Algorithms and Techniques for Computation Offloading in Edge Enabled Cloud of Things (ECoT)–A Primer

Aliza Jamal*, Farhan Ahmed Siddiqui*, Adnan A. Siddiqui#, Nadeem Mahmood*, Muhammad Saeed*, Syed Asim Ali*

*Department of Computer Science (UBIT), University of Karachi, Karachi, Pakistan
#Department of Computing, HIET-FEST, Hamdard University, Karachi, Pakistan

Abstract
Recent Cloud of Things (CoT) applications entail continual transmission of data from the things to the Cloud, which is unable to provide desirable QoS for delay sensitive IoT applications. The uncertain variations in communication delay from the Things to the Cloud become a high-risk factor for these applications. To provide satisfactory computation performance as well as to achieve energy efficient resource allocation/optimal workload allocation in mixed edge-cloud environment for IoT, offloading is an attractive technology that offloads some parts of applications to edge or cloud server to save time and energy for IoT with constrained resources. This paper presents the advanced techniques, strategies, schemes and algorithms used for computational offloading in Edge enabled Cloud of Things named as ECoT. The paper also introduces three typical edge computing technologies, termed as fog computing, cloudlets and Mobile Edge Computing (MEC). After comparing their architecture, principles, applications and standardization efforts, fog is selected as a promising middleware technology for CoT for further algorithm analysis. Furthermore, open research challenges, related issues and future directions of offloading in ECoT paradigm is discussed in detail.

Key words:
Cloud of things, Edge computing, Computation offloading

1. Introduction

As of now, people are intimate with cloud computing (CC) and the extending Internet of Things (IoT). To attain full benefits of the IoT, it is essential to allocate ample computing and networking infrastructure to maintain quick response times and reduce delay for IoT devices and applications [1]. As a result, cloud-based technologies have always given preference over the existing one and IoT has been replaced by the newly introduced Cloud of Things (CoT) paradigm. CoT paradigm is used to produce intelligent applications and services based on the evolution of the cloud by things: (a) Sensor Event as a Service, (b) Database as a Service, (c) Sensing and Actuation as a Service, (d) Identity and Policy Management as a Service, (e) Ethernet as a Service, (f) Video Surveillance.

CoT will take dominate part in the internet's future. Similarly, one can easily perceived that CC fills some gaps of IoT (e.g., limited storage and applications over internet in terms of managing real world services more dynamically and in a distributed way) by offloading the storage and computation from IoT devices to the cloud. Mobile Cloud Computing (MCC) mitigates the limitations of storage and computation and extends the lifetimes of IoT devices [2]. Likewise, some gaps of Cloud Computing (e.g., main issue of limited scope) can be filled by IoT.

Nonetheless, due to its extensive processing and storage capacity CC is the key enabler for IoT devices and applications to manage enormous amounts of data from IoT clusters [1]. Being far from end-users, the transmission of huge amounts of data to and from cloud computers encounters severe challenges which includes low Spectral Efficiency (SE), heavy load on cloud servers, high response time, non-adaptive machine type of communication and lack of global mobility because of limited bandwidth. Consequently, in place of offloading data to the cloud servers, it may be more efficient to offload the processing, storage, and applications nearer to the data source, and edge computing proffers a promising solution to this problem.

Motivating to solve these issues, and for creating complex IoT applications edge computing offers far more vigorous information handling potential than the cloud by shifting the functionality of CC to edge devices. There are various edge computing technologies are instigating from distinct backgrounds to enhance SE, minimize delay, and contribute to immense machine type of communication [3]. Three most promising edge computing technologies are namely, fog computing, cloudlets and MEC.

Furthermore, concurrent computation offloading across different devices is extremely necessary and significant because of the finite capacity of the edge cloud and IoT's massive nature. Therefore, for each and every device, computation offloading must be resource-efficient in order to satisfy the requirements of QoS. Resource efficiency also leads to less usage of resources and as a result cloud service payment declines to the end-user [4]. Although, in existing research, resource efficient offloading is hardly investigated.

With this motivation, a novel integrated ECoT paradigm is presented in this paper, that exploits the benefits of the IoT,

Manuscript received June 5, 2019
Manuscript revised June 20, 2019

CC and edge(fog) in a unified prototype as in Figure 1. The contributions of this paper are summarized as follows:

- Section 2: Presents a general review of our proposed problem and its background.
- Section 3: Recalls the fundamental concepts of Edge Computing technologies and overview the characteristics essential for their integration with CoT.
- Section 4: Describes the various criteria, strategies, techniques and technologies used to offload tasks in ECoT environment.
- Section 5: Provides a comprehensive picture of existing algorithms, research methodologies and prevailing solutions.
- Section 6: Classify open research challenges that must be tackled effectively to progress tasks' offloading reliability, efficiency, and performance in Edge enable CoT.
- Section 7: Concludes our research work and provide future direction of computation offloading.

2. Problem Statement

Limitations of the cloud model are giving birth to a newly developed computing paradigm named as edge computing which intends to process data in close proximity of the network edge to minimize Internet traffic. Since, at the edge tier the servers are not as powerful as in the cloud, therefore it is necessary to maintain a balance to process data in between the cloud and the edge (e.g., fog) [5]. When all data is analyzed and computed in the cloud only, two major problems encountered inevitably are increased latency and wasted resources of distributed datacenters, cloudlets, and mobile edge nodes sited at the edge of network. Computation offloading confers an approach to solve this problem. It migrates intensive computational tasks to other rich computing resources to reduce energy consumption and enhance the processing capacity of the IoT or edge nodes. It can take place between edge devices, sensors, IoT nodes, or fog nodes. However, unpredictable operating environment is one of the most noteworthy challenges towards real world application of computation offloading. Network conditions, application characteristics and platform differences are some sources of variation. The gains achieved through the practical use of computation offloading can be hurt due to the instability of network's bandwidth. Likewise, diversity in application workload on the servers at cloud or on devices can reduce the potential gains from offloading [6]. To enable applications to offload efficiently, the multiple resource allocation, including computation, spectrum, and energy resource as well as user scheduling, plays a quite important role in future networks. However, due to the constrained computing, storage and radio resource of Fog (edge) nodes (FNs), the multiple resource allocation would be a fundamental problem and it should be well investigated.

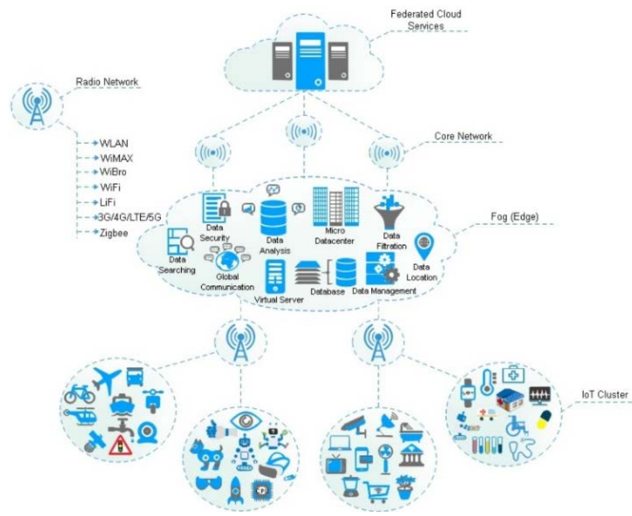

Fig. 1 The ECoT architectural paradigm

3. Edge Computing Technologies

Edge computing refers to the processing and analysis of data within the sufficient proxemics to the source of data collection. In this way, computation and load of data storage is taken from cloud and put to the edge servers. A device or thing, instead of sending all of its data to the cloud, can:

- Become its own mini data center and process data itself.
- Offload data to a nearby resource-rich device, e.g., a computer, a micro data center, a gateway networking device, for analysis.

It is predicted by business gurus that over 5.6 billion devices will make use of edge computing in industries such as energy, manufacturing and transportation. It solves various problems concerning the transmission of data for IoT, such as lessen load on networks and cost of management, latency, protection of data, and unforeseen damage recovery. As there are three main edge computing technologies exist for IoT, it is essential to emphasize the differences among them with reference to their architecture, development and background, as presented in Table 1.

3.1 Fog Computing

The term "Fog" was coined by Cisco with the analogy of real-life fog that conveys the idea of bringing the advantages of the cloud nearer to the ground (data source) [9]. It is defined by OpenFog Consortium [10] "Fog computing is a system-level horizontal architecture that distributes resources and services of storage, computing, networking and control anywhere along the continuum from Cloud to Things."

The main thing drawing difference between fog and cloud is the proximity of the fog to end users i.e. cloud is located within internet whereas fog resides at the network edge. As a result, the latency of data transmission is reduced from IoT to offloaded servers. It takes away data, services, applications, and decision making from concentrated nodes to extremes of a network [7].

3.2 Mobile Edge Computing

MEC is pivotal technology, recognized for its key enable vertical solutions for IoT to reduce network congestion and make application's performance better. MEC provides high bandwidth, and low jitter, proximity, delay and location awareness [8]. It is implemented based on a virtualized platform that leverages recent advancements in software-defined networks (SDN), information-centric networks (ICN) and network functions virtualization (NFVMEC is the key enabler technology for the fifth generation (5G) wireless systems as it enables the computing services at the edge of wireless cellular networks [9].

3.3 Cloudlets

A team at Carnegie Mellon University (CMU) developed the Cloudlets as a research project [10][11]. Cloudlets also represents the middle tier of the 3-tier architecture (IoT-cloudlet-cloud). Cloudlets are aimed to deal with the issues of jitter, wide area network latency, and packet loss as suffered by mobile cloud computing. Application execution, however, will be similar; the IoT application will act as a thin client that offloads significant computations to the nearest cloudlet to save battery and other resources. The cloudlet is completely transparent to the users, and in the case, that no cloudlet is available, the device adapts to either rely on a distant cloud provider or attempt to use its own resources.

A VM-Based cloudlet architecture for offloading computations faces several drawbacks if it is intended to be used for IoT. First, higher synchronization efforts are necessary to preserve a consistent state between the mobile device and the VMs that handle the computations that are being offloaded. Second, as the number of devices that leverage the cloudlet increases problems related to scalability become evident, and the deployment of numerous heavyweight virtual machine images grows unpractical.

These concepts are all essentially the same. All of them utilize "decentralize computing," transferring resources and services from the network core to the network edge to meet the of multiple IoT applications concurrently. Since edge and fog are often used interchangeably and fog computing provides much more feasibility for CoT than cloudlets or MEC therefore, this paper only focus on computation offloading algorithm for integrated fog CoT (IFCoT).

4. Offloading

Computation offloading is a technique of executing tasks on the remote server to save time and energy on resource limited IoT when there is a computing and power restricted environment. It partitions power hungry IoT applications and tasks to make use of cloud resources. The various offloading strategies, techniques, schemes, criteria and application domains are discussed below.

4.1 Offloading Strategies

One of the potential benefits of offloading is the migration of heavily computational components of an application onto a remote server or other resourceful device to diminish the execution delay caused by unavailability of processing resources on core IoT systems. There are various possibilities of offloading process in which an IoT device must consider issues from several aspects, i.e., when, what, where, and how to offload task from user device to the remote server to reduce jitter, bandwidth and energy consumption.

4.1 1 When to Offload

Offloading a task also adds extra costs, so it is not always a good choice to offload it. Therefore, the decision of when to offload depends on not only user's requirement, but also current condition. Consider an example of offloading among multiple fog nodes for load balancing. In this case, offloading is inessential if all fog nodes hold very small amount of load, and if all fog nodes are heavily overloaded, still offloading isn't beneficial to reduce the execution latency remarkedly. Offloading helps

Table 1: Comparison of fog computing, cloudlets and MEC.

Technical Aspect	Cloudlets	Mobile Edge Computing	Fog Computing
Organization	OEC supported by Huawei, Vodafone, and Intel	ETSI MEC launched by Intel, IBM, Nokia Networks, Huawei, and NTT DoCoMo	Cisco, OpenFog Consortium, Dell, Microsoft, Intel, and Princeton University
Standard consortium body	No	Yes	No
Node Device	Datacenter in a box	Server running in base stations	Routers, switches, access points, gateways
Interested business	Mobile computing	5G telecommunications	IoT
Motivation w.r.t applications	Enable delay sensitive and intensive computing mobile applications in cloudlet-based ecosystem	Enable an open Radio Access Network that deploys third party's applications at the edge of the network	Enable interoperability, security and high-performance in a multivendor ecosystem based on fog
Features on openness	OPENSTACK++ expands the functions of Openstack to support cloudlets	Operators open their networks to host innovative services and applications	The OpenFog RA provides a baseline for attaining a fog-based multivendor ecosystem

when there is a high degree of variance in the load among fog nodes [1].

4.1.2 What to Offload

One of the main problems is to identify what tasks could be offloaded and what tasks should be offloaded. The offloadable tasks can be identified manually; 1 - the offloadable tasks are identified by the developer or automatically, 2- the offloadable tasks are identified automatically by the profiler program.

4.1.3 Where to Offload

Most offloading mechanisms consider Cloud as the offloading target while still some other works emphasize the Device-to-Device (D2D) offloading scenario. Despite that the cloud can provide storage and computation capacity, the latency may constrain its usage. An alternative is to offload the tasks to an ad-hoc local network formed by D2D communication i.e. fog or mobile edge.

4.1.4 How to Offload

There are several offloading approaches that offload tasks at different level. Basically, we can classify them as fellow.

- *(a)* <u>System Level</u>: The system level offloading mechanisms move the entire operating system and all its running applications to the server. E.g. CloneCloud.
- *(b)* <u>Application Level</u>: In application level a set of applications run on the remote server such as cloud, and the client can call it as web services. E.g. DAvinCi
- *(c)* <u>Method Level</u>: In method level offloading a program is partitioned into pieces and decide which pieces should run remotely. E.g. Remote Procedure Call (RPC).

4.2 Criteria used in offloading

This section explores principals and criteria used in offloading to decide whether to go with computation offloading or it is beneficial to avoid it.

4.2.1 Load balancing

If a server's capacity of executing tasks reached at maximum, then it affects device efficiency and performance. In this case, server cannot perform additional tasks and they must be offloaded among other servers within the SP's ecosystem to maximize processor throughput, minimize task response time, optimal resource utilization, and to avoid overloading among nodes. For example, a fog micro datacenter would balance the load of incoming requests by distributing computation to multiple servers.

4.2.2 Excessive resource/computation constraint

When execution of sophisticated applications needs computation or resources above the capability of the device or when there is a power and computing limited environment, executing tasks need to be offloaded to a relatively resourceful device to obtain system performance gains.

4.2.3 Permanent or long-term storage

When small IoT device services need a massive long-term storage then it can be allocated at a resourceful device that can be employed as the extension of the storage and computing capabilities of devices.

4.2.4 Data organization and management

When a device stores data that may not be needed at present but, probably required in near future so, it needs to be saved somewhere. Therefore, it must be offloaded to another secure location like cloud or fog node. For example, with Dropbox or Google Drive services, the users can store their data in the cloud and can access anywhere, anytime when needed.

4.2.5 To meet latency requirement

Latency may increase due to network delay or more workload on the servers. In these cases, an edge node in vicinity of the accepting node need to be engaged in task offloading from the distant node to deliver the desired services with least transmission latency [11].

4.2.6 Privacy and security

Offloading may take place because of the secrecy and sensitivity user data. For example, offloading occurs in healthcare environment to move hospital's sensitive data from native devices to a private cloud for security and privacy reasons. Similarly, personal data migrates from mobile devices to a personal mobile edge cloud.

4.2.7 Affordability, feasibility, and maintenance

If computation extensive or storage related tasks require maintenance of the high-end servers, then it is more feasible to offload tasks to the SPs that serve pay-as-you-go services which avoid the hassle of device maintenance and related staff requirements.

4.2.8 Accessibility

In many cases, device that generate data or tasks isn't widely accessible from everywhere. Therefore, to access data efficiently from anyplace, anytime, it should be offloaded to a cloud or fog node.

4.3 Offloading Steps

For integration of IoT and edge cloud applications offloading plays an attractive roIe that can be divided into three steps. (a) application partitioning (b) preparation and (c) offloading decision as depict in figure 3.

4.3.1 Application Partitioning

The partitioning strategies play significant role in offloading deployment. It splits the application into offloadable and non-offloadable components to decide which components to retain on the IoT and which to offload to the remote server to satisfy delay-sensitive tasks' requirements and save energy. Automatically partitioning seems to be a better way to identify the offloadable methods, as compared to partitioning program itself because it is hard to design and generalize to all applications. The process of partitioning is known as an NP-Complete problem. Various partitioning granularity levels affect several difficulties including state transfer, object identity, compatibility, performance overhead and class uploading [10]. There are two types of partitioning to decide what tasks should be offloaded.

- *(a)* Statically: The intensive portions of computation load are partitioned, no matter how the running states and environment condition changes.
- *(b)* Dynamically: The tasks that should be offloaded are decided during the runtime and may change according to current running states and environment context.

4.3.2 Preparation

This step follows all requisite operations that offloadable components need to be used in IoT applications. For example, remote server selection, the installation and transfer of the data or code, as the start of proxy processes that receive and execute tasks on behalf of the smart IoT [10].

4.3.3 Offloading Decision

Usually a decision engine is engaged to decide what and whether components of computation should be uploaded to the server for offloading depending on the ongoing running states and network state. The QoS of an offloading process is greatly affected by the accuracy of the offloading decision making algorithm. Primarily, computation offloading decisions upshot in:

- *(a)* Local execution: Entire computation is treated on local device. The offloading to remote server or other incentive device is not performed. E.g., if offloading does not pay off or resources are not available at remote server.
- *(b)* Partial offloading: Computation is partitioned into offloadable and non-offloadable components, one processed locally at the IoT device, other offloaded to more resource-rich device (e.g., fog nodes, cloud and cloudlets) or gateway respectively.
- *(c)* Full offloading: Entire computation is offloaded and processed by the resource-rich device.

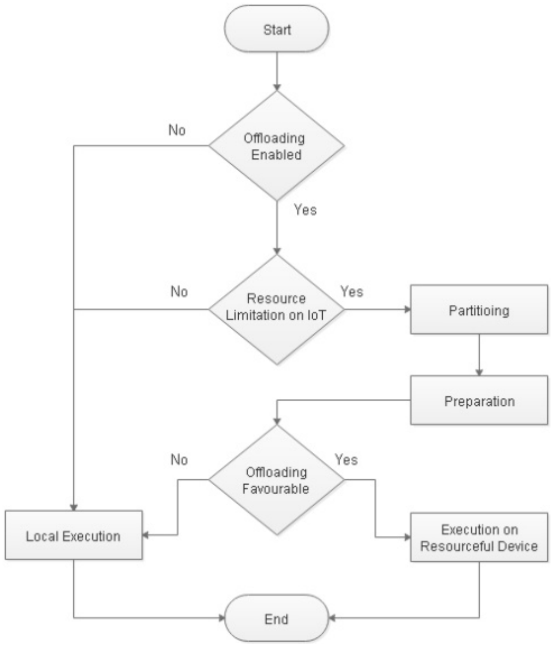

Fig. 2 Flowchart of offloading process

4.4 Offloading Techniques

Offloading techniques can be classified as:

4.4.1 Data Offloading

It is one of the most commonly used offloading techniques which may classify into four groups, i.e., data offloading through opportunistic mobile networks, small cell networks (SCNs), WiFi networks, and heterogeneous networks (HetNets). In many IoT devices, only the data are offloaded to the remote server where different recognition or prediction models are built with data analysis.

4.4.2 Method-level Code Offloading

Some computation constraint algorithms or applications perform method level code offloading in which compiler code must be analyzed to detect significant code segments for offloading which is used to lower the network congestion. ThinkAir and MAUI are popular frameworks for code offloading [12].

4.4.3 Virtual Machine Migration

This technique moves the entire operating system (OS) and all its running applications in place of offloading code or data to remote server. It makes sure that running environment of applications is similar to the one as on IoT devices. For example, CloneCloud creates a clone of the operating system (OS) and its applications on the cloud server, the offloading is implemented using process migration. Though, an expensive setup is required to run a cloned VM on the offloading system which is unsuitable for systems where devices are connected for short period only [12].

4.5 Offloading Schemes

Offloading schemes are beneficial for several application domains. It is frequently explored in existing studies with different frameworks. Table 3 gives a research overview of different application domains that have been revamped by various offloading frameworks and schemes.

Table 2: List of application domains improved by offloading schemes

Domain	Research Work
Artificial intelligence-based applications	[13]–[16]
Graphics and image processing	[17]–[20]
Computer vision	[21]
Healthcare and big data application	[22]–[25]
Virus detection applications	[26], [27]
Games applications	[28], [29]
Agriculture	[30]

5. Related Work

In recent years, computation offloading has been a ho t topic and many researchers have paid attention to offloading in 3-tier architecture (IFCoT) w.r.t fog computing. Nan et al. [5] introduced a Lyapunov optimization based algorithm named Lyapunov optimization on time and energy cost (LOTEC), to adjust tradeoff between average cost and response time to take coherent decision on computation offloading in fog enable cloud of things system. Zhao et al. [31] presented an offloading algorithm that hand-pick fog or cloud server dynamically for offloading to conserve energy as well as to satisfy certain application response time requirement. Al-muhtadi et al. [32] introduced an efficient energy aware allocation strategy for placement of application modules based on DVFS and Improved Round Robbin algorithms on fog devices in 3-tier architecture. Authors compared the proposed strategy with default edge-ward policy to allocate workload in a way that fog devices never go in overloaded or underused state. Liang et al. [33] presented Software as a Service (SaaS) based architecture named OpenPipe for virtual Radio Access Networks (RAN) and Software Defined Networks (SDN) with fog computing to offloads computation at the network edges. Researchers adopted a hybrid control model with two hierarchical control levels, to integrate fog with SDNs. Samie et al. [34] presented a novel technique for offloading process management in a local IoT network under bandwidth constraints to avoid underutilization of gateway's resources due to the fragmentation issue. Lingjun et al. [35] proposed an advanced mobile task offloading framework called D2D Fogging and Lyapunov optimization based an online task offloading algorithm to allocate the communication and computation resources among each other via the control assistance by the network operators dynamically. Deng et al. [36] introduced a framework to tackle tradeoff in power consumption delay and to coin the workload allocation problem in the fog enabled cloud computing system. Du et al. [37] studied the mixed-integer non-linear programming model to reduce the maximal weighted cost of energy consumption and latency in a mixed cloud-fog system by jointly optimizing the allocation of computation resources and offloading decisions. To solve this NP-Hard problem authors proposed Computation Offloading and Resource Allocation algorithm (CORA) based on Semidefinite Relaxation and Randomization (SDR) to obtain offloading decisions and used Lagrangian dual decomposition and fractional programming theory to device Bisection Method for Computation Resource Allocation (BCRA). We have summarized the relevant research papers related to computation offloading approaches and their respective results in the form of table.

Table 3: Comparison of research papers concentrating computation offloading approaches and algorithms.

	Offloding Type	**Objective**	**Proposed solution**	**No. of UE offloading**	**Evaluation Method**	**Results**
[5]	Full	Minimize energy consumption, monetary cost and delay	LOTEC	Multi	Theoretical verification, Discrete Event Simulations	LOTEC has better optimization ability compared with existing offloading schemes
[31]	Full	Minimize energy consumption and delay	Optimal energy consumption algorithm design	Single	Theoretical verification, Simulations using MATLAB	As compared to cloud, offloading to fog reduces up to 87% cost and energy consumption
[32]	Full	Efficient Resource Allocation to reduce energy consumption of Fog devices.	Energy-aware allocation of application modules	Multi	Simulation using iFogSim	Reduces end-to-end latency, energy consumption up to 2.72% and 1.6% as compared to cloud-only and fog-default environment respectively
[33]	Full	SDN flow	OpenPipe: fog-based architecture for SDN and virtualized RANs	Multi	Lab demo of the proposed network architecture	OpenPipe makes the underlying infrastructure transparent to applications.
[34]	Partial	Minimize energy consumption, latency	Iterative BW Allocation on the Gateway	Multi	Theoretical verification, Simulations	Battery life IoT devices is extended approximately 1 or 1.5 hour and utilization of gateway's bandwidth y 40%
[35]	Full	Minimize energy consumption, Incentive	Lyapunov based online task offloading algorithm	Multi	Simulations using Opportunistic Network Environment	Introduced algorithm reduces energy up to 23%, 30%, and 18% over greedy, random, and reciprocal schemes, respectively
[36]	Partial	Tradeoff between transmission latency and energy consumption	Generalized Benders Decomposition (GBD), Hungarian Algorithm	Multi	Theoretical verification, Simulations obtained by MATLAB	Proposed scheme reduces transmission latency and power consumption
[37]	Partial	Cost conservation and resource allocation	CORA, BCRA, Iterative and Suboptimal Power and Bandwidth Allocation Algorithm	Multi	Theoretical verification, Monte Carlo simulation obtained by MATLAB	Improve performance in terms of cost conservation, radio bandwidth and transmission power
[38]	Partial	Improvement in IoT performance measures	Static Mixed Nash Equilibrium (SM-NE) algorithm	Multi	Theoretical verification, simulations	Presented algorithm enhance system performance and provide a feasible solution for coordinating collaborative computation offloading with low signaling overhead.
[39]	Full	Maximize the utilities of cloud service operator	A Stackelberg game approach	Multi	Theoretical verification, simulations	Offloading ratio increases with the incentive
[40]	Full	Incentive, computation	IoT Worker-Device Selection algorithm	Multi	Simulations	Approximately 66% energy consumption is reduced as compared to without offloading scheme

[41]	Full	Improve users' QoE	EDFPC, EDF-LAT, and CS-LAT algorithms.	Multi	Simulations	Users' QoE enhanced by 90% for up to 4 users per small cell, high latency gain and reasonable power consumption.
[1]	Partial	Reduce the service delay for IoT nodes	Delay-minimizing collaboration and fog offloading policy	Multi	Theoretical, verification, simulations	Overall service delay is reduced from 150 to 117 ms and 60 to 18 ms for heavy and light processing tasks respectively
[42]	Partial	Maximizing quality of experience (QoE).	Joint optimal pricing and task scheduling algorithm	Multi	Theoretical, verification, simulations	Significantly, 20% more reduces the computation delay and enables low-latency fog computing services for delay-sensitive IoT applications.
[12]	Full	Decision Support for Computational Offloading	A novel probing approach	Single	Simulations	Proposed approach attained up to 85.5% accuracy for predicting the runtime performance of unknown services
[43]	Full	Multiple resource allocation, including computation, spectrum, and energy resource.	Hybrid computation offloading approach to optimize the offloading decision and multiple resource allocation	Multi	Monte Carlo simulation	Proposed approach achieves a better performance compared with existing schemes
[44]	Partial	Reduce the energy consumption and shorten the latency	Energy and latency optimal partial computation offloading	Multi	Theoretical verification, simulations	Reduced energy consumption up to 36% compared to partial offloading

6. Offloading Research Challenges

Computation offloading in ECoT paradigm faces several challenges that have to be tackled in the future as mentioned below.

6.1 Resource Allocation and Scalability

One of the most noteworthy challenge associated with offloading is how to allocate the accurate amount of resources reasonably and efficiently, desired at the task executing location i.e. local device or edge cloud. Computation resources that are shared by different devices and nodes, need to be allocated elaborately for a better performance which is an NP-hard problem in IFCoT paradigm. For instance, consider a case in which there are more resources. It is obvious that some may be under-utilized. Likewise, consider other case in which there are few resources, then offloading may occur frequently. Therefore, when the resources shouldn't be raised and give preference to offloading, and when and how much the resources should be scaled up to keep away from offloading is the tradeoff that remains a challenge [11].

6.2 High Software Quality

Since partitions of offloading framework and differential users' inputs increase the CPU load on IoT and reduce system performance, an excessive quality of software is required in order to optimize partitioning frameworks and enhance users' QoE respectively.

6.3 Scalability of Application

Scalability of an application depends upon capability of running algorithms to handle requests generated by users interacting with the system. Offloading process involves a great number of assorted parameters e.g., energy and bandwidth consumption, load balancing, latency and QoE etc. Therefore, scaling up the application without affecting offloading process w.r.t above mentioned parameters, also considered as a significant challenge.

6.4 Service Level Agreement

When critical IoT applications performs offloading, ensuring that service level agreement (SLA) isn't breached becomes an essential challenge. Trust on the SP, knowledge about SLAs monitoring, matching and infraction are fundamental facets that must be pondered in offloading process.

6.5 Security and Privacy of Data and User

Offloading exacerbates the security of data and privacy of users. In the offloading framework, the threat of interruption, interception, modification and fabrication of data increases with rise of the data hops count and multiple nodes. Sometimes misuse of sensitive data may lead to crucial loss, for instance manufacturing failures in an industrial environment, threat to life in a healthcare environment and accidents in an Intelligent Transporting System (ITS). Therefore, vigorous and efficient data security measures and a trustworthy partitioning and offloading framework is required to guarantee the trustworthiness of the CoT computing environment [45].

6.6 Integration and Interoperability

Interoperability challenges increases in 3-tier communication (i.e., integration of IoT, cloud and fog) since a vast heterogeneous networked of devices, platforms, OS and services involve in IoT applications[46]. For example, different storage techniques employed by different IoT middleware and cloud SP may have different synchronization, compression, privacy and security mechanisms that affect the size and duration of synchronization of the stored data [11].

6.7 Energy Consumption Trade-Off

Sometimes it becomes a tradeoff that whether to go for offloading or not because computation offloading itself consumes some amount of energy and bandwidth. Hence, energy consumption becomes the most unpromising challenge of offloading. An effective decision needs to be taken to ascertain whether the offloading is beneficial or unprofitable in terms of computation latency and energy conservation. For example, some tasks swiftly voids battery capacity on both the things and the gateway tier limiting the continuous operation of system up to 24 hours [11].

6.8 Incentives for Offloading Service

Designing a security-aware incentive mechanism of offloading has become an intensive challenge in the literature. Some offloading services become liabilities somehow, therefore one should amend the entity that perform the offloading with some rational incentives that must be devised to encourage selfish users to participate in offloading.

6.9 Monitoring

In order to enhance performance measures, SLA's, resource management, security, privacy and troubleshooting, monitoring plays a vital role in CoT offloading environment. While offloading to different entities, monitoring improves the delivery, quality and fault analysis of the service. Monitoring of the overall offloading process often influenced by many factors such as volume, variety, and velocity characteristics of IoT thus, there is a need of explicit and well-proven monitoring strategies.

7. Conclusion

This paper surveys the vision of edge computing technologies as an essential ingredient of the integration of cloud computing and IoT and named this architecture as ECoT. With the goal of reducing energy consumption, service delay, jitter, saving time, and increasing bandwidth, this paper deeply explores computation offloading criteria, decisions, strategies, application domains, algorithms and respective challenges in ECoT. With the help of this paper, the researchers get knowledge to explore edge enabled Cloud of Things in depth. As future work, authors of this paper are intended to apply above algorithms in Wireless Body Area Networks (WBANs) supporting smart healthcare applications. Further, the applications of CoT paradigm and above mentioned algorithms and techniques may be applied for better results.

References

[1] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On Reducing IoT Service Delay via Fog Offloading," in IEEE Internet of Things Journal, 2018, vol. 5, no. 2, pp. 998–1010.
[2] X. Lyu et al., "Selective Offloading in Mobile Edge Computing for the Green Internet of Things," IEEE Netw., vol. 32, no. 1, pp. 54–60, 2018.
[3] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: a primer," Digit. Commun. Networks, vol. 4, no. 2, pp. 77–86, 2018.
[4] X. Chen, Q. Shi, L. Yang, and J. Xu, "ThriftyEdge: Resource-Efficient Edge Computing for Intelligent IoT Applications," IEEE Netw., vol. 32, no. 1, pp. 61–65, 2018.
[5] Y. Nan et al., "Adaptive Energy-Aware Computation Offloading for Cloud of Things Systems," IEEE Access, vol. 5, pp. 23947–23957, 2017.
[6] A. Bhattacharya and P. De, "A survey of adaptation techniques in computation offloading," Journal of Network and Computer Applications, vol. 78, no. November 2016. Elsevier, pp. 97–115, 2017.
[7] A. Munir, P. Kansakar, and S. U. Khan, "IFCIoT: Integrated Fog Cloud IoT," IEEE Consum. Electron. Mag., vol. 6, no. 3, pp. 74–82, 2017.
[8] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," Comput. Networks, vol. 130, no. 2018, pp. 94–120, 2018.
[9] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Communications Surveys and Tutorials, vol. 19, no. 4. pp. 2322–2358, 2017.
[10] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: Issues and challenges," Applied Computing and Informatics, vol. 14, no. 1. King Saud University, pp. 1–16, 2018.
[11] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," Futur. Gener. Comput. Syst., vol. 87, pp. 278–289, 2018.
[12] C. Meurisch, J. Gedeon, T. A. B. Nguyen, F. Kaup, and M. Mühlhäuser, "Decision support for computational offloading by probing unknown services," in 2017 26th International Conference on Computer Communications and Networks, ICCCN 2017, 2017.
[13] J. Zheng, Q. Zhang, S. Xu, H. Peng, and Q. Wu, "Cognition-based Context-aware Cloud Computing for Intelligent Robotic Systems in Mobile Education," IEEE Access, vol. PP, no. c, p. 1, 2018.

[14] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," vol. XX, pp. 1–10, 2018.
[15] M. Aiello, Y. Yang, Y. Zou, and L.-J. Zhang, "Erratum to: Artificial Intelligence and Mobile Services – AIMS 2018," Lect. Notes Comput. Sience, Springer, vol. 10970, pp. E1–E1, 2018.
[16] Z. Wang, Y. Cui, and Z. Lai, "A First Look at Mobile Intelligence: Architecture, Experimentation and Challenges," IEEE Network, no. May, pp. 1–6, 2019.
[17] D. Ho, G. S. Park, and H. Song, "Game-Theoretic Scalable Offloading for Video Streaming Services over LTE and WiFi Networks," IEEE Trans. Mob. Comput., vol. 17, no. 5, pp. 1090–1104, 2018.
[18] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards Privacy-Preserving Content-Based Image Retrieval in Cloud Computing," IEEE Trans. Cloud Comput., vol. 6, no. 1, pp. 276–286, 2018.
[19] Z. Qin, J. Weng, Y. Cui, and K. Ren, "Privacy-Preserving Image Processing in the Cloud," IEEE Cloud Computing, no. April, pp. 48–57, 2018.
[20] C. Bailas, M. Marsden, D. Zhang, N. E. O'connor, and S. Little, "Performance of video processing at the edge for crowd-monitoring applications," in IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings, 2018, vol. 2018–Janua, pp. 482–487.
[21] J. Li, Z. Peng, B. Xiao, and Y. Hua, "Make smartphones last a day: Pre-processing based computer vision application offloading," in 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2015, 2015, pp. 462–470.
[22] P. Sanabria, J. I. Benedetto, J. Navon, and C. Poellabauer, "Code Offloading Solutions for Audio Processing in Mobile Healthcare Applications : A Case Study," pp. 117–121, 2018.
[23] M. Z. Uddin, "A wearable sensor-based activity prediction system to facilitate edge computing in smart healthcare system," J. Parallel Distrib. Comput., vol. 123, pp. 46–53, 2019.
[24] M. G. R. Alam, M. S. Munir, M. Z. Uddin, M. S. Alam, T. N. Dang, and C. S. Hong, "Edge-of-things computing framework for cost-effective provisioning of healthcare data," J. Parallel Distrib. Comput., vol. 123, pp. 54–60, 2019.
[25] A. A. Mutlag, M. K. Abd Ghani, N. Arunkumar, M. A. Mohammed, and O. Mohd, "Enabling technologies for fog computing in healthcare IoT systems," Futur. Gener. Comput. Syst., vol. 90, pp. 62–78, 2019.
[26] D. Deyannis, R. Tsirbas, G. Vasiliadis, R. Montella, S. Kosta, and S. Ioannidis, "Enabling GPU-assisted Antivirus Protection on Android Devices through Edge Offloading," in The 1st International Workshop on Edge Systems, Analytics and Networking, 2018, pp. 13–18.
[27] W. Zhang, Y. Wen, and X. Zhang, "Towards Virus Scanning as a Service in Mobile Cloud Computing: Energy-Efficient Dispatching Policy under N -Version Protection," IEEE Trans. Emerg. Top. Comput., vol. 6, no. 1, pp. 122–134, 2018.
[28] R. S. Schmoll, S. Pandi, P. J. Braun, and F. H. P. Fitzek, "Demonstration of VR / AR offloading to Mobile Edge Cloud for low latency 5G gaming application," in CCNC 2018 - 2018 15th IEEE Annual Consumer Communications and Networking Conference, 2018, vol. 2018–Janua, pp. 1–3.
[29] F. Messaoudi, A. Ksentini, and P. Bertin, "Toward a mobile gaming based-computation offloading," in IEEE International Conference on Communications, 2018, vol. 2018–May, pp. 1–7.
[30] M. Roopaei, P. Rad, and K.-K. R. Choo, "Cloud of Things in Smart Agriculture: Intelligent Irrigation Monitoring by Thermal Imaging," IEEE Cloud Comput., vol. 4, no. 1, pp. 10–15, 2017.
[31] X. Z. B, L. Zhao, and K. Liang, Quality, Reliability, Security and Robustness in Heterogeneous Networks, vol. 74. 2012.
[32] M. M. E. Mahmoud, J. J. P. C. Rodrigues, K. Saleem, J. Al-Muhtadi, N. Kumar, and V. Korotaev, "Towards energy-aware fog-enabled cloud of things for healthcare," Comput. Electr. Eng., vol. 67, pp. 58–69, 2018.
[33] K. Liang, L. Zhao, X. Chu, and H. H. Chen, "An Integrated Architecture for Software Defined and Virtualized Radio Access Networks with Fog Computing," IEEE Netw., vol. 31, no. 1, pp. 80–87, 2017.
[34] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, and D. Soudris, "Computation Offloading Management and Resource Allocation for Low-power IoT Edge Devices," in Internet of Things (WF-IoT), 2016, pp. 7–12.
[35] N. D. D. Collaboration, "D2D Fogging : An Energy-Efficient and Incentive-Aware Task Offloading Framework," IEEE J. Sel. Areas Commun., vol. 34, no. 12, pp. 3887–3901, 2016.
[36] C. Lai, T. H. Luan, R. Lu, H. Liang, and R. Deng, "Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption," IEEE Internet Things J., vol. 3, no. 6, pp. 1–1, 2016.
[37] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-Max Fairness Guarantee," IEEE Trans. Commun., vol. 66, no. 4, pp. 1594–1608, 2018.
[38] S. Jošilo and G. Dán, "Decentralized Algorithm for Randomized Task Allocation in Fog Computing Systems," IEEE/ACM Trans. Netw., vol. 27, no. 1, pp. 85–97, 2019.
[39] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach," Comput. Networks, vol. 129, pp. 399–409, 2017.
[40] R. Hasan, M. Hossain, and R. Khan, "Aura : An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading," Futur. Gener. Comput. Syst., vol. 86, pp. 821–835, 2018.
[41] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in IEEE Vehicular Technology Conference, 2015, vol. 2015.
[42] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: A computation offloading game," IEEE Internet Things J., vol. 5, no. 4, pp. 3246–3257, 2018.
[43] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Hybrid computation offloading in fog and cloud networks with non-orthogonal multiple access," in INFOCOM 2018 - IEEE

Conference on Computer Communications Workshops, 2018, pp. 154–159.
[44] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, “Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling,” IEEE Trans. Commun., vol. 64, no. 10, pp. 4268–4282, 2016.
[45] F. Gu, J. Niu, Z. Qi, and M. Atiquzzaman, “Partitioning and offloading in smart mobile devices for mobile cloud computing: State of the art and future directions,” Journal of Network and Computer Applications, vol. 119, no. June. Elsevier Ltd, pp. 83–96, 2018.
[46] A. Botta, W. De Donato, V. Persico, and A. Pescapé, “Integration of Cloud computing and Internet of Things: A survey,” Futur. Gener. Comput. Syst., vol. 56, pp. 684–700, 2014.