IoT Provisioning QoS based on Cloud and Fog Computing

Abstract

The wide-spread Internet of Things (IoT) utilization in almost every scope of our life made it possible to automate daily life tasks with no human intervention. This promising technology has immense potential for making life much easier and open new opportunities for newly developed applications to emerge. However, meeting the diverse Quality of Service (QoS) demands of different applications remains a formidable topic due to diverse traffic patterns, unpredictable network traffic, and resource-limited nature of IoT devices. In this context, application-tailored QoS provisioning mechanisms have been the primary focus of academic research. This paper presents a literature review on QoS techniques developed in academia for IoT applications and investigates current research trends. Background knowledge on IoT, QoS metrics, and critical enabling technologies will be given beforehand, delving into the literature review. According to the comparison presented in this work, the commonly considered QoS metrics are Latency, Reliability, Throughput, and Network Usage. The reviewed studies considered the metrics that fit their provisioning solutions.

Keywords: *IoT, QoS, Provisioning, Cloud Computing, Fog Computing, Virtualization, SDN*

I. INTRODUCTION

The growth in technological advancement has increased data generated from connected devices to the cloud. The cloud is a large data unit where computing and storing are done and made available to emphasize consumer needs [1]. The world will see a tripling of Internet-connected devices in the next decade, from 11 billion in 2019 to 30 billion by 2030 [2]. These services and software are used worldwide in various scenarios, include smart factories, intelligent farming, and cities [3]. A considerable storage size is required due to this prompt raise in data. This increase also means for data processing, a large bandwidth consumption and higher latency [4].

To enable connecting digital worlds with real worlds, the IoT has been identified as one of the enabling technologies for computing the next age. IoT applications' growth has advanced a range of fields like smart cities, smart health, connected vehicles. By 2025, the global market of IoT will reach $1567 billion, according to Statista Inc.

With this strain on the Internet today, service providers (SPs) have been between two options, either invest more in their networks or implementing stringent regulations. Both options will either lead to increase costs or not satisfying the customers. Besides, SPs are obligated to provide specific QoS according to the Service Level Agreement (SLA). That is why there is much money at stake for SPs due to the enormous excess in the numbers of devices connected to the Internet [5]. At that point, maintaining QoS while efficiently managing the network capital becomes challenging for many SPs or network operators [6].

QoS provisioning stands for the degree of quality granted to the user while carrying out a service. This definition has been receiving a significant focus over the last decades. It became a source for academia and technological solutions such as algorithms, protocols, and commercial products. However, when academia delivers a solution, either new services' criteria or growing users' standards made such a solution insufficient. For example, after thousands of contributions went into the routing area, there is still room for improvement [7].

Currently, adopting remote processing at the cloud with its first subsidiary product referred to as fog is widely agreed up on for meeting QoS requirements of IoT [8]. For this scenario, many technologies and techniques are involved such as Software Defined Network (SDN) [9- 11], Network Function Virtualization (NFV) and 5G mobile networking [12]. Moreover, due to the artificial intelligence (AI) and Machine Learning (ML) ability to solve problems and automate tasks at a

network level, they become of great interest during IoT system development [13-15]. These technologies and techniques could easy-up or complicate finding the right solution for QoS provisioning in IoT systems. For the reasons above, this work's main objective is to review the most recent studies involved proposing QoS provisioning schemes for IoT systems. The next section will provide the reader with background knowledge about key concepts in the topic at hand. The surveyed studies will then be reviewed and compared with tables summarizing the utilized techniques, QoS metrics and baselines. The paper ends with giving conclusions in the last section.

II. Background and Theory

This section gives a brief epitome about the topic key concepts to comprehend IoT's characteristics and architecture with its QoS parameters. Moreover, the introduction of critical enabling technologies will also be mentioned.

A. IoT Concept and Principles

IoT is an advanced framework leveraging modern information technology. It covers a range of technological fields, such as sensor technology, integrated circuit (IC), data transmission, automation, high-end computing, information processing and security [16]. Objects can interact with one another without human involvement in IoT. The four sections of IoT industrial chain are identification, sensing, processing and data transmission [17]. These sections utilize key technologies such as Radio-frequency identification (RFID), on-chip sensor, intelligent chip and wireless communication. For example, objects with RFID tags produce radio wave identification signal detected wirelessly by RFID reader. The reader obtains the object's information and sends it to an information network system middleware through Internet or other communication channel [18]. The object names are usually represented through Object Naming Service (ONS), while Electronic Product Code (EPC) interfaces can provide other variety of object information [19]. The system's whole operation gains support from the Internet, utilizing varieties of description languages and communication protocols. Thus, it can be said that the IoT is a combination of different physical product information services based on the Internet's construction.

B. IoT Devices

Linking computers and "things" to the Internet and other networks has been a commonplace. Technological developments such as automated teller machine (ATM), wireless sensor network (WSN), machine to machine (M2M) systems and similar connections have occurred over the years. The above does not mean that all the systems and devices listed are part of what is currently known as the IoT. IoT devices are not all connected, and not all connected devices are IoT devices. The term 'Internet of Things' is used when referring to uniquely addressable things [20]. There are several IoT definitions, and it is not easy to establish a universal definition. It depends on the approach is taken, such as the technical approach, the application approach, or the business approach. However, the IoT signifies the interconnectivity and interdependence of devices with integrated sensing, actuating, and communication capabilities [21]. A thing can sense the cyber-physical surrounding to generate outcomes which upon it actuates outcomes. Then the thing share with the cyber-physical environment the outcomes that resulted from both sensing and actuating (Fig. 1) [22]. Data in IoT is collected, analyzed, organized, and communicated through hardware, software, and software systems.

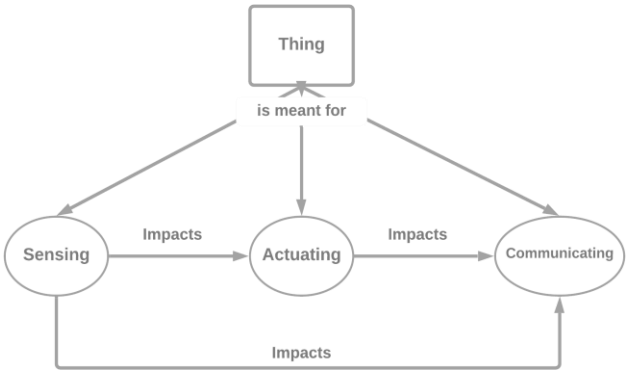

Fig. 1. Thing's duties in IoT model

C. IoT Architecture

IoT is an interconnection of intelligent things in nature and function in coordination over the network [23]. IoT's architecture concerns are network protocols, smart things, security, scalability, and interoperability through diverse devices [24]. The architecture can have three-layer as can be seen in Fig. 2 [25].

The Sensing Layer represents physically interconnected set-up monitor and maintain things remotely. Sensing is the most crucial task in the IoT system [23]. Intelligent sensor nodes and RFID are usually used for the sensing task. In this layer, RFID tags or wireless sensor nodes are designed to sense and exchange data among different things [26]. Superior technology advances IoT sensing and recognition of connecting more devices. Sensing and recognition are essential concerning networks like the IoT [27, 28].

The network layer is the second one which enables all the connected devices/things to exchange information among each other. This layer automatically discovers accessible network devices, and maps each device to a network interface. [29]. It also automatically assigns devices to their roles such as modules for deployment, work scheduling, and when needed, connecting with any other network devices. The IoT network layer's development includes dealing with network management technologies such as mobile or stationary, wireless spectrum license, security and privacy, and service recuperation [30].

The third one is the service layer. Here, IoT communicates using middleware technology that alleviates various functionalities to incorporate unstrained [31]. The main chore of the layer is to cover middleware's stipulates. Different groups industrialize these specifications. The middleware technology brings forth a cost-effective platform for IoT applications. In this platform hardware and software, schemes can be reprocessed. The service-oriented problems processed by this layer are storage administration, search engine, communications, and information transfer. Some of the service layer's components

include service discovery, service composition, trustworthiness management, and services APIs [32].

The last IoT layer is the interface layer. In IoT, unalike industries and companies usually do not adopt similar network protocols [33]. Numerous issues posed in the exchange of information between different things, result from this adaption. This issue is addressed by shortening interrelation of things. Without this layer's existence, the steady increase of IoT devices will become more challenging to communicate, operate, connect and disconnect [34]. An active interface is a set of generalization services that defines the configuration between applications and services.

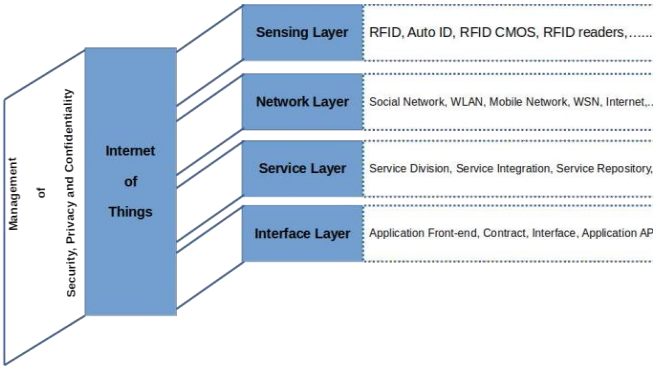

Fig. 2. Illustration of IoT Architecture

D. Cloud Computing

The technology of cloud computing provides services to the user anywhere at any time [35, 36]. Here, resources are shared all around the job for speedy servicing the user. The term "cloud" comes from the different resources pool that offers services to the end-users [37]. The “computing” term refers to the computing done based on the SLA to provide the resources with efficiency to the users [38]. The aggregation of the two terms is referred to as cloud computing. Load balancing is done to increase the utilization of resources [39, 40]. However, it is considered a significant challenge in the cloud. The challenge is to distribute the computing resources effectively among the users [41, 42]. The resources are offered on-demand to meet the SLA’s requirements. Load balancing in cloud system is done through virtualization technology to effectively handle dynamic resources [43, 44]. Cloud services provided to the users can be private, public or hybrid [45, 46]. Businesses usually uses tailored private cloud for internal purposes, while public clouds are used by individuals or organizations based on their need [47]. The integration of public and private clouds provides hybrid services to the users. The SP should guarantee the QoS for each application in the data center while achieving the server’s utilization and energy efficiency [48, 49]. The cloud developers are responsible for fulfilling the users and cloud providers requirements. Lastly, cloud computing is considered a critical enabler to meet IoT applications' demand [50].

E. Fog Computing

Cisco describes Fog Computing (FC) as a cloud expansion that spread from the center to the edge to increase performance and data analytics [51]. This expansion consists of several fog nodes (FNs) distributed in various locations to provide data services and applications [52]. The FNs are each lightweight versions of the cloud server [53]. These assets provide information and processing closer to the end-devices, usually IoT. FC provides a network of collaborating units that automate storage and processing functions in real-time [54].

Moreover, the FNs' hardware and software are customizable according to the application's requirements or environment where it will be deployed [55]. FC offers localized processing services with appropriate latency for enterprises, and because the data are not standardized, the fog analyzes them locally before transmitting them [56, 57]. It executes applications locally because of the scalability and high efficiency of its data storage system. FC is not meant to compete with cloud computing but boost and strengthen cloud computing efficacy [58]. Low latency, mobility, position awareness, scalability, security, and interaction with heterogeneous devices are supported by this technology [59].

Moreover, it reduces traffic between users and the cloud and energy usage while saving the bandwidth [60]. The FNs provide computing power, storage, and networking services for the infrastructure’s applications [61]. These nodes are heterogeneous devices that range from access points, servers, edge routers, base stations, to smart end devices [51]. Scalability of FC can be internal as adding hardware or software to the node [62], or externally by adding more nodes as required to meet service provisioning. Utilizing distributed cloud service development at each node, achieving higher scalability and reliability for the system. The node's performance is influenced by the deployment location and resources allocation among the nodes [63].

F. QoS in IoT

Connecting things to the Internet is the main aim of IoT. This aim is achieved by creating a network of things that communicate with each other [64]. As IoT devices increase, the amount of data being generated would dramatically increase [65]. The devices’ capability to provide several services at once is the reason behind this increase. As a result, various factors required for QoS prediction on the user side have been elucidated [66].

The QoS service can be referred to as a quality assurance service of network connectivity, prioritizing applications across the network [22]. QoS is a crucial enabler of IoT networking because it handles network functionality, resources and offers secure connectivity. QoS systems identify traffic in order to manage delays, bandwidth, and package loss. Delivering data rapidly and with efficiency is an essential goal of IoT and its services [67]. That is why IoT needs to deliver various services and choose the right one based on QoS requirements. These requirements or metrics are diverse in IoT system because of combining things with computing and communication. There are QoS requirements for each one of these components to meet for efficient and effective IoT system. In terms of things, the IoT devices’ QoS may implicate power consumption, coverage, the optimal number of active sensors, sensor quality, data bulk, trustiness, and mobility [68- 71]. Any of the above metrics might not be significant when measured in isolation [70]. However, there is a lot more to consider when considering the vast number of devices involved in delivering the service. For example, the

cumulative power consumption of hundreds of 0.9W sensors can make a real impact on the network's power usage. For communication, the network's QoS would include metrics, like throughput, response time, availability, capacity, repair time, delay and jitter [71- 74]. Relating to computing, the data analysis programming models within the cloud requires QoS metrics that satisfy throughput and response time. However, CPU usage, memory usage, network latency, and network bandwidth represent the cloud infrastructure layer's QoS requirements [70]. From the IoT application perspective, the main QoS requirements change according to the application's field. For example, a health-monitoring application requires privacy, security, precision, durability, responsiveness, robustness, accuracy, reliability and availability [68], [75, 76]. However, time-sensitive applications consider low latency as its highest priority requirement [68, 77], while high priority goes to network utilization and energy efficiency in less time-critical applications like building automation [68, 78].

III. Literature Review

The most recent and related academic works will be reviewed in this section and compared through tables in the next section.

Shaheen et al. [79] pointed out that the considerable distance among users and end-devices expand the number of routers' hops, resulting in rising latency and network utilization. Consequently, infrastructure provisioning in real-time is obstructed, and the QoS is reduced when using remote FNs for outsourced applications. A lightweight location-aware fog system (LAFF) is proposed in this work, using the fog head node model that keeps track of other FNs in terms of user registration and location. The proposed LAFF continuously improves QoS using a location-aware algorithm. In this work, the cloud layer used for data processing and storing for a longer duration. If the fog head struggles to offer user services, the cloud facilitates users. Fog heads are fixed and predetermined physically concerning the geographical region. According to the devised algorithm they worked to identify the user's location and the requested data type. Fog head knows the exact location of all FNs. If any nearest FN is unreachable, then the shortest path is found by implementing the k*-algorithm. The development of LAFF is conducting by using CloudSim to handle the simulation at the cloud, and iFogSim to handle FNs' events. Comparing to state-of-the-art frameworks, LAFF decreased latency by 11.01%, network utilization by 7.51% and service time by 14.8%. Furthermore, given RAM and CPU consumption, the proposed architecture surpasses intelligent FC analytical model (IFAM) and task placement on FC (TPFC) targeting IoT applications.

Rani et al. [80] mentioned that the challenges of densely deployed IoT networks are energy-effective communication, scalability and network coverage. The authors proposed a new IoT QoS infrastructure to combine fault tolerance and effective communication in the transmission of sensitive data. They worked on optimizing IoT's sensing layer in WSN using hierarchical and multi-hop communication protocols (ZSEP/LEACH/SEP and TSEP) to solve scalability in IoT. The network simulated in MATLAB has $200m^2$ area split into four areas. In each region, a sink is used in the middle that gathers data from all the region's nodes and all four sinks forward data to the IoT's base station layer. Moreover, Cluster Heads (CHs) are chosen from within each region for data transmission between the sink and the normal node. CHs are selected according to energy levels and distance, while sinks are provided with unlimited power due to IoT restrictions. The proposed methodology was compared with CBCCP, ME-CBCCP, HCR and ERP protocols. The IoT–QoS scheme took less time for transmission than Genetic HCR and ERP. However, ME-CBCCP received the lowest time among the protocols.

Quedraogo et al. [81] stated that scaling in IoT platforms can answer the QoS requirements when the traffic load is increased. However, it would increase the provisioning costs. Their alternative answer is to scale up the network for end-to-end IoT traffic control using virtualized network functions. They relied on multi-objective optimization problem for planning network function and scaling action according to considered constraints. The planner developed by the author is called QoS for NFV enabled IoT platforms (QoS4NIP). QoS4NIP uses a Genetic Algorithm (GA) to solve the multi-objective optimization problem by making a series of improvements in an iterative process. The scaling action is implemented by deploying Traffic Control Functions (TCF) as Application Network Function (ANF) or Virtualized Network Functions (VNF) on the FNs. The TCFs were evaluated by implementing Java Management Extensions (JMX)-based monitoring tools. Results reveal that TCFs implemented as VNFs use more CPU than ANFs. However, both (ANFs and VNFs) utilize the same RAM. The authors evaluate the QoS4NIP against First-Come-First-Served (FCFS), Auto-scaling (AS), QoSEF, QoSEFe in vehicle-to-network (V2N) communication scenario which implemented in Python using Platypus library. The proposed scheme provided better end-to-end latency, excluding for traffic efficiency, where the auto-scaling scheme provided lower latency figures of 160ms.

Bhandari et al. [82] argued that Routing Protocol for Low-power and Lossy network (RPL) is not efficient for multi-purposes IoT applications which aim for diverse QoS requirements in the network. The reasons for that are the following. First, the RPL default Objective Functions (OFs) depend on a single metric, leading to trade-off in routing performance. Second, while multiple metrics are supported by RPL for parent selection, metric combinations are not defined by any specific guideline. Last reason is the RPL's design is for low data traffic network, so it suffers issues in large scale networks. Therefore, the authors proposed different OFs that ensure the discrimination of QoS at the network level. Ensuring the QoS is done by virtually dividing the physical network into instances of DODAG network topology. Different OFs can be associated with each instance and routed it through the corresponding DODAG. Moreover, a new framework for parent selection is presented in this work. It relied on the approach of multi-attribute decision making to tackle the single routing-metric issue in PRL. They resolved this issue by implementing a grey relational analysis (GRA). Three separate QoS requirements classes are identified: energy consumption, reliability and latency. Cooja simulator was used to examine the effect of network scale and data traffic load on OFs'

performance in various situations. The scheme managed to show significant improvements on the QoS provision, comparing with the default RPL results. The improvements were in terms of reliability, delay, and packet loss while assuring the network's stability and minimal overhead.

Badidi et al. [83] considered selecting a fog service that ensures low latency service delivery because mapping tasks to distributed services is considered an NP-hard class problem. Thus, they presented a FC architecture based on a Fog Broker (FB) element with different scheduling algorithms. The broker receives inquiries from various applications and upon available fog services resources provide a scheduling plan for the different tasks. The application's inquiries are sent to the FB by assigning it with a collection of appropriate FNs to meet their QoS requirements. CloudAnalyst simulation tool simulated a fog cluster scenario with five FNs as proof of concept. This tool utilizes three scheduling policies to determine fog service efficiency. Three broker scheduling policies provided by CloudAnalyst, and they are Reconfigure Dynamically with Load (RDL), Optimise Response Time (ORT) and Closest Fog Node (CFN). According to the results, the average request service time was no more than 2ms for all cluster nodes and the scheduling policies. Consistent average request servicing time across cluster FNs allowed by the CFN scheduling policy. The ORT scheduling policy had the shortest time for average request servicing on almost all FNs.

Badawy et al. [84] mentioned that a dynamic service-oriented environment is essential to meet the QoS requirements while satisfying the user demands. Moreover, in the long run, IoT complex services will suffer from performance debasement and real-time adaptive sensing. Thus, relying on the Backtracking Search Optimization Algorithm (BSOA), they designed a dynamic QoS Provisioning Framework (QoPF) for service-oriented IoT. The QoPF's main objective is to optimize complex service quality in the IoT application layer through balancing service reliability with a reasonable computational time cost. Assessed, intrinsic and perceived QoS are three QoS models classified by the authors. The performance metrics used to evaluate the framework efficacy are throughput, jitter, delay time, and packet delivery ratio. NS2.35 simulator was used for evaluation, while the benchmark algorithms were GA, PSO, ACA and Differential evolution (DE). The BSOA significantly outperforms all the benchmark algorithms for all metrics except the packet delivery ratio metric against PSO algorithm.

Asad et al. [85] argued that the QoS parameters might differ between the access network and the core network. Furthermore, network-based QoS provisioning schemes usually require the end-devices to inform the network devices about their QoS requirements. To tackle the points mentioned above, the authors developed a QoS aware selection scheme for multi-radio access technologies (M-RAT). The IoT nodes with M-RAT can connect to one or more AP simultaneously. For optimal access device selection, the optimization problem runs separately at each node. The problem had four constraints. First constrain is to ensure the parameters considered for QoS provisioning satisfy the predefined thresholds. The second one is to limit the number of access devices that a node can connect to simultaneously. Constrain number three limits the number of nodes that can connect to an access device. The last one limits the workload at the access devices from all connected nodes. Mixed-integer linear programming (MILP) and binary possibilities were used to solve the problem. The Mininet emulation environment was used because it requires low computing power. The proposed scheme's performance was compared to best-SNR and maximum bandwidth selection methods in average throughput and delay. The results illustrated that the proposed scheme was closer to the ideal system than the others in terms of throughput. However, it was closer to the best-SNR selection method in terms of delay.

In another work by Asad et al. [86], the authors also worked on a QoS aware selection scheme for a M-RAT client. They found by reviewing the literature that the selection techniques are only client-centric RAT or network-centric QoS provisioning. Thus, they presented a novel hybrid end-to-end QoS provisioning technique that combines client-centric and SDN based network-centric approaches. The proposed architecture for the QoS scheme has four layers. The first layer is the end-devices layer that contains clients with M-RAT. The second one is the access layer for M-RAT access devices. The fourth layer composites from SDN controllers. The core layer is the last one where interconnecting devices such as routers are responsible for carrying data between networks. The core-QoS algorithm is implemented in the controller layer. The access-QoS algorithm implemented by the client device to select an access device by a single parameter. On the other hand, the core network's minimum cost path is calculated by the core-QoS algorithm according to the client's requirements. Mininet-WiFi network emulator was used to emulate a scenario of an indoor wireless LAN network with two WiFi APs. Moreover, two Raspberry Pi 4 equipped with 2.4GHz IEEE 802.11ac network interface cards were used in an experiment as WiFi APs, while three Android-based smartphones and a tablet used as end-devices. The emulation results showed that the proposed methods outperformed the AP selection approach based on the Received Signal Strength Indicator (RSSI) in the hardware experiment.

Ali et al. [87] considered ensuring QoS for IoT mission-critical application or services while providing wireless channel access to every connecting object. Accordingly, accommodating the demand for IoT over a limited wireless spectrum is a new challenge for communication. This work's primary focus is priority differentiation among secondary users (SUs) in cognitive Radio IoT. The authors worked on reducing high priority SU call blocking probability and increasing channel utilization efficiency. Thus, they developed a scheme for priority-based call admission and channel allocation by using traffic-aware dynamic channel reservation. First, they surveyed the available licensed channels based on the traffic patterns of its primary users. Second, for queuing analysis, the SU traffic rate is estimated by a Markov Chain model. According to it, the channels are reserved for each priority. The workflow of the scheme is as the following. Different SU application with different priorities contacts the secondary base station (SBS) which decide to block or allow the channel allocation. Here the allocation is based on priority class and the total available channel, which detected according to the primary user (PU) traffic activities probability. The proposed scheme's performance was evaluated and compared with greedy non-

priority and fair proportion schemes in call-blocking, call-dropping, channel utilization and throughput. According to simulation results, the proposed priority scheme surpasses the baseline schemes. However, the baseline schemes' figures fell between the four priority classes for the SU application suggested by the authors.

Yousefpour et al. [88] introduced a framework for QoS-aware Dynamic Fog Service Provisioning (QDFSP) and called it FOGPLAN. It is based on dynamically deploy application services on FNs, or releasing previously deployed ones on FNs to meet QoS requirements while minimizing cost. Dynamically placing fog services on either FNs or cloud servers has an essential effect on network utilization and end-to-end delay. The framework does not make any assumptions about IoT devices' capabilities. Integer Nonlinear Programming (INLP) formulation and two greedy algorithms were used to address the optimization problem of QDFSP. The proposed framework's performance evaluation was done through simulation of real work traffic traces and a Discrete-Time Markov Chain (DTMC)-based traffic generator. The asymptotic complexity was the same for both minimum-delay and minimum-cost algorithms. However, according to the results, minimum-cost is faster than the minimum-delay algorithm, particularly for more FNs and services case. Except for the optimum execution reached by INLP, minimal-delay algorithm had the lowest average operation delay and average delay violations. It was concluded that minimum delay output comes at a slower run-time rate.

Yao et al. [89] addressed the failure issue during virtual machines (VMs) renting by fog provisioning to manages tasks and reduce device cost. Scaling VMs should boost reliability and QoS, but it will increase device cost. The authors investigated reliability maximization while reducing the system cost for providing fog resources in IoT networks. They formulated an Integer Linear Programming (ILP) problem. However, it suffered from complex computation. Thus, another algorithm was designed to accomplish sub-optimal solutions with improved time efficiency. Fog resource provisioning formulated as a multi-objective problem, then converted into a single-objective problem by weighted sum method. The principle here is that the different computing tasks of IoT devices are offloaded to the FN. Then the FN schedules these tasks to be processed on several VMs. The authors designed a Modified Best Fit Decreasing (MBFD) algorithm to attain sub-optimal solutions for the scheduling problem. MBFD was simulated in MATLAB, and the outcomes were compared against the the IBM CPLEX Optimizer's optimal solution. Moreover, they benchmarked the proposed algorithm with another from a past work called (Bench), which only considered the system cost. The simulation demonstrated that MBFD provides near-optimal solutions. However, it performed similarly to the Bench algorithm in terms of reliability.

Yao et al. [90] also worked on leasing and releasing VMs by the FN in an on-demand fashion. They focused on power management to sustain stable wireless transmission rate and acceptable QoS. This work addresses jointly optimize the number of rented VMs and power management problem for system cost minimization whilst guarantee QoS requirements. The Mixed-Integer Non-Linear Programming (MINLP) to formulate the optimization problem. Then it was converted to a convex optimization problem solved by the gradient projection algorithm through relaxing its integer variables. An adequate solution is obtained by an integer recovery scheme. The proposed system architecture consists of FN connected to IoT gateway and mobile IoT devices, that move within the gateway's coverage. The proposed QoS scheme was simulated and compared with the problem's lower bound. The convex problem is solved to obtain the bound after relaxing the number of rented VMs at a given location. The comparison was also made with a Fog Provisioning Problem (FPP) scheme that selects a fixed transmission power during the connection period. According to the outcomes the proposed algorithm performed similarly to the relaxed MINLP's lower bound and surpassed the FPP scheme.

Verma et al. [91] considered the hot-spot problem in multi-hop communication among the IoT-based Wireless Sensor Network (WSN). This issue occurs when the nodes nearest to the sink node get burdened by the other nodes' traffic data. Thus, they presented two QoS provisioning-based routing protocols based on multiple WSN-based IoT sinks. The authors called them Optimized Energy and Threshold Sensitive Stable Election Protocol (O-ETSSEP), and Multiple data Sinks-based Optimized-ETSSEP (MSO-ETSSEP). They relied on energy threshold, residual energy, distance and node density variables for optimizing Cluster Head (CH) selection in both protocols. For network energy balancing, the protocols use three energy heterogeneity levels. Also, MSO-ETSSEP uses four data sinks along each square-shaped network periphery to minimize hot-spot problems by surrounding multi-hop communication. MATLAB simulations evaluated the protocols through considering multiple scenarios. The QoS provisioning performance metrics were; stability period, network lifetime, network efficiency, networks remaining energy, throughput, latency and reliability. The performance of O-ETSSEP was validated against the TSEP38 and ETSSEP protocols. MS-ETSSEP and MS-SEP were compared against the MSO-ETSSEP. The results pointed out that integrating multiple data sinks into the network improves its reliability and stability. Moreover, the observed increase in performance of the MSO-ETSSEP was related to the proposed selection of CH and it achieved enhanced stability compared to MS-ETSSEP and MS-SEP.

Srinidhi et al. [92] utilized the multi-objective optimization problem to approximate the network's outage performance and lifetime. They combined quantum particle swarm optimization (QPSO) and improved non-dominated sorting genetic algorithm (NGSA) to produce Hybrid Energy Efficient and QoS Aware (HEEQA) algorithm. The HEEQA algorithm is designed to balance the devices by tuned MAC layer parameters to reduce energy consumption. To solve the multi-objective optimization problem, NSGA was applied, while the QPSO algorithm is used to get the best suitable combination. This work stress more on finding equilibrium between network lifetime and QoS provisioning. NS-2 simulator was used to evaluate the HEEQA algorithm, which compared to the QPSO. The comparison's metrics were the maximizing residual energy, end-to-end delay, packet delivery ratio (PDR), transmission overhead, maximizing network lifetime and throughput. Tuning up of MAC layer parameters reduced energy consumption of each node in the IoT network. The HEEQA outperforms QPSO in terms of all

performance metrics. However, it could perform poorly in energy conservation when nodes are mobile with different moving speeds.

Li et al. [93] discussed that spectrum shortages contributed to the changing of spectrum use from an exclusive to a sharing mode due to the increase of wirelessly connected IoT. However, it is not easy to assure QoS while using a shared spectrum due to its unpredictable availability. Thus, the authors suggested metric that guarantee the QoS statistically by evaluating how much data can be delivered during a session period via a shared band, and called it probabilistic link capacity (PLC). A Distributionally Robust (DR) data-driven approach was developed based on the first and second-order statistics to estimate the PLC's value. The DR-PLC was formulated into a semi-definite programming problem based on the worst-case conditional-value-at-risk (CvaR) to calculate it for each case. Accordingly, a service-based spectrum aware data transmission scheme was designed to satisfy the various IoT service by allowing efficient use of different spectrum. They also proposed a network model named a cognitive capacity harvesting network (CCHN), that ease the IoT data transmissions over a shared spectrum. This architecture aimed to enhance the existent cellular network by transforming it into an ultra-dense network similar to the 5G design. It includes Macro-cell Base Station (MBS), femtocell Base Station (FBS), and Cognitive Radio Router (CRR). Finally, it was numerically evaluated and compared the PLC under different probability distribution and DR-PLC for under exact data-driven statistics or uncertain ones. According to the results, PLC and DR-PLC cannot accomplish similar confidence levels, while the gap among them becomes more extensive due to historical data fluctuations. DR-PLC provided an efficient way to insure QoS while utilizing the shared spectrum.

Khan et al. [94] considered the security of the relay nodes in multi-hop communication while assuring QoS. They suggested a secured communication scheme that is QoS-aware (QoS-IoT). The scheme is based on a Sybil attack detection mechanism for identifying compromised nodes and their counterfeit identities. The scheme selects an optimal contention window (CW) after detection to efficiently utilize the available bandwidth and achieve per-flow fairness. The detection mechanism is a signal-print based on the node's obtained signal strength information to detect malfunctioning nodes. The size of CW depends upon the actual to fair bandwidth allocation ratio. The Binary Exponential Back-off (BEB) mechanism was used to select the optimal CW. The proposed scheme is based on the following network model. An area of $100x100m^2$ was split into smaller IoT networks, where each one dwell of static, mobile, Sybil and high-powered nodes. Thus, only delay and throughput were considered as QoS requirements because they are deeply affected by Sybil nodes' existence. The Sybil nodes block actual or genuine nodes from the use of network services with various forged identities. The network model is simulated in NS-2. The scheme was evaluated and compared with First-In-First-Out (FIFO), Round Robbin (RR) scheduling, and Cross-layer based on Utilization evaluation to Contention Window (CUCW) schemes in terms of throughput, fairness and the utilization of link. By increasing the offered load, the QoS-IoT received better fairness index compared to the other schemes. However, it performed similarly to CUCW in term of throughput. The QoS-IoT received smaller queue length by increasing the offered load than the other schemes.

Guo et al. [95] stated that queueing delay is nun-negligible in IoT applications due to the scarce edge server's computation resource. They also argued that due have workload at the edge of the network, the cloud energy consumption can be lower than in the edge servers. Therefore, to achieve green computing while providing QoS for end-users, they formulated a problem for the Delay-Based Workload Allocation (DBWA). The problem is based on optimal workload allocation between local edge, neighboring edge-servers, and the cloud to reduce energy consumption while guaranteeing the delay. A DBWA algorithm was proposed for solving the problem and it was based on the theory of Lyapunov drift-plus-penalty. The proposed scheme's network model was structured as IoT devices pushing computation jobs stochastically to a layer of edge nodes containing edge servers and edge communication infrastructures to connect to the cloud layer. The edge nodes make workload allocation decisions to offload the arrival jobs to a neighbor edge or the cloud or execute it locally. The ping-pong effect was avoided by not offloading already offloaded jobs again. The event-based simulator combines MATLAB and C++ to simulate a scenario with three IoT-devices regions, three edge nodes and the cloud. The scheme is compared with the edge-only and cloud-only offloading versions. The DBWA surpassed the other energy consumption schemes and obtained average end-to-end delay by increasing job generation rate or size.

End-to-End Delay (E2ED) estimators are significant for designing efficient QoS provisioning scheme for IoT systems. Therefore, Maslouhi et al. [96] proposed real-time evaluation metrics and addressed varying packet payload (PP) size effects in multi-hop wireless IoT networks through counting hops from source to destination. The authors considered the following four elements (Radio propagation delay, Transmission delay, Queueing delay and Signal processing delay) that contribute to the end-to-end packet delay in one direction from source to destination in their theoretical study. IP6, IP4 and ATM network protocols evaluated in terms of packet transmission delay vs packet number. Because of E2ED strongly dependent on the message size, this work concentrates on the message's average length and header. In MATLAB simulation, the IoT wireless network is considered and a single source node is transmitting packets to a single destination node across several IoT nodes. The results are compared with Ethernet's use and the speed of the Internet using fixed values. According to the results, the estimator provided reasonable estimates of payload packets, End-to-End delay and jitter. Thus, it provided valuable insight into multi-hop wireless networks' QoS provisioning.

To optimize sharing resources among IoT services, Skarlat et al. [97] presented a system model called fog landscape. It consisted of fog cells, fog colonies, and a FC management system. Fog colonies are micro data centers that are created by the accumulation of fog cells. Each fog colony has a control node that provision resources by coordinating fog cells. Also, it communicates with other colonies to coordinate extra resources if needed. The colonies connect to a middleware running in the cloud called FC management system. Also, the authors introduced the Fog Service Placement Problem (FSPP) scheme

to address the placement of IoT services on virtualized fog resources. The placement considered QoS constraints such as deadlines on the execution time of applications. FSPP was implemented as an ILP problem and solved using IBM CPLEX solver. The solution was evaluated in terms of the execution cost and QoS support. The fog landscape environment was simulated using iFogSim, and the FSPP was compared to execution in the cloud. According to the results, 70% of services were utilized when FSPP included in the fog-landscape. This lead to a 35% reduction in the execution cost comparing to the execution in the cloud. The application's deadline was not violated by the FSPP scheme, unlike the baseline approach.

Muralidharan et al. [98] mentioned a promising paradigm to handle the exponential increase in the global IoT traffic volume, called Named Data Networking (NDN). The NDN traditional version only supported PULL traffic, where interest pulls Data packets from the IoT devices. However, PULL traffic as well PUSH traffic is required by IoT applications. For effective exchange of data in IoT applications, the authors presented a hybrid PUSH-PULL Traffic (PPT) model that uses NDN's efficient qualities to amend the IoT QoS parameters. The NDN's data exchange model is altered to push data as soon as IoT devices generate it without the need to remain online and check for an inbound request. The authors define the taxonomy of the network model as three entities. The IoT devices are smart sensors that can name Data packets. IoT gateway delivers messages and works as a point for entering and exiting from a network to another one. The third entity is the NDN cache router (CR) to hold and execute the proposed PPT algorithm. A Building Management System (BMS) was considered by this work in a smart building to evaluate the proposed model's performance. The simulations implemented in Visual C/C++ and the PPT model results were compared with traditional NDN and IPv6 protocol. PPT results showed that the generated network load is 50% lower than the IPv6. This helped deliver almost 98% of the packets. Also, the PPT model was 50% higher than the IPv6 in terms of average throughput.

IV. Discussion and Comparison

The technologies and techniques used by the surveyed studies will be discussed and compared in this section. At the end of this section, three comparison tables for the reviewed studies that focused on QoS provisioning for IoT. Table I present the problems considered by the surveyed studies and the techniques used for solving them. Table II summaries the considered QoS metrics with the corresponding references. The third table includes the baseline algorithms or approach considered by the corresponding authors in their evaluation. The solutions presented by all the mentioned studies addressed their legit corresponding problems. According to the comparison table (Table I), the commonly used QoS metrics were Latency, Energy efficiency, Throughput, Availability and Reliability. However, the reviewed studies did not settle on using all the metrics mentioned in the background knowledge section. Instead, each one used the metrics that fit their provisioning solutions. Moreover, some studies introduced their metrics, usually a combination of fundamental QoS metrics [93, 94]. Some works were done on ready protocols or standards such as PRL and NDN to make them more feasible for provisioning QoS in IoT system [82, 98]. In terms of the network model, most of the reviewed studies relied on FC paradigm to propose their schemes [79, 81, 83, 88- 90, 97]. The reviewed studies also included provisioning schemes for IoT environments that needed resources allocation for NFV [81, 89, 90, 97]. These studies shared with the other ones, the necessity to solve objective optimization problems, which usually done by linear or nonlinear integer programming [86, 88]. However, others used a Markov chain model to formulate their problems [87], [88]. Towards modern communication techniques, a selective number of studies designed QoS provisioning schemes for IoT devices with M-RAT or the ability to share the spectrum [85-87, 93]. Two studies out of the reviewed studies focused on multi-hop communication, while one considered security during designing the QoS provisioning scheme [91, 94]. Finally, comparing the solutions' effectiveness presented in the reviewed papers is out of the scope of this work. However, this is difficult to do because the authors considered different baselines and QoS metrics.

TABLE I. Problems and Techniques Considered by Recent Studies that Focused on QoS Provisioning for IoT

Ref.	Problems	Techniques
[79]	The distance among users and end devices increases the number of routers/hops, resulting in higher latency and network utilization	A lightweight location-aware fog system (LAFF) based on fog head node model
[80]	The challenges of densely deployed IoT networks are energy-efficient communication, network coverage and scalability.	Optimize IoT's sensing layer in WSN using hierarchical and multi-hop communication protocols (ZSEP/LEACH/SEP and TSEP) to solve IoT's scalability.
[81]	Scaling in IoT platforms can answer the QoS requirements when the traffic load increases, but it would increase the provisioning costs.	Scaling up the network for end-to-end IoT traffic management using VNF.
[82]	RPL protocol is not efficient for multipurpose IoT applications	Virtually dividing the physical network into instances of DODAG network topology. Each instance can be associated with the different objective function.
[83]	Selecting a fog service that ensures low latency service delivery because mapping tasks to distributed services is considered an NP-hard class problem.	a FC architecture based on a fog broker element with several scheduling algorithms
[84]	In the long run, IoT complex service will suffer from performance degradation and real-time adaptive sensing.	A Dynamic QoS provisioning framework (QoPF) for service-oriented IoT based on BSOA algorithm
[85]	IoT's heterogeneous characteristic causes the QoS requirements to differ from one IoT node to another	a QoS aware selection scheme for IoT nodes with multi-radio access technologies (RAT)

[86]	Past literature focused only on network-centric QoS provisioning or client-centric RAT.	A novel hybrid end-to-end QoS provisioning technique that combines client-centric and SDN based network-centric approaches.
[87]	Accommodating the demand for IoT over a limited wireless spectrum is a new challenge for communication	A scheme for priority-based call admission and channel allocation by using traffic-aware dynamic channel reservation.
[88]	Ensuring Quality of Service (QoS) for delay-sensitive complex applications is challenging.	A framework for QoS-aware Dynamic Fog Service Provisioning (QDFSP) called FOGPLAN.
[89]	Fail issue during VMs renting by fog provisioning to manages tasks and reduce device cost.	Formulating reliability maximization while reducing the system cost to provide fog resources in IoT networks using ILP problem
[90]	The QoS may be degraded for the power limited mobile IoT devices because the conditions of the wireless channel are not consistent.	Jointly optimize how many VMs can rent and power control problems for system cost minimization while ensuring QoS requirements.
[91]	The hot-spot problem in multi-hop communication among the IoT-based Wireless Sensor Network (WSN).	Two QoS provisioning-based routing protocols based on multiple WSN-based IoT sinks. They called them Optimized Energy and Threshold Sensitive Stable Election Protocol (O-ETSSEP), and Multiple data Sinks-based Optimized-ETSSEP (MSO-ETSSEP).
[92]	Reducing energy utilization in industrial IoT network without compromising the QoS.	Combining quantum particle swarm optimization (QPSO) and improved non-dominated sorting genetic algorithm (NGSA) to produce Hybrid Energy Efficient and QoS Aware (HEEQA) algorithm.
[93]	The challenge of ensuring QoS while using a shared spectrum due to its unpredictable availability	A Distributionally Robust (DR) data-driven approach was developed based on the first and second-order statistics to estimate the value of probabilistic link capacity (PLC).
[94]	Ensuring the security of the relay nodes in multi-hop communication while assuring QoS.	a QoS-aware secured communication scheme (QoS-IoT) based on a Sybil attack detection mechanism for identifying compromised nodes and their counterfeit identities.
[95]	Achieve green computing while providing QoS for end-users is a challenge	A Delay-Base Workload Allocation (DBWA) algorithm based on Lyapunov drift-plus-penalty theory
[96]	Accurate and efficient End-to-end delay (E2ED) estimators are significant for designing efficient QoS provisioning scheme for IoT systems.	A real-time evaluation metrics and addressed varying packet payload (PP) size effects in multi-hop wireless IoT networks through counting hops from source to destination.
[97]	Optimizing sharing resources among IoT services by using FC	Fog Service Placement Problem (FSPP) scheme designed to address the placement of IoT services on virtualized fog resources
[98]	The exponential increase in the volume of global IoT traffic	A hybrid PUSH-PULL Traffic (PPT) model uses NDN's efficient qualities to amend the IoT QoS parameters.

TABLE II. QOS METRICS CONSIDERED BY RECENT STUDIES THAT FOCUSED ON QOS PROVISIONING FOR IOT

QoS metrics	Reference
Latency	[79, 81, 82, 84 - 86, 88, 91 - 97]
Network Usage	[79, 87, 91 - 94, 97, 98]
Service Time	[79, 83, 97]
RAM Consumption	[79]
CPU Utilization	[79]
Stability	[80, 91, [94]
Scalability	[80, 88]
Energy efficiency	[80, 82, 90, 93, 95]
Throughput	[81, 82, 85, 87, 91-94, 98]
Availability	[81, 86- 88, 90- 92]
Reliability	[82, 84, 87- 93, 96, 98]
Response Time	[83, 97]
jitter	[84, 96]
Device/Network Cost	[88- [90, 97]

TABLE III. EVALUATION BASELINES CONSIDERED BY RECENT STUDIES THAT FOCUSED ON QOS PROVISIONING FOR IOT

Ref.	Baselines
[79]	Intelligent FC Analytical Model (IFAM) and Task Placement on FC (TPFC) model
[80]	CBCCP, ME-CBCCP, HCR and ERP protocols.
[81]	First-Come-First-Served (FCFS), Auto-scaling (AS), QoSEF, QoSEFe
[82]	Default RPL
[83]	Optimize Response time (ORT), Closest Fog Node (CFN), and Reconfigure Dynamically with Load (RDL).
[84]	GA, PSO, ACA and Differential evolution (DE) algorithms
[85]	best-SNR and maximum bandwidth selection methods
[86]	the Received Signal Strength Indicator (RSSI) AP selection approach
[87]	Greedy non-priority and fair proportion schemes
[88]	All IoT's requests to the cloud and Static Fog approach where the services are deployed statically at the beginning
[89]	The IBM CPLEX Optimizer's optimal solution and Bench algorithm.
[90]	The problem's lower bound acquired by solving the convex problem through relaxing the number of rented VMs at a given location and fixed transmission power approach.
[91]	O-ETSSEP is performed versus the ETSSEP and TSEP38 protocols, while MSO-ETSSEP compared against MS-ETSSEP and MS-SEP.
[92]	QPSO algorithm
[93]	PLC under different probability distribution (normal, uniform and Gamma distribution)
[94]	First-In-First-Out FIFO, Round Robbin (RR) scheduling, and Cross-layer based on Utilization evaluation to Contention Window (CUCW) schemes
[95]	Edge-only and cloud-only offloading approaches
[96]	IP6, IP4 and ATM network protocols
[97]	Execution in the cloud.
[98]	Traditional NDN and IPv6 protocol

V. Conclusion

All the mentioned studies had legit problems to solve, and they addressed it with brilliant solutions. According to Table II, the commonly considered QoS metrics are Latency, Reliability, Throughput, and Network Usage. However, these studies did not settle on using all the metrics mentioned in the background knowledge section. Instead, each one used the metrics that fit their provisioning solutions. Moreover, most of the reviewed studies considered FC paradigm as their network model for the proposed schemes which required resources allocation for NFV. Finally, due to the IoT system's heterogeneous characteristics, the metrics for QoS provisioning cannot be unified. Thus, there is no one solution fits all cases. To conclude, the academic community will still have many cases to go through while new communication technologies are coming up or still in the pipeline, such as LiFi and 6G.

References

[1] B. Paharia and K. Bhushan, "Fog Computing as a Defensive Approach Against Distributed Denial of Service (DDoS): A Proposed Architecture," in 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Jul. 2018, pp. 1–7, doi: 10.1109/ICCCNT.2018.8494060.

[2] N. H. Mahmood et al., "White Paper on Critical and Massive Machine Type Communication Towards 6G," arXiv, Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.14146v2.

[3] I. Sittón-Candanedo, R. S. Alonso, S. Rodríguez-González, J. A. García Coria, and F. De La Prieta, "Edge Computing Architectures in Industry 4.0: A General Survey and Comparison," in Advances in Intelligent Systems and Computing, vol. 950, 2020, pp. 121–131.

[4] I. S. Abdulkhaleq and S. Askar, "Evaluating the impact of network latency on the safety of blockchain transactions," Int. J. Sci. Bus., vol. 5, no. 3, pp. 71–82, 2021, doi: 10.5281/zenodo.4497512.

[5] A. V. Bataev, I. Zhuzhoma, and N. N. Bulatova, "Digital Transformation of the World Economy: Evaluation of the Global and Russian Internet of Things Markets," in 2020 9th International Conference on Industrial Technology and Management (ICITM), Feb. 2020, pp. 274–278, doi: 10.1109/ICITM48982.2020.9080392.

[6] A. Constantin and I. B. Bacîș, "Performance targets and QoS requirements for the service provided to users/subscribers of public IP networks," in Advanced Topics in Optoelectronics, Microelectronics and Nanotechnologies X, Dec. 2020, no. December 2020, p. 32, doi: 10.1117/12.2570968.

[7] M. Molnár, "QoS Routing for Data Gathering with RPL in WSNs," in Advances in Intelligent Systems and Computing, vol. 1132, 2020, pp. 87–111.

[8] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, "Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges," IEEE Wirel. Commun., vol. 23, no. 5, pp. 10–16, Oct. 2016, doi: 10.1109/MWC.2016.7721736.

[9] S. Askar, "SDN-Based Load Balancing Scheme for Fat-Tree Data Center Networks," Al-Nahrain J. Eng. Sci., vol. 20, no. 5, pp. 1047–1056, 2017.

[10] F. S. Fizi and S. Askar, "A novel load balancing algorithm for software defined network based datacenters," in 2016 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom), Sep. 2016, pp. 1–6, doi: 10.1109/COBCOM.2016.7593506.

[11] G. Aziz and S. Askar, "Software Defined Network Based VANET," Nature, vol. 5, no. 3, pp. 83–91, 2021, doi: 10.5281/zenodo.4497640.

[12] P. Krishnan, S. Duttagupta, and K. Achuthan, " SDN/NFV security framework for fog-to-things computing infrastructure," Softw. Pract. Exp., vol. 50, no. 5, pp. 757‐800, May 2020, doi: 10.1002/spe.2761.

[13] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, "Artificial Intelligence Enabled Internet of Things: Network Architecture and Spectrum Access," IEEE Comput. Intell. Mag., vol. 15, no. 1, pp. 44–51, Feb. 2020, doi: 10.1109/MCI.2019.2954643.

[14] C. M. Mohammed and S. Askar, "Machine Learning for IoT HealthCare Applications : A Review," Int. J. Sci. Bus., vol. 5, no. 3, pp. 42–51, 2021, doi: 10.5281/zenodo.4496904.

[15] K. D. Ahmed and S. Askar, "Deep Learning Models for Cyber Security in IoT Networks: A Review," Int. J. Sci. Bus., vol. 5, no. 3, pp. 61–70, 2021, doi: 10.5281/zenodo.4497017.

[16] M. A. M.Sadeeq, S. R. M. Zeebaree, R. Qashi, S. H. Ahmed, and K. Jacksi, "Internet of Things Security: A Survey," in 2018 International Conference on Advanced Science and Engineering (ICOASE), Oct. 2018, no. October, pp. 162–166, doi: 10.1109/ICOASE.2018.8548785.

[17] S. I. Saleem, S. R. M. Zeebaree, D. Q. Zeebaree, and A. M. Abdulazeez, "Building smart cities applications based on IoT technologies: A review," Technol. Reports Kansai Univ., vol. 62, no. 3, pp. 1083–1092, 2020.

[18] L. M. Haji, O. M. Ahmad, S. R. M. Zeebaree, H. I. Dino, R. R. Zebari, and H. M. Shukur, "Impact of cloud computing and internet of things on the future internet," Technol. Reports Kansai Univ., vol. 62, no. 5, pp. 2179–2190, 2020.

[19] H. Aftab, K. Gilani, J. Lee, L. Nkenyereye, S. Jeong, and J. Song, "Analysis of identifiers in IoT platforms," Digit. Commun. Networks, vol. 6, no. 3, pp. 333–340, Aug. 2020, doi: 10.1016/j.dcan.2019.05.003.

[20] K. Ali and S. Askar, "Security Issues and Vulnerability of IoT Devices," Int. J. Sci. Bus., vol. 5, no. 3, pp. 101–115, 2021, doi: 10.5281/zenodo.4497707.

[21] H. Raad, Fundamentals of IoT and Wearable Technology Design. Wiley, 2020.

[22] A. Qamar, M. Asim, Z. Maamar, S. Saeed, and T. Baker, "A Quality-of-Things model for assessing the Internet-of-Things' nonfunctional properties," Trans. Emerg. Telecommun. Technol., Jun. 2019, doi: 10.1002/ett.3668.

[23] L. Da Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," IEEE Trans. Ind. Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.

[24] P. P. Ray, "A survey on Internet of Things architectures," J. King Saud Univ. - Comput. Inf. Sci., vol. 30, no. 3, pp. 291–319, Jul. 2018, doi: 10.1016/j.jksuci.2016.10.003.

[25] T. Poongodi, A. Rathee, R. Indrakumari, and P. Suresh, Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm, vol. 174. Cham: Springer International Publishing, 2020.

[26] S. Fosso Wamba, A. Anand, and L. Carter, "A literature review of RFID-enabled healthcare applications and issues," Int. J. Inf. Manage., vol. 33, no. 5, pp. 875–891, Oct. 2013, doi: 10.1016/j.ijinfomgt.2013.07.005.

[27] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor Technologies for Intelligent Transportation Systems," Sensors, vol. 18, no. 4, p. 1212, Apr. 2018, doi: 10.3390/s18041212.

[28] Z. J. Hamad and S. Askar, "Machine Learning Powered IoT for Smart Applications," Int. J. Sci. Bus., vol. 5, no. 3, pp. 92–100, 2021, doi: 10.5281/zenodo.4497664.

[29] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," IEEE Internet Things J., vol. 4, no. 5, pp. 1125–1142, Oct. 2017, doi: 10.1109/JIOT.2017.2683200.

[30] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," Comput. Commun., vol. 54, pp. 1–31, Dec. 2014, doi: 10.1016/j.comcom.2014.09.008.

[31] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A Survey," IEEE Internet Things J., vol. 3, no. 1, pp. 70–95, Feb. 2016, doi: 10.1109/JIOT.2015.2498900.

[32] M. A. A. da Cruz, J. J. P. C. Rodrigues, J. Al-Muhtadi, V. V. Korotaev, and V. H. C. de Albuquerque, "A Reference Model for Internet of Things Middleware," IEEE Internet Things J., vol. 5, no. 2, pp. 871–883, Apr. 2018, doi: 10.1109/JIOT.2018.2796561.

[33] P. Gokhale, O. Bhat, and S. Bhat, "Introduction to IoT," Int. Adv. Res. J. Sci. Eng. Technol., vol. 5, no. 1, pp. 41–44, Jan. 2018, doi: 10.17148/IARJSET.2018.517 41.

[34] C.-L. Zhong, Z. Zhu, and R.-G. Huang, "Study on the IOT Architecture and Gateway Technology," in 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Aug. 2015, pp. 196–199, doi: 10.1109/DCABES.2015.56.

[35] M. Xu, W. Tian, and R. Buyya, "A Survey on Load Balancing Algorithms for VM Placement in Cloud Computing," Concurr. Comput. Pract. Exp., vol. 22, no. 6, pp. 685–701, Jul. 2016, doi: 10.1002/cpe.4123.

[36] Z. N. Rashid, S. R. M. Zeebaree, and A. Shengul, "Design and Analysis of Proposed Remote Controlling Distributed Parallel Computing System Over the Cloud," in 2019 International Conference on Advanced Science and Engineering (ICOASE), Apr. 2019, pp. 118–123, doi: 10.1109/ICOASE.2019.8723695.

[37] Z. N. Rashid, S. R. M. Zebari, K. H. Sharif, and K. Jacksi, "Distributed Cloud Computing and Distributed Parallel Computing: A Review," in 2018 International Conference on Advanced Science and Engineering (ICOASE), Oct. 2018, pp. 167–172, doi: 10.1109/ICOASE.2018.8548937.

[38] C. M. Mohammed and S. R. M. Zeebaree, "Sufficient Comparison Among Cloud Computing Services : IaaS , PaaS , and SaaS : A Review," Int. J. Sci. Bus., vol. 5, no. 2, pp. 17–30, 2021, doi: 10.5281/zenodo.4450129.

[39] Z. J. Hamad and S. R. M. Zeebaree, "Recourses Utilization in a Distributed System : A Review," Int. J. Sci. Bus., vol. 5, no. 2, pp. 42–53, 2021, doi: 10.5281/zenodo.4461813.

[40] H. I. Dino, S. R. M. Zeebaree, O. M. Ahmad, H. M. Shukur, R. R. Zebari, and L. M. Haji, "Impact of Load Sharing on Performance of Distributed Systems Computations," Int. J. Multidiscip. Res. Publ., vol. 3, no. 1, pp. 30–37, 2020.

[41] B. Muthulakshmi and K. Somasundaram, "A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment," Cluster Comput., vol. 22, no. S5, pp. 10769–10777, Sep. 2019, doi: 10.1007/s10586-017-1174-z.

[42] H. Shukur et al., "A State of Art Survey for Concurrent Computation and Clustering of Parallel Computing for Distributed Systems," J. Appl. Sci. Technol. Trends, vol. 1, no. 4, pp. 148–154, Dec. 2020, doi: 10.38094/jastt1466.

[43] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, "Cloud Computing Virtualization of Resources Allocation for Distributed Systems," J. Appl. Sci. Technol. Trends, vol. 1, no. 3, pp. 98–105, 2020, doi: 10.38094/jastt1331.

[44] S. Askar, "Adaptive Load Balancing Scheme For Data Center Networks Using Software Defined Network," Sci. J. Univ. Zakho, vol. 4(A), no. 2, pp. 275–286, 2016, doi: 10.25271/2016.4.2.118.

[45] L. M. Haji, S. R. M. Zeebaree, O. M. Ahmed, A. B. Sallow, K. Jacksi, and R. R. Zeabri, "Dynamic Resource Allocation for Distributed Systems and Cloud Computing," test Eng. Manag. J., vol. 83, no. May-June, pp. 22417–22426, 2020.

[46] Y. S. Jghef and S. R. M. Zeebaree, "State of Art Survey for Significant Relations between Cloud Computing and Distributed Computing," Int. J. Sci. Bus., vol. 4, no. 12, pp. 53–61, 2020, doi: 10.5281/zenodo.4237005.

[47] P. Y. Abdullah, S. R. M. Zeebaree, H. M. Shukur, and K. Jacksi, "HRM System using Cloud Computing for Small and Medium Enterprises (SMEs)," Technol. Reports Kansai Univ., vol. 62, no. 4, pp. 1977–1987, 2020.

[48] J. Upadhyaya and N. J. Ahuja, "Quality of service in cloud computing in higher education: A critical survey and innovative model," in 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Feb. 2017, pp. 137–140, doi: 10.1109/I-SMAC.2017.8058324.

[49] G. A. Qadir and S. R. M. Zeebaree, "Evaluation of QoS in Distributed Systems : A Review," Int. J. Sci. Bus., vol. 5, no. 2, pp. 89–101, 2021, doi: 10.5281/zenodo.4462245.

[50] B. BAGIROZ, M. GUZEL, U. YAVANOGLU, and S. OZDEMIR, "QoS Prediction Methods in IoT A Survey," in 2019 IEEE International Conference on Big Data (Big Data), Dec. 2019, pp. 2128–2133, doi: 10.1109/BigData47090.2019.9006523.

[51] Y. Chen, E. Sun, and Y. Zhang, "Joint optimization of transmission and processing delay in fog computing access networks," in 2017 9th International Conference on Advanced Infocomm Technology (ICAIT), Nov. 2017, pp. 155–158, doi: 10.1109/ICAIT.2017.8388906.

[52] B. H. Husain and S. Askar, "Survey on Edge Computing Security," Int. J. Sci. Bus., vol. 5, no. 3, pp. 52–60, Jun. 2021, doi: 10.5281/zenodo.4496939.

[53] A. Khakimov, A. Muthanna, and M. S. A. Muthanna, "Study of fog computing structure," in 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Jan. 2018, vol. 2018-Janua, pp. 51–54, doi: 10.1109/EIConRus.2018.8317028.

[54] C.-C. Lin and J.-W. Yang, "Cost-Efficient Deployment of Fog Computing Systems at Logistics Centers in Industry 4.0," IEEE Trans. Ind. Informatics, vol. 14, no. 10, pp. 4603–4611, Oct. 2018, doi: 10.1109/TII.2018.2827920.

[55] W. Steiner and S. Poledna, "Fog computing as enabler for the Industrial Internet of Things," e i Elektrotechnik und Informationstechnik, vol. 133, no. 7, pp. 310–314, Nov. 2016, doi: 10.1007/s00502-016-0438-2.

[56] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying Fog Computing in Industrial Internet of Things and Industry 4.0," IEEE Trans. Ind. Informatics, vol. 14, no. 10, pp. 4674–4682, Oct. 2018, doi: 10.1109/TII.2018.2855198.

[57] K. D. Ahmed and S. R. M. Zeebaree, "Resource Allocation in Fog Computing : A Review," Int. J. Sci. Bus., vol. 5, no. 2, pp. 54–63, 2021, doi: 10.5281/zenodo.4461876.

[58] S. Shukla, M. F. Hassan, L. T. Jung, and A. Awang, "Architecture for Latency Reduction in Healthcare Internet-of-Things Using Reinforcement Learning and Fuzzy Based Fog Computing," in Advances in Intelligent Systems and Computing, vol. 843, 2019, pp. 372–383.

[59] A. B. Manju and S. Sumathy, "Efficient Load Balancing Algorithm for Task Preprocessing in Fog Computing Environment," in Smart Innovation, Systems and Technologies, vol. 105, Springer Singapore, 2019, pp. 291–298.

[60] J. Grover, A. Jain, S. Singhal, and A. Yadav, "Real-Time VANET Applications Using Fog Computing," in Smart Innovation, Systems and Technologies, vol. 79, 2018, pp. 683–691.

[61] A. Mebrek, L. Merghem-Boulahia, and M. Esseghir, "Efficient green solution for a balanced energy consumption and delay in the IoT-Fog-Cloud computing," in 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), Oct. 2017, vol. 2017-Janua, pp. 1–4, doi: 10.1109/NCA.2017.8171359.

[62] Z. Á. Mann, "Notions of architecture in fog computing," Computing, vol. 103, no. 1, pp. 51–73, Jan. 2020, doi: 10.1007/s00607-020-00848-z.

[63] X. Yuan, Y. He, Q. Fang, X. Tong, C. Du, and Y. Ding, "An Improved Fast Search and Find of Density Peaks-Based Fog Node Location of Fog Computing System," in 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Jun. 2017, vol. 2018-Janua, pp. 635–642, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.100.

[64] G. White, A. Palade, C. Cabrera, and S. Clarke, "Quantitative Evaluation of QoS Prediction in IoT," in 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Jun. 2017, pp. 61–66, doi: 10.1109/DSN-W.2017.26.

[65] H. F. Atlam, R. J. Walters, and G. B. Wills, "Internet of Things: State-of-the-art, Challenges, Applications, and Open Issues," Int. J. Intell. Comput. Res., vol. 9, no. 3, pp. 928–938, Sep. 2018, doi: 10.20533/ijicr.2042.4655.2018.0112.

[66] G. White, A. Palade, C. Cabrera, and S. Clarke, "IoTPredict: Collaborative QoS Prediction in IoT," in 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), Mar. 2018, pp. 1–10, doi: 10.1109/PERCOM.2018.8444598.

[67] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog Orchestration for Internet of Things Services," IEEE Internet Comput., vol. 21, no. 2, pp. 16–24, Mar. 2017, doi: 10.1109/MIC.2017.36.

[68] C.-L. Fok, C. Julien, G.-C. Roman, and C. Lu, "Challenges of satisfying multiple stakeholders: quality of service in the internet of things," in

Proceeding of the 2nd workshop on Software engineering for sensor network applications - SESENA '11, 2011, p. 55, doi: 10.1145/1988051.1988062.

[69] J.-P. Calbimonte, M. Riahi, N. Kefalakis, J. Soldatos, and A. Zaslavsky, "Utility Metrics Specifications. OPENIoT Deliverable D4.2.2.," 2014. [Online]. Available: https://infoscience.epfl.ch/record/210923/files/OpenIoT-WP4-D422-EPFL-140114-V26-QR.pdf.

[70] P. P. Jayaraman, K. Mitra, S. Saguna, T. Shah, D. Georgakopoulos, and R. Ranjan, "Orchestrating Quality of Service in the Cloud of Things Ecosystem," in 2015 IEEE International Symposium on Nanoelectronic and Information Systems, Dec. 2015, pp. 185–190, doi: 10.1109/iNIS.2015.64.

[71] R. Duan, X. Chen, and T. Xing, "A QoS Architecture for IOT," in 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Oct. 2011, pp. 717–720, doi: 10.1109/iThings/CPSCom.2011.125.

[72] B. Li and J. Yu, "Research and Application on the Smart Home Based on Component Technologies and Internet of Things," Procedia Eng., vol. 15, pp. 2087–2092, 2011, doi: 10.1016/j.proeng.2011.08.390.

[73] A. Alqahtani, Y. Li, P. Patel, E. Solaiman, and R. Ranjan, "End-to-End Service Level Agreement Specification for IoT Applications," in 2018 International Conference on High Performance Computing & Simulation (HPCS), Jul. 2018, vol. 49, no. 12, pp. 926–935, doi: 10.1109/HPCS.2018.00147.

[74] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms, and Open Problems," IEEE Commun. Surv. Tutorials, vol. 19, no. 1, pp. 550–586, 2017, doi: 10.1109/COMST.2016.2610578.

[75] B. N. Silva, M. Khan, and K. Han, "Internet of Things: A Comprehensive Review of Enabling Technologies, Architecture, and Challenges," IETE Tech. Rev., vol. 35, no. 2, pp. 205–220, Mar. 2018, doi: 10.1080/02564602.2016.1276416.

[76] R. M. Savola, P. Savolainen, A. Evesti, H. Abie, and M. Sihvonen, "Risk-driven security metrics development for an e-health IoT application," in 2015 Information Security for South Africa (ISSA), Aug. 2015, pp. 1–6, doi: 10.1109/ISSA.2015.7335061.

[77] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing," J. Netw. Comput. Appl., vol. 67, pp. 99–117, May 2016, doi: 10.1016/j.jnca.2016.01.010.

[78] M. Moreno, B. Úbeda, A. Skarmeta, and M. Zamora, "How can We Tackle Energy Efficiency in IoT BasedSmart Buildings?," Sensors, vol. 14, no. 6, pp. 9582–9614, May 2014, doi: 10.3390/s140609582.

[79] Q. Shaheen, M. Shiraz, M. U. Hashmi, D. Mahmood, Z. Zhiyu, and R. Akhtar, "A Lightweight Location-Aware Fog Framework (LAFF) for QoS in Internet of Things Paradigm," Mob. Inf. Syst., vol. 2020, pp. 1–15, Sep. 2020, doi: 10.1155/2020/8871976.

[80] S. Rani, N. Saravanakumar, S. Rajeyyagari, V. Porkodi, and S. H. Bouk, "QoS aware cross layer paradigm for urban development applications in IoT," Wirel. Networks, vol. 26, no. 8, pp. 6203–6214, Nov. 2020, doi: 10.1007/s11276-020-02430-z.

[81] C. A. Ouedraogo, S. Medjiah, C. Chassot, K. Drira, and J. Aguilar, "A Cost-Effective Approach for End-to-End QoS Management in NFV-enabled IoT Platforms," IEEE Internet Things J., pp. 1–1, 2020, doi: 10.1109/JIOT.2020.3025500.

[82] K. S. Bhandari, I.-H. Ra, and G. Cho, "Multi-Topology Based QoS-Differentiation in RPL for Internet of Things Applications," IEEE Access, vol. 8, pp. 96686–96705, 2020, doi: 10.1109/ACCESS.2020.2995794.

[83] E. Badidi and A. Ragmani, "An Architecture for QoS-Aware Fog Service Provisioning," Procedia Comput. Sci., vol. 170, pp. 411–418, 2020, doi: 10.1016/j.procs.2020.03.083.

[84] M. M. Badawy, Z. H. Ali, and H. A. Ali, "QoS provisioning framework for service-oriented internet of things (IoT)," Cluster Comput., vol. 23, no. 2, pp. 575–591, Jun. 2020, doi: 10.1007/s10586-019-02945-x.

[85] M. Asad, A. Basit, S. Qaisar, and M. Ali, "Beyond 5G: Hybrid End-to-End Quality of Service Provisioning in Heterogeneous IoT Networks," IEEE Access, vol. 8, pp. 192320–192338, 2020, doi: 10.1109/ACCESS.2020.3032704.

[86] M. Asad, S. Qaisar, and A. Basit, "Client Based Access Layer QoS Provisioning in Beyond 5G IoT Networks," in 2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet), Sep. 2020, pp. 1–8, doi: 10.1109/CommNet49926.2020.9199612.

[87] A. Ali et al., "Quality of Service Provisioning for Heterogeneous Services in Cognitive Radio-Enabled Internet of Things," IEEE Trans. Netw. Sci. Eng., vol. 7, no. 1, pp. 328–342, Jan. 2020, doi: 10.1109/TNSE.2018.2877646.

[88] A. Yousefpour et al., "FOGPLAN: A Lightweight QoS-Aware Dynamic Fog Service Provisioning Framework," IEEE Internet Things J., vol. 6, no. 3, pp. 5080–5096, Jun. 2019, doi: 10.1109/JIOT.2019.2896311.

[89] J. Yao and N. Ansari, "Fog Resource Provisioning in Reliability-Aware IoT Networks," IEEE Internet Things J., vol. 6, no. 5, pp. 8262–8269, Oct. 2019, doi: 10.1109/JIOT.2019.2922585.

[90] J. Yao and N. Ansari, "QoS-Aware Fog Resource Provisioning and Mobile Device Power Control in IoT Networks," IEEE Trans. Netw. Serv. Manag., vol. 16, no. 1, pp. 167–175, Mar. 2019, doi: 10.1109/TNSM.2018.2888481.

[91] S. Verma, N. Sood, and A. K. Sharma, "QoS provisioning-based routing protocols using multiple data sink in IoT-based WSN," Mod. Phys. Lett. A, vol. 34, no. 29, p. 1950235, Sep. 2019, doi: 10.1142/S0217732319502353.

[92] N. N. Srinidhi, J. Lakshmi, and S. M. Dilip Kumar, "Hybrid Energy Efficient and QoS Aware Algorithm to Prolong IoT Network Lifetime," in Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, vol. 276, Springer International Publishing, 2019, pp. 80–95.

[93] X. Li, H. Ding, M. Pan, J. Wang, H. Zhang, and Y. Fang, "Statistical QoS Provisioning Over Uncertain Shared Spectrums in Cognitive IoT Networks: A Distributionally Robust Data-Driven Approach," IEEE Trans. Veh. Technol., vol. 68, no. 12, pp. 12286–12300, Dec. 2019, doi: 10.1109/TVT.2019.2946834.

[94] F. Khan et al., "A Quality of Service-Aware Secured Communication Scheme for Internet of Things-Based Networks," Sensors, vol. 19, no. 19, p. 4321, Oct. 2019, doi: 10.3390/s19194321.

[95] M. Guo, L. Li, and Q. Guan, "Energy-Efficient and Delay-Guaranteed Workload Allocation in IoT-Edge-Cloud Computing Systems," IEEE Access, vol. 7, no. c, pp. 78685–78697, 2019, doi: 10.1109/ACCESS.2019.2922992.

[96] I. Maslouhi, E. Miloud, K. Ghoumid, and K. Baibai, "Analysis of End-to-End Packet Delay for Internet of Things in Wireless Communications," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 9, pp. 338–343, 2018, doi: 10.14569/IJACSA.2018.090944.

[97] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards QoS-Aware Fog Service Placement," in 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), May 2017, pp. 89–96, doi: 10.1109/ICFEC.2017.12.

[98] S. Muralidharan, B. J. R. Sahu, N. Saxena, and A. Roy, "PPT: A Push Pull Traffic Algorithm to Improve QoS Provisioning in IoT-NDN Environment," IEEE Commun. Lett., vol. 21, no. 6, pp. 1417–1420, Jun. 2017, doi: 10.1109/LCOMM.2017.2677922.