ENERGY-EFFICIENT AND DELAY-AWARE OFFLOADING SCHEME USING D2D-ENABLED MOBILE EDGE COMPUTING

RAMTIN RANJI

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

ENERGY-EFFICIENT AND DELAY-AWARE OFFLOADING SCHEME USING D2D-ENABLED MOBILE EDGE COMPUTING

RAMTIN RANJI

DISSERTATION SUBMITTED IN PARTIAL
FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: RAMTIN RANJI

Registration/Matric No.: WOA160013

Name of Degree: Master of Computer Science

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Energy-efficient and Delay-aware Offloading Scheme using D2D-enabled Mobile Edge Computing

Field of Study: Wireless Network

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:
Designation:

ENERGY-EFFICIENT AND DELAY-AWARE OFFLOADING SCHEME USING D2D-ENABLED MOBILE EDGE COMPUTING

ABSTRACT

Energy efficient operation of mobile/Internet of Things (IoT) devices is a major challenge due to the limited capacity of their batteries. Also, because of their limited processing power, many of them cannot perform computationally intensive applications like face recognition in a timely manner. Device to Device (D2D) communication and Mobile Edge Computing (MEC) are two technologies to mitigate these limitations by offloading the computationally intensive tasks. With D2D, mobile/IoT devices can cooperate directly without the intervention of the Base Station. MEC provides computing services at the edge of network, that is close to the user.

Most of the current studies on the energy efficient offloading, put the focus either on the offloading to the edge server, or to a device in the proximity. The problem of MEC solution is scalability, because, edge servers would be overloaded in dense networks. On the other hand, to find a proper offloading destination in D2D networks, devices must consume excessive amount of energy. Recently, a few numbers of integrated schemes were proposed to mitigate those problems. However, there is lack of study to propose both MEC and D2D, as the target of offloading tasks for execution while considering the energy required for offloading and its delay.

In this work, we study Energy-Efficient and Delay-aware Offloading Scheme (EEDOS). In the proposed scheme, energy constraint devices and those with low computational power, have two options to offload their work. They can either use MEC, or D2D, and the computational power of the edge server is leveraged to find a proper candidate. For

EEDOS network topology, we integrate D2D communication capability in the user layer of mobile networks, so that, mobile users can communicate with MEC layer. The research problem is formulated with consideration of the required energy for task offloading and completing the task execution, under the required deadline.

The EEDOS, MEC and D2D offloading schemes have been simulated to evaluate the proposed scheme and to validate the findings with the existing schemes. The numerical results showed that EEDOS was capable to save the energy of mobile devices up to 95 % in comparison to local task execution, as well as reducing the execution delay. The proposed EEDOS, outperformed the existing schemes in terms of task offloading energy consumption and execution delay. This is due to the integration of the computational capability of MEC and idle devices in the network. The resource-limited mobile devices can save more energy, because, the proposed EEDOS, used edge servers to find the proper offloading destination and took into account the high computational power of edge servers and computational resources of large number of idle devices in the network. In this scheme, the load on the edge server was decreased dramatically in comparison to current MEC offloading schemes, because of participating idle devices in the network through D2D communication.

ABSTRAK

Skim Nyahmuatan yang Cekap-Tenaga dan Sedar-Kelewatan menggunakan D2D yang dibolehkan dengan Pengkomputeran Edge Mudah-alih

Operasi cekap tenaga mudah alih / Internet of Things (IOT) merupakan cabaran besar disebabkan oleh kapasiti bateri yang terhad. Selain itu, banyak peranti mudah alih tidak dapat melaksanakan aplikasi intensif pengiraan seperti pengenalan wajah tepat pada masanya. Komunikasi kepada Peranti kepada Peranti (D2D) dan Pengkomputeran Edge Mudah-alih (MEC) adalah dua teknologi untuk mengurangkan batasan-batasan ini dengan mengira tugas intensif pengkomputeran, dan mereka akan dilaksanakan dalam rangkaian Generasi Kelima (5G). Dengan D2D, peranti mudah alih / IOT boleh bekerjasama secara langsung tanpa campur tangan Stesen Pangkalan. MEC menyediakan keupayaan pengkomputeran awan di pinggir rangkaian, dekat dengan pengguna.

Kebanyakan kajian semasa mengenai penyiasatan enjin tenaga, meletakkan perhatian sama ada pada pelayan pelayan tepi, atau ke peranti berdekatan. Masalah penyelesaian MEC adalah skalabilitas, kerana pelayan tepi akan terlalu banyak dalam rangkaian yang padat. Di sisi lain, untuk tujuan destinasi yang betul dalam rangkaian D2D, peranti mesti mengambil jumlah tenaga yang berlebihan. Baru-baru ini, beberapa skim bersepadu yang dicadangkan untuk mengurangkan masalah tersebut. Walau bagaimanapun, terdapat kekurangan kajian untuk mencadangkan kedua-dua MEC dan D2D sebagai sasaran tugas-tugas oo sambil menimbangkan tenaga yang diperlukan untuk memberi keterangan dan kelewatannya.

Dalam karya ini, kita mengkaji Skim Nyahmuatan yang Cekap-Tenaga dan Sedar-Kelewatan menggunakan (EEDOS). Dalam skim yang dicadangkan, peranti kekangan tenaga dan mereka yang mempunyai kuasa pengiraan yang rendah, mempunyai dua pilihan untuk

mengerjakan kerja mereka. Mereka boleh sama ada menggunakan MEC, atau D2D, dan kuasa pengiraan pelayan kelebihan dimanfaatkan untuk mendapatkan calon yang tepat. Kami memperoleh pesawat pengguna D2D yang boleh berkomunikasi dengan pelayan tepi, sebagai topologi rangkaian EEDOS. Masalah tugas yang dikemukakan digubal dengan pertimbangan tenaga yang diperlukan untuk tugas dan pelaksanaan tugas, di bawah syarat akhir tugas.

Skim pemunggahan EEDOS, MEC dan D2D telah disimulasikan untuk menilai skim yang dicadangkan dan untuk mengesahkan penemuan dengan skema yang sedia ada. Keputusan berangka menunjukkan bahawa EEDOS mampu menjimatkan tenaga peranti mudah alih sehingga 95% dibandingkan dengan pelaksanaan tugas tempatan, serta mengurangkan masa pelaksanaan. EEDOS yang dicadangkan, mengatasi skim sedia ada dari segi tugas mengimbangi penggunaan tenaga dan jumlah masa pelaksanaan. Ini berlaku kerana integrasi keupayaan pengiraan MEC dan peranti terbiar dalam rangkaian. Peranti mudah alih yang terhad untuk sumber daya dapat menjimatkan lebih banyak tenaga kerana EEDOS yang dicadangkan, menggunakan pelayan tepi untuk mencari destinasi pemunggahan yang betul dan mengambil kira kuasa pengiraan tinggi pelayan tepi dan sumber pengiraan sejumlah besar peranti terbiar dalam rangkaian. Dalam skema ini, beban pada pelayan tepi menurun secara dramatik berbanding dengan skema pemunggahan MEC semasa kerana peranti yang terbiar yang mengambil tugas dalam bahagian rangkaian melalui komunikasi D2D.

ACKNOWLEDGEMENTS

I would like to express my deepest gratiude to my supervisors Dr Ali Mohammed Mansoor and Dr Asmiza Abdul Sani for their kind suggestions and help throughout my research journey.

TABLE OF CONTENTS

Abstract iii

Abstrak v

Acknowledgements vii

Table of Contents viii

List of Figures xii

List of Tables xiv

List of Symbols and Abbreviations xv

List of Appendices xvii

CHAPTER 1: INTRODUCTION 1

1.1 Background 1

1.2 Problem Statement 3

1.3 Research Objectives 5

1.4 Research Significance 6

1.5 Research Scope 6

1.6 Research Steps 7

1.7 Dissertation Organisation 8

1.8 Conclusion 9

CHAPTER 2: LITERATURE REVIEW 10

2.1 Introduction 10

2.2 Energy efficiency techniques in cellular networks 12

  2.2.1 Fog and Mobile Edge Computing 12

  2.2.2 Energy efficiency by MEC 13

2.2.3 Device-to-Device Communication (D2D) 17
2.2.4 Challenges of separate D2D and MEC techniques studies 19
2.2.4.1 Device Discovery in D2D 20
2.2.4.2 Malicious Nodes 20
2.2.4.3 Overloaded MEC 20
2.3 Integrating D2D and MEC for energy efficiency 21
2.3.1 Benefits of integration 21
2.3.1.1 Benefits for the service provider 21
2.3.1.2 Benefits for users 22
2.3.2 Applications of D2D enabled MEC 22
2.4 Research trends on D2D enabled MEC 23
2.4.1 Problem-solving studies 27
2.4.2 D2D and MEC integration architecture 29
2.4.2.1 Massive D2D systems 29
2.4.2.2 Small world IoT 33
2.4.3 Energy efficient integrated framework 35
2.5 Related Work 35
2.6 Research Gap 36
2.7 Conclusion 37

CHAPTER 3: RESEARCH METHODOLOGY 39

3.1 Introduction 39
3.2 Energy-Efficient and Delay-aware Offloading Scheme (EEDOS) 39
3.2.1 System Architecture 40
3.2.2 Problem Formulation 40

3.2.2.1 User/IoT device model 40
3.2.2.2 MEC server computing capacity model 43
3.2.2.3 Task execution model 44
3.2.3 Optimisation problem 47
3.2.4 Offloading selection algorithms 48
3.3 Simulation 52
3.3.1 Approach 53
3.3.2 Scenario 56
3.3.3 Required Software Tools 56
3.3.4 Hardware Used 57
3.4 Conclusion 57

CHAPTER 4: RESULTS AND DISCUSSION 59

4.1 Introduction 59
4.2 Comparison of EEDOS to current studies 59
4.2.1 Energy Saving 59
4.2.2 Energy Consumption 61
4.2.3 Execution time 63
4.2.4 Successful offloading 65
4.2.5 Meeting Deadline 66
4.2.6 Edge server resource 68
4.3 Conclusion 69

CHAPTER 5: CONCLUSION 70

5.1 Introduction 70
5.2 Objectives and Findings 70

5.3 Results Summary 72

5.4 Limitation of the study 72

5.5 Future research direction 73

  5.5.1 Blockchain 74

    5.5.1.1 Consensus Mechanism 75

    5.5.1.2 Smart Contracts 76

    5.5.1.3 Types of Blockchain 76

    5.5.1.4 Is Blockchain applicable in Cooperative MEC? 76

  5.5.2 Token-based Cooperative platform for Mobile Edge Computing 77

    5.5.2.1 Extension to the offloading scheme 79

References 82

Appendices 93

LIST OF FIGURES

Figure 1.1: MEC offloading scheme 4

Figure 1.2: D2D offloading scheme 5

Figure 1.3: Research Methodology 7

Figure 2.1: Reviewed studies and their relation to the research objectives 11

Figure 2.2: Integrated architecture of D2D enabled MEC 23

Figure 3.1: Network topology of EEDOS 42

Figure 3.2: Workflow of EEDOS scheme 49

Figure 3.3: Decision graphs that built in EEDOS workflow 50

Figure 3.4: SimuLTE simulation environment in OMNET++ 54

Figure 3.5: SimuLTE sample setting 55

Figure 3.6: Simulation Steps 55

Figure 4.1: Energy Saving ratio with increasing number of devices 60

Figure 4.2: Energy Saving ratio with increase in the task frequency 61

Figure 4.3: Energy Consumption with increasing number of devices 62

Figure 4.4: Energy Consumption with increase in the task frequency 63

Figure 4.5: Execution Delay with increasing number of devices 64

Figure 4.6: Execution Delay with increase in the task frequency 65

Figure 4.7: Offloaded requests 66

Figure 4.8: Missed deadlines 67

Figure 4.9: Edge CPU required to handle all offloading requests and meet deadline 68

Figure 5.1: Blockchain in a Peer to Peer network with MEC 75

Figure 5.2: Blockchain operation for incentivize the cooperation 80

Figure A.1: Matching in a non-bipartite graph. 94

Figure A.2: Augmenting path from free vertex 3 to free vertex 5. 95

Figure A.3: Matching after inverting the augmenting path. 96

Figure A.4: Layered BFS tree 96

Figure A.5: Supernode creation 98

Figure A.6: Invert the augmenting path 98

Figure A.7: Shrinking in Edmond's Blossom algorithm 99

Figure A.8: Reconstruction of augmenting path clockwise 100

Figure A.9: Reconstruction of augmenting path counter-clockwise 100

Figure A.10: Edmond's Blossom example: steps a to f 102

Figure A.11: Edmond's Blossom example: steps g to l 103

Figure A.12: Edmond's Blossom example: steps m to s 104

LIST OF TABLES

Table 2.1: Energy efficiency by offloading to MEC 13

Table 2.2: D2D Energy efficiency techniques 18

Table 2.3: Studies on both D2D and MEC 25

Table 2.4: Research gaps in Chen et al.,(2017) 37

Table 2.5: Research gaps in Lyu et al.,(2018) 37

Table 3.1: Nomenclature used in this section 41

Table 3.2: Simulation Parameters 56

LIST OF SYMBOLS AND ABBREVIATIONS

B_i^{dl} : Required cellular downlink traffic for task i.
B_i^{ul} : Required cellular uplink traffic for task i.
E_i^{cc} : Energy required for device i to offload the task to the cloud computing server.
E_{ij}^{d2d} : Energy required for device i to offload the task to j.
E_i^{l} : Energy required for device i to perform the task locally.
E_i^{mec} : Energy required for device i to offload the task to the edge server.
I_i : Input size of task i.
N : Set of devices in the communication range of MEC server.
O_i : Output size of task i.
P_i^{dl} : Receive power of i.
P_i^{d} : D2D transmission power of device i.
P_i^{ul} : Transmission power of i.
R_{ij}^{d2d} : Data rate of communication from device i to j.
R_i^{dl} : Downlink data rate of i.
R_i^{ul} : Uplink data rate of i.
T_{ij}^{d2d} : Total time of offloading task of i to j.
T_i^{l} : Total time of performing task i locally.
T_i^{mec} : Total time of offloading task i to MEC server.
T_i^{req} : Deadline for task i.
W_i^{t} : Current energy level of device i.
W_i^{tr} : Predefined energy threshold for device i to enter energy sensitive mode.
Z_i : Required CPU cycles of task i.
μ_i : Decision about scheduling task of i in MEC server $\mu_i \in \{0, 1\}$.
ρ_i : Energy cost per CPU cycle for device i.
θ : Current CPU load of $i, 0 < \theta \leq 1$.
f_i^0 : CPU clock of the device $i \in N$.
f_{mec}^0 : Initial computation capacity of MEC server dedicated for offloaded tasks.
f_i^l : Current computation capacity of device i.
f_j^{d2d} : Allocated resources of j to perform task i.
f_{mec} : Current computation capacity of MEC server.
mu_{it} : Decision about whether task of i is delay-sensitive or not.

3GPP	:	3rd Generation Partnership Project.
5G	:	Fifth Generation.
AES	:	Advanced Encryption Standard.
BBU	:	Base Band Unit.
CQI	:	Channel Quality Indicator.
CR	:	Cognitive Radio.
CSMA	:	Carrier Sense Multiple Access.
D2D	:	Device to Device.
EE	:	Energy Efficiency.
EEDOS	:	Energy-Efficient and Delay-aware Offloading Scheme.
eMBMS	:	Evolved multimedia broadcast/multicast service.
Fog-AP	:	Fog Access Point.
HPN	:	high power node.
IoT	:	Internet of Things.
KGC	:	Key Generation Center.
LTE	:	Long Term Evolution.
MANET	:	Mobile Ad Hoc Network.
MEC	:	Mobile Edge Computing.
METIS	:	Mobile and wireless communications Enablers for Twenty-twenty Information Society.
mmWave	:	Millimetre Wave.
O-MEC	:	Overloaded Mobile Edge Computing.
PoW	:	Proof of Work.
ProSe	:	Proximity Service.
QoS	:	Quality of Service.
RAN	:	Radio Access Network.
RB	:	Resource Block.
RDNA	:	Robust Dynamic Network Architecture.
RSA	:	Rivest–Shamir–Adleman.
SDN	:	Software Defined Networking.
SE	:	Spectral Efficiency.
SINR	:	Signal to Interference plus Noise Ratio.
SSDNet	:	Small-World Super-Dense Device-to-Device Wireless Network.
UE	:	User Equipment.
V2I	:	Vehicle to Infrastructure.
V2V	:	Vehicle to Vehicle.
V2X	:	Vehicle to Everything.
VM	:	Virtual Machine.

LIST OF APPENDICES

Appendix A: Edmond’s Blossom Algorithm 93

CHAPTER 1: INTRODUCTION

The primary aim of this study is to increase the energy efficiency of mobile/IoT devices and reducing the task execution delay. In this chapter, the author first identified the problem and defined the research objectives. Then the author proved the importance of this study and described the leveraged research methodology. Finally, the organisation of this dissertation discussed in the last section.

1.1 Background

In the last decade, mobile phones transformed from a simple device for making phone calls and a multimedia gadget, to a fully functional computing device. Today, using smartphones have become part of every day's activity, and the users need mobile phone to operate for extended hours throughout the day. However, their portability and design constraint, made them less capable for today's resource-hungry applications. Augmented reality, face recognition and online gaming are few examples of applications that need high computation capability and low latency network (Soyata, Muraleedharan, Funai, Kwon, & Heinzelman, 2012). Improving the processing power of mobile devices to support more features, without a significant evolution in the design of their batteries is responsible for the fast battery drainage, and make them incapable of long hours operating (Ali, Simoens, Verbelen, Demeester, & Dhoedt, 2016).

IoT devices are deployed widely and the number of devices are expected to grow exponentially in the coming years (Akpakwu, Silva, Hancke, & a.M. Abu-Mahfouz, 2017). IoT devices are extremely resource limited in terms of processing power, storage, and battery capacity. However, in some applications, they need a low latency response like gas leakage scenario (Varma, Prabhakar, & Jayavel, 2017). As another example, Vehicle to Everything (V2X) cannot perform well without a real-time response (Hou et al., 2016).

Therefore, low delay operation of mobile/IoT devices is an important challenge for hardware and network engineers.

Cloud computing is one of the technologies that can help to save the energy of mobile and IoT devices and it is deployed broadly. In cloud computing, devices can offload their computation-intensive tasks to the cloud server, where the immense volume of computing power exists to facilitate the work of resource limited devices (Gubbi, Buyya, Marusic, & Palaniswami, 2013)(Vallati, Virdis, Mingozzi, & Stea, 2015). However, the delay imposed by cloud computing is not suitable for many of today's delay sensitive applications like the ones mentioned before (Dolui & Datta, 2017). Devices can communicate with cloud servers through WiFi or broadband wireless technologies like Long Term Evolution (LTE). In this study, the author selected cellular network as the communication network. Fifth Generation (5G) is the newest mobile wireless standard, based on the IEEE802.11ac standard of cellular technology. In the 5G design, energy consumption of the network is expected to decrease by the factor of 10 (Chávez-Santiago et al., 2015). MEC and D2D communication are two technologies in the 5G to save the energy of mobile/IoT devices and decrease their communication and computing delay.

MEC is proposed to overcome the long delay of task offloading in cloud computing. There is no need for time-consuming transmission of data through the core network in MEC design. MEC collects data from the end user or IoT devices and processes it at the edge of the network without sending it to the traditional cloud (Reznik et al., 2017). Offloading the computation hungry tasks to the MEC will lead to saving the energy of those devices with a low delay. Also, real-time applications can be supported by the task offloading.

D2D communication was first introduced in the LTE Rel-12 standard of cellular network as a Proximity Service (ProSe) (Jaffry, Hasan, Gui, & Kuo, 2017). It is a promising technique in 5G to provide service for a resource-limited device by utilizing communication resources of a more powerful nearby device without the intervention of base station (Feng, Lu, Yuan-Wu, Li, & Li, 2014). Devices in the 5G network are heterogeneous and have different computation and storage capabilities. As the number of devices connecting to a single cell increased in the 5G, there is a huge probability that, many powerful devices become available in the proximity of a resource limited device. As a result, D2D collaboration is able to fully exploit task offloading. In this technique, devices with limited resources like CPU and energy, can offload their task to a more powerful device to save their energy and decrease the computation delay (Bonomi, Milito, Zhu, & Addepalli, 2012). It must be noted that, this study, put the attention mostly on the benefits of task execution out of the local hardware of IoT/Mobile devices.

1.2 Problem Statement

As depicted in Figure 1.1, in MEC offloading scheme that presented in (Lyu, Tian, Jiang, et al., 2018) and (K. Zhang et al., 2016), latency requirements and capabilities of MEC servers are taken into account to design an energy efficient framework. However, collaboration with proximate idle devices in the network is overlooked in their framework. Additionally, their solutions cannot scale well in densely populated networks, because, after certain number of requests, the edge server will be overloaded (Satria, Park, & Jo, 2017). By integrating D2D collaboration into their framework, the author can leverage the capabilities of idle devices in the network, to contribute to more energy saving while decreasing the load on the edge server.

Enabling low delay applications like face recognition, is a primary motivation for developing 5G networks. However, in the existing D2D offloading schemes, (Chen & Zhang, 2017) and (Chen, Pu, Gao, Wu, & Wu, 2017), only the energy consumption of the device is taken into account, which is not applicable for delay-sensitive tasks. Moreover, the

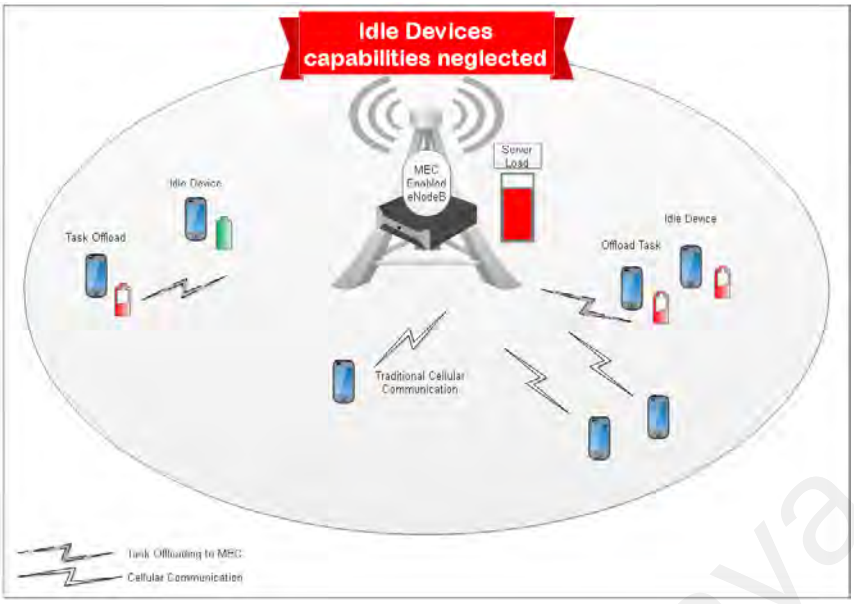


Figure 1.1: MEC offloading scheme

computation capability of MEC for offloading was overlooked in (Chen et al., 2017), while it can be adopted to save the energy of mobile devices and decrease the offloading time. In Figure 1.2, the author showed that, in D2D offloading schemes, the computational power of the server was not employed for task execution, as a result, the server load would be low. Enhanced NodeB or eNodeB is the communication node, that connects the user devices in the cellular network to the rest of the network that is called base station in the legacy networks (Larmo et al., 2009). Since in 5G the edge servers are existing in eNodeB or base station, we can leverage their resources for task offloading. In addition, transmitting data consumes much more energy than receiving it. However, in their studies, they overlooked it for simplicity and do not differentiate between them. As a result, their framework will produce the same results for the tasks that have a different ratio of input/output.

The author concludes that, there is lack of comprehensive study that jointly considers D2D and MEC as an offloading destination for task execution, with consideration of energy consumption and task deadline as two primary metrics.

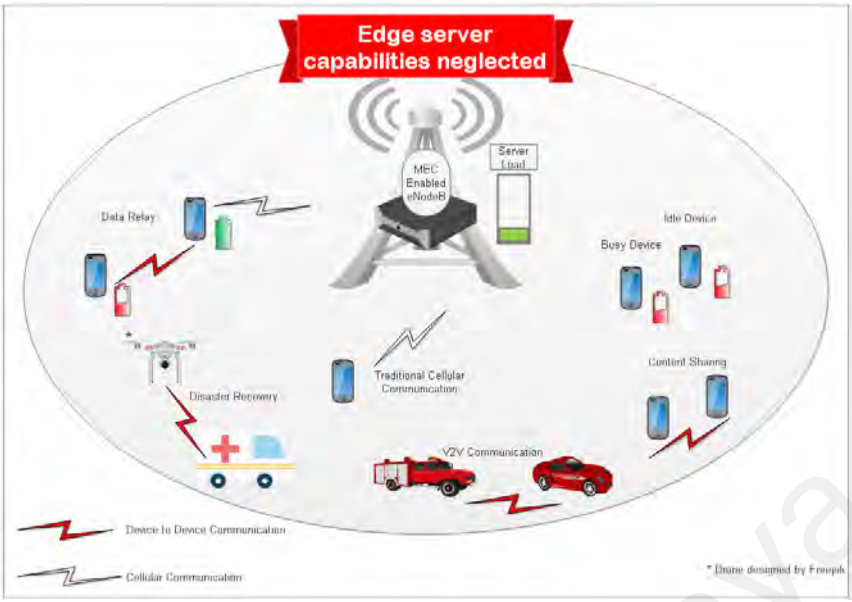


Figure 1.2: D2D offloading scheme

1.3 Research Objectives

The current problem is that how we can join D2D and MEC into an energy efficient and low latency scheme to overcome the problems of each solution. So, the main aim of this study, is to propose an energy efficient and delay-aware scheme, that use D2D capabilities in the MEC network for task execution. Since these two techniques are going to be included in the 5G, more research is needed to develop an effective scheme for cooperation of devices in 5G network. Following is the list of the study's research objectives:

- To investigate energy and delay challenges of IoT/mobile devices and current methods to mitigate them with D2D collaboration and offloading in MEC.
- To develop an offloading scheme by leveraging D2D collaboration method in MEC with consideration of energy and delay parameters.
- To evaluate the performance of the proposed scheme concerning energy efficiency and total execution time, and compare it with the current D2D collaboration and MEC offloading studies.

1.4 Research Significance

The main aim of this study is to improve the energy efficiency and delay of IoT and mobile devices in the task execution. The author proposed a scheme to reduce the cost of task execution in terms of delay and energy. This effort paved way to implement an energy efficient and delay aware mobile network. Through this, users can enjoy running computation-intensive applications without the need for constant device upgrade and worrying about their battery drainage. At the same time, the developers can develop new cutting-edge applications for variety of domains like transportation, entertainment, public health and social security.

1.5 Research Scope

In this research, the author designed an energy-efficient and delay-aware offloading scheme (EEDOS) to help mobile/IoT devices to perform their computational hungry tasks. The primary focus of this study is to find a proper offloading destination to perform and complete the offloaded task within the deadline. Underlying communication details in the physical layer were not included in this study. So, the goal of this study is not to improve D2D communication technology and its performance. Instead, finding the most energy efficient neighbour, or selecting the edge server to run and complete the task is the primary technique to achieve energy efficiency in this study. In addition, the author assumed the general hypothesis as used in the similar studies (Chen et al., 2017) that, devices are willing to cooperate. Therefore, the malicious behaviour and incentive mechanisms were not considered in the scheme design. However, the author refers the interested reader to Section 5.5, where a solution based on the Blockchain technology discussed briefly, as the future research direction.

1.6 Research Steps

The steps of the research is presented in Figure 1.3. To best understand the current state of D2D communication and computation offloading paradigm, and to accomplish the first research objective, the author conducted a literature review on them. The author then classify studies based on the leveraged technology to achieve energy efficiency. From the literature review, the author have identified the research gaps regarding the energy efficient offloading scheme of mobile/IoT devices with low latency.

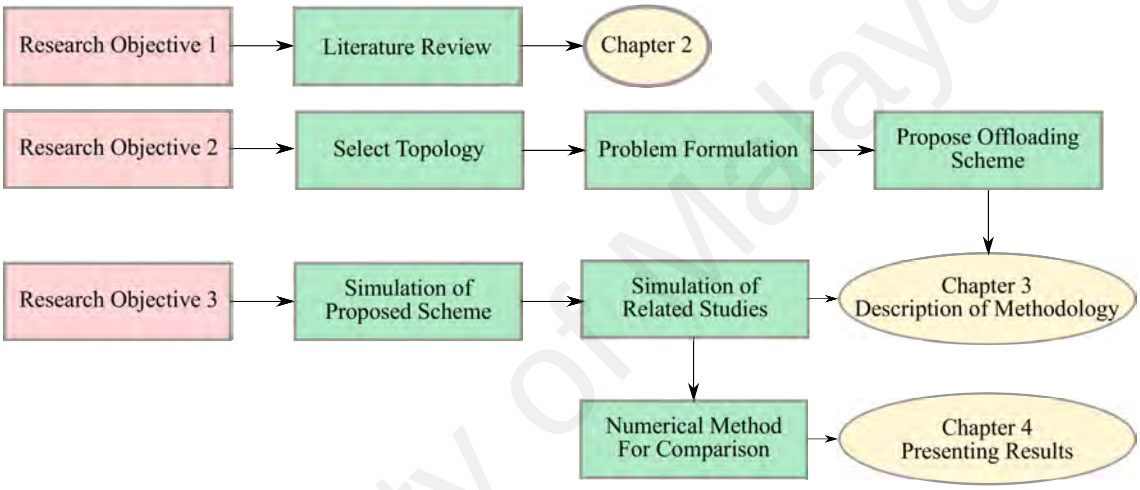


Figure 1.3: Research Methodology

The author proposed a novel and comprehensive scheme with consideration of idle device, edge and cloud servers' capabilities. To propose the computation offloading decision algorithm, the author first defined the study's network topology, then formulated the problem for user/IoT device, MEC server, task parameters and task execution. The author proposed two algorithms to find the proper offloading destination. One for the delay sensitive tasks and the other for energy constraints devices. Therefore, the second research objective met with an extensive scheme, problem formulation and offloading decision algorithms.

The author has used simulation to accomplish the last research objective. The author has reviewed numerous simulation tools like NS-2, NS-3, OMNET++, MATLAB and ONE.

However, it has been identified that, there is a lack of comprehensive tool that supports all the features of MEC offloading in LTE networks, task execution battery consumption, and D2D communications. As a result, the author has used SimuLTE (Virdis, Stea, & Nardini, 2015) simulation tool in Omnet++ to simulate the movement of the nodes in a realistic LTE channel with D2D communication capability. The Signal to Interference plus Noise Ratio (SINR) of each D2D and cellular communication can be extracted from the log file of the simulation in SimuLTE (Virdis et al., 2015). From the SINR, the author calculated the realistic D2D and cellular data rate. Then the author developed a JAVA code to run the proposed scheme, based on the defined formulas and offloading selection algorithms. To compare the results of this study with the current studies, the author implemented them in JAVA, and tested them with the same parameter values as EEDOS. Finally, the author compared the results of the proposed scheme with the current studies, based on the results of the numerical values of the simulation results. The primary metrics to compare the schemes are energy consumption of the mobile devices, and the execution time of the tasks. These metrics and some additional ones discussed in details in Chapter 4

1.7 Dissertation Organisation

The organisation of this dissertation follows the conventional format. In Chapter 2, the author presented the literature review and concluded it with the identification of related studies and resolution of the research gap. In Chapter 3, the author presented how the study achieved the research objectives. The author proposed EEDOS and presented its design in detail. Then, simulation approach to validate it, and requirements to implement the solution were discussed. Then, the author discussed about the results of the simulation in Chapter 4 where extensive comparison of EEDOS to the existing studies was presented. Finally, the author concluded this study in Chapter 5, and the author suggested cooperation stimulation mechanism in that chapter for future research's direction.

1.8 Conclusion

Mobile/IoT devices are operating on batteries. Therefore, their energy efficient operation is always attracted researchers in the industry and academia. On the other hand, real-time applications like Face Recognition require heavy computation capability that most of the mobile/IoT devices cannot afford. In this chapter, the author introduced MEC and D2D, as two technologies to mitigate these problems. However, the author argued that, current solutions hasd some limitations. Based on that, the author identified and defined research objectives and presented the methodology to achieve them. The rest of this dissertation presents the integrated MEC and D2D scheme that address energy efficiency of mobile/IoT devices and their task deadlines.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In developed countries, existing communication systems usually consume around five percent (5%) of the total energy (Morley, Widdicks, & Hazas, 2018). European Mobile and wireless communications Enablers for Twenty-twenty Information Society (METIS) project forecasted that, 5G would have at most 100 times more connected devices, therefore, more energy would be consumed. Ideally, the capacity of 5G data traffic that is increased by the factor of 1000 should have an energy efficiency of the system at 1/1000 of the current energy per bit consumption (Datang Wireless Mobile Innovation Center, 2013). With the calculation of line of sight and non-line of sight coverage of the mobile devices, there will be up to 2,650 to 190,000 connected devices around a single device in a populated city (Cheng, Yu, Zhao, & Cheng, 2018). Having a significant number of devices in the proximity will bring collaboration opportunities to save the energy for smart devices.

D2D communication and MEC are the two promising techniques in the 5G cellular networks that enable collaboration and energy efficiency. Some reviewed studies either put the focus on the D2D communication (Asadi, Wang, & Mancuso, 2014), or Mobile Edge/Fog computing (J. Hu, Heng, Li, & Wu, 2017)(Mao, You, Zhang, Huang, & Letaief, 2017)(Abbas, Zhang, Taherkordi, & Skeie, 2018), and particularly in (H. Wang & Fapojuwo, 2017) authors provide an extensive review of edge computing and caching technologies. D2D communication presented in existing work deals with computation offloading without studying its effect on the energy consumption. Therefore, there is lack of comprehensive study on integrated solutions of D2D communication and MEC. In this chapter, the author is focusing on MEC and D2D techniques to provide an overview of energy-efficiency methods and to discuss the most relevant research challenges that need to be addressed in

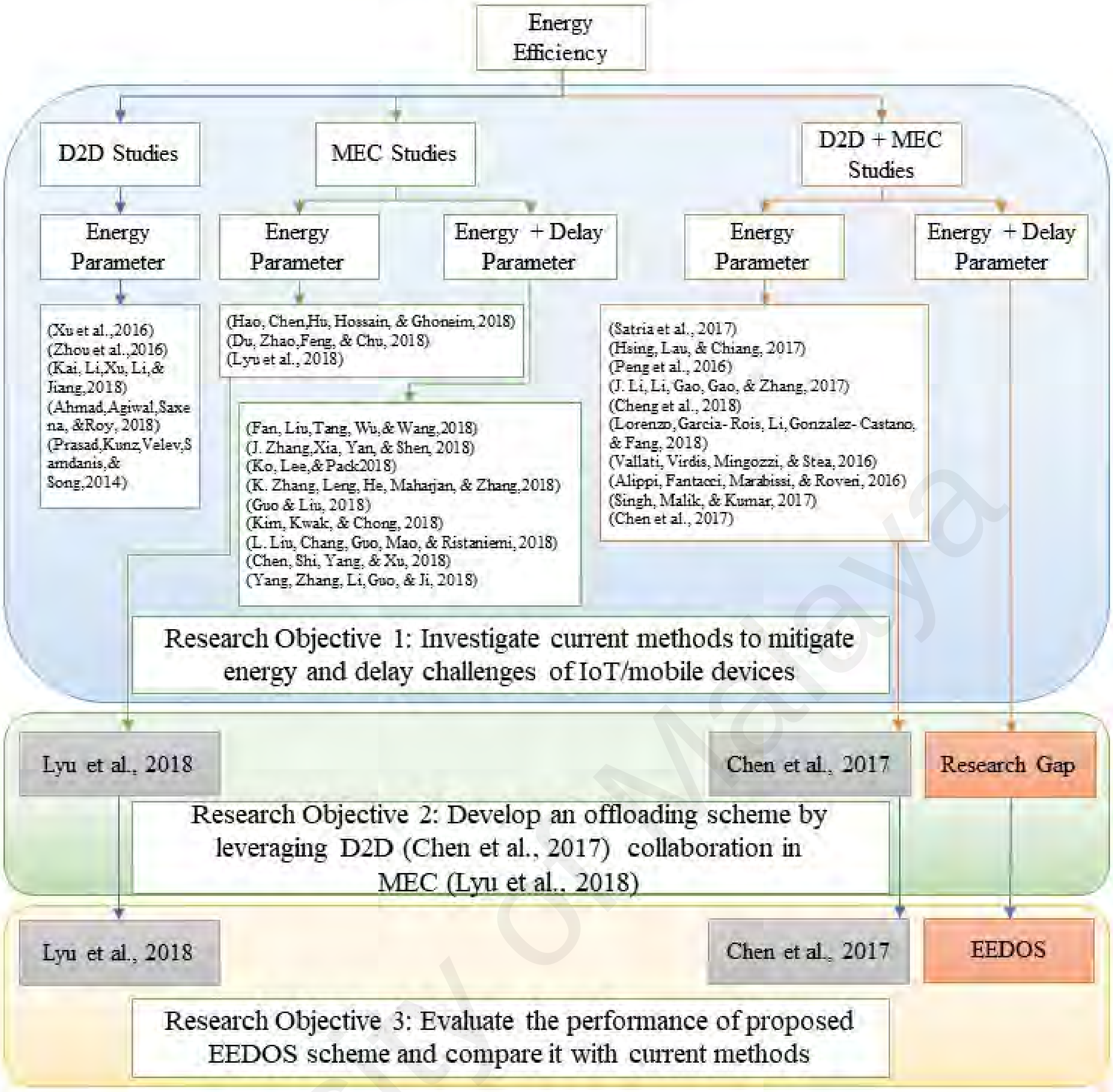


Figure 2.1: Reviewed studies and their relation to the research objectives

the future, to achieve the first research objective. Based on the reviewed studies, related studies have been selected to identify the research gaps and to propose a solution for reaching the research objective. For the third research objective, the author compared the results of the proposed solution evaluation, with the current schemes. Figure 2.1, presents the relation between reviewed studies and the research objectives.

2.2 Energy efficiency techniques in cellular networks

2.2.1 Fog and Mobile Edge Computing

Currently, cloud is used to leverage the computation limitation and battery capacity of Internet of Things (IoT) and mobile devices (Gubbi et al., 2013)(Vallati et al., 2015), but this is not a scalable solution since the future growth of the number of connected devices are predicted to be steep, and it is not a proper solution, especially for delay sensitive tasks (Dolui & Datta, 2017). Cisco predicted that the global IP traffic would reach 278 EB per month by 2021 (Cisco, 2017). While some of the services can be provided at the edge of the network closer to the operating devices, this exponential growth of IoT devices will rapidly add to the overall traffic of the internet.

Until publishing the reference fog architecture by OpenFog consortium (OpenFog, 2017), there was no clear differentiation between MEC and fog computing. Many of the prior studies erroneously used these two technologies interchangeably. According to the new definition, fog is the continuum of cloud computing that brings resources and services of computing, storage, control and networking anywhere close to the devices while edge computing places applications, data and processing to the logical extreme of the network. Hence, edge computing is a subset of fog computing system-level architecture.

In the new network architecture, the MEC resides within the Radio Access Network (RAN) in the proximity of the user. Here, users can use MEC services without the need to communicate with servers on the Internet.

The primary motivations for MEC are to reduce the latency and increase the efficiency of service delivery, which all would lead to better user experience (Y. C. Hu, Patel, Sabella, Sprecher, & Young, 2015). The characteristics of MEC such as low latency, high bandwidth, and proximity to the user can empower mobile service providers with new services. Cisco suggested using fog in the following scenarios (Cisco Systems, 2016):

- When data is needed to be collected at the edge of the network like Vehicle to Infrastructure (V2I) communications
- In scenarios in which IoT devices are densely deployed (IoT big data application)
- When we need to analyse data collected by devices like augmented reality promptly

2.2.2 Energy efficiency by MEC

Energy consumption reduction in User Equipment (UE) and IoT can be achieved by computation offloading. In this technique, devices with limited resources can offload their computation-intensive tasks to the MEC server. Existing review studies (Aazam, Zeadally, & Harras, 2018)(Mouradian et al., 2018) cover energy efficiency in fog computing. Authors in (Aazam et al., 2018) specifically address the task offloading technique in fog computing. Since the focus of this section is to address the challenges of MEC-only studies, the author briefly reviewed the state-of-the-art works in Table 2.1. The following abbreviation used for the sake of table readability. Approaches: A-Algorithm, Sc- Scheme. Evaluation: S-Simulation, R-Real Equipment Measurement. Metrics: EE-Energy Efficiency, DR- Delay Reduction, RD-Required Delay, CR-Complexity Reduction, Ue-Number of beneficial UEs, PC-Payment Cost.

Table 2.1: Energy efficiency by offloading to MEC

Paper	Approach	Evaluation	Major Contribution	Metrics
(Fan, Liu, Tang, Wu, & Wang, 2018)	A	S	Propose a scheme to improve the computation offloading by further offloading the extra tasks of one MEC to another MEC	EE, DR

Table 2.1: Energy efficiency by offloading to MEC continue

Paper	Approach	Evaluation	Major Contribution	Metrics
(J. Zhang, Xia, Yan, & Shen, 2018)	A-Sc	S	Propose a low complexity scheme in MEC enabled heterogeneous networks	EE, DR, CR
(Ko, Lee, & Pack, 2018)	A-Sc	S	Propose a Markov decision process in which a mobile device decides where and when performs task offloading in edge cloud-enabled heterogeneous networks	EE, DR
(Hao, Chen, Hu, Hossain, & Ghoneim, 2018)	A	S	They consider task caching with task offloading. Propose algorithm to determine which task should be cached and how much task should be offloaded.	EE, RD
(K. Zhang, Leng, He, Maharjan, & Zhang, 2018)	Sc	S	Present a mobility-aware MEC framework for IoT	EE, DR

Table 2.1: Energy efficiency by offloading to MEC continue

Paper	Approach	Evaluation	Major Contribution	Metrics
(Du, Zhao, Feng, & Chu, 2018)	A	S	With consideration of user fairness and maximum tolerable delay, they propose an algorithm to optimize offloading decision and the allocation of computation resource, transmit power, and radio bandwidth in a mixed fog/cloud system	EE, RD, Ue
(Guo & Liu, 2018)	A	S	Introduce hybrid fiber-wireless (FiWi) network to include the MEC servers and centralized cloud server in their framework while satisfy the delay requirements of task offloading	EE, DR, Ue
(Kim, Kwak, & Chong, 2018)	A	S-R	Propose a model for offloading and study two scenarios: Cooperation and competition between mobile devices and MEC server	EE, DR, Ue

Table 2.1: **Energy efficiency by offloading to MEC continue**

Paper	Approach	Evaluation	Major Contribution	Metrics
(L. Liu, Chang, Guo, Mao, & Ristaniemi, 2018)	A	S	Use queuing theory to study offloading processes in a fog computing system.	EE, DR, PC
(Chen, Shi, Yang, & Xu, 2018)	A-Sc	S	Study task offloading to MEC in ultra-dense network with utilization of Software Defined Networking (SDN)	EE, DR
(Lyu, Tian, Jiang, et al., 2018)[1]	A-Sc	S	Independent request and admission framework between devices and MEC	EE, RD
(Yang, Zhang, Li, Guo, & Ji, 2018)	A-Sc	S	Energy optimization problem with consideration of fronthaul and backhaul energy cost	EE, DR

---

[1] The author selected this paper as the reference study. This paper was published in IEEE Network journal with impact factor of 7.197 (2017)

2.2.3 Device-to-Device Communication (D2D)

Device-to-device communication is one of the technologies in 5G mobile network that significantly improves the latency and energy efficiency of a cellular network (Boccardi, Heath, Lozano, Marzetta, & Popovski, 2014). D2D communication enables two cellular devices in the vicinity to communicate with each other without the need to transmit through the cellular base station or eNodeB (Feng et al., 2014). Node B or base station is the communication node in the legacy cellular network that connects users to the rest of the network through cellular communication. In the LTE networks, the functionalities of Node B enhanced, and it is called enhanced Node B or eNodeB in short (Larmo et al., 2009). The current 3rd Generation Partnership Project (3GPP) specification for D2D communication is ProSe (Proximity Services), which defined a 1-to-1 and 1-to-many direct communications of UE in Rel-12 (Specification, 2015). In D2D-enabled architecture, base station serves the UEs like before, but here, some devices can communicate directly with each other which lead to Spectral Efficiency (SE) and Energy Efficiency (EE). Moreover, the shorter distance communication causes less delay, as well as, communicating without the intervention of the base station will decrease the load of the network. Table 2.2 summarize the reviewed studies.

There are two general approaches to study the energy consumption reduction of D2D communication. Some researchers put the attention on the fact that direct communication of the devices can decrease the load on the base station, so the consumed energy by the base station will be decreased. Meanwhile, others argue that devices are closer to each other so the transmission power and the time of the communication will be decreased, and as a result, devices can conserve their energy. In (Xu et al., 2016), authors put the attention on the fact that this technique can cut down the traffic from the eNodeB so it will help to decrease the energy consumption of the eNodeB as well. Though, one of the critical

Table 2.2: D2D Energy efficiency techniques

Approach	Study	Problem Addressed	Metrics
Matching theory	(Xu et al., 2016)	Energy consumption of base station	Relay cost
Distributed energy efficient resource allocation algorithm	(Zhou et al., 2016)	Energy Efficiency of UE	SE, EE
Heuristic algorithm for sub-carrier assignment, convex optimization problem for power allocation	(Kai, Li, Xu, Li, & Jiang, 2018)	Energy consumption of D2D and cellular UEs	EE, Data Rate
New entities in eNodeB coordinated device discovery and D2D link setup	(Ahmad, Agiwal, Saxena, & Roy, 2018)	Prolong battery life for disaster situations	Power Allocation
Efficient device discovery process	(Prasad, Kunz, Velev, Samdanis, & Song, 2014)	Energy efficiency of UE	Proximity Area

challenges of D2D communication is that the users need some incentives to relay others' data. They define the users who need the data as the buyers and potential users who can provide the data as sellers, while the eNodeB can play a role of the broker for them. They formulate and solve the problem of minimizing the energy usage of the network by having separate contract-based pricing and matching theories. Finding a solution by looking into integrating these two mechanisms to provide more energy efficiency in the network is worthy.

Another group of studies focuses on the energy consumption of the mobile devices. In (Zhou, Dong, Ota, Wu, & Sato, 2014), authors proved that with the help of a distributed algorithm and small amount of spectral efficiency loss while putting the constraint on the maximum power, a D2D connection can gain the significant amount of energy efficiency. They defined cooperation problem as a no-cooperative game in which every device concerns its energy only. In their work, the D2D connection uses the same allocated channels for

UEs. Authors in (Kai et al., 2018) consider the high data rate transmission requirements of smart cities to describe the increasing energy consumption and gas emission problems. They propose D2D communication as a solution for the high data rate applications like real-time monitoring through green communication. In their approach, they divide green communication into two sub-problems: (1) energy efficient sub-carrier assignment, and (2) power allocation to D2D communication to both provide required data rate and energy efficiency for both cellular communication and D2D communication. With the help of a successive convex approximation approach, they solved the power allocation problem and showed the energy efficiency of their approach by an extensive numerical study and simulation results.

2.2.4 Challenges of separate D2D and MEC techniques studies

Ericsson (Vestberg, 2010) and Cisco (Evans, 2011), predicted around 50 billion connected devices on the internet by 2020. This brings challenges on transmission and processing of enormous volume of generated data. As presented in the previous sections, there are two ways to address these challenges. For decreasing the demand for the transmission of all generated data, the D2D solution was introduced to eliminate the need to send all of the generated data directly to the Base Station. Meanwhile, other studies suggested MEC offloading as a technique to provide low latency and energy efficient task processing. The author identified that each of these techniques has limitations in term of the following:

- Device Discovery in D2D
- Malicious behaviour of nodes in D2D
- Overloading of MEC servers

In the following section, the author further explained these limitations.

2.2.4.1 Device Discovery in D2D

In D2D solution, it is a challenge to find the appropriate neighbouring devices that can offload the data. Device discovery (Feng et al., 2014) is an essential part of a D2D connection. A survey conducted by Tsolkas, Passas, and Merakos (2016) looked into different device discovery protocols and concluded that when a large number of devices are deployed in LTE, available resources become insufficient to perform efficient device discovery. As a result, complex protocols are needed to perform the device discovery process efficiently. Increasing protocol complexity causes more energy consumption. Therefore, there is a need for developing a distributed protocol (Qing, Zhu, & Wang, 2006)(Singh, Chiu, Tsai, & Yang, 2017)(Z. Liu, Zheng, Xue, & Guan, 2012) , or we can let the MEC to be involved in the process as an alternative solution.

2.2.4.2 Malicious Nodes

Misbehaving nodes can reduce the performance of the network in a D2D communication significantly (Marti, Giuli, Lai, & Baker, 2000). There are several proposed mechanisms based on the selection of cluster heads to monitor the behaviour of nodes to prevent any nodes from benefiting from network services without cooperating (Robert, Otrok, & Chriqi, 2012) or possibly attack the network (Denko, 2012). The problem of these approaches is the excessive energy consumption of cluster heads due to their monitoring and decision-making tasks. Therefore, having a powerful node with unlimited electricity source is required to effectively deploy complicated incentive algorithms for a large number of devices in D2D mode.

2.2.4.3 Overloaded MEC

In MEC, when devices are deployed densely in a network, it may cause overloading at eNodeB (Satria et al., 2017), as previously shown in Figure 1.1. It is highly possible that the number of service demands exceed the resources of the edge server. One way to reduce the overloading is to delegate other devices in the proximity to help in the task offloading.

2.3 Integrating D2D and MEC for energy efficiency

D2D communication and edge computing are two promising technologies of 5G that can contribute to a solution for energy and delay reduction in devices. Considering the capabilities of edge computing frameworks and the fact that many heterogeneous devices with different computation and transmission capability are in the proximity of each device, academic scientists have considered both of these technologies as an integrated solution of improving energy efficiency.

2.3.1 Benefits of integration

A D2D enabled MEC architecture aggregates the benefits of both technologies. Moreover, several new energy efficient applications can be defined as discussed in the Section 2.3.2. In the previous section, the author described the limitations as device discovery, handling malicious nodes and overloading the edge server. In an integrated architecture, these challenges can be mitigated. On the other hand, both service providers and users can benefit from this new integrated approach.

2.3.1.1 Benefits for the service provider

Increasing network throughput, improved spectral efficiency (Bhushan et al., 2014), and extended network coverage (L. Wang, Tang, Wu, & Stüber, 2017) are some of the benefits of D2D communication. By mitigating the heavy load on the eNodeB (Peng, Yan, Zhang, & Wang, 2016), the energy of the radio access network devices conserved and thus reducing the service provider cost. Therefore, service providers can provide better applications and services to attract new users.

2.3.1.2 Benefits for users

There is no need for the users to upgrade their mobile devices to use new services constantly. The heterogeneous nature of 5G, enables users to offload their computation hungry tasks to a more powerful neighbour or edge server. Authors in (Pu, Chen, Xu, & Fu, 2016) use the term D2D collaboration for offloading the resource hungry tasks to a neighbour. This offloading provides access to a large number of available resources with low delay and less energy consumption. Authors in (Chen et al., 2017)(Hong, Wang, Cai, & Leung, 2017)(Sabella, Vaillant, Kuure, Rauschenbach, & Giust, 2016) called this process D2D assisted cloud offloading. Moreover, users can benefit from mobile data offloading in which devices with poor connectivity can use stronger connected devices in their proximity as a relay to send their data (C. Liu & Natarajan, 2017). However, in this study, the author did not include data relaying. Instead, the author put the attention on the task execution benefits when it performs by neighbour devices or MEC.

2.3.2 Applications of D2D enabled MEC

By using D2D enabled MEC architecture, many new applications can be invented, and existing applications will be improved. Smart vehicles are one of the cases that can significantly benefit from these technologies. Vehicles generate a massive amount of data from combinations of different sensors and cameras and most of them require real-time processing. Offloading such data into the cloud is not a feasible solution because of the delay imposed by the core network. Communication with roadside units (V2I) and other vehicles (Vehicle to Vehicle (V2V)) are features that enable smart vehicles to operate (OpenFog, 2017). V2I and V2V communication are the best example of Edge computing and D2D communication as fog nodes. Unmanned vehicles, smart traffic control systems and connected vehicles using D2D and MEC are discussed in detail in (Mao et al., 2017) and (Baccarelli, Naranjo, Scarpiniti, Shojafar, & Abawajy, 2017).

Public safety (Doumi et al., 2013), proximity-based applications like content sharing advertisement (Huynh, Chen, Huynh, & Hai, 2017) and processing of video streams in surveillance systems (Khan, Parkinson, & Qin, 2017) are some of the D2D and fog architecture implementation in smart cities. Without fog computing and D2D communication, the smart city is not realisable (Baccarelli et al., 2017). Augmented reality, face recognition, and natural language processing are some other new applications that can be benefited from D2D communication and MEC (Chen et al., 2017). As shown in Figure 2.2, combining these two techniques, allows many D2D applications to exist in the network while some users demand services from MEC servers. As a result, the MEC servers will not be heavily occupied because adjacent devices can perform some services.

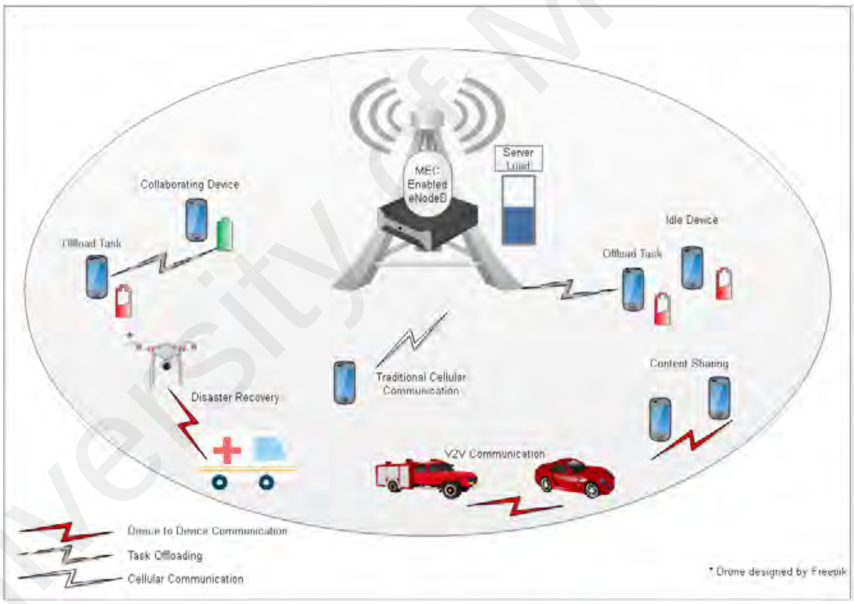


Figure 2.2: Integrated architecture of D2D enabled MEC

2.4 Research trends on D2D enabled MEC

After reviewing numerous publications on D2D enabled MEC, based on the classification criteria, the author classified research trends into three categories: (1) problem-solving studies, (2) D2D and MEC integration architectures, and (3) energy efficient framework. The following sub-sections presented the findings and discussion of research trends according to the previously mentioned classification. The author summarised the findings in Table 2.3.

Table 2.3: Studies on both D2D and MEC

Category	Study	Summary	Limitation
Problem solving	(Satria et al., 2017)	Alleviate the burden on the overloaded MEC by D2D relaying to a free MEC	Heuristic algorithms must be integrated for disconnected relay nodes to allocate relay nodes with lower latency
	(Hsing, Lau, & Chiang, 2017)	Cooperation stimulation by social relationship algorithm deployed in MEC	Unrealistic assumptions
D2D and MEC integration	(Peng et al., 2016)	Define D2D enabled fog architecture	Proposed architecture is not compatible with conventional wireless architectures
	(J. Li, Li, Gao, Gao, & Zhang, 2017)	Overcome energy wastage during Mobile Ad Hoc Network (MANET) link establishment by D2D communication with cloudlet	SNR and other major channel factors not considered for D2D energy consumption

Table 2.3: Studies on both D2D and MEC Continue

Category	Study	Summary	Limitation
D2D and MEC integration continue	(Cheng et al., 2018)	Define Small World Super Dense D2D wireless network with fog computing	Based on the architecture, simulation or real-world scenario not provided
	(Lorenzo, Garcia-Rois, Li, Gonzalez-Castano, & Fang, 2018)	Propose robust dynamic network architecture of D2D enabled IoT network with fog computing	Energy Model not provided
	(Vallati, Virdis, Mingozzi, & Stea, 2016)	Investigate effect of D2D and edge computing in smart home architecture	Energy model not provided, and results are only based on simulation
	(Alippi, Fantacci, Marabissi, & Roveri, 2016)	Define D2D communication in fog computing environment for IoT and implementation in real-world scenario	H/W layer definition omitted

Table 2.3: Studies on both D2D and MEC Continue

Category	Study	Summary	Limitation
	(Singh, Malik, & Kumar, 2017)	Implement D2D relay network in edge computing and compare with cloud image loading	Energy efficiency of the structure not outlined specifically
Energy efficient integrated framework	(Chen et al., 2017)[2]	Propose energy efficient task offloading framework by D2D collaboration and MEC assistance	Overhead for local execution decision making in Edge server, task delay overlooked

2.4.1 Problem-solving studies

One of the main problems of MEC architecture is when a massive number of devices request service from eNodeB and cause it to overload. Based on the availability of neighbouring MEC in the proximity of the Overloaded Mobile Edge Computing (O-MEC), in (Satria et al., 2017) authors identify two cases and proposed solutions to tackle each problem. The first case happens when there is a non-overloaded MEC, and the other happens when there is no any free MEC available within proximity.

In the first case, the O-MEC merely offload its work to the neighbouring MEC, but in the latter case, it uses serving users as a relay to reach to the non-overloaded MEC. For both solutions, based on the geographical position, mobile users are grouped into clusters. To perform the recovery process, one MEC becomes the cluster head. For the first recovery process, if a MEC identifies an overloading condition, it sends a request to the cluster

---

[2] The author selected this paper as the second reference study. This paper was published in IEEE Wireless journal with impact factor of 9.202 (2017). The color of this study in table is different for clarification on selected studies.

head for help. Cluster head searches to find a free MEC in the communication range of O-MEC and notify the O-MEC by the address of the free MEC. Then offload its work to prevent corruption of the service. If there is no free MEC in the range, the second recovery scheme will be employed. In this scheme, the connected devices to the adjacent MEC will perform as D2D relay nodes. The users of O-MEC are grouped based on their location in proximity of non-overloaded MEC's users. The cluster head will notify the O-MEC users with a possible service MEC address, and then they use Floyd-Marshall algorithm to find a pass among relay nodes to discover the new MEC. Their proposed methods can be further investigated by employing novel methods of route discovery for ad hoc relaying part to increase the energy efficiency process.

One of the primary challenges of D2D enabled fog architecture, is how to stimulate the cooperation between devices. This is a traditional problem of ad hoc networking (Buttyan & Hubaux, 2001), and since the nature of the problem is almost similar, we can adapt existing solutions to use high computation power, low latency and battery-less characteristics of Fog nodes such as eNodeB. Another approach, as discussed in (Hsing et al., 2017), is to apply the principle of social relations between humans. They proposed two ways of social paradigm for cooperation stimulation. The first is social trust that obtains from the number of friendship hops between users in social applications, and they suggest a Facebook or Twitter plugin. Second is social reciprocity, which means that human tends to participate in a give-and-take relationship. In this case, a device would relay data of proximate node if other node relays its data. Specific to solving cooperation between devices problem, the first step is choosing between two approaches. Then, if the second one is selected, we need to know how we can group the devices to efficiently relaying the tasks for cooperative D2D communication in fog networking. They define a coalitional game theory framework to tackle these issues. They first showed that coalitional game admits the top-coalition property. Based on the coalitional game theory it means that for any non-empty set of devices, there exists a top-coalition for any non-empty subset of devices (Banerjee, Konishi, & Sönmez, 2001). Based on that, they design a core relay selection algorithm for computing the core solution to the game and finally develop a network assisted mechanism to solve the problem. The evaluation of the performance based on the ER[3] social graphs and real data trace-based social graphs validate the performance gain of 122% over the non D2D cooperation.

2.4.2 D2D and MEC integration architecture

In the previous section, the author has discussed how the integration of D2D and MEC is beneficial in solving energy consumption problems in mobile networks. From the literature, the author classified proposed integration architectures into *massive D2D systems* and *small world IoT*.

2.4.2.1 Massive D2D systems

In this group of studies, authors propose D2D enabled MEC environment that has a massive number of devices try to communicate with the network and use its services.

The first ever published work in the context of Fog/Edge computing architecture in 5G radio access network has been done by (Peng et al., 2016). They defined the architecture of the system and principles behind the design of D2D and MEC. They discussed that D2D connection is most appropriate for user-centric services and Fog/MEC can bring a significant amount of storage, management, configuration, computation and measurement capabilities next to the user. In their structure, Fog Access Points (Fog-APs) are used to process signals and managing resources locally for fog node. Fog-AP provides interference prevention and spectral sharing for D2D connection. On the other side, they compress and

---

[3] Erdos-Renyi

forward the UEs traffic intended for the core network to the Base Band Unit (BBU) pool. Putting two functions of signal processing and resource management for collaborative communications on Fog-APs cause decreasing the load on the fronthaul.

In their architecture, Fog-UEs defined as D2D enabled UEs that have access to Fog-APs. In a small area, Fog-APs and high power nodes (HPNs) can become active, and D2D relay fog nodes can leverage the need for increasing the capacity when there is a considerable load. If the distance of two D2D enabled device is further than their supported range, a third Fog-UE can become a relay in between. Fog-APs and HPNs can sleep when the load is below the threshold.

MANET is a distributed system of mobile devices that can move without the limitation and they organise dynamically, temporary and arbitrary (Chlamtac, Conti, & Liu, 2003)(Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, 2002). The characteristics of MANET, which is high mobility and low capabilities of mobile devices cause the frequent link breakage between devices. Establishing a new link may need high power consumption and service delay that can lead to low Quality of Service (QoS) for users. To overcome the problem of energy wastage during re-establishing the link to the network for MANET, fog computing architecture of 5G integrated into the solution by using the cloudlets (J. Li et al., 2017). Their mechanism can meet the green network in two aspects. First, by establishing a D2D connection between a mobile device and a cloudlet, mobile devices can benefit from more efficient communication. Second, by using the computation capabilities of a cloudlet, mobile devices can offload computation hungry tasks to their nearest cloudlet to conserve their energy. In their proposed solution, every mobile device will hold a temporary file for holding the identity and online route information. To enhance the ability of information sharing between devices, they use D2D as a critical promising technology in 5G. For storing the topology and cooperation mechanisms, they use cloudlets as a small-scale data centre next to the mobile devices. Although mobile devices are frequently moving as a primary characteristic of MANET, they can request service and route information quickly from the network. By emerging D2D communication capability in 5G and advantages of an ultra-dense network, devices can communicate with cloudlets more efficiently and use their computation power to conserve their energy. As a result, mobile devices in a MANET can have a longer life and can move more freely.

Authors in (Cheng et al., 2018) argue that massive D2D systems are different from MANET. They describe a Small-World Super-Dense Device-to-Device Wireless Network (SSDNet) in 5G and define its challenges and benefits for deployment and compare it with MANET technology. The authors pointed out that SSDNet is different from MANET because in the current ad hoc networks there are at most 30 nodes for research and about ten nodes for the real implementations. Super dense characteristics of the SSDNet requires more advanced technologies to be deployed in 5G such as Cognitive Radio (CR) (Haykin, 2005) and Millimetre Waves (mmWaves) (Roh et al., 2014) to better use of the spectrum and increase the capacity of the network. They have shown that super dense feature of their proposed network causes a higher success rate in D2D communication, and small world feature enables deterministic optimisation, and both of them lead to energy efficiency. However, there are some critical challenges that must be addressed before the real implementation of such a network.

(a) ***Physical and link layer challenges***

The smallest unit of resource that can be allocated to a user is a resource block. With the maximum bandwidth of 20MHz, only 100 resource blocks are available for allocation (Lai & Tang, 2013). This is a significant number for WiFi and LTE networks, but super dense networks with frequent D2D communications must be able to handle many more. Otherwise, the intra-interference will be a big challenge for SSDNet. Besides, D2D communication should avoid imposing interference with other non-D2D communications. Broadcasting and multicasting are efficient techniques for information sharing, but current solutions like LTE Evolved multimedia broadcast/multicast service (eMBMS) (Lecompte & Gabin, 2012) is controlled by the base station which is not applicable to emergency situations that only D2D connection is possible. Since peer discovery is one of the main foreseen capability, it cannot be based on Carrier Sense Multiple Access (CSMA) like WiFi or Bluetooth, because this technique is time and energy consuming, as well as suffers from a lack of reliability in many wireless network applications (Z. Zhang, Wang, & Zhang, 2017). Instead, considering time-multiplexing technique can make the congestion avoidance procedure more efficient.

(b) *Network layer challenges*

The general assumption is that the users in proximity have similar interests. In the small world of SSDNet, multi-hop routing is not an issue because most of the devices are in a single or two hops away, but node degree in a super dense environment is high. Legacy ad-hoc routing protocols are based on broadcast route discovery and thus, impose high overhead. As a result, they are not suitable for SSDNet.

In (Lorenzo et al., 2018) a Robust Dynamic Network Architecture (RDNA) was proposed. In this architecture, wireless devices use the state-of-the-art technologies to help lightweight IoT devices perform their tasks more efficiently in term of energy usage. RDNA consists of physical, access, networking, application and business level. D2D communication is defined at the network level for collaborative data sharing among devices. Devices can use short range and energy-efficient D2D communication to share the same requested data. Without proper incentive mechanisms, any cooperative architecture would fail because typically, devices have concerns about their energy consumption. In RDNA design, cooperation stimulation is considered at the business level, while fog and edge computing paradigm are considered at the network level for performing resource hungry tasks of IoT devices with a reasonable delay. They concluded that the reliability, energy efficiency, latency in their RDNA architecture and mitigating the network congestion made it a global architecture for IoT.

2.4.2.2 Small world IoT

Another group of researchers consider the D2D communication in a small world IoT with edge computing enabled environment (Vallati et al., 2016)(Alippi et al., 2016). Their goal is to investigate possible approaches that can bring MEC logic into the proximity of IoT devices. In this case, researchers expect to achieve lower latency in decision making situations and providing more communication bandwidth for IoT devices and reduce the energy consumption.

In (Vallati et al., 2016), the smart home is defined as representative of small-scale IoT system. They discuss three MEC deployment alternatives. One, placing MEC node at the edge of the network next to the eNodeB to offer high computation and storage resources by the service provider, and as other alternatives they placed MEC server into either femtocell or a UE. After that, they compare MEC-enabled smart home deployment options. They conclude that the best performance is achieved by using the D2D communication in a house that employed mobile edge computing. Similarly, in (Alippi et al., 2016), a fog-IoT paradigm is discussed. They pointed out that transmitting data to the cloud is an inefficient solution and instead, by transmitting data directly to the powerful fog nodes, or through nearby fog nodes, users can conserve their energy and reduce the latency. In their proposed architecture, there are four approaches for D2D communication:

- No-contention mode: Resource blocks are assigned to each D2D link individually by the small cell base station.

- Random mode: Each device competes for predefined resource blocks for D2D links.
- Underlay mode: To avoid interference, each D2D link uses resources same as communications that are very far.
- Overlay Mode: Devices use unused resources for D2D communication.

Although the two latter approaches are complex in term of computation, they provide the most spectral efficiency. In their architecture, smart objects exchange data locally, and unnecessary data will not be sent to the cloud. Hence, the result is increasing the energy efficiency and prevent bandwidth wastage. They implemented their structure in a real-world scenario of rock collapsing and land sliding prediction and proved that their infrastructure is effective and energy efficient.

In (Singh, Malik, & Kumar, 2017) the authors proposed a MEC-based framework and implement it in a lab using D2D communication. They put several wireless access points as relay gateways in their scenario between the core network and user devices. In their design, every relay gateway is a member of the mobile edge computing environment and serves the users locally. Relay gateways communicate through a D2D connection. There are two modules in each relay gateways:

- Request handler: Manages the service requests from users.
- Service handler: Provides a balance between relay gateways. It will forward service to the neighbour relay gateway with D2D connection or broker it to the server.

They test their framework in an image loading scenario and compared it with loading the image from Google Drive as a cloud service. They concluded that their solution outperforms in term of delay, throughput and user experience. Their framework can be used in tourist information sites, sporting events and advertisements.

2.4.3 Energy efficient integrated framework

In the third category of research trends, researchers are looking into energy efficient integrated framework. The HyFog framework presented by (Chen & Zhang, 2017), and it is the integration of D2D connection and fog computing as a hybrid task offloading framework for energy efficiency. In HyFog, each device can choose between three types of task execution, 1) local execution, 2) D2D offloading the task to a more powerful proximate device, and 3) offload the task to a resource-rich edge computing server. They first build a three-layer graph and then transform it into a minimum weight matching problem and finally impose a revised version of the Edmond's Blossom algorithm (Edmonds, 1965) to find which option is optimal for task execution. The author presented the full details of this algorithm in Appendix A. They enhance their work and introduce the D2D crowd system (Chen et al., 2017). They have considered task offloading by the collaboration of heterogeneous devices in term of connection quality and processing power. The base station helps to identify the most efficient task offloading assignment, again, by solving a graph-matching based algorithm. They achieve 50 per cent of energy consumption reduction in compare of local task execution in the simulation. Decreasing the delay is the primary motivation for edge offloading, but both of the mentioned studies neglected it in their proposed model. Moreover, the edge server decides about the efficiency of local execution in their model. This approach imposes unnecessary overhead, and in fact, it can be done more efficiently by the users.

2.5 Related Work

As discussed before, authors in (Chen et al., 2017) proposed D2D collaboration scheme for task offloading to achieve energy efficiency. On the other hand, in (Lyu, Tian, Jiang, et al., 2018), integration of cloud, MEC and IoT was proposed to provide energy efficient task offloading framework under the latency requirement. Based on these studies, the author proposed an integrated framework that leverage offloading capabilities of both studies, while overcoming their limitations that the study discuss in the next section. The main reason of choosing these two studies in the current study was that, both of them addressed task offloading problem in the same way. In (Chen et al., 2017), energy efficiency problem was formulated for D2D collaboration and in (Lyu, Tian, Jiang, et al., 2018), delay optimisation of task executing by MEC was formulated. By combining their optimisation problems and impose some improvements in the problem formulation, the author was able to propose an energy-efficient and delay-aware offloading scheme that outperformed better in comparison to both of them. To validate the proposed solution, the author presented the results of the comparison with these two studies in Chapter 4.

2.6 Research Gap

As mentioned in the Section 1.2, in Chen and Zhang (2017) and Chen et al. (2017), only the energy consumption of the device was taken into account. Therefore, the effect of their scheme on execution time was not studied. In addition, the computation capability of MEC was overlooked in Chen et al. (2017), while it could be employed for task offloading, to save the energy of mobile devices and to provide low latency. As another limitation in Chen et al. (2017), the authors did not differentiate between uploading and downloading data for computing the energy consumption. But, transmitting data consumes much more energy than receiving it. To clarify the mentioned research gaps, the author discussed them in Table 2.4.

In Lyu, Tian, Jiang, et al. (2018), capabilities of MEC servers was taken into account, to design an energy efficient offloading framework. However, there are a vast amount of opportunities for collaboration with proximate devices in the network, that was overlooked in their framework. Table 2.5, summarise the research gaps in their work and current study's approach to mitigate them.

Table 2.4: Research gaps in Chen et al.,(2017)

Research Gap	Importance of the problem	Solution
Authors did not differentiate between uploading and downloading in their model	Transmitting data consumes much more energy than the energy consumes for receiving	Include separate parameters in task model
Devices send their computation capability info to MEC to see if they can perform the task locally	Unnecessary Overhead	In the designed scheme, devices calculate their own capability
MEC simulation was not provided	Evaluation of the solution was incomplete	Evaluate their solutions by simulation
Delay parameter was not considered	Delay is a key parameter for real-time applications	Propose a joint energy and delay task model

Table 2.5: Research gaps in Lyu et al.,(2018)

Research Gap	Importance of the problem	Solution
Device to Device Collaboration was not considered	Edge servers becomes overloaded in dense networks (Scalability)	Proposed an integrated solution and include D2D collaboration in the scheme
Simulation results was not provided in the realistic LTE network	Simulation is needed to validate the solution	Evaluate their solutions by simulation

2.7 Conclusion

In the previous sections, the author presented findings on the D2D collaboration and MEC offloading schemes. In D2D studies, devices offload their computational hungry tasks to their idle neighbours for computation. However, in 5G, we can leverage the computational capability of edge servers to save the energy of mobile/IoT devices. Also, device discovery consumes excessive energy on mobile devices. On the other hand, in MEC studies, devices can offload their tasks to the edge server. In this type of framework, massive amount of computational capability of idle devices is neglected. From the review of existing studies, the author concluded that there is lack of comprehensive study on an integrated D2D and MEC scheme. The author highlighted the research gaps in the previous section and the rest of this study aims to propose a comprehensive scheme, that mitigates the mentioned research gaps. In the next chapter, the author shall present the research methodology to propose the solution.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This chapter presents the research methodology. The author has discussed the research steps in Section 1.6. The outcome of this study is an extensive Energy-Efficient and Delay-aware Offloading Scheme (EEDOS). In the next section, the author presented the characteristics of EEDOS. In the Section 3.2.1 the author described the topology of the EEDOS network. Before presenting the proposed offloading selection algorithm, the author formulated the offloading problem in the Section 3.2.2. To validate the correctness and effectiveness of EEDOS, extensive simulation has been performed, that the details discussed in the Section 3.3.

3.2 Energy-Efficient and Delay-aware Offloading Scheme (EEDOS)

After performing the literature review, the author designed a comprehensive scheme with consideration of idle device, edge and cloud servers' capabilities. To propose the computation offloading decision algorithm, the author first formulated the entities in the proposed scheme as follows:

- User/IoT device:
    - Computing capacity
    - Uplink and downlink communication
    - D2D link communication
- MEC Server:
    - Computing capacity
- Task execution:
    - Local

  - Neighbour device offloading by D2D communication
  - Edge server offloading by cellular communication
  - Cloud server offloading by cellular communication

In the next step, the author shall propose two algorithms, to find the proper offloading destination. The first algorithm is for delay sensitive tasks, in a three-layer graph. The vertices of the graph are the time taken to offload and execute the task. Similarly, the second algorithm creates an energy graph, but the vertices are based on the energy consumed for offloading the task to a neighbour device through D2D communication or sending it to MEC plus the energy required for task execution by the neighbour or MEC.

3.2.1 System Architecture

In EEDOS, if a device cannot meet the deadline of a delay sensitive task, it can offload the task to a proximate device or edge server. If a device suffers from lack of sufficient energy, it can offload its task to three alternative options: *a neighbour device*, *edge server* or *cloud server*. The decision about either option can be made efficiently by MEC server. Figure 3.1 illustrates the network topology that integrate D2D enabled user layer with MEC and Cloud computing layers.

3.2.2 Problem Formulation

In this section, the study describes the time and energy model for task execution and offloading to the different layers of the proposed scheme. Table 3.1 represents the definition of the formulas' notations.

3.2.2.1 User/IoT device model

In EEDOS, user/IoT devices with the capability of D2D communication are in the first layer of the architecture. The author formulated the computation capacity and

Table 3.1: Nomenclature used in this section.

Description ⇒ (Notation)
Set of devices in the communication range of MEC server (N)
CPU clock of the device $i \in N$ (f_i^0)
Current computation capacity of device i (f_i^l)
Allocated resources of j to perform task i (f_j^{d2d})
Initial computation capacity of MEC server dedicated for offloaded tasks (f_{mec}^0)
Current computation capacity of MEC server (f_{mec})
Current CPU load of $i, 0 < \theta \leq 1$ (θ)
Decision about scheduling task of i in MEC server $\mu_i \in \{0, 1\}$ (μ_i)
Decision about whether task of i is delay-sensitive or not (mu_{it})
Uplink data rate of i (R_i^{ul})
Downlink data rate of i (R_i^{dl})
Transmission power of i (P_i^{ul})
Receive power of i (P_i^{dl})
Data rate of communication from device i to j (R_{ij}^{d2d})
D2D transmission power of device i (P_i^d)
Energy cost per CPU cycle for device i (ρ_i)
Input size of task i (I_i)
Output size of task i (O_i)
Required CPU cycles of task i (Z_i)
Required cellular uplink traffic for task i (B_i^{ul})
Required cellular downlink traffic for task i (B_i^{dl})
Deadline for task i (T_i^{req})
Total time of performing task i locally (T_i^l)
Total time of offloading task i to MEC server (T_i^{mec})
Total time of offloading task of i to j (T_{ij}^{d2d})
Energy required for device i to perform the task locally (E_i^l)
Energy required for device i to offload the task to the edge server (E_i^{mec})
Energy required for device i to offload the task to the cloud computing server (E_i^{cc})
Energy required for device i to offload the task to j (E_{ij}^{d2d})
Current energy level of device i (W_i^t)
Predefined energy threshold for device i to enter energy sensitive mode (W_i^{tr})

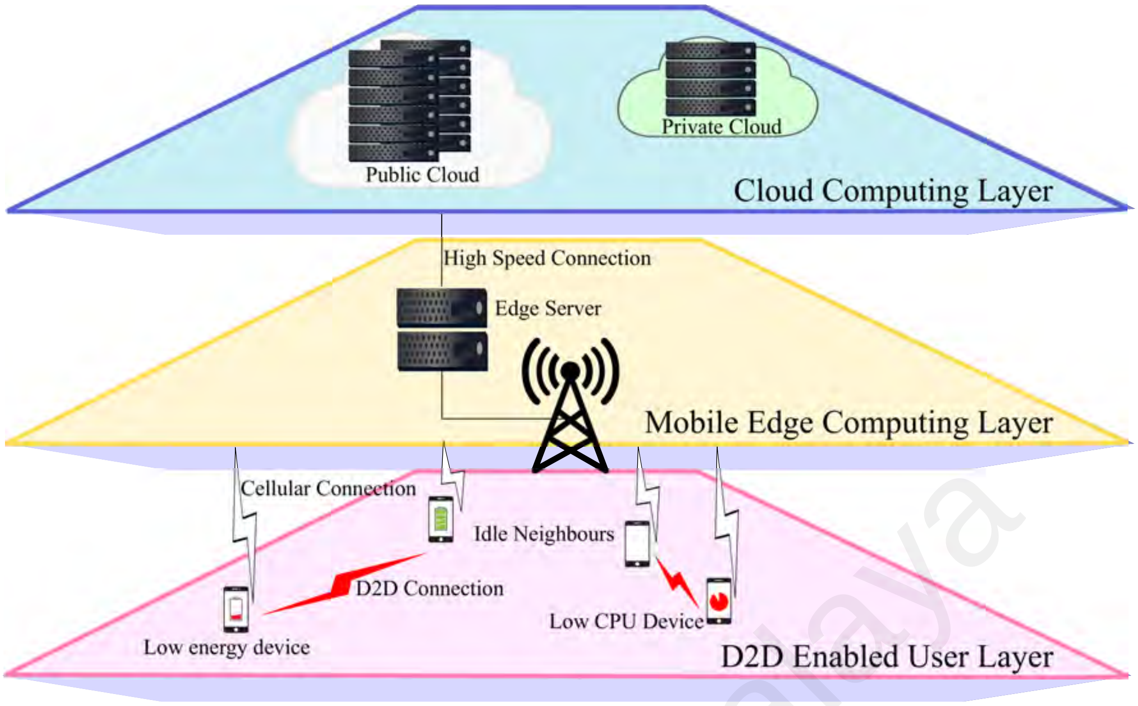


Figure 3.1: Network topology of EEDOS

communication details of the devices. In the first step of communication, devices can only communicate with the MEC layer. If MEC server decides about D2D collaboration, they can communicate directly without the intervention of base station.

(a) ***Computing capacity model***

The author indicated the background services and tasks that the device cannot offload them, occupy the CPU of the device by θ. As a result of this, the current computation capacity of the device i is as follows (Chen et al., 2017).

$$f_i^l = (1 - \theta) f_i^0 \tag{3.1}$$

(b) ***Uplink and downlink***

Devices continuously provide feedback to the Base Station. Based on the transmission power of the device and its current location coupled with the current noise of the channel, every device has its unique SINR value at a given time. Devices in the network periodically report these values to the Base Station as Channel Quality Indicator (CQI). As it is not energy-efficient that all users provide CQI feedback to the base station for all Resource Blocks (RBs), authors in (Salman, Mansoor, Jalab, Sabri, & Ahmed, 2018) proposed a partial CQI feedback to save the energy. In (Chen et al., 2017), authors defined the required time for cellular traffic as $T_i^{cellular} = B_i^{cellular}/R_i^{average}$. But the study assumed the cellular traffic for sending data (uplink - ul) and receiving data (downlink - dl) separately. Because the transmit power and receive power of mobile devices are different and can affect the energy consumption calculations significantly. Therefore, the required time for cellular traffic were obtained as follow:

$$T_i^{ul} = B_i^{ul}/R_i^{ul} \tag{3.2}$$

$$T_i^{dl} = B_i^{dl}/R_i^{dl} \tag{3.3}$$

(c) ***D2D link model***

In EEDOS architecture, devices can communicate with the MEC and proximate devices simultaneously. D2D SINR values are reported to the base station. According to the following Shannon Formula (C for maximum capacity, B for bandwidth, S signal power and N noise power), MEC can calculate the R_{ij}^{d2d} for the data rate of communication from device i to j.

$$C = R_{ij}^{d2d} = B\log_2(1+\frac{S}{N}) \tag{3.4}$$

3.2.2.2 MEC server computing capacity model

MEC layer is located near to the user and IoT devices. MEC servers can communicate with the user and IoT devices directly on one side, and cloud computing layer on the other side. MEC layer brings the capabilities of the cloud computing within the RAN. As a result of this, mobile/IoT devices that need real-time response can use the services with low latency. In EEDOS, the author has considered the traditional cloud computing layer. Powerful and resource-rich servers are located in this layer to perform complex tasks. There are different Virtual Machines (VMs) in any MEC server to support different services simultaneously. Network manager can adjust the computation capacity of each VM statically or allocate it dynamically. Based on the requirements of each task, the author should specify the minimum capacity of CPU to provide scalability. The remaining computation capacity of the VM is obtainable through Formula 3.5 where μ_i denoted whether a task is offloaded to the MEC or not.

$$f_{mec} = f_{mec}^{0} - \sum_{i=1}^{n} \mu_i f_i^{mec} \tag{3.5}$$

f_i^{mec} is the reserved resource in MEC to execute task of device i. Same as (Lyu, Tian, Jiang, et al., 2018), the study assign the minimum required computing resource in MEC for execution of the task of device i, because it must maintain its resources to serve more users.

3.2.2.3 Task execution model

Task model in defined as the following tuple in (Chen et al., 2017):

$$(I_i, O_i, Z_i, B^{cellular}) \tag{3.6}$$

Where I_i is the input size, O_i is the output size, Z_i is the required CPU cycles to complete the task, $B_i^{cellular}$ is the required cellular traffic of the task. However, in EEDOS, the study extended this model by adding the delay constraints (deadline) of the task as T_i^{req}, to jointly consider the energy efficiency and delay of the task in the scheme. As we have mentioned before, the study separate the cellular traffic to calculate the required energy and delay more accurately. So, the task execution tuple is defined as follow:

$$(I_i, O_i, Z_i, B_i^{ul}, B_i^{dl}, T_i^{req}) \tag{3.7}$$

In EEDOS, there are four ways to complete a task. If a device has sufficient energy and can meet the deadline, it can perform the task locally. Otherwise, it should offload the task to a neighbour or MEC server. If a task is delay-tolerant and MEC server is busy, there are two options to perform the task. The MEC either candidates a neighbour for offloading or forwards the task to the cloud. Following this, the study presented the model of different task execution options.

(a) ***Local Execution***

If a device runs the task by its CPU, the total time of execution is the cumulative time of processing and cellular traffic as follows:

$$T_i^l = Z_i / f_i^l + T_i^{ul} + T_i^{dl} \tag{3.8}$$

Let ρ_i be the energy cost per CPU cycle for device i. Accordingly, the energy required to perform the task locally is obtained as $Z_i \rho_i$. Also, let the P_i^{ul} be the transmission power and P_i^{dl} be the receive power of device i. Then, the energy required for the cellular traffic of the task can be obtained from $P_i^{ul} T_i^{ul} + P_i^{dl} T_i^{dl}$. So, the required energy to perform the task locally, as E_i^l, obtains from Formula 3.9. On the other hand, based on two conditions, a device cannot run the task locally. If $T_i^l > T_i^{req}$ device cannot meet the deadline so it should send the request for offloading decision to the MEC server. Similarly, if the current energy of the device is below the predefined threshold ($W_i^t \leq W_i^{tr}$) it should send the offloading request.

$$E_i^l = Z_i \rho_i + P_i^{ul} T_i^{ul} + P_i^{dl} T_i^{dl} \tag{3.9}$$

(b) ***Offload Execution***

When a device determines that it cannot meet the deadline or it does not have sufficient energy to perform the task, it sends the offloading request to the MEC server. The server can decide between three options based on the following calculations:

If device i needs to offload its task to device j it takes time to transfer the input of the task and receive the output. Also, the overhead for j to perform the task must be considered. The total time and energy required for D2D offloading is calculated as follows:

$$T_{ij}^{d2d} = I_i/R_{ij} + O_i/R_{ji} + Z_i/f_j^{d2d} + B_i^{ul}/R_j^{ul} + B_i^{dl}/R_j^{dl} \tag{3.10}$$

$$E_{ij}^{d2d} = P_i^d(I_i/R_{ij}) + P_j^d(O_i/R_{ji}) + \rho_j(Z_i/f_j^{d2d}) + P^u l_j(B_i^{ul}/R_j^{ul}) + P^d l_j(B_i^{dl}/R_j^{dl}) \tag{3.11}$$

If $T_{ij}^{d2d} \leq T_i^{req}$ device j would be a candidate for D2D offloading. The first difference between EEDOS D2D offloading and the work in (Chen et al., 2017) is considering different parameters for sending and receiving data. Also, the previous study did not consider the time taken for D2D offloading in candidating a neighbour device for offloading.

As an alternative offloading candidate, MEC server should assign its resources to delay sensitive tasks with higher priority and also saves its resources to support more users. As a result, it should adjust allocated resources to meet the delay required by the task as in Formula 3.12. Therefore, the required time for MEC offloading is calculated as Formula 3.13 and the energy required for devices to offload their task is calculated as 3.14.

$$f_i^{mec} = Z_i/(T_i^{req} - T_i^{ul} - T_i^{dl}) \tag{3.12}$$

$$T_i^{req} \leq T_i^{mec} = Z_i/f_i^{mec} + T_i^{ul} + T_i^{dl} \tag{3.13}$$

$$E_i^{mec} = P_i^{ul}T_i^{ul} + P_i^{dl}T_i^{dl} \tag{3.14}$$

The difference between MEC offloading model in EEDOS and (Lyu, Tian, Jiang, et al., 2018) is that they did not expand and explain the cellular traffic delay. If edge server cannot perform the task and offloading the task to a proximate device is not energy efficient, it should send the task to a cloud. Because cloud server cannot meet the deadline of the delay sensitive tasks, this can be done only for delay tolerant tasks. Accordingly, the study just considered the energy required by the device i to offload its task to the MEC server. In EEDOS, devices can only communicate with servers in the MEC layer, so the cost of cloud offloading for mobile devices could be same as MEC offloading as $E_i^{mec} = E_i^{cc}$.

The energy efficiency in EEDOS achieves by offloading the task to the most energy efficient destination. Based on the results of the Equations 3.11 and 3.14, either an efficient neighbour or the MEC would be selected for task execution. It must be noted that running the task by the most efficient offloading destination, is the primary technique that has been leveraged by this study to achieve energy efficiency. In this study, the energy consumption of the MEC has not been taken into account, because, the goal is to decrease the energy consumption of mobile devices and edge servers are connected to the power grid.

3.2.3 Optimisation problem

By combining the optimisation problem in (Chen et al., 2017) with MEC offloading scheme in (Lyu, Tian, Ni, et al., 2018), the author can define EEDOS optimisation problem as 3.15. The current study's objective is to minimise the energy consumption of energy constrained devices for task execution. Meanwhile, for delay sensitive tasks, the study needs to minimize the execution time considering the task deadline.

$$min \sum_{i=1}^{n} (\mu_{it}(\mu_i T_i^{mec} + \sum_{i \neq j, \mu_i = 0} T_{ij}^{D2D}) + (1 - \mu_{it})(\mu_i E_i^{mec} + \sum_{i \neq j, \mu_i = 0} E_{ij}^{d2d})) \tag{3.15}$$

In 3.15, first part is related to optimisation of delay for all tasks while reserve the MEC resources for scalability and μ_{it} is a binary decision, that denotes whether the task of device i is delay sensitive or not. The second part is for minimising the required energy for completing all tasks while meeting deadlines.

3.2.4 Offloading selection algorithms

The optimisation problem is complex to be solved by one algorithm. Therefore, the study divides it to two sub-problems. The first sub-problem is for delay sensitive tasks and the study needs to assign the higher priority to it to not miss the deadline. The second sub-problem is for energy constraint devices that suffer from the lack of energy.

The decision about sending to proper offloading destination is a two-step process. Figure 3.2 demonstrates the workflow. Instead of transmission overhead and delay were imposed in (Chen et al., 2017) to decide on local execution by MEC, the proposed scheme let the device make this decision efficiently by itself. In the first step, each device considers its task parameters and its resources, and based on the following conditions it would decide to perform the task locally or it would send the offloading request.

$$W_i^t - E_i^l > W_i^{tr} and T_i^l \leq T_i^{req} \tag{3.16}$$

If the result of 3.16 is true, it means that device has sufficient energy to perform the task and it can meet the delay requirement. In the other situation, the device would send an offloading request to the MEC server.

In the second step, the server uses the proposed method to find the best option for offloading. Based on the possible D2D communication and the remaining capacity of MEC, it builds two weighted graphs. There are three types of nodes (vertices) in each graph: 1) devices that has a task to offload, 2) idle devices, and 3) allocated resources in VM to a particular device, as illustrated in Figure 3.3. The difference between two graphs is how the author calculates the weight of each edge.

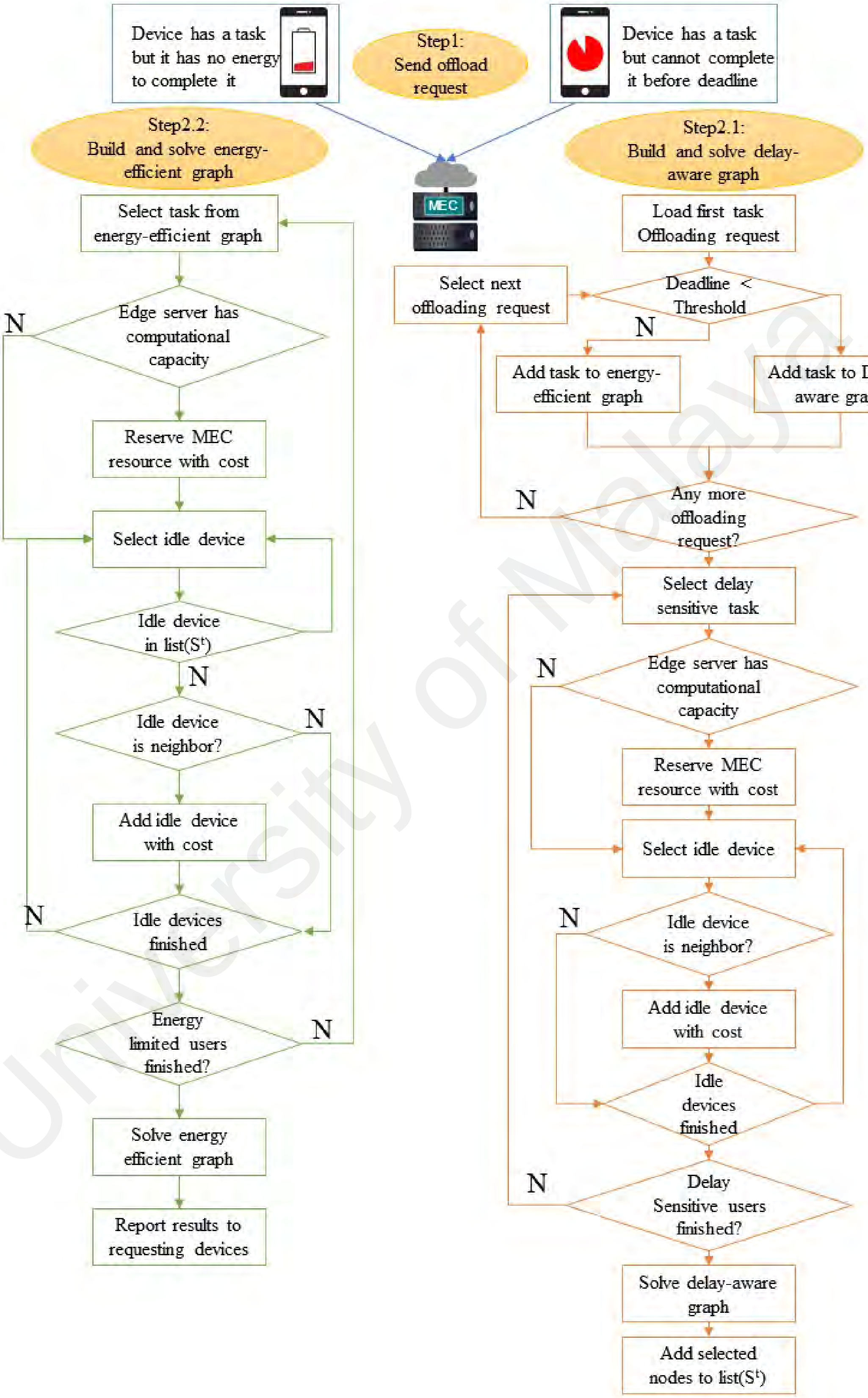


Figure 3.2: Workflow of EEDOS scheme

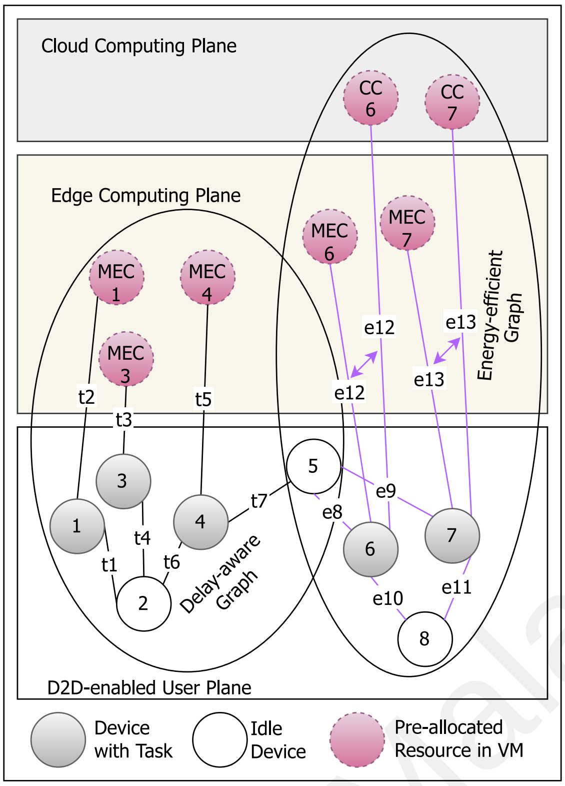


Figure 3.3: Decision graphs that built in EEDOS workflow

The first graph builds by EEDOS offloading decision algorithm for delay sensitive tasks (Delay-aware graph) as shown in Algorithm 1. The value of ϵ as the task deadline threshold is defined in the MEC to decides whether a task is delay sensitive or not. The only reason that the EEDOS chooses to process the delay sensitive tasks first is because MEC needs to assign its resources effectively to meet the deadline of tasks. Therefore, the edges of the graph are the time required for offloading the task plus the time taken for task execution at the destination (The destination can be a proximate device or MEC). After constructing the graph, EEDOS solves it by Edmond’s Blossom algorithm (Edmonds, 1965) to find the maximum matching with minimum cost answer. The description of the algorithm is described in Appendix A. Because of the complexity of this algorithm, this study adopted the current most efficient implementation that was developed by (Kolmogorov, 2008). According to the result, MEC either reserves its resources for the task execution or selects D2D collaboration. For the second graph, EEDOS considera the devices that do not have the sufficient energy to run their task (Energy-efficient graph). Based on the result of the previous algorithm, some of the idle nodes in the proximity of requested users would be omitted because they are already selected to perform delay-sensitive tasks. The energy-efficient graph was created and solved by the use of EEDOS offloading decision algorithm for energy sensitive devices as shown in Algorithm 2. In the example, device 5 has not been selected for offloading in the first step, as a result, MEC includes it in the energy-efficient graph again. If MEC becomes out-of-resource, it should forward the task to the cloud. Therefore, it pre-allocates resources in the cloud to prevent overloading. Similarly, MEC finds the maximum matching with minimum cost and inform all of the requested devices to offload their task to the selected neighbour or the MEC.

```
foreach $i \in \overline{N}$ do
    if $T_i^{req} \leq \epsilon$ then
        add $i$ to $G^t$
    end
    else
        add $i$ to $G^e$
    end
end
foreach $i \in G^t$ do
    if $f_{mec} \geq \frac{Z_i}{T_i^{req} + T_i^{mec}}$ then
        add $vm_i$ to $G^t$;
        $f_{mec} \leftarrow f_{mec} - \frac{Z_i}{T_i^{req} + T_i^{mec}}$;
        $V(i, vm_i) = Equation 3.13$;
    end
    foreach $j \in N \ \& \ j \notin G^t$ do
        if $j = neighbour(i)$ then
            temp $\leftarrow$ Equation 3.10;
            if $temp \leq T_i^{req}$ then
                add $j$ to $G^t$;
                $V(i, j) = temp$;
            end
        end
    end
end
Solve $G^t$ by BlossomV (Kolmogorov, 2008);
$S^t \leftarrow$ Selected nodes (vertices);
```

Algorithm 1: EEDOS offloading decision algorithm for delay sensitive tasks

(a) *Complexity of the algorithm*

The proposed algorithms include two steps. The first step is for graph creation that runs $O(n + nm)$, which n is the number of requesting devices and m is the number of idle devices and MEC reserved resources. The second step is solving the graph by Edmond's Blossom algorithm. Since we leveraged the currently most efficient implementation by (Kolmogorov, 2008), this section runs in $O(n * mLogn)$, where n is the number of nodes in the graph and m is the number of vertices. Hence, the complexity of the proposed EEDOS algorithms are $O(n * mLogn)$.

```
foreach $i \in G^e$ do
    if $E_i^{mec} < W_i^t$ then
        if $f_{mec} > f_{min}$ then
            add $vm_i$ to $G^e$;
            $f_{mec} \leftarrow f_{mec} - f_{min}$;
            $V(i, vm_i) = Equation 3.14$
        end
        else
            add $cloud_i$ to $G^e$;
            $V(i, cloud_i) = Equation 3.14$;
        end
    end
    foreach $j \in N \;\&\; j \notin G^t \;\&\; j \notin S^t$ do
        if $j = neighbour(i)$ then
            add $j$ to $G^e$;
            $V(i, j) = Equation 3.11$;
        end
    end
end
Solve $G^e$ by BlossomV (Kolmogorov, 2008);
$S^e \leftarrow$ Selected nodes (vertices);
```

Algorithm 2: EEDOS offloading decision algorithm for energy sensitive devices

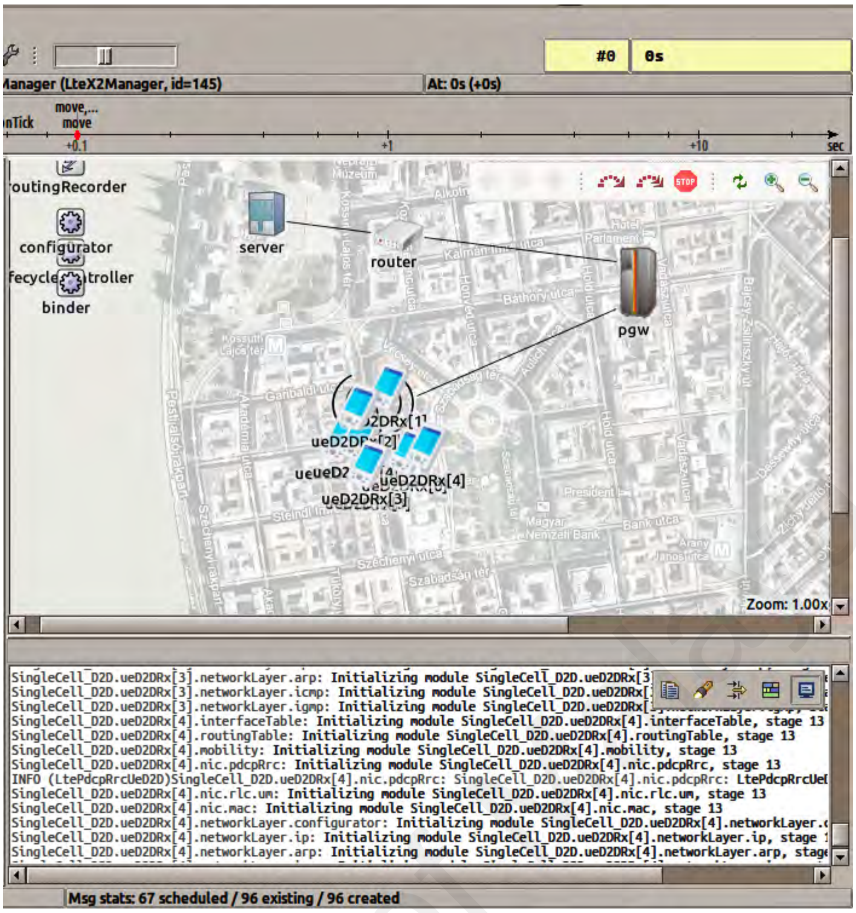
3.3 Simulation

To accomplish the last research objective, the author needed a tricky simulation approach. The author has reviewed numerous simulation tools like NS-2, NS-3, OMNET++, MATLAB and ONE. However, there was no comprehensive tool that support all the required features namely MEC offloading in LTE networks, task execution battery consumption, and D2D communications. Therefore, the author performed the following steps to simulate and to validate the proposed EEDOS scheme.

3.3.1 Approach

The study used Omnet++ to simulate the movement of the mobile nodes. Omnet++ is an object-oriented discrete event simulator. It is modular and provides infrastructure for writing tools. SimuLTE (Virdis et al., 2015) simulation tool is well-known to simulate the realistic LTE channel with D2D communication capability. So the author developed the simulation in this tool as shown in figure 3.4 and Figure 3.5. The SINR of each D2D and cellular communication can be extracted from the log file of the simulation. From the SINR, the author calculated the realistic D2D and cellular data rate based on the Shannon Formula (3.4). Then, the author developed a JAVA code to simulate the devices and assigning the task to them. In the simulation, the study considered two types of mobile phones with real energy consumption that were measured by Kwak, Kim, Lee, and Chong (2015). The study assigned tasks randomly according to the parameters in 100 iterations to gain the average runtime and energy saving parameters. The maximum distance of each device from the base station was considered 100 m. According to the results of the D2D and cellular links, the study created the energy efficient and delay aware graphs. In the next step, the study solved graphs with the help of Edmond's Blossom algorithm. Currently, the most efficient implementation of this algorithm was proposed in (Kolmogorov, 2008) that the author adopted it in this study. Based on the results, the study calculated the energy consumption and computation time. Figure 3.6 shows the simulation steps.

To compare the results of the proposed EEDOS scheme with the current studies, the author implemented two scenarios in JAVA whereas the same communication details were used in EEDOS. In the first scenario, the author considered a framework that resource

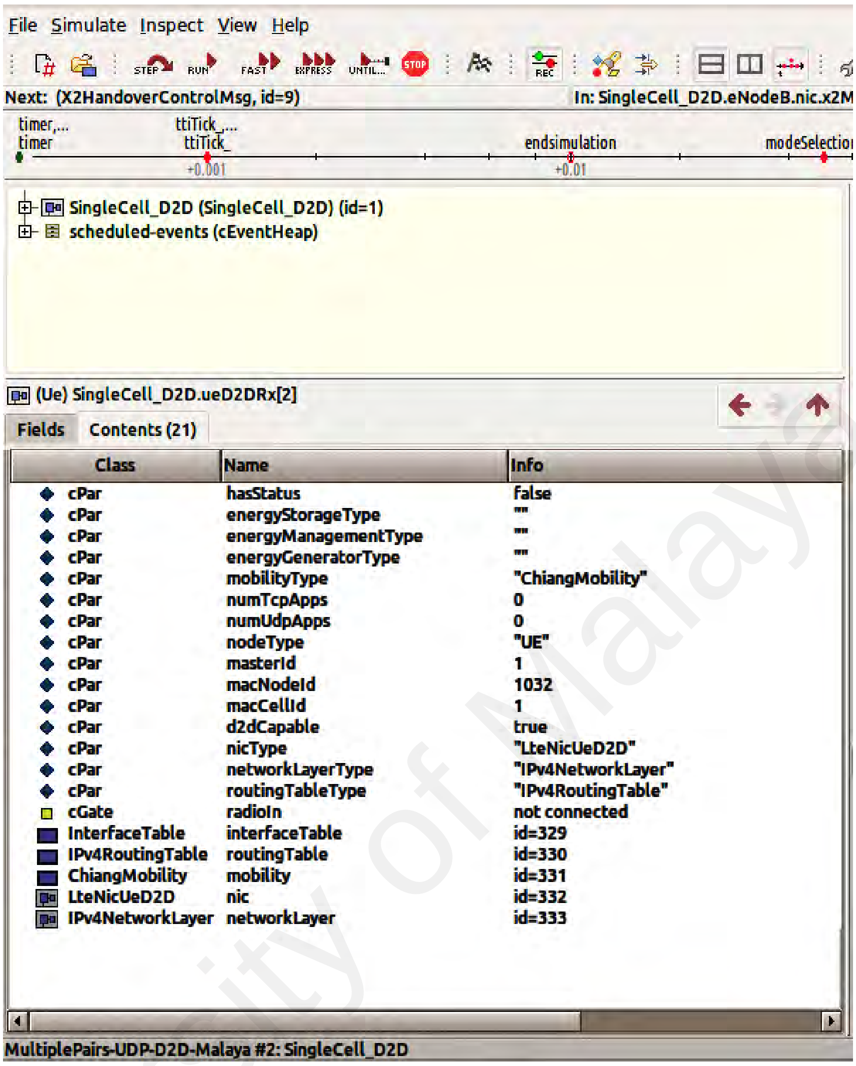


```
SingleCell_D2D.ueD2DRx[3].networkLayer.arp: Initializing module SingleCell_D2D.ueD2DRx[3
SingleCell_D2D.ueD2DRx[3].networkLayer.icmp: Initializing module SingleCell_D2D.ueD2DRx[
SingleCell_D2D.ueD2DRx[3].networkLayer.igmp: Initializing module SingleCell_D2D.ueD2DRx[
SingleCell_D2D.ueD2DRx[4].interfaceTable: Initializing module SingleCell_D2D.ueD2DRx[4].interfaceTable, stage 13
SingleCell_D2D.ueD2DRx[4].routingTable: Initializing module SingleCell_D2D.ueD2DRx[4].routingTable, stage 13
SingleCell_D2D.ueD2DRx[4].mobility: Initializing module SingleCell_D2D.ueD2DRx[4].mobility, stage 13
SingleCell_D2D.ueD2DRx[4].nic.pdcpRrc: Initializing module SingleCell_D2D.ueD2DRx[4].nic.pdcpRrc, stage 13
INFO (LtePdcpRrcUeD2D)SingleCell_D2D.ueD2DRx[4].nic.pdcpRrc: SingleCell_D2D.ueD2DRx[4].nic.pdcpRrc: LtePdcpRrcUe
SingleCell_D2D.ueD2DRx[4].nic.rlc.um: Initializing module SingleCell_D2D.ueD2DRx[4].nic.rlc.um, stage 13
SingleCell_D2D.ueD2DRx[4].nic.mac: Initializing module SingleCell_D2D.ueD2DRx[4].nic.mac, stage 13
SingleCell_D2D.ueD2DRx[4].networkLayer.configurator: Initializing module SingleCell_D2D.ueD2DRx[4].networkLayer.
SingleCell_D2D.ueD2DRx[4].networkLayer.ip: Initializing module SingleCell_D2D.ueD2DRx[4].networkLayer.ip, stage 
SingleCell_D2D.ueD2DRx[4].networkLayer.arp: Initializing module SingleCell_D2D.ueD2DRx[4].networkLayer.arp, stage
```

Msg stats: 67 scheduled / 96 existing / 96 created

Figure 3.4: SimuLTE simulation environment in OMNET++

constraint devices can only offload their work to the edge server as proposed by Lyu, Tian, Jiang, et al. (2018). In the second case, the author implemented a D2D collaboration framework that mobile devices can offload their task to an idle device. In this case, the study make use of edge server capabilities for offloading decision as described in Chen et al. (2017). In a pure D2D collaborative framework without any edge server in the network, devices should perform device discovery and decision making by themselves. Such a framework requires sophisticated algorithms and excessive resource usage. So, the performance difference would be entirely different from an integrated framework that leverages both D2D collaboration and MEC. When the study use edge server capabilities for offloading decision making, the author were able to precisely measure the effect of using the edge server to perform offloaded tasks.



Next: (X2HandoverControlMsg, id=9) In: SingleCell_D2D.eNodeB.nic.x2M

SingleCell_D2D (SingleCell_D2D) (id=1)
scheduled-events (cEventHeap)

(Ue) SingleCell_D2D.ueD2DRx[2]

Fields Contents (21)

Class	Name	Info
cPar	hasStatus	false
cPar	energyStorageType	""
cPar	energyManagementType	""
cPar	energyGeneratorType	""
cPar	mobilityType	"ChiangMobility"
cPar	numTcpApps	0
cPar	numUdpApps	0
cPar	nodeType	"UE"
cPar	masterId	1
cPar	macNodeId	1032
cPar	macCellId	1
cPar	d2dCapable	true
cPar	nicType	"LteNicUeD2D"
cPar	networkLayerType	"IPv4NetworkLayer"
cPar	routingTableType	"IPv4RoutingTable"
cGate	radioIn	not connected
InterfaceTable	interfaceTable	id=329
IPv4RoutingTable	routingTable	id=330
ChiangMobility	mobility	id=331
LteNicUeD2D	nic	id=332
IPv4NetworkLayer	networkLayer	id=333

MultiplePairs-UDP-D2D-Malaya #2: SingleCell_D2D

Figure 3.5: SimuLTE sample setting

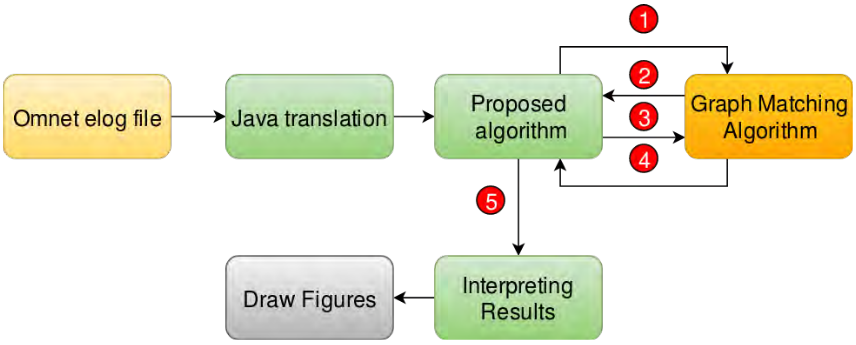


Figure 3.6: Simulation Steps

Table 3.2: **Simulation Parameters**

Parameter	Value
Edge CPU cycles	15GHz
Cell size	100m
Maximum Cellular Transmit Power	600mW
Maximum D2D Transmit Power	200mW
Maximum D2D bandwidth	20MHz
Maximum D2D distance	200m
Device CPU cycles	[1.4,1.6]GHz
CPU load	[30-70]%

3.3.2 Scenario

For task assignment, the study applied face recognition algorithm with 420KB of input image size that needs 1000 MCycles of computational resource (Soyata et al., 2012). Table 3.2 represents the simulation parameters. In the initialisation step, the study assigned a random load to devices, then tasks were generated based on the variable frequencies (30% to 70% of devices in the network). Devices first calculated their capabilities, and if they found out that they cannot meet the deadline, and/or they did not have sufficient energy to complete the task locally, they sent an offloading request to the MEC.

3.3.3 Required Software Tools

Following are the software tools that we used for the simulation.

- Device movement, D2D and cellular communication
  - OMNET++ 5.2.1, SimuLTE V1.0.1 (Virdis et al., 2015), supported platforms
    - 64-bit versions of Windows 7 and Windows 10
    - macOS 10.12
    - Ubuntu 16.04 LTS (The study's Choice)
    - Fedora Core 25
    - Red Hat Enterprise Linux Desktop Workstation 7.x
    - OpenSUSE 42

- Task assignment, energy consumption
  - NetBeans IDE 8.2, recommended platforms
    - Windows 7/ Windows 8
    - Ubuntu 15.04
    - OS X 10.10
  - JAVA Development Kit 8
- Edmonds Blossom Algorithm Source Code
  - C++ , Blossom V (Kolmogorov, 2008)

3.3.4 Hardware Used

The study run the simulation codes on a Lenovo laptop with Core i5-4200M CPU and 6GB of RAM. We selected Ubuntu 16.04 LTS because of the excellent support of the required software tools. However, to run the simulation for a high number of devices (70-100) much more powerful hardware are required to run the simulation faster.

3.4 Conclusion

In this chapter, the author explained how the study achieved the research objectives by designing EEDOS and its simulation. As in the second research objective, the author developed an offloading scheme by leveraging D2D collaboration method in MEC with consideration of energy and delay parameters. To fulfil the evaluation and validation research objectives, the author implemented and simulated EEDOS. In addition, the simulation of related studies have been performed. As the name of EEDOS scheme suggest, the performance metrics for comparing the effectiveness of proposed solution was energy consumption of mobile/IoT devices and the total time required to performing a task (offload time, computation and receive time of result). Moreover, the author applied other metrics to prove that EEDOS stands out from the existing studies, such as the number of successfully offloaded tasks, and the required edge server computational capability. In the next Chapter, the author presents the results of simulation, and explains how EEDOS outperforms the existing studies.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the author presents the results of EEDOS simulation. The primary goal of this study is to decrease the energy consumption of mobile devices while meeting the deadline of the tasks in the delay-sensitive applications. Therefore, the study presents the results of energy saving and energy consumption of mobile devices first. Then, the author investigate the effect of EEDOS on the execution time of the tasks. Finally, the author investigates other parameters to prove the superior of EEDOS against current studies. Whenever it is relevant, the author compares EEDOS with D2D offloading scheme that was proposed in Chen et al. (2017) and with MEC offloading scheme that was proposed in Lyu, Tian, Jiang, et al. (2018).

4.2 Comparison of EEDOS to current studies

To measure the performance of EEDOS and compare it with existing studies, the author performed the following steps:

- Investigate the performance of EEDOS through simulation
- Simulate the D2D collaboration scheme proposed in Chen et al. (2017)
- Simulate the MEC offloading scheme proposed in Lyu, Tian, Jiang, et al. (2018)
- Validate the superior of EEDOS by comparing the results of previous steps

In the following, the author presents the results and explains their meanings.

4.2.1 Energy Saving

In Figure 4.1, the author depicts the energy saving ratio of EEDOS and compares it with two existing studies.

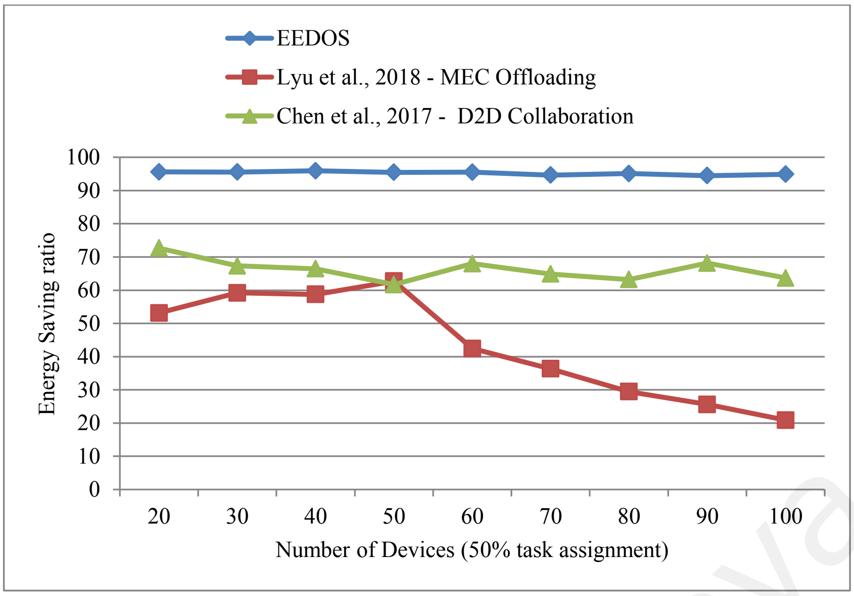


Figure 4.1: Energy Saving ratio with increasing number of devices

In this figure, Y-axis is the percent of energy saving when we compare the offloading scheme with a case that devices execute the task on their local hardware. The X-axis is the number of devices in the simulation. In this case, the study assigned a task to 50% of mobile devices randomly. EEDOS imposed steady energy saving in comparison to MEC offloading. Although change margin for the D2D collaboration scenario is just 16%, but with the increase in the number of nodes, the energy saving of the edge-only scheme degrade dramatically to 21%. The reason is the edge server becomes overloaded by the increase in the number of offloading requests and it cannot accept any other request, while in D2D enabled schemes like EEDOS and Chen et al. (2017), proximate idle devices will handle the task execution in case of overloading the edge server.

Similarly, Figure 4.2 presents the energy saving ratio but the number of devices is 50 nodes in all the simulation iterations. However, this study increases the possibility of having a task from 30% to 70% to see the effect of task assignment ratio on the energy saving.

There is a critical logical difference between Figure 4.1 and Figure 4.2. In the former figure, when the author add devices, the number of idle devices in the proximity will be

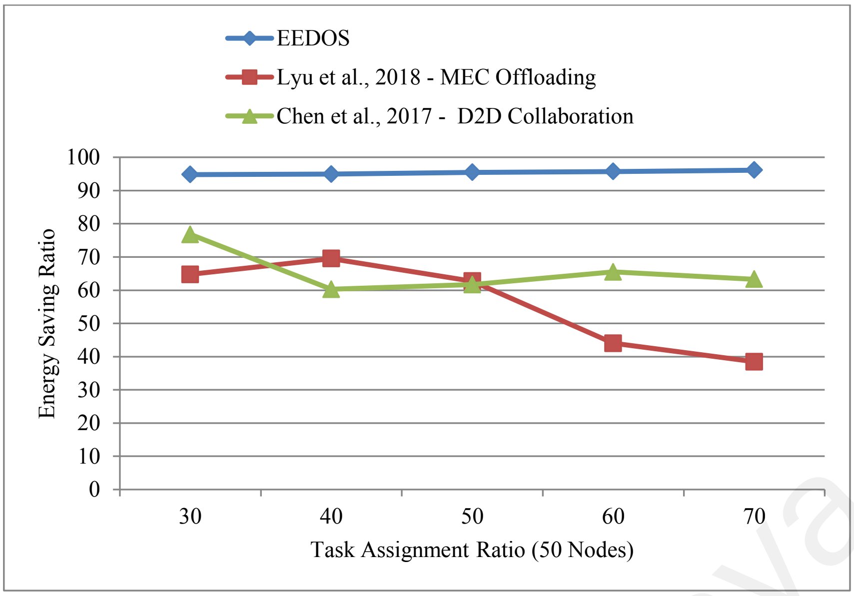


Figure 4.2: Energy Saving ratio with increase in the task frequency

increased. In the latter, when the study increased the task assignment ratio, the number of idle devices in the proximity will be decreased. However, in both cases, EEDOS shows a steady energy saving from 94.8% to 95.9%.

Authors in Chen et al. (2017) presented 50% of energy efficiency for D2D offloading when they compared running the task on local hardware. However, in this study, the author achieved the average of 60% energy efficiency for their scheme. The reason for this difference is that, in their scheme, if a user can perform its task, it will send its parameters to the edge server. Then the edge server informs the device that they can do it by themselves. In this study, we considered that all of the devices that have a task do not have sufficient energy to run it. So, they cannot complete the task, and they need to offload the task to MEC or a neighbour device. This approach is more common in the MEC offloading studies such as Lyu, Tian, Jiang, et al. (2018).

4.2.2 Energy Consumption

Average energy consumption to perform a task is shown in Figure 4.3 and Figure 4.4. The calculations are based on the real smartphone energy consumption measurement

(Kwak et al., 2015). Similar to the previous section, Figure 4.3 is based on the increase in

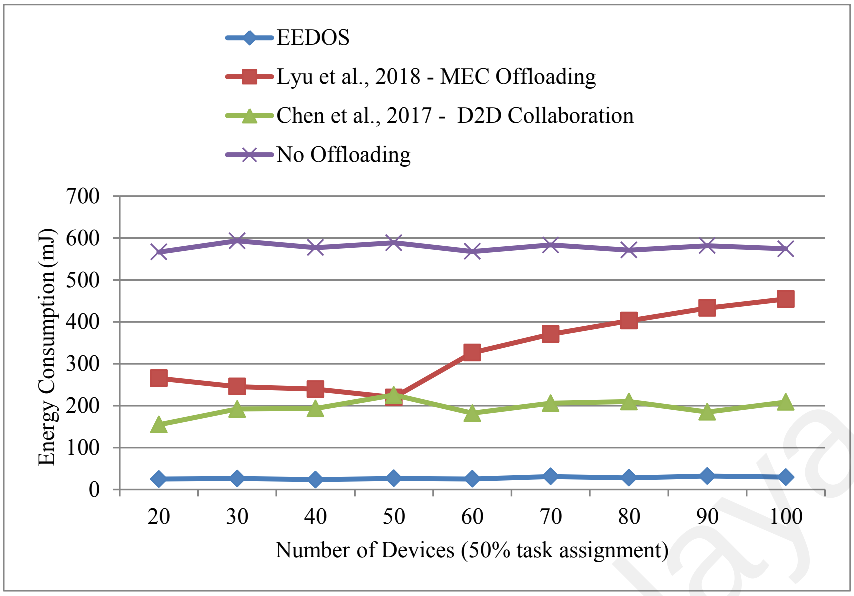


Figure 4.3: Energy Consumption with increasing number of devices

the number of devices. The study compared four scenarios a) No offloading: means that devices run the task on their local hardware without getting help form the network, b) MEC offloading: based on the work presented in the Lyu, Tian, Jiang, et al. (2018), a device can run the task locally or offload it to the MEC if it consume less energy, c) D2D offloading: based on the work in the Chen et al. (2017), a device can run the task locally or offload to a neighbour, if the MEC decides that it consumes less energy, and d) EEDOS: a device can run the task locally, offload to MEC, or a neighbour, if offloading consumes less energy.

The study shows that mobile devices in EEDOS consumed as low as 21.8mJ to 29.60mJ. After EEDOS, D2D offloading scheme performed better with 133.3mJ to 225.3mJ, in comparison to the MEC offloading scheme that mobile devices consume high energy from 168.5mJ to 349mJ. The reason of better performance in EEDOS in comparison with D2D offloading was leveraging the computational capability of edge server in the network. Again, as the number of devices increase, the performance of the MEC offloading scheme was degraded dramatically. It was because overloading the MEC with high number of offloading requests. However, in EEDOS, idle devices in the network would be a proper candidate to perform and complete the task if the MEC can not handle more tasks becuase of overloading.

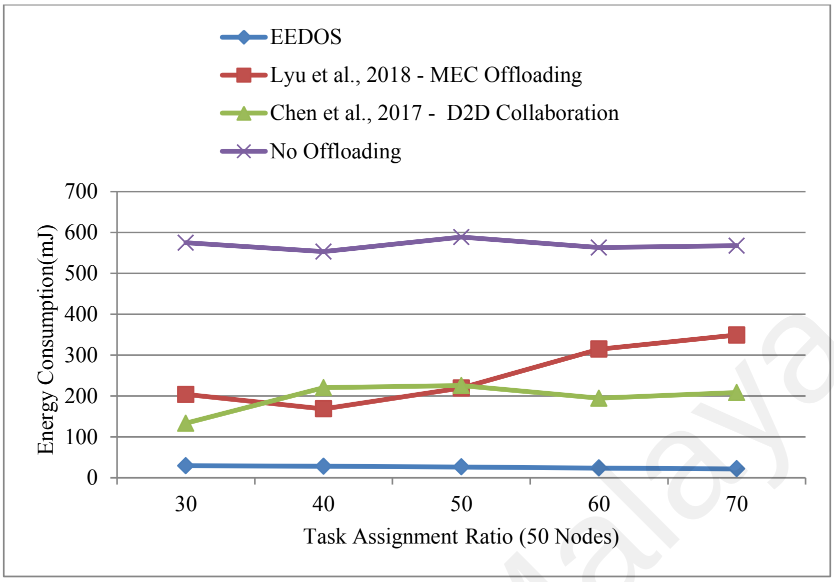


Figure 4.4: Energy Consumption with increase in the task frequency

Figure 4.4 presents the same metric of energy consumption for the mobile devices, but with the increase in the task assignment ratio for the fixed number of devices in the network. It must be noted that, running the task on local hardware as no offloading legend in both figures, consumed considerably higher energy than all offloading schemes (552.9 mJ to 593.2 mJ).

4.2.3 Execution time

Figure 4.5 demonstrates the average time to execute the task with the increase in the number of requesting devices (From 20 to 100 device). The study assigned a task to 50 % of the devices in the network. With increase in the number of devices, more idle nodes were available to participate in the collaboration. As a result of this, in EEDOS, the average execution time will be reduced when the number of devices is more than 30. In the study's simulation, the author supposed that, all idle nodes would dedicate all of their remaining CPU power to perform the task while edge server adjusts its resources based on the delay requirements. It means that, for delay sensitive tasks, MEC collaboration takes precisely 1.0s, but for D2D, it might be less, if the offloading destination (idle node) has high computation capacity.

In Figure 4.5, MEC offloading performed worst, because edge servers would maintain their resources to support more users. As a result of this, running the task by MEC took longer time. After MEC, D2D offloading performed better, because, idle devices in the network would dedicate all of their computing resources to run the offloaded task. EEDOS, outperformed other schemes, because, in comparison of MEC it leveraged idle devices, so, the execution delay and the time of transmitting the task would be decreased. In comparison of D2D offloading, the number of successful offloading increased in EEDOS, as discussed in the next section. Therefore, EEDOS decrease the total execution time by offloading more tasks. To understand this, consider a case that a device cannot offload the task because of the lack of proper offloading destination in D2D scheme and running the task on its local hardware takes several seconds. As a result of this, the total execution time would be increased in comparion of EEDOS scheme.

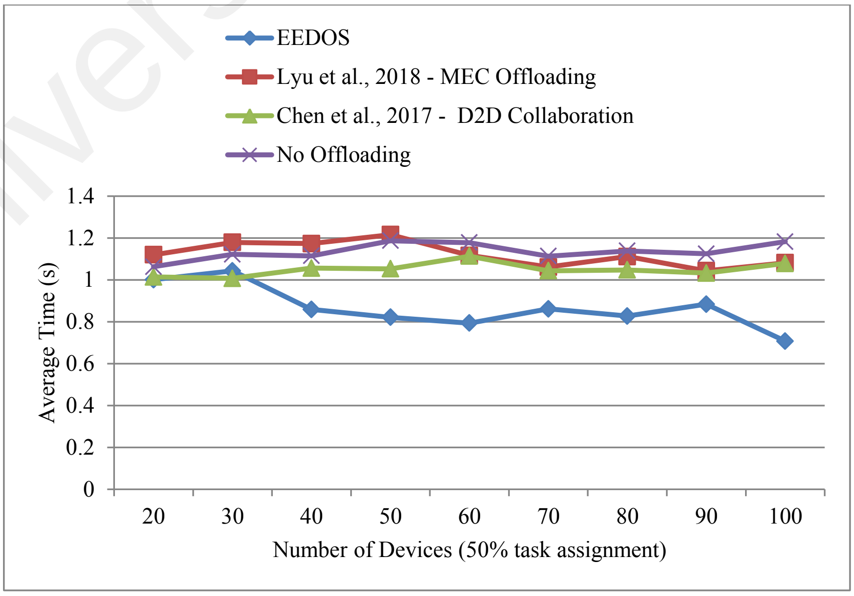


Figure 4.5: Execution Delay with increasing number of devices

In Figure 4.6, the author presented the results of the execution delay with the increase in the task assignment ratio for the fix number of 50 nodes in the network (From 30 % to 70 %). It must be noted that, the author present the no offloading results which are the devices that run the task on their local hardware, as a reference for comparison. If a device does not have sufficient energy to perform a task, obviously it cannot produce the output and contribute to the average execution time. However, the author calculated it to show that, beside energy saving in EEDOS, the execution time is improved as well.

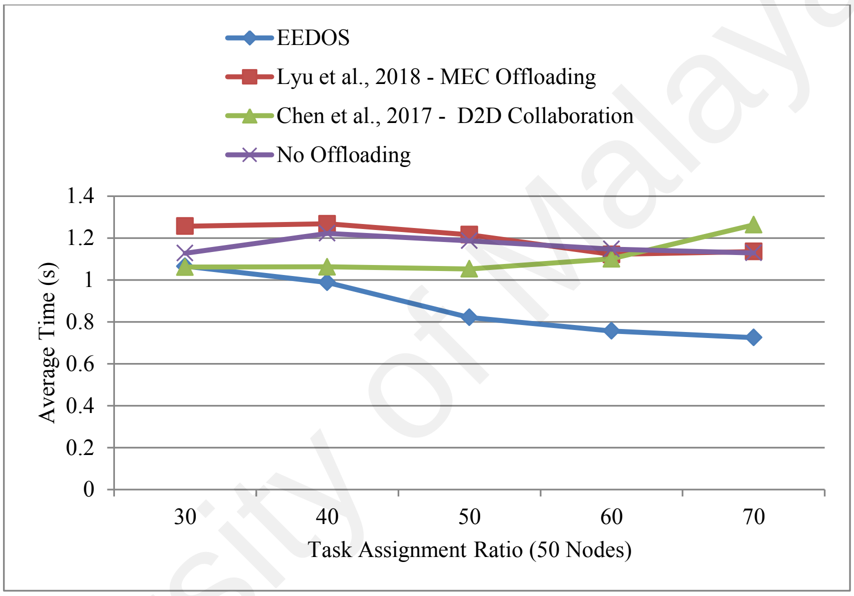


Figure 4.6: Execution Delay with increase in the task frequency

4.2.4 Successful offloading

In this section, the author compared the number of successful task offloading in EEDOS and MEC offloading scheme that presented in Lyu, Tian, Jiang, et al. (2018). The author did the comparison with the increase in the number of devices (From 20 to 100 devices) and the increase in the task assignment frequency (F= 30,50,70 %). In Figure 4.7, blue lines are number of tasks that were offloaded with increase in the number of devices in EEDOS. It is obvious that, with the increase in the number of devices, the number of successful tasks is increased at the same frequency in EDDOS. In contrast, in MEC offloading scheme that is illustrated with red colour, it can perform maximum 18 number of offloading requests. EEDOS outperforms MEC offloading because when the author simulated their work, the number of tasks that were offloaded was related to the computation capacity of the edge server. As a result, after 18 numbers of pre-assigning the resources for offloading, it is not possible to accept any more request. In contrast, in EEDOS, the author integrated D2D collaboration for task execution. Therefore, idle nodes in the proximity of the requested device can help to execute the task and increase the chance of successful offloading.

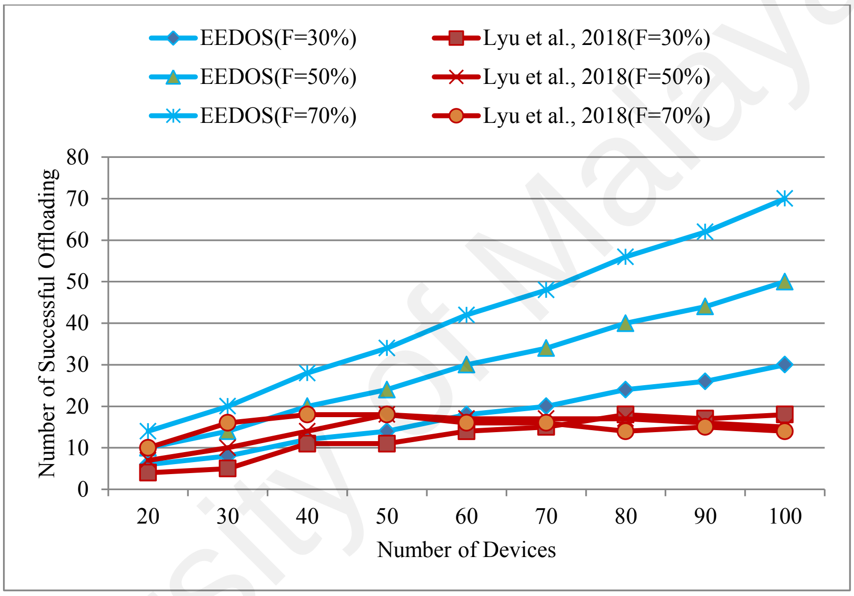


Figure 4.7: Offloaded requests

4.2.5 Meeting Deadline

In Figure 4.8, the author enumerated the number of missed deadlines. This means that, because of the failure in offloading and lack of enough computation power, some number of execution times exceeded the deadline. As mentioned previously, the author presented the no offloading values to compare with offloading schemes. It is clear that, EEDOS and D2D offloading scheme that leverage the computational capabilities of the massive number of devices in the network, performs better than executing the tasks locally (We assume that all devices have sufficient energy to perform their task) and MEC offloading.

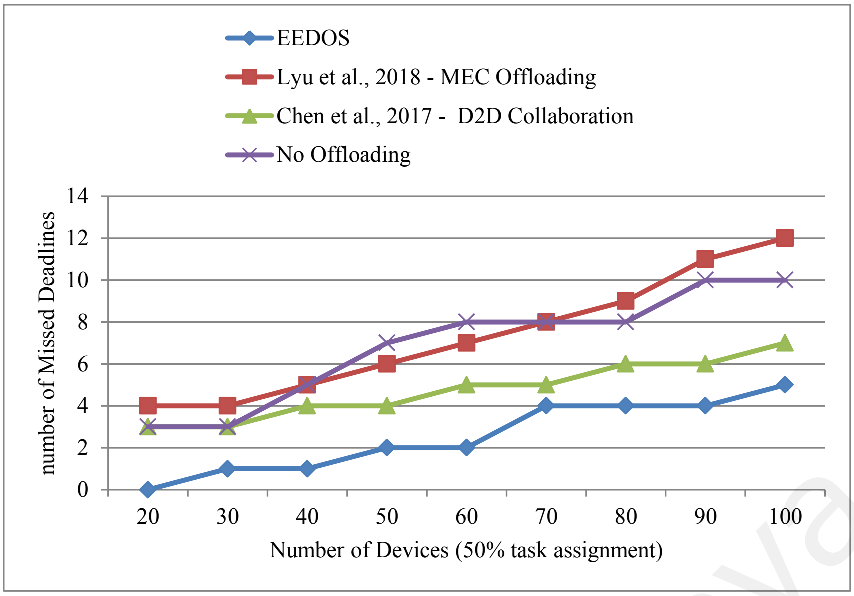


Figure 4.8: Missed deadlines

The study show that, EEDOS, has the best performance in term of meeting deadlines. When the author performed the simulation with 20 nodes, only EEDOS could meet all the deadlines while D2D offloading miss 3 and MEC offloading miss 4 deadlines. It is more dramatic with 100 nodes in the network that, MEC offloading missed 12 deadlines but D2D offloading perform better with 7 misses. However, the best result achieved by EEDOS with only 5 missed deadlines with 100 nodes in the network when the author assigned a task to 50 % of them. The reason that D2D offloading is outperforms MEC offloading is because of leveraging the computational capability of large number of idle devices in the network by this method, while MEC offloading relies only on the computational capability of the edge server. As the number of devices increase in the network, there is a higher chance to find an idle device in the proximity that can execute the offloaded task. So, it is clear that EEDOS and D2D offloading methods perform better than MEC offloading. In EEDOS, the author integrated the capabilities of idle devices in the network and Edge servers, so there are more resources in the network that can contribute in the offloading. As a result of this, EEDOS achieved the best results in term of completing the tasks in comparison with both methods.

4.2.6 Edge server resource

One of the primary motivations of this work is to alleviate the workload on the edge computing server. Overloading the edge server is one of the main limitations of the MEC offloading schemes. By combining D2D offloading with MEC offloading, the author mitigated this problem. As depicted in Figure 4.9, more than 61GHz of computation capacity was required in the edge-only scheme which refers to the Lyu, Tian, Jiang, et al. (2018) study, when the author run the simulation with 100 devices that 70% of them have a task to offload. However, in EEDOS, D2D communication and collaboration have decreased this amount to the value of 17.5GHz. By reviewing the current studies on MEC offloading, the author claims that 61GHz of CPU resource is not realistic to be available in edge servers. For example, in (K. Zhang et al., 2016), they assumed 4 GHz of CPU or 10 GHz in (Lyu, Tian, Jiang, et al., 2018). Similarly, when the network is not busy (20 devices with 30% task assignment), EEDOS needed only 2.72GHz of CPU, but MEC offloading required 17.22GHz CPU resource in the edge server.

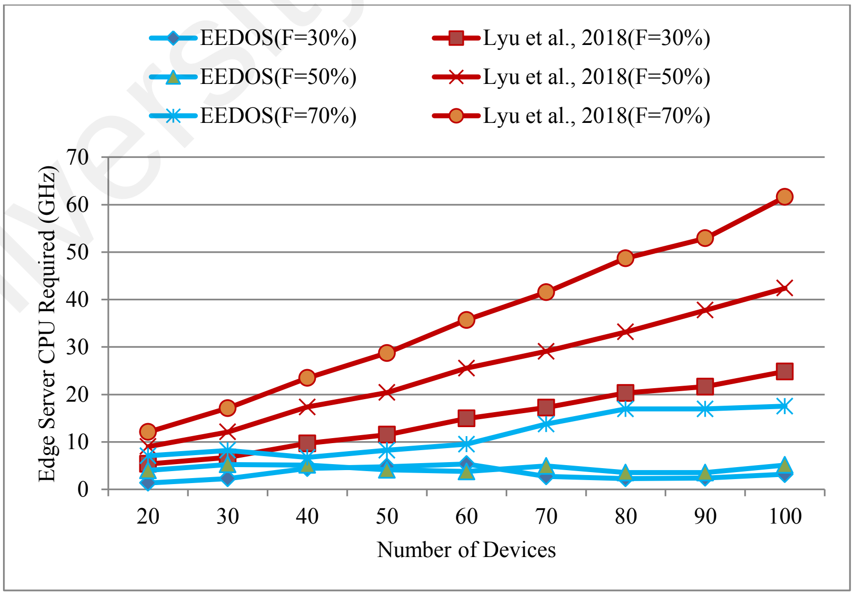


Figure 4.9: Edge CPU required to handle all offloading requests and meet deadline

It can be seen from the Figure 4.9 that, MEC offloading "required edge server CPU", is a linear function that has a direct relation with the number of devices. This means with the increase in the number of devices, the amount of required CPU capacity increased with the same rate. However, in EEDOS, the required edge server CPU is relatively same when the author increased the number of devices. This is because massive number of idle devices are available in the network to cooperate in the offloading scheme. This approach decreases the load on the edge server.

4.3 Conclusion

In this chapter, the author observed that EEDOS gained 95% of energy saving in comparison to local task execution. The author compared it with edge computing task offloading and D2D collaboration schemes, and the results showed that EEDOS outperformed existing approaches on task offloading in terms of energy efficiency of mobile/IoT devices and decreasing the task execution delay. Specifically, EEDOS required far less server computation capacity in comparison to MEC offloading scheme. Last but not least, EEDOS improved the number of missed deadlines in comparison to existing studies. In the next chapter, the author conclude the work and briefly discuss the study's research objectives and the findings. According to the scope of this research, the author assumed that all of the devices in the network are willing to cooperate. However, for the real implementation of the proposed scheme, some incentives must be considered for cooperation. The author discuss some suggestions in the next chapter.

CHAPTER 5: CONCLUSION

5.1 Introduction

In this research, the author has presented the novel energy efficient and delay-aware offloading scheme (EEDOS) to perform energy efficient task offloading while meeting the delay requirements of resource-limited IoT and mobile devices. Through the literature review, several research gaps were identified, and their solution were applied in the proposed scheme. The author has validated the scheme by the simulation of EEDOS and related studies. The next section, the author recap what has been done to achieve the research objectives of this study.

5.2 Objectives and Findings

The author has conducted the study according to the steps in Figure 1.3. The following is the summary of the findings regarding the research objectives (RO).

(a) ***RO1: To investigate energy and delay challenges of IoT/mobile devices and current methods to mitigate them with D2D collaboration and offloading in MEC.***

The author has done the literature review in Chapter 2. According to the problem statement, the author has selected two of the best current studies that addressed the same problems. The study's criteria to select the related studies was based on their citations and journal ranking. The first study presented in (Chen et al., 2017) leveraged the computational capability of edge servers to find a proper D2D offloading destination. The second study by the authors in (Lyu, Tian, Jiang, et al., 2018), used the edge server computational power to perform the task of devices in the network. The author highlighted the research gaps in these studies as the current study's contribution to this field.

(b) ***RO2: To develop an offloading scheme by leveraging D2D collaboration method in MEC with consideration of energy and delay parameters.***

In Chapter 3, the author explained how the EEDOS was designed, and discussed the study's method to simulate it. The study presented the problem formulation of task execution and task offloading to MEC and proximate devices. Moreover, the author proposed two algorithms to find the proper offloading destination. In EEDOS, when a mobile device cannot perform a task, it sends an offloading request to the edge server. The edge server builds two graphs. The nodes of the graphs are:

- Requesting device that has a task and cannot complete it
- MEC reserved resources to perform the offloaded task
- Idle device in the proximity of the requested task

The first graph was weighted based on the time required to offload the task and complete it in the neighbour device or MEC, coupled with the required time for offloading. The second graph was weighted based on the energy required to offload the task and the energy that the idle device consumes to complete the task. Because the study's goal was to decrease the energy consumption of mobile devices, the author did not consider the energy required for the MEC to perform the task. In this study, it is assumed that, MEC was connected to the power grid and did not rely on battery. The author used maximum matching with minimum cost algorithm to solve the graphs. Finally, the edge server sent the results to the requesting devices.

(c) ***RO3: To evaluate the performance of the proposed scheme concerning energy efficiency and total execution time, and compare with to the current D2D collaboration and MEC offloading studies.***

The author applied several metrics in the study's simulation to validate the solution. The amount of energy that was consumed to offload plus completing the task, task execution delay, number of missed deadlines and the amount of required edge server CPU capacity were among the metrics. The author discussed the study's simulation method to evaluate the performance of EEDOS in Chapter 3. In Chapter 4, the author presented the results of EEDOS simulation and compared it with current studies.

5.3 Results Summary

To evaluate the effectiveness of the proposed scheme, the author implemented a simulation. Here, the author is able to show that the proposed EEDOS scheme achieved 95% of energy saving in comparison to local task execution. The proposed scheme saves more energy on mobile/IoT devices than the existing studies so they can operate longer. Authors in (Chen et al., 2017) claimed 50% of energy saving for D2D collaboration. EEDOS performed much better because the computational capabilities of MEC was taken into account for executing the offloaded tasks. Similarly, the EEDOS scheme performs better than MEC offloading scheme (Lyu, Tian, Jiang, et al., 2018), because computational capabilities of large number of idle devices in the network was taken into account as well. Additionally, the EEDOS scheme improved the task execution delay, so mobile/IoT devices could run delay-sensitive applications smoothly. Finally, by leveraging the computation power of idle devices in the network, there was no need for deploying expensive edge servers with huge computational capacity to handle all the offloading requests. Therefore, the implementation cost of this current scheme is lower than the existing MEC offloading schemes for network operators.

5.4 Limitation of the study

In this work, physical layer communication details were not included. It must be noted that, this study's goal was not to improve energy efficiency of D2D communication or MEC. Instead, the author put the attention on the potential of these technologies to save the energy of mobile devices and improve the task execution delay. As another limitation, the EEDOS system was static and movement of the mobile devices were not considered. Additionally, the author supposed that every idle device in the proximity was willing to collaborate on the offloading. Although this is a common hypothesis in research, it is not a realistic consideration. In the real world, users have more concern about their energy, so an incentive mechanism must be considered to motivate them for collaboration. In the next section, the author presented the current solution to this problem and proposed a novel solution based on the Blockchain as a research direction for the interested researchers on this topic. Following is the brief indication of EEDOS limitations:

- Physical layer details such as available resource blocks are overlooked.
- Dynamic of the mobile devices are not considered.
- we assume that every idle device is willing to share its computational resources without any remuneration.

Each of the mentioned limitations, could be an attractive research topic for the interested researcher in this field.

5.5 Future research direction

As the author discussed in the previous section, EEDOS system is static. Therefore, investigating the performance of this scheme in a dynamic mobile network that, movement of devices affects the parameters is an attractive research direction.

Basically, D2D communication and MANET are different in the physical layer details and operation. But, in term of cooperation stimulation, both of these techniques require some incentives for participating devices. Cooperation stimulation is a legacy problem in MANET (Chlamtac et al., 2003) and there are a large number of studies that addressed this problem in MANET. However, these solutions are not applicable in D2D communication, because of the scale of the network. By integrating MEC and D2D, researchers can integrate complex solutions to solve the issue, because edge servers with high computational power can manage the network. Therefore, leveraging the capabilities of MEC to manage the incentive mechanism is another important study.

Recently, social networks gained much attention to solve cooperation stimulation issue in D2D enabled networks (Yi, Huang, & Cai, 2018)(Huynh, Wang, Duong, Vo, & Chen, 2018)(Y. Meng, Jiang, La, Quek, & Ren, 2017)(Xiao, Niyato, Chen, & Han, 2016)(C. Zhang, Sun, Mo, Zhang, & Bu, 2016)(K. Zhang et al., 2016). Therefore, integrating social-aware techniques in this EEDOS could be an interesting topic.

On the other hand, we cannot leverage the social-aware solutions for IoT devices, because of their nature. Additionally, IoT devices are more vulnerable against hackers because of their limited computation power. By borrowing the idea of using Blockchain technology in MEC for IoT network that presented in (R. Li et al., 2018), the author suggests providing an incentive mechanism for offloading. In the next section, the author first presents basic concepts of Blockchain, and then propose a novel solution to the cooperation stimulation.

5.5.1 Blockchain

Blockchain is a distributed ledger that is shared among users in the network. The main characteristic of the Blockchain is the lack of central authority to validate the transactions. Instead, every transaction must be validated by the eligible users, called miners, in the network through a consensus mechanism. Every user is determined by its public key that acts as an address in the Blockchain and it can create a transaction by sign it with its private key. After the creation of a transaction, users must verify it to reside in a block. Every block is a composite of some transactions, a block header and the hash of previous block (Christidis & Devetsikiotis, 2016) as depicted in the Figure 5.1.

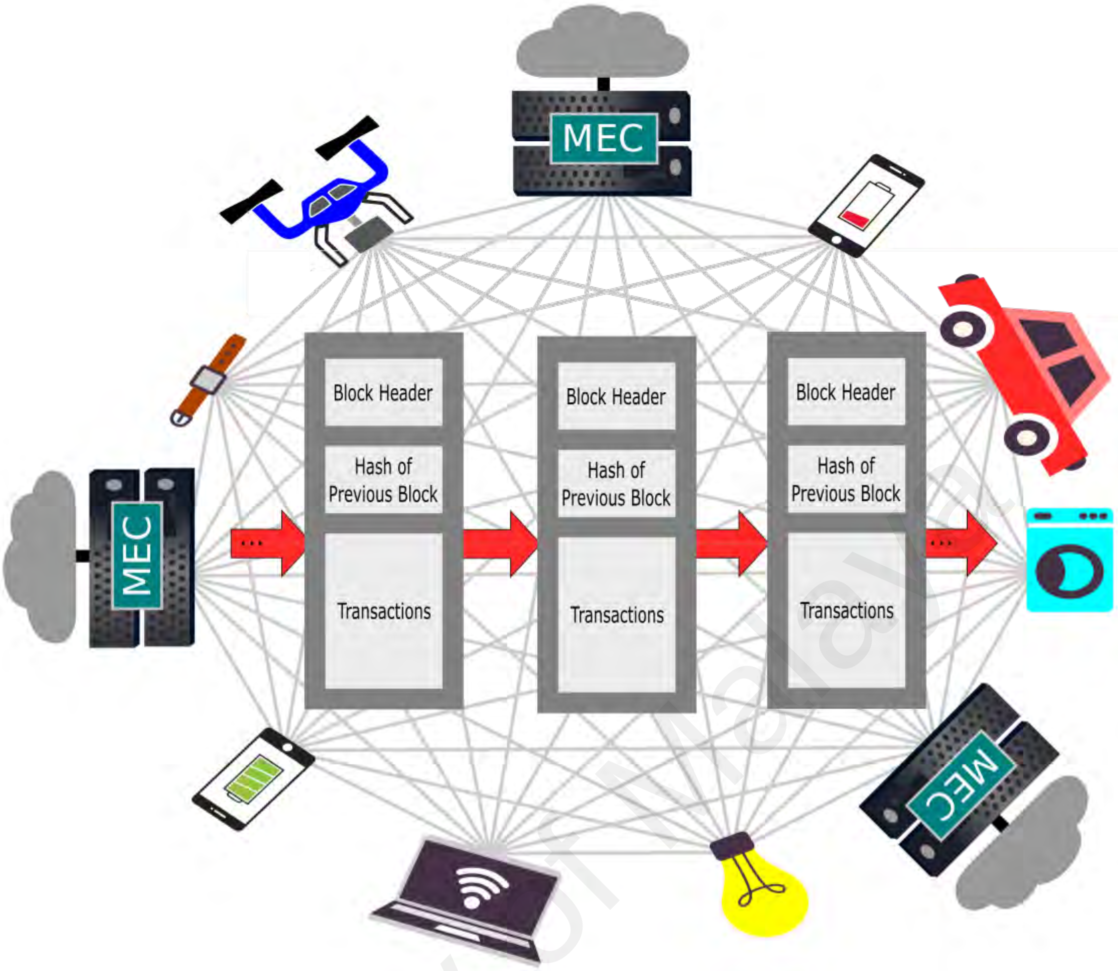


Figure 5.1: Blockchain in a Peer to Peer network with MEC

5.5.1.1 Consensus Mechanism

There is no central authority to validates the transactions and controls the security of the Blockchain. Therefore, there must be a mechanism to control the flow of the transactions and their validations. In the Blockchain, nodes must be agreed on a protocol to update the ledger. Consensus mechanism is used to create a valid block and update the distributed ledger. Nodes that can create a block called miners. Bitcoin uses the Proof of Work (PoW) as the consensus mechanism. In PoW, expensive calculations required to find the next valid hash in the network. A miner that can solve this puzzle, can add a block to the network and also is eligible to verify the transactions. However, PoW is extremely costly and become more complicated as the Blockchain grows.

5.5.1.2 Smart Contracts

Smart contract paradigm is not dedicated to Blockchain technology. It is proposed by Szabo (W. Meng, Tischhauser, Wang, Wang, & Han, 2018) and it brings a possibility to run an automated agreement without any kind of trust between parties. In a Turing-complete Blockchain project like Ethereum, any kind of code can be written to run automatically in the network if certain conditions were met.

5.5.1.3 Types of Blockchain

In the public or permission-less Blockchain like Bitcoin and Ethereum, anyone can join without the approval of a third-part. In contrast, in private or permissioned network, only whitelisted nodes can join (Christidis & Devetsikiotis, 2016). However, it is possible to design a Blockchain platform that uses a public technology like Ethereum in a private network and create a whitelist of nodes who can transact, mine or deploy the smart contract.

5.5.1.4 Is Blockchain applicable in Cooperative MEC?

Based on the flow diagram presented in the (Fernández-Caramés & Fraga-Lamas, 2018), the author investigate if the Blockchain is a proper solution for incentivize the D2D offloading scheme in MEC.

(a) ***Is there a need to keep a redundant copy of the ledger in distributed computers?***

Yes. In the current study's case, there is a need to minimize the interaction between mobile/IoT devices with the cloud server to support the real-time applications. Mobile devices and vehicles may roam between different Base Stations. Therefore, there is a need to store a copy of the ledger in every MEC server to use it in the offloading scheme.

(b) ***Do mobile/IoT devices trust each other?***

No. From the user viewpoint, a peer in the network may want to use the network service without any cooperation. Besides, devices need to make sure that their contribution is valued by the others in the network.

(c) ***Is the deploying network is restricted?***

Yes. The offloading scheme will be employed by a network operator that enabled the MEC service and D2D communication. As a result, the network operator has a full control on deciding who can join the network.

(d) ***Is it critical to decide who can update and maintain the Blockchain?***

It depends. If the primary aim of the mobile/IoT platform is to provide computation power and conserve their energy, it is not logical to let them participate in the expensive process of mining. However, every node must be able to create a transaction for paying its offloaded task. In this case, a private permissioned Blockchain is the selected technology. In contrast, if a researcher intends to design a robust cooperative scheme that supports real-time application for every mobile/IoT device without any concern about their energy, a private permissionless Blockchain is a good choice. In this case, mobile/IoT device can do the mining process in their idle times to earn tokens for their future use.

5.5.2 Token-based Cooperative platform for Mobile Edge Computing

In the previous section, the author inferred that, a private permissioned Blockchain is a proper solution to manage the incentives for D2D cooperation in the current study's offloading scheme. We can leverage the computation power of MEC servers to perform the costly task of mining and validating the transactions. Now, if a device intends to send an offloading request to the MEC server, it must deposit some of its tokens (based on the computation complexity). In the following, the author introduced the characteristics of the solution.

(a) *Cryptographic Algorithms*

IoT devices are resource constraints and cannot handle complex cryptographies like Rivest–Shamir–Adleman (RSA). Public-key cryptography is a critical need to provide security and privacy of Blockchain. Using RSA for IoT is not energy efficient, as a result, some researchers suggest using Elliptic Curve Cryptography based algorithms because of better speed and energy efficiency in IoT devices (Yeow, Gani, Ahmad, Rodrigues, & Ko, 2017).

Nodes require a hash function to sign the transactions. The selected function must be fast and consumes the smallest amount of energy. SHA-256 hash algorithm is the most popular in the Blockchain. However, its energy requirements are not suitable for IoT devices. Therefore, researchers suggest using Advanced Encryption Standard (AES) which consumes less energy and still is secure.

In the proposed solution for future research direction, we suggest certificateless cryptography (Al-Riyami & Paterson, 2003) which the user public key is generated by the user's identity and some secret that Key Generation Center (KGC) is not aware. As a result, no one knows about the private key of the user, but its public key still can be verified by its ID. In a traditional Public Key Infrastructure approach, Certification Authority signs a certificate to assure the public key. However, this approach consumes excessive bandwidth and CPU. Therefore, certificateless cryptography is a proper choice for the IoT environment. As proposed by (R. Li et al., 2018), in the study's solution, the edge server can play the role of KGC to reduce the load on the IoT device.

(b) ***Message Timestamping***

To prevent injecting fake transactions in the middle of the blocks, transactions must be timestamped correctly in order. In the current study's case, miners are belonged to the network operator and they are definitely synced to provide signalling service for the cellular devices. Therefore, all of the nodes can trust MEC servers as timestamp server.

(c) ***Consensus Mechanism***

In the private Blockchain, network operator controls the user access, so the risk of Sybil attack is low. As a result, costly mining algorithms like PoW and PoS are not required.

5.5.2.1 Extension to the offloading scheme

Mobile/IoT devices that intend to use the offloading service from the network must first register in the Blockchain. The author has shown the process in Figure 5.2. The edge server has the role of KGC and has a secret key to combine with the ID_{device} to produce the partial private key (PPK_{device}). Device A starts this process by contacting the edge server and send the ID_A to get the PPK_A. Then, the device can produce the private key by the combination of the PPK_A and a secret value (X_A). Meanwhile, the device creates a public key, by a function that uses the X_A and the ID_A. Now, miners can verify the device A transactions by checking if the public key is derived from the ID_A or not, and if the signed transaction can be verified by the public key. Upon registering to the Blockchain, the edge server creates a transaction to send the user's initial tokens. In the proposed solution, tokens are not necessarily bound to a cryptocurrency. It must be noted that some of the IoT devices are extremely resource constraint and cannot contribute to the shared computation of the network. Therefore, if they spend their initial tokens, they cannot use the offloading service afterwards. To solve this issue, MEC can create a smart contract to send tokens for IoT devices periodically (like n number of tokens every 24-hours to the

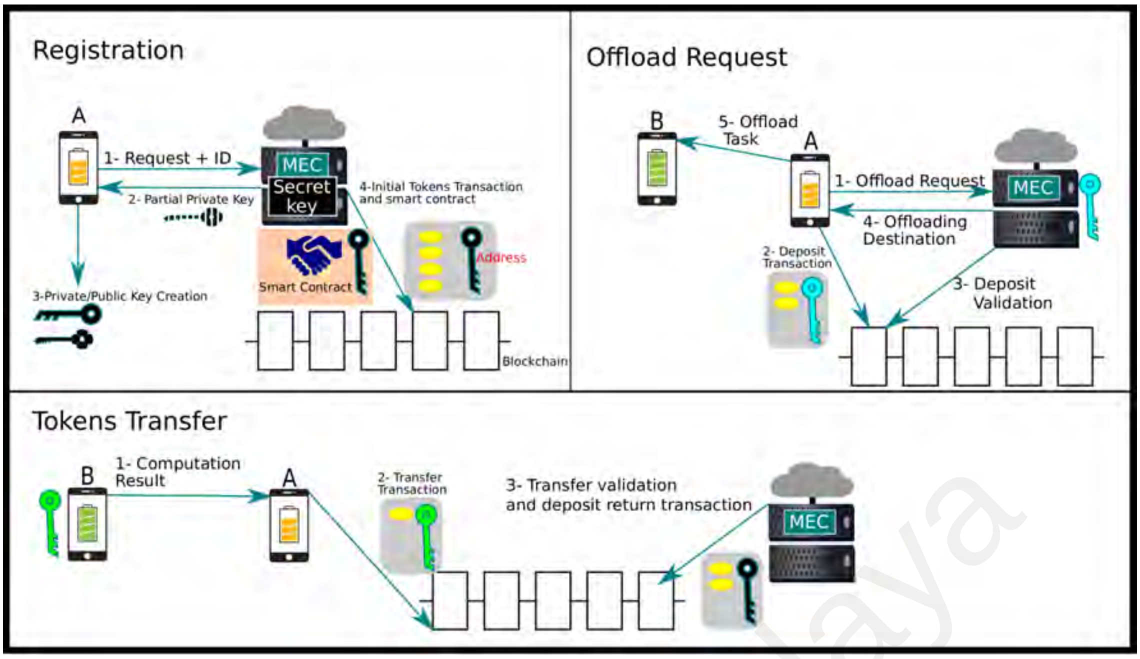


Figure 5.2: Blockchain operation for incentivize the cooperation

devices that are out of tokens). On the other side, this smart contract must be adjusted to prevent overloading the MEC server and idle devices in the network.

Now, if a device A sends an offloading request to the MEC, it must create a transaction to deposit $m + x$ number of tokens, where m is determined based on the complexity of the task and x is the penalty that imposed to the user who refuses to pay the cost of its offloaded task. To encourage the users to use the offloading service out of the peak hours, a pricing policy can be leveraged to adjust the price, based on the overall load of the network. Miners validate the transaction and meanwhile, MEC solves the offloading graph and sends the results to the requested devices. When device A has notified about the destination of its task (device B), it offloads its input data to the B. When the neighbour performed the task and sent the result back to the requested device, requested device must create another transaction to transfer the tokens. In the current study's case, device A creates a transaction for decreasing m tokens from itself. Meanwhile, it creates another transaction to increase the account of device B by m tokens, signs the transaction and locks it against device B's public key and sends them to the Blockchain. Miners are able to verify the validity of transactions and the MEC releases the deposited tokens. Then, m number of tokens will be sent to the device B as the reward of its cooperation to the offloading scheme.

REFERENCES

Aazam, M., Zeadally, S., Harras, K. A. (2018, oct). Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. *Future Generation Computer Systems*, *87*, 278–289.

Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T. (2018, feb). Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal*, *5*(1), 450–465.

Ahmad, H., Agiwal, M., Saxena, N., Roy, A. (2018, aug). D2D-based Survival on Sharing for critical communications. *Wireless Networks*, *24*(6), 2283–2295.

Akpakwu, G., Silva, B., Hancke, G., a.M. Abu-Mahfouz. (2017). A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges. *IEEE Access*, *6*, 3619–3647.

Ali, F. A., Simoens, P., Verbelen, T., Demeester, P., Dhoedt, B. (2016). Mobile device power models for energy efficient dynamic offloading at runtime. *Journal of Systems and Software*, *113*, 173–187.

Alippi, C., Fantacci, R., Marabissi, D., Roveri, M. (2016). A Cloud to the Ground: The New Frontier of Intelligent and Autonomous Networks of Things. *IEEE Communications Magazine*, *54*(12), 14–20.

Al-Riyami, S., Paterson, K. (2003). Certificateless public key cryptography. *Advances in Cryptology-ASIACRYPT 2003*, 452–473.

Asadi, A., Wang, Q., Mancuso, V. (2014). A Survey on Device-to-Device Communication in Cellular Networks. *IEEE Communications Surveys & Tutorials*, *16*(4), 1801–1819.

Baccarelli, E., Naranjo, P. G., Scarpiniti, M., Shojafar, M., Abawajy, J. H. (2017). Fog of Everything: Energy-Efficient Networked Computing Architectures, Research Challenges, and a Case Study. *IEEE Access*, *5*, 9882–9910.

Banerjee, S., Konishi, H., Sönmez, T. (2001, jan). Core in a simple coalition formation game. *Social Choice and Welfare*, *18*(1), 135–153.

Bhushan, N., Li, J., Malladi, D., Gilmore, R., Brenner, D., Damnjanovic, A., . . . Geirhofer, S. (2014). Network densification: The dominant theme for wireless evolution into 5G. *IEEE Communications Magazine*, *52*(2), 82–89.

Boccardi, F., Heath, R., Lozano, A., Marzetta, T., Popovski, P. (2014, feb). Five disruptive technology directions for 5G. *IEEE Communications Magazine*, *52*(2), 74–80.

Bonomi, F., Milito, R., Zhu, J., Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 13–16.

Buttyan, L., Hubaux, J.-p. (2001). Nuglets : a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks. *Technical Report DSC*, 1–15.

Chávez-Santiago, R., Szydełko, M., Kliks, A., Foukalas, F., Haddad, Y., Nolan, K. E., . . . Balasingham, I. (2015). 5G: The Convergence of Wireless Communications. *Wireless Personal Communications*, *83*(3), 1617–1642.

Chen, X., Pu, L., Gao, L., Wu, W., Wu, D. (2017). Exploiting Massive D2D Collaboration for Energy-Efficient Mobile Edge Computing. *IEEE Wireless Communications*, *24*(4), 64–71.

Chen, X., Shi, Q., Yang, L., Xu, J. (2018). ThriftyEdge: Resource-Efficient Edge Computing for Intelligent IoT Applications. *IEEE Network*, *32*(1), 61–65.

Chen, X., Zhang, J. (2017). When D2D meets cloud: Hybrid mobile task offloadings in fog computing. *IEEE International Conference on Communications*, 0–5.

Cheng, W., Yu, J., Zhao, F., Cheng, X. (2018). SSDNet: Small-World Super-Dense Device-to-Device Wireless Networks. *IEEE Network*, *32*(1), 186–192.

Chlamtac, I., Conti, M., Liu, J. J. (2003). Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Networks*, *1*(1), 13–64.

Christidis, K., Devetsikiotis, M. (2016). *Blockchains and Smart Contracts for the Internet of Things* (Vol. 4).

Cisco. (2017). Cisco Visual Networking Index: Global Mobile Data Traffic Forecast

Update, 2016–2021 White Paper. *Cisco*, 2016–2021. Retrieved from `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf`

Cisco Systems. (2016). Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. *Www.Cisco.Com*, 6.

Datang Wireless Mobile Innovation Center. (2013). *Evolution, Convergence, and Innovation 5G* (Tech. Rep.). DATANG Telecom. Retrieved from `http://www.datanggroup.cn/upload/accessory/201312/2013129194455265372.pdf`

Denko, M. K. (2012). Detection and Prevention of Denial of Service (DoS) Attacks in Mobile Ad Hoc Networks using Reputation-Based Incentive Scheme. *", Journal Systemics, Cybernetics and Informatics, Volume 3 - Number 4, pp 1-9.*, *3*(046940), 1–9.

Dolui, K., Datta, S. K. (2017). Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. *GIoTS 2017 - Global Internet of Things Summit, Proceedings*.

Doumi, T., Dolan, M. F., Tatesh, S., Casati, A., Tsirtsis, G., Anchan, K., Flore, D. (2013). LTE for public safety networks. *IEEE Communications Magazine*, *51*(2), 106–112.

Du, J., Zhao, L., Feng, J., Chu, X. (2018, apr). Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems With Min-Max Fairness Guarantee. *IEEE Transactions on Communications*, *66*(4), 1594–1608.

Edmonds, J. (1965). Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, *69B*(1 and 2), 125.

Evans, D. (2011). *The Internet of Things - How the Next Evolution of the Internet is Changing Everything* (Tech. Rep. No. April). Cisco. Retrieved from `https://www.cisco.com/c/dam/en{_}us/about/ac79/docs/innov/IoT{_}IBSG{_}0411FINAL.pdf`

Fan, W., Liu, Y., Tang, B., Wu, F., Wang, Z. (2018). Computation Offloading Based on Cooperations of Mobile Edge Computing-Enabled Base Stations. *IEEE Access*, *6*, 22622–22633.

Feng, D., Lu, L., Yuan-Wu, Y., Li, G., Li, S. (2014). Device-to-device communications in cellular networks. *IEEE Communications Magazine*(April), 49–55.

Fernández-Caramés, T. M., Fraga-Lamas, P. (2018). *A Review on the Use of Blockchain for the Internet of Things* (Vol. 6).

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660.

Guo, H., Liu, J. (2018, may). Collaborative Computation Offloading for Multiaccess Edge Computing Over Fiber–Wireless Networks. *IEEE Transactions on Vehicular Technology*, *67*(5), 4514–4526.

Hao, Y., Chen, M., Hu, L., Hossain, M. S., Ghoneim, A. (2018). Energy Efficient Task Caching and Offloading for Mobile Edge Computing. *IEEE Access*, *6*(c), 11365–11373.

Haykin, S. (2005, feb). Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, *23*(2), 201–220.

Hong, Z., Wang, Z., Cai, W., Leung, V. C. (2017). Connectivity-aware task outsourcing and scheduling in D2D networks. *2017 26th International Conference on Computer Communications and Networks, ICCCN 2017*.

Hou, X., Li, Y., Chen, M., Wu, D., Jin, D., Chen, S. (2016). Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures. *IEEE Transactions on Vehicular Technology*, *65*(6), 3860–3873.

Hsing, T. R., Lau, V. K. N., Chiang, M. (2017). *FOG FOR 5G AND IoT WILEY SERIES ON INFORMATION.*

Hu, J., Heng, W., Li, X., Wu, J. (2017). Energy-Efficient Resource Reuse Scheme for D2D Communications Underlaying Cellular Networks. *IEEE Communications Letters*, *21*(9), 2097–2100.

Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., Young, V. (2015). Mobile Edge Computing A key technology towards 5G. *ETSI White Paper No. 11 Mobile*(11), 1–16.

Huynh, D. T., Chen, M., Huynh, T. T., Hai, C. H. (2017). Energy consumption optimization for green Device-to-Device multimedia communications. *Future Generation Computer Systems*.

Huynh, D. T., Wang, X., Duong, T. Q., Vo, N. S., Chen, M. (2018). Social-aware energy efficiency optimization for device-to-device communications in 5G networks. *Computer Communications*, *120*(August 2017), 102–111.

Jaffry, S., Hasan, S. F., Gui, X., Kuo, Y. W. (2017). Distributed device discovery in ProSe environments. *IEEE Region 10 Annual International Conference, Proceedings/-TENCON*, *2017-Decem*, 614–618.

Kai, C., Li, H., Xu, L., Li, Y., Jiang, T. (2018, apr). Energy-Efficient Device-to-Device Communications for Green Smart Cities. *IEEE Transactions on Industrial Informatics*, *14*(4), 1542–1551.

Khan, S., Parkinson, S., Qin, Y. (2017). Fog computing security: a review of current applications and security solutions. *Journal of Cloud Computing*, *6*(1).

Kim, Y., Kwak, J., Chong, S. (2018, feb). Dual-Side Optimization for Cost-Delay Tradeoff in Mobile Edge Computing. *IEEE Transactions on Vehicular Technology*, *67*(2), 1765–1781.

Ko, H., Lee, J., Pack, S. (2018). Spatial and Temporal Computation Offloading Decision Algorithm in Edge Cloud-Enabled Heterogeneous Networks. *IEEE Access*, *6*, 18920–18932.

Kolmogorov, V. (2008). Blossom V : A new implementation of a minimum cost perfect matching algorithm 1 Introduction 2 Background : Edmonds ' blossom algorithm. *Math. Prog. Comp.*, *1*, 1–14.

Kwak, J., Kim, Y., Lee, J., Chong, S. (2015). DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems. *IEEE Journal on Selected Areas in Communications*, *33*(12), 2510–2523.

Lai, W. K., Tang, C. L. (2013). QoS-aware downlink packet scheduling for LTE networks. *Computer Networks*, *57*(7), 1689–1698.

Larmo, A., Lindström, M., Meyer, M., Pelletier, G., Torsner, J., Wiemann, H. (2009). The

LTE Link-Layer Design. *IEEE Communications Magazine*, *47*(4), 52–59.

Lecompte, D., Gabin, F. (2012). Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: Overview and Rel-11 enhancements. *IEEE Communications Magazine*, *50*(11), 68–74.

Li, J., Li, X., Gao, Y., Gao, Y., Zhang, R. (2017). Dynamic Cloudlet-Assisted Energy-Saving Routing Mechanism for Mobile Ad Hoc Networks. *IEEE Access*, *5*, 20908–20920.

Li, R., Song, T., Mei, B., Li, H., Cheng, X., Sun, L. (2018). Blockchain For Large-Scale Internet of Things Data Storage and Protection. *IEEE Transactions on Services Computing*, *1374*(c).

Liu, C., Natarajan, B. (2017). Power-Aware Maximization of Ergodic Capacity in D2D Underlay Networks. *IEEE Transactions on Vehicular Technology*, *66*(3), 2727–2739.

Liu, L., Chang, Z., Guo, X., Mao, S., Ristaniemi, T. (2018, feb). Multiobjective Optimization for Computation Offloading in Fog Computing. *IEEE Internet of Things Journal*, *5*(1), 283–294.

Liu, Z., Zheng, Q., Xue, L., Guan, X. (2012). A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks. *Future Generation Computer Systems*, *28*(5), 780–790.

Lorenzo, B., Garcia-Rois, J., Li, X., Gonzalez-Castano, J., Fang, Y. (2018, jan). A Robust Dynamic Edge Network Architecture for the Internet of Things. *IEEE Network*, *32*(1), 8–15.

Lyu, X., Tian, H., Jiang, L., Vinel, A., Maharjan, S., Gjessing, S., Zhang, Y. (2018). Selective Offloading in Mobile Edge Computing for the Green Internet of Things. *IEEE Network*, *32*(1), 54–60.

Lyu, X., Tian, H., Ni, W., Zhang, Y., Zhang, P., Liu, R. P. (2018). Energy-Efficient Admission of Delay-Sensitive Tasks for Mobile Edge Computing. *IEEE Transactions on Communications*, *66*(6), 2603–2616.

Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K. B. (2017). A Survey on Mobile Edge

Computing: The Communication Perspective. *IEEE Communications Surveys and Tutorials*, *19*(4), 2322–2358.

Marti, S., Giuli, T. J., Lai, K., Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, 255–265.

Meng, W., Tischhauser, E. W., Wang, Q., Wang, Y., Han, J. (2018). When intrusion detection meets blockchain technology: A review. *IEEE Access*, *6*, 10179–10188.

Meng, Y., Jiang, C., La, Q. D., Quek, T. Q., Ren, Y. (2017). Dynamic social-aware peer selection scheme for cooperative device-to-device communications. *IEEE Wireless Communications and Networking Conference, WCNC*.

Morley, J., Widdicks, K., Hazas, M. (2018). Digitalisation, energy and data demand: The impact of Internet traffic on overall and peak electricity consumption. *Energy Research and Social Science*, *38*(February), 128–137.

Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., Polakos, P. A. (2018). A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. *IEEE Communications Surveys and Tutorials*, *20*(1), 416–464.

OpenFog. (2017). OpenFog Reference Architecture for Fog Computing. *OpenFogConsortium*(February), 1–162.

Peng, M., Yan, S., Zhang, K., Wang, C. (2016). Fog Computing based Radio Access Networks: Issues and Challenges. *IEEE Network*(July 2016), 21.

Prasad, A., Kunz, A., Velev, G., Samdanis, K., Song, J. (2014, dec). Energy-Efficient D2D Discovery for Proximity Services in 3GPP LTE-Advanced Networks: ProSe Discovery Mechanisms. *IEEE Vehicular Technology Magazine*, *9*(4), 40–50.

Pu, L., Chen, X., Xu, J., Fu, X. (2016). D2D Fogging: An Energy-Efficient and Incentive-Aware Task Offloading Framework via Network-Assisted D2D Collaboration. *IEEE Journal on Selected Areas in Communications*, *34*(12), 3887–39014.

Qing, L., Zhu, Q., Wang, M. (2006). Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Computer Communications*, *29*(12), 2230–2237.

Reznik, A., Arora, R., Cannon, M., Cominardi, L., Featherstone, W., Frazao, R., . . . Zheng, Z. (2017). ETSI White Paper #20: Developing Software for Multi-Access Edge Computing. *Etsi*(20). Retrieved from `www.etsi.org`

Robert, J. M., Otrok, H., Chriqi, A. (2012). RBC-OLSR: Reputation-based clustering OLSR protocol for wireless ad hoc networks. *Computer Communications*, *35*(4), 487–499.

Roh, W., Seol, J.-y., Park, J., Lee, B., Lee, J., Kim, Y., . . . Aryanfar, F. (2014, feb). Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results. *IEEE Communications Magazine*, *52*(2), 106–113.

Sabella, D., Vaillant, A., Kuure, P., Rauschenbach, U., Giust, F. (2016). Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things. *IEEE Consumer Electronics Magazine*, *5*(4), 84–91.

Salman, M. I., Mansoor, A. M., Jalab, H. A., Sabri, A. Q. M., Ahmed, R. (2018, jan). A Joint Evaluation of Energy-Efficient Downlink Scheduling and Partial CQI Feedback for LTE Video Transmission. *Wireless Personal Communications*, *98*(1), 189–211.

Satria, D., Park, D., Jo, M. (2017). Recovery for overloaded mobile edge computing. *Future Generation Computer Systems*, *70*, 138–147.

Singh, S., Chiu, Y. C., Tsai, Y. H., Yang, J. S. (2017). Mobile Edge Fog Computing in 5G Era: Architecture and Implementation. *Proceedings - 2016 International Computer Symposium, ICS 2016*, 731–735.

Singh, S., Malik, A., Kumar, R. (2017). Energy efficient heterogeneous DEEC protocol for enhancing lifetime in WSNs. *Engineering Science and Technology, an International Journal*, *20*(1), 345–353.

Soyata, T., Muraleedharan, R., Funai, C., Kwon, M., Heinzelman, W. (2012). Cloud-Vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. *Proceedings - IEEE Symposium on Computers and Communications*, 000059–000066.

Specification, T. (2015). Universal Mobile Telecommunications System (UMTS); LTE; Proximity-services (ProSe) User Equipment (UE) to ProSe function protocol aspects; Stage 3 (3GPP TS 24.334 version 14.0.0 Release 14). , *1*.

Tsolkas, D., Passas, N., Merakos, L. (2016). Device discovery in LTE networks: A radio access perspective. *Computer Networks*, *106*, 245–259.

Vallati, C., Virdis, A., Mingozzi, E., Stea, G. (2015). Exploiting LTE D2D communications in M2M Fog platforms: Deployment and practical issues. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, 585–590.

Vallati, C., Virdis, A., Mingozzi, E., Stea, G. (2016). Mobile-Edge Computing Come Home Connecting things in future smart homes using LTE device-to-device communications. *IEEE Consumer Electronics Magazine*, *5*(4), 77–83.

Varma, A., Prabhakar, S., Jayavel, K. (2017). Gas Leakage Detection and Smart Alerting and Prediction Using IoT. , 327–333.

Vestberg, H. (2010). Ceo to shareholders : 50 billion connections 2020. *Ericsson Press Release*, 4–6.

Virdis, A., Stea, G., Nardini, G. (2015). Simulating LTE/LTE-advanced networks with simuLTE. *Advances in Intelligent Systems and Computing*, *402*(January), 83–105.

Wang, H., Fapojuwo, A. O. (2017). A Survey of Enabling Technologies of Low Power and Long Range Machine-to-Machine Communications. *IEEE Communications Surveys and Tutorials*, *19*(4), 2621–2639.

Wang, L., Tang, H., Wu, H., Stüber, G. L. (2017). Resource Allocation for D2D Communications Underlay in Rayleigh Fading Channels. *IEEE Transactions on Vehicular Technology*, *66*(2), 1159–1170.

Xiao, Y., Niyato, D., Chen, K. C., Han, Z. (2016). Enhance device-to-device communication with social awareness: A belief-based stable marriage game framework. *IEEE Wireless Communications*, *23*(4), 36–44.

Xu, L., Jiang, C., Shen, Y., Quek, T. Q., Han, Z., Ren, Y. (2016). Energy Efficient D2D Communications: A Perspective of Mechanism Design. *IEEE Transactions on Wireless Communications*, *15*(11), 7272–7285.

Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, J.-P. S. (2002). The broadcast stormproblemin a mobile ad hoc network. *Wireless Networks*, *8*(2-3), 153–167.

Yang, L., Zhang, H., Li, M., Guo, J., Ji, H. (2018, jul). Mobile Edge Computing Empowered Energy Efficient Task Offloading in 5G. *IEEE Transactions on Vehicular Technology*, *67*(7), 6398–6409.

Yeow, K., Gani, A., Ahmad, R. W., Rodrigues, J. J., Ko, K. (2017). Decentralized Consensus for Edge-Centric Internet of Things: A Review, Taxonomy, and Research Issues. *IEEE Access*, *6*, 1513–1524.

Yi, C., Huang, S., Cai, J. (2018). An Incentive Mechanism Integrating Joint Power, Channel and Link Management for Social-Aware D2D Content Sharing and Proactive Caching. *IEEE Transactions on Mobile Computing*, *17*(4), 789–802.

Zhang, C., Sun, Y., Mo, Y., Zhang, Y., Bu, S. (2016). Social-aware content downloading for fog radio access networks supported device-to-device communications. In *2016 ieee international conference on ubiquitous wireless broadband, icuwb 2016.*

Zhang, J., Xia, W., Yan, F., Shen, L. (2018). Joint Computation Offloading and Resource Allocation Optimization in Heterogeneous Networks With Mobile Edge Computing. *IEEE Access*, *6*, 19324–19337.

Zhang, K., Leng, S., He, Y., Maharjan, S., Zhang, Y. (2018, may). Mobile Edge Computing and Networking for Green and Low-Latency Internet of Things. *IEEE Communications Magazine*, *56*(5), 39–45.

Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., . . . Zhang, Y. (2016). Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks. *IEEE Access*, *4*, 5896–5907.

Zhang, Z., Wang, L., Zhang, J. (2017). Energy efficiency of D2D multi-user cooperation. *Sensors (Switzerland)*, *17*(4).

Zhou, Z., Dong, M., Ota, K., Wu, J., Sato, T. (2014). Energy efficiency and spectral efficiency tradeoff in device-to-device (D2D) communications. *IEEE Wireless Communications Letters*, *3*(5), 485–488.

Zhou, Z., Ma, G., Dong, M., Ota, K., Xu, C., Jia, Y. (2016). Iterative Energy-

Efficient Stable Matching Approach for Context-Aware Resource Allocation in D2D Communications. *IEEE Access*, *4*, 6181–6196.