^2}) \tag{1}$$

where w is the channel bandwidth, P_M^k is the transmission power from EU k to the MRRH, H_M^k is the channel gain between UE k and the MRRH. The denominator is the interference caused by other UEs using the same channel for transmission. The variable σ^2 denotes the background noise power. The total uplink data rate of UE k to the MRRH is calculated as:

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

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

D. Computation Model

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

1. Local Computation: Local computation is when the smartphone k executes its computation task T_k locally. Denote F_l^k as the computation capability of the smartphone in MIPS. It is assumed that k can have various computation capabilities.

The execution time (t_k^l) and energy consumed (e_k^l)) for executing task T_k for k can be expressed as:

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

where P_a is the power consumed by k when active.

2. MEC Computation: When k chooses computing its task by the MEC server, the input data can be transmitted to the VM through the MRRH or the PRRH. This means k would incur the extra overhead in terms of time and energy for transmitting the computation input data via 5G wireless access. In case k offloads T_k via MRRH, the total time duration (t_k^M) can be calculated as transmission time plus time during MEC cloud execution of task T_k

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

$$t_k^M = \frac{B_k}{r_k^M} + \frac{D_k}{F_k^{mec}} \tag{4}$$

Where F_k^{mec} is the computation ability of the MEC server VM. There are many VMs in the MEC cloud that can process tasks of an application in parallel. The total energy consumed by k via offloading through the MRRH can then be calculated as

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

Similarly, for offloading via the PRRH,

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

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

3. *Nearby D2D offloading:* In case the UE offload a task to a nearby device, the time of execution denoted t_k^{nd} is 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 is D2D uplink data rate. Similarly, the energy consumption is denoted as $e_k^{nd} = t_k^{nd} * Pa$.

E. Problem Formulation

The aim is to minimise energy consumption in the smartphone by offloading some application tasks to the 5G MEC cloud such that, transmission delay, propagation delay, task processing time, energy consumption in both MEC server and smartphone are minimised while transmission data rate is maximised. The optimization problem is formulated as follows in (11).

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

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

$$\text{C2: } \sum_{k=1}^{N_{NU}}\sum_{c=1}^{N_{ch}} a_{k,m,c} \leq N_{ch},\ m = \{2,3\} \tag{9}$$

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

The function 1(x) is an indicator function which is equal to 1 when x is true and zero otherwise. The first constraint *C1* insures that the delay constraints are met. The second constraint insures that the total number of channels allocated to smartphones does not exceed the total number of channels in the system. Constraint *C3* states that only one channel can be allocated to only one smartphone.

F. 5GREM Offloading Framework

This section will describe the 5GREM offloading framework in detail. The framework is shown in Fig. 3.

The smartphone comprises of an elastic application and other components that enables partitioning and offloading application tasks. The system components are as follows: Device profiler: Collects smartphone hardware context at runtime and pass the information to the offloading agent. The hardware context includes the battery State of Charge (SoC), average CPU utilisation and memory usage. ii) Resource monitor: Resides in both the smartphone gateway and the allocated VM. It collects network related context at runtime and pass the information to relevant modules like the offloading agent. The network context includes network connection state, bandwidth, and signal strength.

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

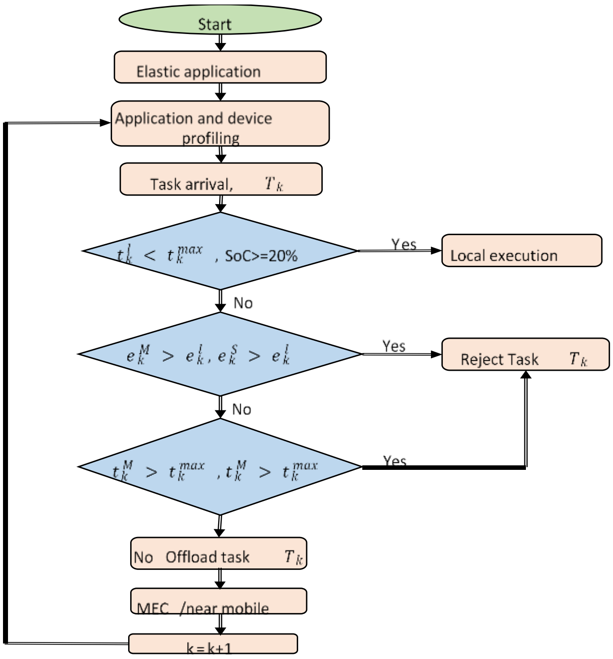
Start
Elastic application
Application and device profiling
Task arrival, T_k
$t_k^l < t_k^{max}$, SoC>=20%
Yes
Local execution
No
$e_k^M > e_k^l, e_k^S > e_k^l$
Yes
Reject Task T_k
No
$t_k^M > t_k^{max}$, $t_k^M > t_k^{max}$
Yes
No Offload task T_k
MEC /near mobile
k=k+1

Fig. 2. The 5GREM offloading agent flow chart

Offloading Agent: Consists of a set of cost estimation models like the delay or execution time and the energy models. Based on the received context above, the offloading agent decides on when, where and how to offload the task. If there are MEC VMs with enough processing power, they are registered in the directory services to execute the tasks, the offload-able class codes task is then uploaded to the class loader of the MEC server VM which then execute the class code in the recipient OS, and after execution, the results are loaded back to the offloading agent in the client device. Also, on the other hand, if there are nearby smartphone device with enough battery power and compute requirements, the task class codes are loaded to the client device and executed in the class loader. The flow chart in Fig. 2 shows operation of the offloading agent.

The offloading agent start with the arrival of a task, T_k. If local execution time is less than the maximum delay tolerable and the battery $SoC \geq 20\%$, the task is executed locally. Else if the offloading delay deadlines are met and energy is saved using offloading, the task is offloaded to the VM in the MEC server or in the nearby mobile device. The next task then follows the same order in the flow chart.

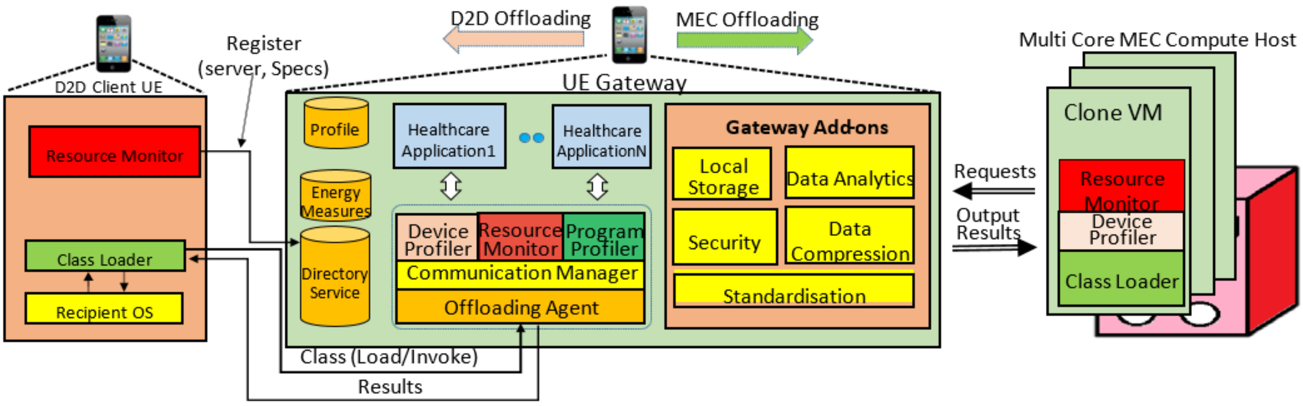

Fig. 3. Proposed 5GREM offloading framework.

Communication manager: It creates and maintains connection between the client and the server side. It serialises the code on the client side and serialises the request from the client at the cloud side. It also keeps the client and the server VM in sync. The communication manager checks if the required files and programs exist in the server side, else it contacts the client device to fetch the files and related libraries for remote execution.

Healthcare gateway add-ons: The healthcare gateway can also be incorporated with some add-ons like local temporary storage in case the network is down, data analytics, security, standardisation of the smart device data using healthcare standards like HL7, HL7 2, or HL7 3 and data compression before uploading to a remote cloud storage eg at the hospital to reduce data size and maximise 5G cellular bandwidth utilisation. All these add-ons can have their tasks processed in the MEC server to save energy in the smartphone.

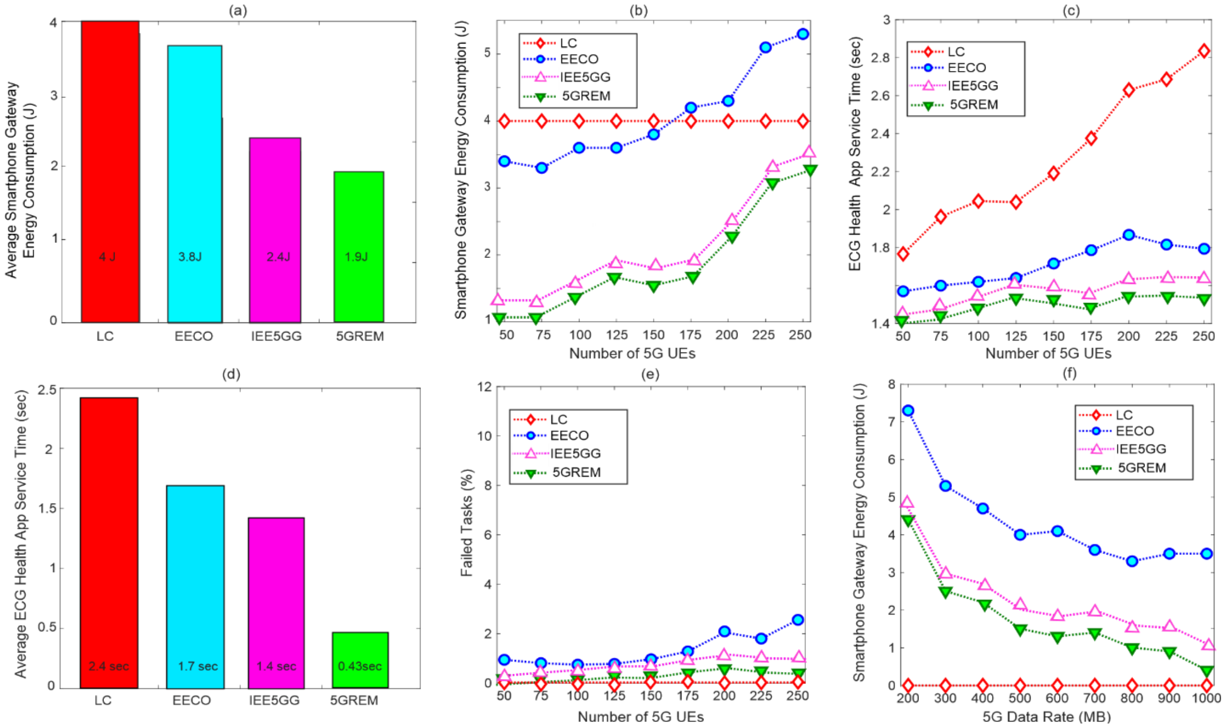

Fig. 4. Simulation results of (a) Average smartphone gateway energy consumption (b) Effects of increased network UEs on smartphone gateway energy consumption(c) Health app task service time (d) Average service time of tasks (e) Failed tasks due to mobility or VM overload (f) Effects of 5G data rate on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

III. Results and Discussions

A. *Results Simulation Tools*

To analyse the performance of the proposed 5GREM framework, one MRRHs overlaid by 12 PRRHs per MRRH is considered. The proposed 5GREM framework is compared with: 1) Local Computation scheme denoted LC in the results where all health tasks are processed locally in the gateway, 2) our previous work in [5] introduced in Section I) the EECO scheme introduced in Section II in [11]. In EECO, the MEC servers are of limited processing capacity and standalone. Table 1 shows the simulation parameters. The application to be considered is the heavy computation health application that uses electrocardiogram (ECG) device connected to the patient's smartphone to monitor heat beat rhythms. The ECG application is broken into two tasks to be executed (Task1 and Task2) with their parameters shown in Table 1. The simulation was performed using an open source Edgecloudsim simulation tool [15]. The Edgecloudsim is a toolkit for modelling and simulation of resource management techniques in IoT and MEC taking into consideration 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 shows the average energy consumed in the smartphone for all the schemes. On average, compared to the local computation, the IEE5GG and the proposed 5GREM scheme saves 40% and 52.5% respectively, this is because both the IEE5GG and the proposed 5GREM schemes involves computation offloading with 5GREM extending the offloading to nearby mobile devices. Fig. 4(b) shows that for the EECO, IEE5GG and 5GREM schemes, as the number of networks UEs increases, the energy consumption in the smartphone gateway increases since more UEs share the bandwidth which causes the uplink data rate of the smartphone to be lower.

Fig. 4(c) shows that as the number of smartphones increases, the service time increases compared to the local computation scheme due to sharing of bandwidth. As shown in Fig. 4(d) the EECO, IEE5GG and the proposed 5GREM schemes saves on average 33%, 41% and 82% respective with 5GREM 49% better due to high processing speed in the centralised MEC servers augmented by D2D offloading. Fig. 4(e) shows that as the more UEs occupies the network, the failure dropping probability of tasks increases but for local computation, fail rate is not affected by the increase in network UEs. The user mobility in D2D has not been included as devices were assumed to be static as such request drop in D2D is omitted, in future user mobility will be considered. Fig. 4(f) shows that as the gateway transmission rate increases, the energy consumed slightly decreases since the transmission time is decreased. Also, the higher the device processing power, the lower the energy consumed since less time is taken when executing some tasks locally.

IV. Conclusions

This paper 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's suffering from chronic heart attacks. The 5GREM framework adopt the 5G technology, Mobile Edge Computing (MEC) and D2D task offloading to reduce energy consumption in the mobile gateway. 5GREM shows that deaths from heart attacks can be significantly reduced.

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/nsp/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 systems 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 networks. 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 Systems and Control (ISCO), 2016 10th International Conference. pp. 1–8.

[9] Y. Zhang, D. Niyato, and P. Wang. (2015). Offloading in mobile cloudlet systems 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 networks. 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 networks. In Communication Software and Networks (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 networks. 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 systems. In Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on. IEEE, pp. 39–44.