Rostami and Goli-Bidgoli
Journal of Cloud Computing (2024) 13:89
https://doi.org/10.1186/s13677-024-00651-7

Journal of Cloud Computing:
Advances, Systems and Applications

REVIEW

Open Access

An overview of QoS-aware load balancing techniques in SDN-based IoT networks

Mohammad Rostami[1] and Salman Goli-Bidgoli[1*]

Abstract

Increasing and heterogeneous service demands have led to traffic increase, and load imbalance challenges among network entities in the Internet of Things (IoT) environments. It can affect Quality of Service (QoS) parameters. By separating the network control layer from the data layer, Software-Defined Networking (SDN) has drawn the interest of many researchers. Efficient data flow management and better network performance can be reachable through load-balancing techniques in SDN and improve the quality of services in the IoT network. So, the combination of IoT and SDN, with conscious real-time traffic management and load control, plays an influential role in improving the QoS. To give a complete assessment of load-balancing strategies to enhance QoS parameters in SDN-based IoT networks (SD-IoT), a systematic review of recent research is presented here. In addition, the paper provides a comparative analysis of the relevant publications, trends, and future areas of study that are particularly useful for the community of researchers in the field.

Keywords Internet of Things, Load-balancing, Quality of service, SD-IoT, Software-defined networking, Systematic review

Introduction

Today, applying the Internet of Things (IoT) has become one of the most important and attractive topics in the network realm and has attracted the attention of many researchers [1]. Applying IoT in making smart homes, cities, and industries has an impact on health, productivity, and energy. It has made many changes in our lifestyle and provided desirable solutions to address the tasks of users [2–4].

IoT is an interconnected network of things that can interact via a network infrastructure to provide or receive services [5]. Service means doing a task on the network to fulfill a set of objectives, such as maximizing reliability and minimizing execution time, and resource cost. The service requirements are separated into functional and non-functional categories. Functional characteristics are those that the service is required to perform. Non-functional features relate to service quality, such as availability, scalability, cost, response time, energy consumption, and security. Non-functional features are sometimes contrasting [6]. So, it is important to create a suitable framework to maintain Quality of Service (QoS) in the IoT network [7]. On the other hand, with the continuous growth of IoT devices, data traffic exponentially increases. This increase in simultaneous data production requires improving QoS [7–9]. Consequently, the significance of enhancing QoS parameters to optimize the overall performance of the network and provide novel technologies and communication schemes has multiplied [7, 10].

The IoT domain starts with smart objects with constrained resources in terms of accuracy and data rate, computing power, energy, memory, and storage [11]. These limitations can lead to heavy traffic in some applications, that require enormous computing, storage, and

*Correspondence:
Salman Goli-Bidgoli
salmangoli@kashanu.ac.ir
[1] Department of Computer Engineering, Faculty of Electrical and Computer Engineering, University of Kashan, Kashan 8731753153, Iran

communication resources such as industry, healthcare, and smart cities [12]. The lack of these resources in devices and IoT infrastructure has led to service response time as one of the significant challenges of these applications. Meanwhile, cloud computing in the IoT has found an effective role in solving many of these problems [13].

Cloud-based IoT includes heterogeneous and intelligent devices that constantly exchange heavy traffic from the network to servers. Servers strive to improve the QoS to end users by providing real-time and reliable services [14]. Nevertheless, any delay in response time would negatively influence QoS, particularly in real-time applications [15]. Inadequate traffic distribution across cloud servers also leads some of them to become overloaded, resulting in a QoS decrease. To achieve this objective, it is required to control and balance the traffic load in both the network infrastructure and IoT servers [14]. To solve this problem, fog computing is proposed to meet the real-time requirements of the IoT [16]. Fog servers may be overloaded and unable to handle all requests in terms of resource constraints and an increased number of object requests. It needs dynamic workflow management techniques with significant processing resources that are not available on fog servers [9].

In this regard, the Software-Defined Networking (SDN) model introduces a favourable solution to improve QoS and increase the flexibility of the IoT network. It can manage the infrastructure as well as heterogeneous IoT devices by separating the logical control layer (traffic management decision-making) and the hardware data layer (traffic transfer mechanism to the intended destination) [9, 17]. SDN may forward requests (tasks) of IoT applications such as industry, health, and smart city to the fog or cloud server as near as feasible to address the issue of simultaneous delay and load-balancing while managing the dynamic traffic flow [16, 18]. This feature is obtained by SDN characteristics, such as global network visibility, programmability [19], openness, and virtualization [20], monitoring global network resources. SDN can find optimal load assessment [12] and the optimal location for task processing [21], manage dynamic incoming traffic to network nodes, balance traffic load [22], and thus increase network efficiency and QoS [16]. It is used to develop the IoT network as the SDN-based IoT network (SD-IoT) [16]. In general, the IoT and SDN have a set of complementary requirements and capacities. IoT can strengthen SDN by solving the issues related to scalability, mobility, and real-time requirements. On the other hand, users have a series of QoS requirements that can be answered by combining the IoT and SDN.

The IoT alone cannot solve the challenges related to the QoS. Numerous studies show that SDN is an effective approach to the IoT. By separating the control layer from the infrastructure layer, simplifying the hardware used in the network as much as possible, and having real-time information about all the network conditions (links status and load on the hardware), it is possible to define the QoS from the direction of the user, service provider and infrastructure in SDN. Therefore, new load-balancing algorithms based on QoS can be proposed. The integration of IoT and SDN is helpful to load-balancing, QoS improvement, and traffic engineering techniques applications [20, 23]. SDN can serve a distinct role in the IoT environment, enabling the network to be programmable and dynamically adjustable while ensuring interoperability across heterogeneous IoT networks. By integrating both technologies, the IoT network will have a full view of network resources and task requirements. Network resources can then be efficiently assigned based on the task requirements during a specific time based on load balancing. This study is to find out the increasing demand to handle the workload on resources present at the SD-IoT and to enquire about load-balancing approaches. A comprehensive review has been conducted to evaluate load-balancing approaches of SD-IoT, and these approaches were compared based on different metrics. The following Motivations particularly inspired this article:

- The increased need to understand load distribution techniques in SD-IoT for proper utilization of resources.
- Since the concept of load balancing addresses the improvement of QoS from the directions of users, service providers, and infrastructure providers, it motivates us to focus on load balancing techniques.
- Based on the increasing workload in IoT, the need for load-balancing among resources has been recognized. Despite available research, the existing work has been identified and summarized in a systematic way that depicts issues and challenges for future research work.

Therefore, to express the importance of load-balancing in providing and ensuring the QoS of IoT networks, significant contributions have been performed in this study:

- Highlighting key and influential elements of SDN and IoT in QoS.
- Describing the SD-IoT network architecture and load-balancing problem in SD-IoT.
- Studying and comparing available load-balancing techniques in SD-IoT aiming at improving QoS.
- Defining the QoS parameters used in studied load-balancing approaches.

- Introduction of simulators/tools in designing and testing load-balancing algorithms in SD-IoT networks.
- Future research opportunities in QoS-based load-balancing in SD-IoT.

The background of the SD-IoT network, an introduction to the SD-IoT architecture, and the function of load-balancing in the SD-IoT network to enhance QoS parameters are all covered in Background Sect. 2. The research approach is described in Research method Sect. 3. Research questions and highlights section 4 includes responses to research concerns about analyzing and comparing various load-balancing approaches, as well as QoS factors and their simulators. Discussion section 5 discusses research questions and load-balancing techniques. Future research trends and opportunities section 6 presents future research opportunities. In Conclusion Sect. 7, the conclusion and finally, the references are provided.

Background

We first provide an overview of the concept of SD-IoT. Then, the SD-IoT architecture is explained, and afterwards, the load-balancing technique and some QoS parameters are discussed.

Description of SD-IoT network

SDN network is one of the cost-effective and compatible architectures, with the ability to network reconfiguration (e.g., implying a controller restart based on network traffic), scheduling flexibility, scalability, flow-based, ease of access, and optimal management. SDN has provided opportunities for IoT networks to develop agile networking, load-balancing processes, and QoS improvement to separate the control layer from the data layer [22, 24, 25].

SDN controllers store data regarding fog/cloud servers as well as incoming and outgoing traffic from IoT devices. Based on the load on the servers and certain specified criteria, the SDN controller determines the sort of collaboration between the servers and ensures load-balancing and efficient use of computing, storage, and communication resources [26].

SD-IoT architecture

IoT architecture should be scalable and efficient. It should be able to manage enormous tasks with high QoS [14]. Due to the increasing number of wireless devices, the IoT architecture covers a variety of communication technologies from Long Range Networks (LoRa) and cellular networks to wireless sensor networks [4]. Each potential IoT and SDN design is explained individually in Fig. 1, followed by the final hybrid architecture.

The most popular IoT network architecture includes three layers of cloud computing, fog computing, and end devices [27]. The cloud layer provides flexible and efficient computing resources for IoT applications [8, 28]. The cloud is responsible for providing services that require more computing or are not supported by the fog layer [17].

The cloud has QoS restrictions due to the great distance to network devices, including higher costs, delay, energy consumption, carbon emissions, and inefficient resource utilization [11, 17]. The fog layer is a computational fog model that improves communication and processing performance in a variety of applications by putting computing, storage, and communication closer to IoT devices [11, 17]. This leads to QoS improvements, such as reducing service delivery delays, and increasing data rates and bandwidths at the edge of the network [8, 26, 29]. Fog computing seeks to reduce communication and processing overload between edge devices and cloud data centers, thereby preventing network performance (QoS) degradation [17, 30]. However, due to resource constraints and the locality of fog servers, only a limited number of things can be serviced [31]. The end-devices layer is made up of heterogeneous devices with unique IDs and diverse functionalities, as well as users who may connect to the network at any time and place, exchange data, and need high-quality services [8, 22].

At cloud/fog layers, some resources may become overloaded with an increasing number of requests from end devices. Utilizing SDN architecture and the ability of the controllers the manage incoming traffic and allocation to network resources, requests are assigned to the best and closest server in the fog or cloud to establish load-balancing in the network [32, 33]. The SDN network architecture is divided into three layers: data, control, and application [34].

The data layer consists of a collection of packet components (tasks). The tasks are directed to the intended destination. Data transfer between end users occurs according to the rules set by the control layer [26, 32, 35]. The control layer sets network transfer rules and manages workflows [26, 35, 36]. With a global view of the network and workflow awareness, the control layer can monitor network conditions and act as a decision-maker in offloading input tasks to fog/cloud servers to improve QoS [37]. The control layer consists of controllers and is responsible for routing, security, load-balancing, and monitoring [19, 38]. The application layer implements network control logic and strategies and designs services such as analysis, monitoring, transferring plans, manageability, traffic engineering, load-balancing, and security [36, 39].

The ability to design SDN networks may give incentives for optimizing traffic management in IoT networks

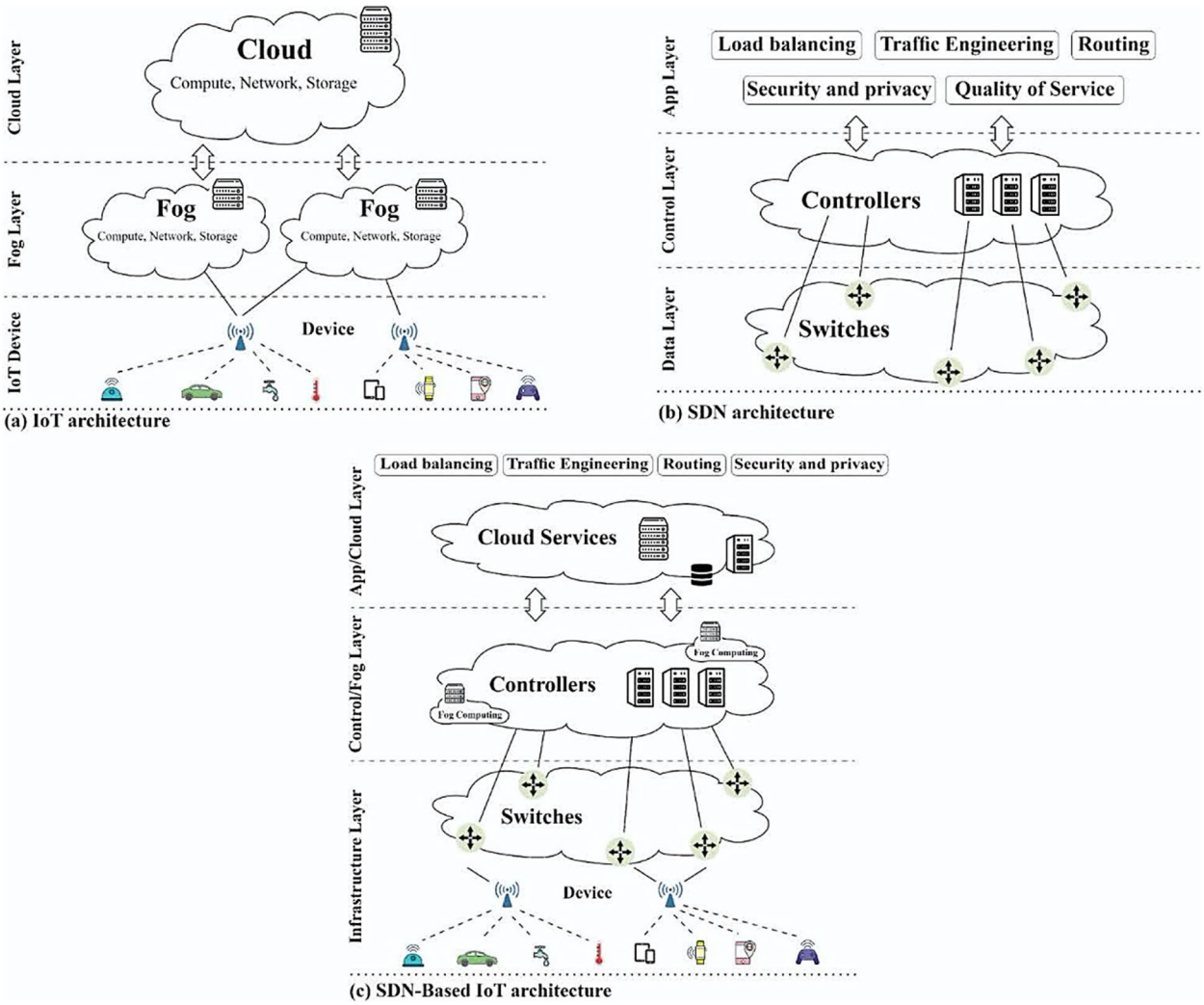

Fig. 1 Types of IoT, SDN, and SDN-Based IoT architectures

to deliver services to end devices [12, 40]. On the SD-IoT network, QoS and network performance can be enhanced at the same time [18, 23]. The SD-IoT architecture comprises three levels for IoT QoS management [38]. The layer of the infrastructure consists of network elements, including devices, gateways, and switches that are distributed in different geographical locations [41]. At this layer, transmission elements, such as switches are used to send the input data flow to the upper layers for further processing.

The control/fog layer includes controllers and fog servers as the main components of this layer that are geographically distributed [41, 42]. The controller creates flow rules and policies for the flow tables to manage the workload in the infrastructure layer [41]. It's worth noting that the control layer maintains QoS requirements in terms of network status monitoring and topology discovery, as well as making load-balancing choices based on the application layer's specifications [43, 44]. The application/cloud layer, the highest layer is allocated to IoT services and applications [42]. This layer interacts with the controllers to apply load-balancing, flexibility, and performance optimization [6, 28, 43].

IoT queries/requests are routed over a gateway to SDN switches. The flow route is determined by SDN switches. The switch asks the associated SDN controller for routing information. The flow tables of switches are updated with new rules. Then, the request is routed to a fog/cloud server which has the desired service and tolerable load. Finally, the server provides services to users [36, 45, 46]. Figure 2 shows the service delivery process on the SD-IoT network.

Load-balancing in SD-IoT

In this section, the importance of the load-balancing technique to improve the QoS of IoT is explained. Then

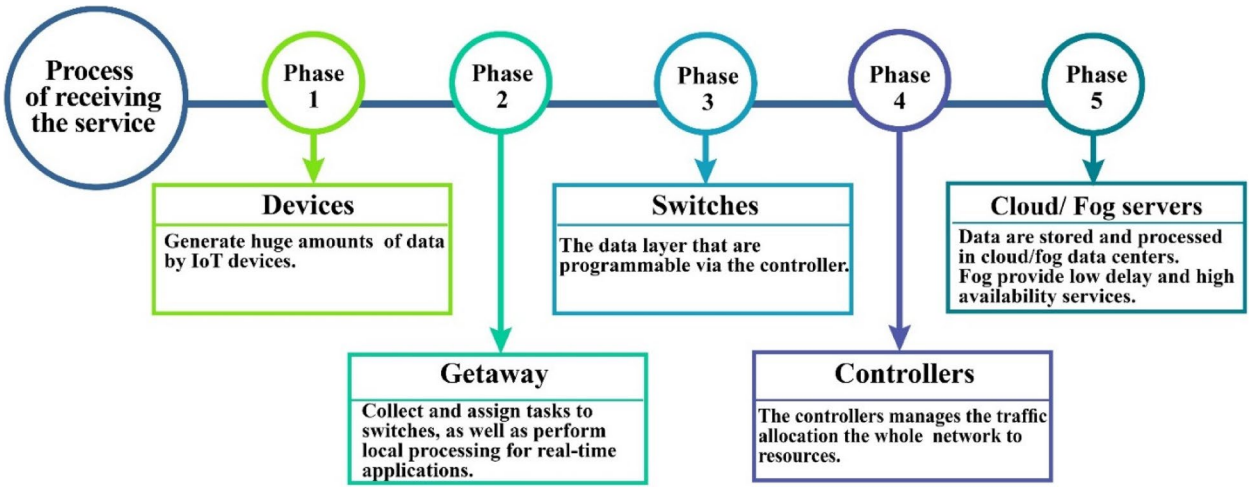

Fig. 2 Service delivery process on the SD-IoT network

the role of the SDN controller in load-balancing in the IoT network is discussed. Finally, the types of SDN controller architecture and the introduction of appropriate controller architecture to improve load-balancing and QoS are described.

The workload balance between network resources is the most essential problem that the service provider must address [47]. It is vital to stress QoS criteria while assigning resources to activities. As a result, it is critical to building effective load-balancing strategies for adjusting network traffic flows to minimize network congestion and fulfill the QoS requirements of IoT applications [48]. In load-balancing studies, the distribution of received tasks as well as resource utilization rates is commonly used to make decisions about load distribution [49]. One way to deal with network resource overload is to transfer the load from the overloaded resources to the underloaded ones [44].

The controller may evaluate real-time traffic rules for switches using a global view of the network (for example, current load status, the residual capacity of cloud/fog resources, and congestion level). Tasks are routed to cloud/fog servers in this instance, and the network meets its load-balancing aim [11, 42, 44]. The SDN controller is responsible for managing the load distribution between network nodes and improving the QoS parameters [45]. The architecture of existing controllers can be categorized into two general categories, centralized and decentralized, as shown in Fig. 3.

The centralized architecture consists of a single controller to manage the entire network and in terms of the problems, such as scalability, accessibility, and reliability, it is not able to meet the requirements of IoT QoS [50, 51]. Of the increase in IoT network traffic, this strategy will be unable to fulfill users' expanding demands and may become a single point of failure [6, 12, 45, 52].

Decentralized architecture, is based on the hierarchy of controllers as primary and secondary controllers. The primary controller can be referred to as the root controller and the secondary controllers as local controllers where the primary controller assigns control to the secondary controller [53, 54]. decentralized controllers can be divided into multi-controller and distributed controller architectures.

A multi-controller architecture is required to manage the traffic of SD-IoT Networks, which can provide scalability and reliability and yet preserve the simplicity of the control function [22]. Through east–west interfaces, there is communication between each controller and the network management [55]. In a large-scale network, the network is divided into multiple domains. Each domain is managed by a controller. Multi-controllers have been implemented to support mobility management, flow processing and flow forwarding in distributed environments [9]. However multiple controller placements are a problem with the issue of the required number of controllers and controller placement to balance the traffic load at minimum delay [9, 56].

Service providers use a distributed controller architecture as a special type of decentralized controller, to manage traffic load and distribute tasks to appropriate resources to control network parameters. Through the interaction of controllers, network utilization and service delivery performance are improved [9, 46, 47]. The architecture of distributed controllers is important for large-scale SD-IoT load-balancing, which improves QoS parameters, such as reliability, scalability, and accessibility [37].

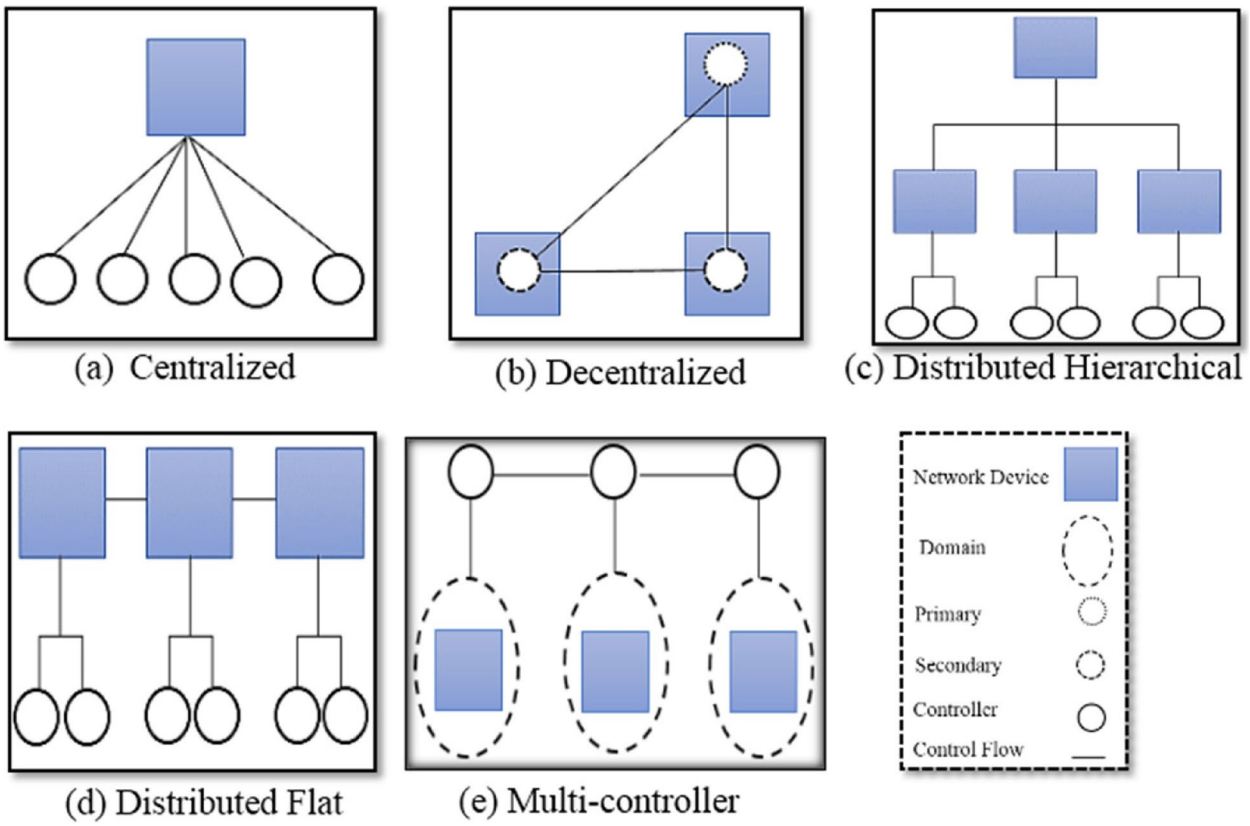

Fig. 3 Types of SDN controller architecture

Each controller is linked to several switches. Uneven load distribution across controllers is caused by the nature of static mapping between switches and controllers, unanticipated network traffic, and dynamic topology change [37]. To prevent the overload of controllers, load distribution is known as a load-balancing technique [25]. Each controller is responsible for a specific geographic area that will have a local view of the network status. Information about service requests, workflow transmission, and resource allocation is managed via the coordination of distributed controllers [46].

In a distributed architecture, distributed controllers can be divided into flat and hierarchical architectures. In flat architecture, controllers in the same layer have the same tasks and communicate directly with each other. Controllers are present at numerous layers of a hierarchical design and have various tasks in each layer. Communication and coordination among the lower controllers are the responsibility of the top controllers [20, 57]. To increase the scalability of the network, it is recommended to distribute the traffic load of IoT devices among the controllers and reduce the computational delay in SDN networks under QoS requirements; So, the design of a hierarchically distributed SDN controller system is proposed [10, 45].

The controllers in a distributed and decentralized architecture periodically exchange network control information with each other, but due to the controller's static connection to the switch, as well as changes in the traffic load of the switches, unfavourable load distribution between the controllers can occur and negatively affecting QoS parameters like response time and network throughput [58]. In general, load balancing may aim at preventing overload, removing the overload, or a combination of both. Neural networks [59], prediction [38], and virtualization of network functions [10] were introduced as solutions to prevent overloading. Moreover, informing the network support [60] and periodic tracking [25] [46] have been devised as solutions to eliminating overload in the network.

QoS parameters

The main purpose of load-balancing is to improve the optimal QoS parameters in the network. To assess load-balancing solutions, researchers looked at many factors. To find a better load-balancing algorithm and identify the

Table 1 Some QoS parameters in related studies

Parameter	Explanation	Ref
Response time	Time elapsed from acceptance to successful response to the task on the server	[44]
Delay	Time spent to transfer and process the request on the server	[5, 61]
Resource productivity	Using network resources (bandwidth, processor, and memory)	[62, 63]
Throughput	Tasks performed per unit of time or fair maximum use of resources and allocation of resources to workflows at the moment of their arrival	[7, 64]
Load-balancing	Rate of workload distribution on the network elements	[63]
Loss rate	The ratio of lost packets to the sent packets	[61]
Packet delivery rate	The number of packets safely delivered to destinations	[65]
Overload	Percentage of using resources more than the threshold	[61]
Energy consumption	Consumed energy by the network nodes	[64, 66]
Scaling	The ability of the network to support changes in the number of devices and network workload traffic as well as green computing	[17]

[a] Load Balancing

[a] Fog of Things

[a] http://mininet.org/

[a] https://iperf.fr/

[a] Distributed Internet Traffic Generator

advantages and disadvantages, several QoS metrics are utilized. Table 1 introduces the most used QoS parameters in various studies to investigate the effect of load-balancing on QoS in the SD-IoT network.

Some other QoS parameters include; Jitter (Deviation from the average data reception delay) [14], stability (distribution of network traffic among resources to maintain service continuity) [16], cost (payment of service cost by the user) [6], processing time (duration of service operation on CPU resources) [22], waiting /transfer time (time required to transfer the task to the server to receive the service) [22], security (protection against attacks to maintain the accuracy of information exchanged in the network) [2], reliability (correct and timely performance of the task) [28], and network lifetime (energy consumed by the network) [67]. These parameters have received less attention in almost all reviewed articles.

Research method

The Systematic Literature Review (SLR) strategy is used to collect and categorize load-balancing techniques in SD-IoT. SLR is a method to find, evaluate, interpret, and combine existing studies related to specific areas and report findings [68, 69]. This section describes the SLR method. This study was performed to increase understanding of load-balancing techniques in SD-IoT.

Data resources

Searching was conducted in September 2023 without any specific time limit and based on the article's title. As a result, 62 articles were found between 2015 and 2023. Research articles in journals and conferences were considered by the IEEE,[1] Springer,[2] Science Direct,[3] Wiley,[4] ACM,[5] MDPI,[6] and Google Scholar[7] to extract related articles.

Searching strategy

Based on a routine literature review, this paper evaluates current efforts and trends and lays the groundwork for future research on load-balancing in the SD-IoT network to enhance QoS. To begin the search, Google Scholar is chosen as the primary search engine. The search terms are identified based on the planned study subject and queries as a first step in shaping the search field. The search keywords of SD-IoT, IoT, SDN, load-balancing, and QoS were used, and the "AND" and "OR" logical operators were used to link keywords and find related articles. The related studies were thoroughly analyzed and summarized based on the main feature of the study, main tasks of the proposed algorithm, research objective environment, major participation, evaluation tool, data set, and criteria used for evaluation.

Figure 4 shows the selection process for related articles. Search in individual publications is the first step. In the second step, the initial search leads to selecting 310

[1] http://ieeexplore.ieee.org

[2] http://link.springer.com

[3] http://www.sciencedirect.com

[4] http://onlinelibrary.wiley.com

[5] http://www.acm.org

[6] https://www.mdpi.com

[7] http://Scholar.google.com

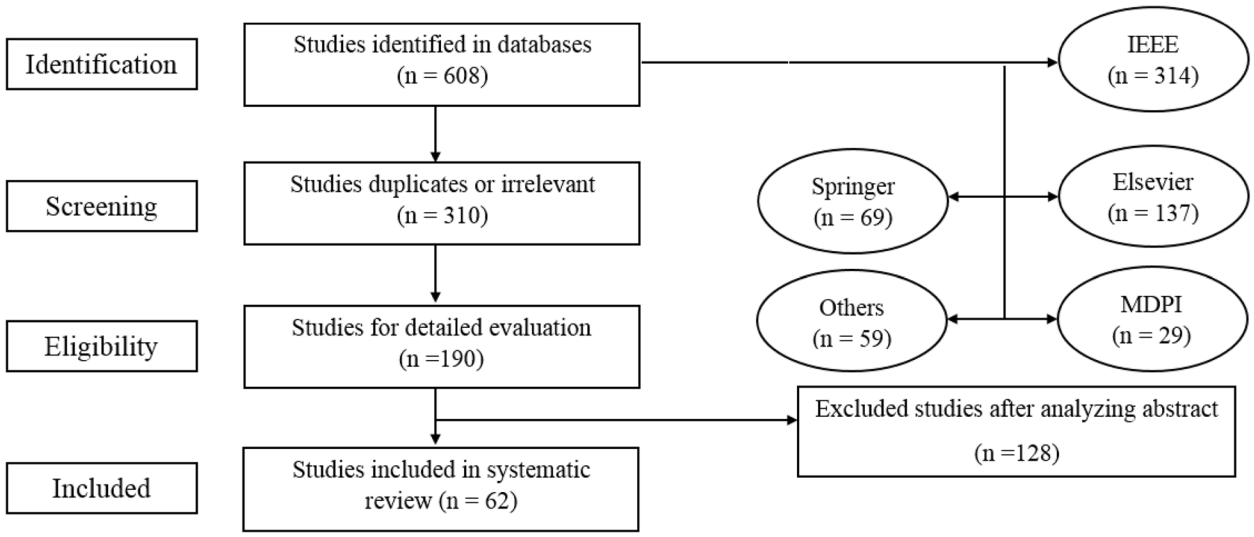

Fig. 4 Article identification process overview

articles. It does not make sense to read all of these articles because some of them are not directly related to the topic or are of low quality. Therefore, in the third step, those articles are carefully studied and the most appropriate ones are selected for deeper analysis.

Pre-2015 journals rarely addressed load balance-related issues in SD-IoT. So in this step, other types of studies such as reviews, reports, working articles, and non-English articles are ignored. As a result, 129 articles were chosen based on the following criteria: published between 2015 and 2023, English language, and subject relevance. The quality evaluation was the fourth phase. Following an examination of the abstracts and, in some instances, the whole papers, 62 relevant studies were identified. Those papers directly address load-balancing in SD-IoT and the improvement of specific QoS metrics. Related papers on load-balancing in SD-IoT are analyzed and extract significant concerns on optimization challenges.

Highlights and research questions

The highlights and research questions are part of the SLR. To analyze load-balancing in SD-IoT, related research highlights and questions are listed along with the motivations for such highlights and questions. They are as follows:

Highlight 1. Listing the challenges in IoT networks that led to the use of SDN. The growth of traffic from IoT devices leads to congestion and reduced QoS. Therefore, management and control of network resources, scalability, flexibility, and load-balancing seem to improve the QoS. This highlight is explored throughout the paper.

Highlight 2. In SD-IoT, load balancing is critical. The rising demand for network services puts more strain on the network, lowering its efficiency and using more energy. Load balancing helps with network traffic control and QoS, as discussed in Sect. 4.

Question 1. What are the present approaches for load-balancing in SD-IoT to improve QoS parameters? In Sect. 4, load-balancing techniques are categorized based on the policies used in the selected articles.

Question 2. Which QoS parameters are emphasized to evaluate load-balancing techniques in SD-IoT? The answer given to this question in Sect. 4 helps researchers evaluate and recognize their innovations.

Question 3. What are QoS requirements expected of the user, service provider, and infrastructure's directions? Simultaneous QoS optimization for user entities, service providers, and infrastructure may provide mutual advantages and improve network efficiency. This problem is discussed in Sect. 4.

Question 4. What is the frequency of SD-IoT network applications in using load-balancing techniques? Responses to this question are discussed in Sect. 4.

Question 5. Which common simulation tools are used for load-balancing in SD-IoT? To model and simulate load-balancing techniques, some basic aspects should be considered, including simulation scenarios, data set type and format, data storage, and communication protocol to control data traffic between nodes. This question will also be discussed in Sect. 4.

Question 6. What are the future research opportunities and open issues in load-balancing in SD-IoT? Suggestions help researchers identify future research trends and

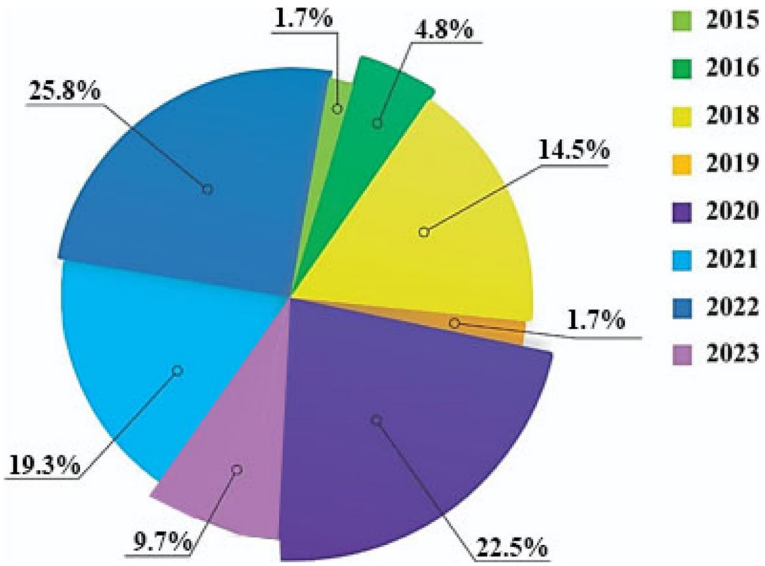

Fig. 5 Publication of selected articles in the journal by year of publication

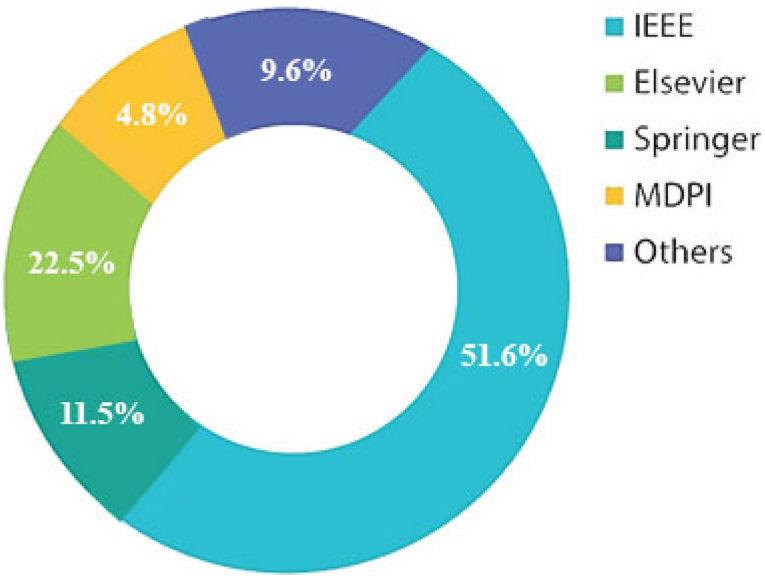

Fig. 6 Circular chart of articles published

opportunities in improving QoS in the SD-IoT network. A description of research opportunities is provided in Sect. 5.

Qualitative evaluation

The frequency of related publications in each journal over time is shown in Fig. 5. Figure 6 shows the proportion of articles published by each publisher.

Research questions and highlights

The highlights seek to clarify the role of load-balancing in the SD-IoT network and identify challenges and techniques applied to improve QoS. Questions also help identify future research areas. In the following, the symbol “RH”s are used for Research Highlights, and "RQ"s are used for Research Questions to answer the above-mentioned highlights and research questions.

RH1. The IoT makes it possible to control and monitor numerous interconnected intelligent objects, such as physical devices and sensors, from remote locations [19]. IoT devices have limited resources in terms of processing power, memory capacity, bandwidth, and battery life (energy) and they would reduce QoS, especially delays in real-time services [7, 70]. In terms of storage and processing power, cloud computing infrastructure has been suggested. The workload offloading to strong cloud resources through the network is intended to overcome hardware limitations and conserve device energy [71]. Users and service providers alike may profit from cloud computing [47].

Offloading tasks in the cloud for real-time applications due to long distances between devices and cloud resources leads to increased workload processing delay, high power consumption, less mobility support, lack of location awareness, security, privacy, and bandwidth congestion [7, 8]. To address these issues, fog-based computing infrastructure is used to move network resources closer to devices that transmit data to delay-sensitive applications, allowing the network to achieve some sort of balance [2, 72]. Fog computing minimizes resource costs, filters raw data, increases service access, and reduces delay to support real-time applications with lower operating costs [8, 11, 73]. The delay-sensitive workload is locally performed at the edge of the network by fog servers and heavy computational load with lower delay sensitivity in remote cloud data centers [41].

To increase network performance and QoS, SDN technology efficiently distributes network resources to workloads. As the number of service requests increases, network nodes become overloaded, and load-balancing techniques are introduced by controllers to reduce traffic congestion and eliminate overload [36, 39, 74]. Offloading of tasks to resources is performed by SDN controllers that could fully program the network with the aim of providing real-time services, fast and reliable data transfer and, in short, improving the QoS [17, 26, 75]. In general, the SDN-based solution attempts to maximize network capacity while simplifying the management of the IoT [19].

RH2. Increasing demand for IoT applications requires improved resource management to protect QoS, which requires a centralized view of all available network resources. SDN provides a centralized view for controlling network resources and network flows [76]. SD-IoT networks may go beyond the processing capacity of nodes by increasing the demand for IoT applications, causing network congestion and network node overload, and reducing QoS [16, 17]. Traffic management involves dynamic load-balancing strategies to adapt to network circumstances, regulate network nodes, and enhance QoS to take advantage of SD-IoT global visibility and flexible control. Various aspects of load-balancing for IoT tasks can be optimized using the approaches provided by SDN [6]. In SD-IoT, the workload should be balanced between the resources by the SDN controller to provide the desired level of QoS. Load balancing is considered an important component in distributed computing technology that directly affects the availability of system applications and services [77, 78]. The classification of reviewed studies is shown in Fig. 7 based on the main purpose and strategy used.

RQ1. The concept of load-balancing in the SD-IoT network has been the subject of much research. Load balancing plays a significant role in increasing network QoS parameters. In much research, the relationship between the controller and the transmission nodes is considered to control traffic.

The SDN controller is an important component for load-balancing and distributing resources. So far, a variety of load-balancing approaches including migration, routing, demand response, scheduling, offloading, clustering, classification, allocation, admission control, aggregation, virtualization, placement, flow change, and architecture have been presented to improve QoS in the SD-IoT environment. Due to the importance of the load-balancing technique in Table 2, a column entitled load-balancing method has been added in each article to provide the roadmap to the reader. This is one of the novelties in this research paper. Figure 8 presents the percentage of load-balancing techniques in SD-IoT covered by various reviewed articles.

Some techniques based on artificial intelligence and meta-heuristic algorithms are proposed for routing, traffic engineering, resource allocation, management, security, traffic classification, and ultimately QoS optimization. In the event of network congestion, load-balancing algorithms split the traffic load across various flow channels. The load-balancing has been done at the server level in most of the research, and SDN controllers may be utilized to choose servers to transfer tasks. The performance objectives as well as the mechanism used in the studies were reviewed. In Tables 2, 3, 4, 5, 6, 7, 8, 9 and 10 load-balancing techniques based on the centralized or distributed architecture of SDN controllers were considered. The incoming traffic is balanced using the SDN controller and shares the load, which results in guaranteeing quality of service parameters. Controllers direct real-time traffic to resources based on QoS. In general, IoT uses SDN to maximize resource capacity utilization and thereby maintain QoS.

The comparisons of selected load-balancing techniques in the SD-IoT network were thoroughly

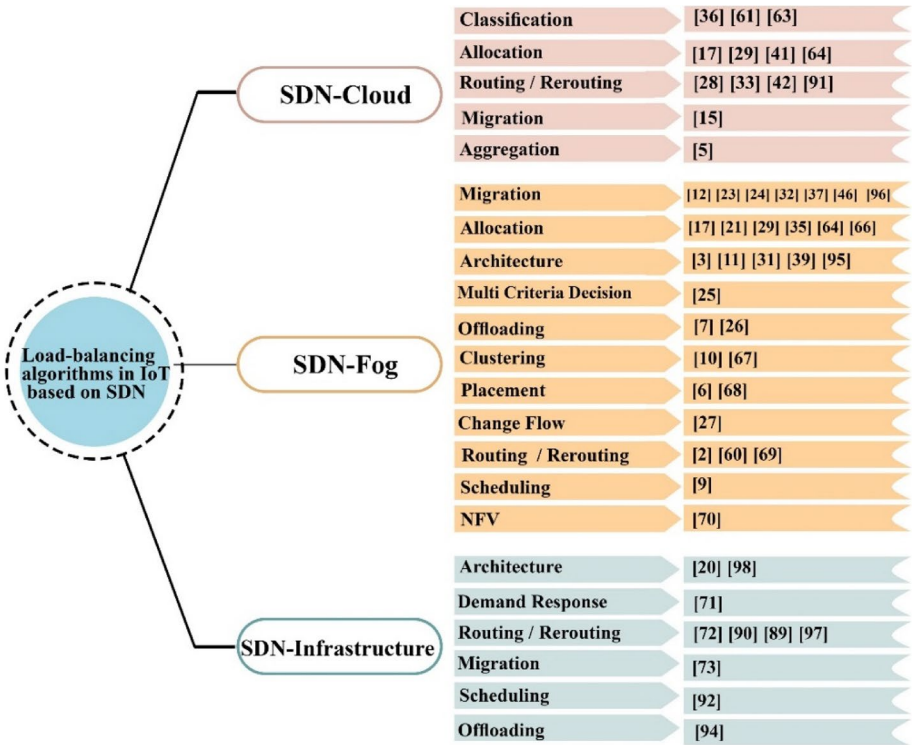

Fig. 7 Classification of studies reviewed based on the main purpose and strategy used

Table 2 Characteristics of routing-based load balancing techniques

Application	Objectives	Architecture	Balance entity	LB[a] method	Network type	Year	Ref
Multimedia	Delay, jitter, packet loss ratio	Centralized	Links	Routing	SDN-IoT	2018	[14]
Large scale	Delay, security, accessibility	Centralized	Switches, middleboxes	Routing	SDN-IoT	2018	[79]
Large scale	Scalability, security	Centralized	Fog resources	Rerouting	SDN-IoT/Fog	2018	[2]
Industry	Delay, throughput, resource utilization	Centralized	Edge servers	Routing	SDN-IIoT	2018	[80]
M2M	Response time	Centralized	Cloud server	Traffic detection and rerouting	SDN- M2M	2018	[32]
Industry	Throughput, delay	Centralized	Cloud server	Routing	SDN-IoT	2020	[38]
Large scale	Load-balancing	Distributed	Cloud servers	Routing	SDN-IoT	2020	[48]
FANET	Throughput, packet delivery ratio, delay	Distributed	Flying nodes	Routing	SDN- Ad-Hoc	2022	[65]
Health, Face recognition, lighting, and home sensor	Energy consumption, delay, cost	Centralized	Base stations	Routing	SD-WSN	2022	[67]
-	Link utilization, overhead, throughput	Centralized	Links	Rerouting	SDN- DCN	2022	[81]
Smart city	Throughput, energy consumption, delay	Centralized	Links	Rerouting	SDN- Fog	2022	[82]
High traffic	Throughput, resource utilization, response time	Centralized	Cloud servers	Routing	SDN- DCN	2022	[83]
Real-time	Resource utilization, response time, throughput	Distributed	Network flows	Routing	SD-IoT	2023	[84]

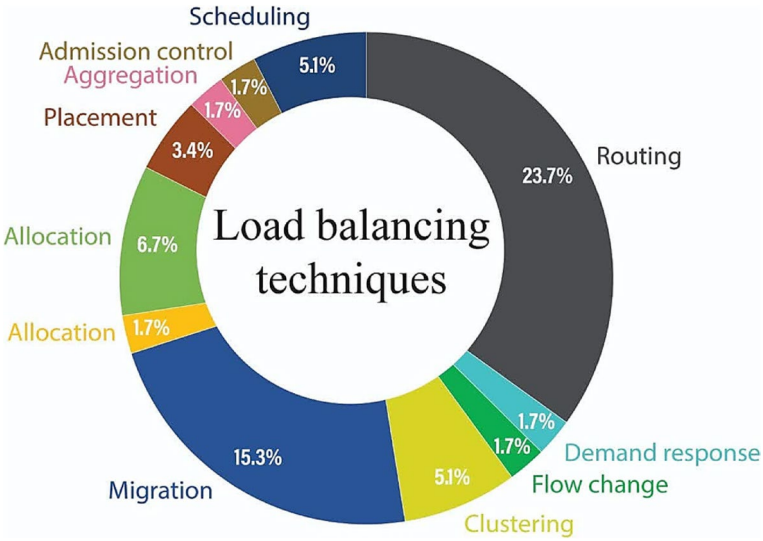

Fig. 8 Percentage of techniques considered in reviewed papers

Table 3 Characteristics of offloading-based load balancing techniques

Application	Objectives	Architecture	Balance entity	LB method	Network type	Year	Ref
5G	Delay, resource utilization, throughput	Centralized	Network cells	Offloading	SDN-WiFi	2015	[49]
Mobile devices	Throughput, load-balancing	Centralized	Cloudlets	Offloading	SDN- Cloudlet	2020	[7]
Vehicles, 5G	Response time, throughput	Centralized	Fog server	Offloading	SDN-Fog	2021	[29]
-	Resource utilization, delay	Centralized	Edge servers	Offloading	SD- Block Edge	2022	[85]

reviewed and analyzed, and the observations are summarized in Tables 2, 3, 4, 5, 6, 7, 8, 9 and 10. Load-balancing techniques have specific QoS parameters. Some researchers have developed a single criterion called the single-objective criterion, while others have found that several criteria, known as multi-objective and many-objective criteria, are more appropriate. Tables 2, 3, 4, 5, 6, 7, 8, 9 and 10 show the type of network and architecture, the load-balancing method, the desired QoS parameters, the balancing entity, and the application.

Table 11 lists the review studies on the issue of load-balancing in the IoT and SDN networks. Each research is represented by parameters such as review type, publication year, article identification process, taxonomy, network type, comparative analysis, future trends, and covered years. Only four articles have used the SLR method to study load-balancing methods, five articles on the IoT, and six articles on SDN. Therefore, the present study is the first study to review load-balancing methods in SD-IoT using the SLR method.

Based on the studies related to the SD-IoT network, some load-balancing techniques to improve QoS parameters in the SD-IoT environment were thoroughly studied and analyzed, along with the most important advantages and disadvantages. These approaches are applied to fog, and cloud layers for load balancing and to achieve better resource utilization. Based on the description in Table 12, we can say that there is a need to work more in the area of load-balancing in the fog computing environment, which mainly considers the processing power and overload of the resources. Table 12 shows the journal or conference type with the name of the publication and reference number, as well as the main subject, key contribution, advantages, and disadvantages, which compares existing load balancing techniques in detail based upon the approaches used.

RQ2. At each layer of the network architecture, QoS parameters affect the QoS of the entire network. All QoS parameters for analyzing the load-balancing efficiency are presented in this research so that network performance can be evaluated and recognized, as well as the benefits and drawbacks of load-balancing approaches. Some optimization parameters are in conflict with each other e.g., scalability, resource efficiency, reliability and bandwidth, power consumption, delay, and cost. To support load-balancing decisions, effective and efficient forecasting of QoS values, are important. Any change in network status is a reason to predict QoS before making

Table 4 Characteristics of architecture-based load balancing techniques

Application	Objectives	Architecture	Balance entity	LB method	Network type	Year	Ref
Vehicles	Delay	Centralized	Cloud, Fog servers	Architecture/ Allocation	SDN-IoV	2016	[33]
Real-time face recognition	Delay	Centralized	Cloud, Fog servers	Architecture/ Allocation	SDC-FN	2016	[86]
Large scale	Response time, resource utilization	Distributed	Controllers	Hierarchical architecture	SDN-IoT	2019	[20]
Large scale	Bandwidth, load-balancing	Distributed	Link/Server	Architecture	SDN- Fog/Cloud	2020	[11]
VoIP, Video	Scalability, delay	Distributed	Controllers	Hierarchical Controllers/ Allocation	SDN- Edge/Cloud	2020	[45]
Wi-Fi	throughput, packet loss ratio	Centralized	Access points	Architecture	SDN -Wi-Fi	2020	[44]
Image processing	Waiting, turnaround, processing times	Distributed	Device clusters	Hierarchical architecture of the control layer	SDN-IoT	2021	[22]
Critical scenarios	Response time, packet loss ratio, processing time	Distributed	Gateway	FoT[a] pattern	SDN- FoT	2021	[17]
Smart city	Response time, throughput	Distributed	Controllers	Architecture	SDN-IoT	2021	[36]
Industry	Throughput, packet loss ratio, response time	Distributed	Controllers	Architecture	SDN/NFV -IoT	2022	[3]
-	Response time, energy consumption, delay	Centralized	Fog nodes	Architecture	SDN- Fog	2022	[87]
Dense networks	Throughput, delay, packet loss rate	Centralized	Base stations	Architecture	SDN- IOMT	2022	[88]
Industry	Throughput, response time, delay, resource utilization	Distributed	Cloud servers	Architecture	SDN-IIoT	2023	[15]

Table 5 Characteristics of classification-based load balancing techniques

Application	Objectives	Architecture	Balance entity	LB method	Network type	Year	Ref
-	Response time, throughput	Distributed	Links	Task classification	SDN-Cloud	2016	[77]
5G	Response time, resource utilization	Centralized	Cloud server	Service classification	SDN-Cloud	2018	[89]
Multimedia	Transmission time, load-balancing	Centralized	Service functions	Packets classification	SDN/SFC-IoT	2018	[42]

Table 6 Characteristics of migration-based load balancing techniques

Application	Objectives	Architecture	Balance entity	LB method	Network type	Year	Ref
Health	Response time, packet delivery ratio, delay, throughput	Centralized	Controllers	Job migration	SDN- Edge	2020	[26]
High traffic	Response time, communication overhead	Distributed	Fog server, Controller	Switch migration	SDN- Edge	2020	[37]
5G	Response time, resource utilization	Distributed	Cloud server, Controller	Switch migration	SDN-IoT (Cloud)	2021	[25]
Dynamic scenarios	Response time, load-balancing, cost	Distributed	Controllers	Switch migration	SDN-IoT	2021	[52]
Vehicles	Delay, load-balancing	Distributed	Controllers	Switch migration	SD-VN	2021	[12]
Vehicles	Resource utilization, throughput, response time	Distributed	Range of switches	Switch migration	SDN/NFV -IoT	2022	[90]
Real-time	Response time, migration cost	Distributed	Controllers	Switch migration	SD-IoT	2022	[91]
Real-time	Delay, CPU utilization, Response time, cost	Distributed	Controllers	Switch migration	SDN-IoT	2023	[43]

Table 7 Characteristics of allocation-based load balancing techniques

Application	Objectives	Architecture	Balance entity	LB method	Network type	Year	Ref
6LoWPAN	Response time, reliability	Centralized	Gateways	Multi-criteria decision/ allocation	SDN-Fog	2020	[28]
Game	Delay, resource utilization	Distributed	Cloud servers	Allocation	SDN-Cloud	2020	[47]
5G	Load balancing	Distributed	Controllers	Allocation	SDN-5G	2022	[92]
Vehicles	Delay	Distributed	Cloud, Fog servers	Allocation	SDN-IoV	2022	[93]
5G	Bandwidth, response time, delay, packet loss	Distributed	Controllers	Allocation	SDN- IoT	2023	[94]
-	Cost, response time, energy consumption, CPU utilization	Centralized	Cloud servers	Allocation	SDN- IoT	2023	[95]

Table 8 Characteristics of scheduling-based load balancing techniques

Application	Objectives	Architecture	Balance entity	LB method	Network type	Year	Ref
Large scale	Throughput, Delay, Jitter	Distributed	Access points	Scheduling	SDN-IoT	2020	[9]
Real-time	Delay, energy consumption	Centralized	Edge server	Workload scheduling	SDN- Edge	2021	[41]
5G	Reliability, delay, energy consumption	Centralized	Edge nodes	Offloading/ Scheduling	SDN- IoT	2022	[96]
High load	Throughput, delay, packet loss rate	Centralized	Links	Scheduling	SDN-Cloud	2022	[97]
-	Load-balancing, delay, response time	Distributed	Fog nodes	Scheduling	SDN- IoT/Fog	2023	[98]

Table 9 Characteristics of clustering-based load balancing techniques

Application	Objectives	Architecture	Balance entity	LB method	Network type	Year	Ref
Smart city	Delay, throughput	Distributed	Controllers	Clustering	SDN/NFV-IoT	2020	[10]
Vehicles	Delay, accessibility, energy consumption, load-balancing	Distributed	Cloud, Fog servers	Hierarchical clustering	SDN-5G IoV	2021	[16]
Smart city	Communication cost	Distributed	Controllers	Clustering	SDN-IoT	2021	[56]

Table 10 Characteristics of other reviewed load balancing techniques

Application	Objectives	Architecture	Balance entity	LB method	Network type	Year	Ref
SG, Industry	Delay, packet delivery ratio	Centralized	Infrastructure	Demand response	SDN- AMI	2018	[99]
Critical scenarios	Response time, packet loss ratio	Centralized	Gateways	Flow change	SDN- FoT	2018	[30]
5G	Resource utilization, overhead	Centralized	Network slices	Network Function virtualization	SDN-NFV	2020	[100]
Vehicles	Delay	Centralized	Cloud, Edge servers	Aggregation	SDN- Edge/Cloud	2020	[5]
Smart City, Industry	Delay, packet delivery ratio, packet loss ratio	Distributed	Access points	Admission control	SDHW-IoT	2021	[101]
Industry	Energy consumption, cost, run time	Centralized	Edge nodes	Placement of tasks	SDN-Cloud	2021	[6]
-	Delay, load-balancing	Distributed	Controllers	Controller placement	SD-IoT	2022	[102]

load-balancing decisions. By predicting the QoS in the IoT, it is possible to increase the utilization of resources.

Numerous studies have been conducted on load-balancing techniques to reveal critical research issues. To improve QoS, various parameters were introduced by researchers in the research background in the realm of single-objective, two-objective, three-objective, or four-objective optimization problems. Having numerous QoS

Table 11 Related surveys in the field of load-balancing in SD-IoT networks

Ref	Year/ Publication	Survey	SLR	Question	Article identification process	Taxonomy	SDN	IoT	Comparative analysis	Future trends	Covered years
[64]	2017/ Elsevier	✓				✓		✓	✓	✓	2008–2017
[61]	2018/IEEE		✓	✓	✓	✓	✓		✓	✓	2008–2017
[103]	2018/ACM	✓				✓		✓	✓	✓	1997–2017
[63]	2019/ WILEY		✓	✓	✓	✓	✓		✓	✓	1988–2018
[104]	2019/ACM	✓		✓	✓	✓		✓	✓	✓	2001–2018
[68]	2020/ Springer		✓	✓	✓	✓		✓	✓	✓	2009–2019
[62]	2020/ Springer	✓				✓	✓		✓		2008–2020
[69]	2020/IEEE		✓	✓	✓	✓	✓		✓	✓	2008–2020
[105]	2020/ MDPI	✓					✓		✓	✓	2007–2020
[55]	2021/ Elsevier	✓				✓	✓		✓	✓	2008–2020
[106]	2021/ Springer	✓		✓	✓	✓		✓	✓	✓	2000–2020
[107]	2022/IEEE		✓	✓	✓	✓		✓	✓	✓	2002–2022
[108]	2022/IEEE	✓						✓	✓	✓	1996–2022
This Survey	2024/ Springer		✓	✓	✓	✓	✓	✓	✓	✓	2015–2024

parameters, maintaining a trade-off between parameters from the direction of users, service providers, and infrastructure is another novelty in this paper. A set of Pareto non-dominated solutions is formed as a result of parameter trade-offs. The SDN controller should select which solutions are best for load-balancing. Finally, the user may choose the preferable solution from this set depending on the criteria supplied.

By reducing QoS constraints, a multi-objective/many-objective model can be created based on maximum workflow [109]. In our study, 62 articles were selected to study the QoS parameters used in load-balancing techniques in the SD-IoT network. During the review, 18 parameters have been identified. The parameters used by different researchers in each of the techniques used by load balancing are listed in Table 13. Studies also focused on providing a load-balancing approach with optimal delay.

The QoS metrics considered in the load balancing approaches are grouped into two broad categories; Qualitative metrics and Quantitative metrics. Also, the metrics may be either dependent or independent. The taxonomy of the load balancing metrics is shown in Fig. 9.

Figure 10 graphically shows the percentage of load-balancing criteria considered in the articles under review. The most used criterion was the delay with 20.5%, followed by the response time with 15.9%. Other criteria such as throughput, resource efficiency, load-balancing, and packet loss rates included 15.1%, 11.4%, 6.8%, and 5.4%, respectively. Then, energy consumption and packet delivery rate accounted for 5.3%, and 3% respectively. Meanwhile, the least used criteria included reliability and network lifetime with approximately 0.7%. Figure 11 shows the percentage of considered parameters in centralized and distributed architectures. In centralized techniques, 58% and 39% of the researchers have attempted to improve delay and throughput, respectively. Also, response time, delay, and resource utilization have the highest attention by the researchers in the distributed techniques.

RQ3. QoS parameters help network infrastructure providers to improve network performance and infrastructures. Users can evaluate their needs by evaluating QoS parameters, and service providers can manage the performance and quality of their services with an emphasis on increasing satisfaction and attracting more users.

Many researchers are attempting to offer desired solutions for users, service providers, and infrastructure providers individually. But, the QoS may be analyzed from a combination of a variety of perspectives or directions, including users, service providers, and network infrastructure. Table 14 lists several QoS parameters in each direction, as well as a brief discussion of each.

Figure 12 shows some QoS parameters in each direction. For example, network architecture is important from the user, service provider, and infrastructure directions. According to the most common load-balancing techniques shown in Fig. 8, parameters such as delay, response time, resource efficiency, and load-balancing are improved from the user and service provider directions.

RQ4. As shown in Fig. 13, most studies were conducted in smart cities with high-traffic loads, such as industry and multimedia-related applications. It shows the importance of IoT applications in improving the quality of

Table 12 An overview of the various load-balancing solutions for SD-IoT

Disadvantages	Advantages	Key Contribution	Main Subject	Conference/Journal
- Not considering other aspects of QoS such as security, capacity - Lack of evaluation of energy consumption	+ Delay-sensitive task processing + Improved delay and QoS	Cloud/fog network architecture	Improved delay for real-time service processing	China Communications (IEEE) [33]
- Inefficiency of load-balancing scheme for saturation scenarios - Not using virtualization in FoT-Gateway	+ Reduced response time and lost samples	Programming to select a virtual machine	Load-balancing for FoT -Gateways and network links	International Conference on Internet of Things (IEEE) [30]
- Starvation in medium and low-priority applications	+ Acceptance control to ensure QoS of high-priority applications + Load-balancing between the routes and selection of the route with the maximum bandwidth + Reduced delay, jitter, and packet loss + Improved average end-to-end flow performance	Admission control	Application-aware QoS routing	Symposium on Computers and Communications (IEEE) [14]
- Lack of cost management	+ Improving resource efficiency and response time + Considering the types of services (service classification)	Type of service request	Load-balancing among cloud servers	IEEE Communications Magazine [89]
- Increased overhead at the data layer with frequent rerouting - Lack of attention to other criteria such as security - Dependence on the transfer rate - Maintenance of backup paths	+ Improved response time + Being used in human/machine networks + Reducing communication overhead	Traffic-aware load-balancing	Improved QoS by detecting and rerouting traffic	IEEE Internet of Things Journal [32]
- Single-point of failure - The lack of evaluation of other criteria of QoS such as congestion, overload, and security	+ Improved end-to-end delays and packet delivery rates + Reliable, scalable, and secure communication network	Traffic routing optimization	Global load-balanced routing the problem in the AMI network	IEEE Internet of Things Journal [99]
- Single-point of failure	+ Reduced delay and ensured safe network execution + Improved network security and stability	Deploying the middlebox in the right place	SDN-based data transfer security model in IoT based on middlebox	IEEE Internet of Things Journal [79]
- Lack of evaluation of energy consumption	+ Timely identification of attack models + Network scale support	Network partitioning and fog resource allocation	Large-scale intrusion detection with minimal delay	IEEE Access [2]
- The lack of evaluation of energy consumption and carbon emissions	+ Improved delay, throughput, and resource efficiency + Improved network performance	Data transfer architecture	Management of communication resources	IEEE Internet of Things Journal [80]
- Checking other parameters of QoS such as security and energy consumption	+ Improved response time, resource efficiency	Vertical (hierarchical) structure of the controller pool	Large-scale control layer load-balancing	IEEE Access [20]

Table 12 (continued)

Disadvantages	Advantages	Key Contribution	Main Subject	Conference/Journal
- Single-point of failure - Not using machine learning at max-overload - Not Considering heterogeneous resources	+ Improved response time, cost, resource utilization, and energy consumption + Increase task acceptance rate	Workload tolerance	QoS-aware load balancing	IEEE Access [95]
- Considering the incentive mechanism for well-behaved devices	+ Reduce task completion time + Resource utilization at edge devices	Token-based resource management	Efficient resource allocation of edge nodes	IEEE Sensors Journal [85]
-The migration process leads to increased network delay - Suitable for a limited number of target controllers to choose from - The lack of attention to the heterogeneity of tasks and resources - The high cost of migration for a large-scale environment (migration overhead)	+ Faster achievement of load-balancing at the control layer + Lower communication overhead and reduced response time	Multi-criteria decision making	Load-balancing in the control plane	IEEE Internet of Things Journal [37]
- Lack of conscious mechanisms for load-balancing of servers - Lack of QoS management at the layer of distributed SDN control and multi-domain network - Requires Network Function virtualization (NFV) for energy management and QoS	+ IoT traffic classification + Scalability of IoT infrastructure with maintaining QoS + Achieving justice and reducing the impact of corruption in QoS + Increasing throughput, and resource efficiency	Resource and QoS-aware framework	Scalable traffic management	IEEE Internet of Things Journal [38]
-Controller bottleneck used - The lack of privacy protection - Need to predict malicious activity with ML techniques - High migration overhead - Low scalability	+ Increased load-balancing and optimal use of resources + Increased security + Improved response time, packet delivery rate, delay, throughput, and overhead	Secure edge computing framework	Lightweight authentication scheme	IEEE Access [26]
- Lack of server and network integration using virtualization techniques - Not considering large-scale networks - Need for processing requests based on priority and resource allocation - Not evaluating traffic classification to ensure QoS	+ Minimization of the bandwidth costs + Link and server load-balancing + Considering load-balancing at network and server levels + Consider homogeneous and heterogeneous networks + Suitable for evaluating any fog computation topology	Cooperative Fog-Cloud Computing Architecture	Load-balancing to manage resources	IEEE Access [11]

Table 12 (continued)

Disadvantages	Advantages	Key Contribution	Main Subject	Conference/Journal
- Lack of extensive control of wireless parameters - Controller bottleneck used	+Improved packet loss rate, received signal strength, and throughput +Reduced dependence on the controller +No controller overload	QoS-aware load-balancing	Solving network congestion problems based on the load level	IEEE Access [44]
- Using queues and their effect on delay - The lack of evaluation of energy consumption	+Improved response time and reliability +Accelerated user access to sensor data	Load-balancing based on multi-criteria decision-making	Achieving load fairness and reducing service processing delays	IEEE Internet of Things Journal [28]
- Using queues and their effect on delay - No cost analysis - The lack of evaluation of energy consumption	+System stability in high current input fluctuations +Ensuring fairness in resource allocation	Cloud-edge hierarchical system	Increased scalability and reduced computational delay	IEEE Systems Journal [45]
- Single point of failure controller	+Avoid congestion and E2E delay +QoS guarantee, improving resource efficiency +Overhead reduction	Traffic engineering framework	Resource management among slices	IEEE Network [100]
- Non-consideration of other aspects of QoS such as scalability, network lifetime, and energy consumption	+Reduced data redundancy and service response delay +Mobility support	Cloud/edge computing	Service synchronization and data aggregation	IEEE Internet of Things Journal [5]
- Non-consideration of QoS criteria	+Mobility management, handover optimization +Improved scalability	Distributed hash-based monitoring structure	Flow control and mobility management in heterogeneous urban networks	IEEE Transactions on Parallel and Distributed Systems [9]
- Increased delay in providing almost optimal routing solutions - The lack of appropriate algorithms for traffic forecasting - Non-consideration of effective network performance parameters	+Improved load-balancing	Approximate routing algorithms	Routing optimization problem with TCAM capacity constraint	Journal of Communications and Networks (IEEE) [48]
- Using queues and their effects on delay - The lack of focus on resource efficiency - Need for scalability improvement	+Minimizing queues, request processing time, and balancing the controller load +Reduce immigration costs	Multi-objective optimization	Self-Adaptive Load-Balancing	International Conference on Autonomic Computing and Self-Organizing Systems (IEEE) [52]

Table 12 (continued)

Disadvantages	Advantages	Key Contribution	Main Subject	Conference/Journal
- Lack of improved switching efficiency among IoV services in fog clusters	+ Four-objective optimization + Minimum delay and energy consumption + Maximum load-balancing and service stability + Mobility support + Using heterogeneous computational resources + Improving real-time scalability	Architecture based on cloud-fog computing	Resource allocation in fog clusters	IEEE Transactions on Intelligent Transportation Systems [16]
- Increased energy consumption due to handover functions	+ Mobility support + Improved load-balancing, service response, and handover rates + Reduced congestion and increased service availability + Considering a heterogeneous network	Link assignment	Load-balancing at the control layer	14th International Conference on Communication Systems & Networks (IEEE) [92]
- Data redundancy in neighbouring tables sent to the controller - Lack of QoS management in the mode of distributed control	+ Reduced the number of messages + Reduced energy consumption + Prolong the network's lifetime	Load-balancing-based routing and clustering	Reduced load distribution and increased network lifetime	IEEE Access [67]
- Non-anticipation of QoS criteria with artificial intelligence techniques - Inattention to scalability - Non-examination of the heterogeneity of tasks and resources	+ Improved throughput, response time, and resource efficiency + Maximum CPU usage and minimum memory usage + Checking the migration cost and load-balancing rate	QoS -aware load-balancing framework	Improved QoS for network stability	IEEE Transactions on Green Communications and Networking [90]
- Further investigation to reduce the response time of the controller when a failure occurs - Non-consideration of the large scale	+ Increase link utilization, balance traffic loads, conserve table space + Reduce blocked packets, and alleviate table-full events	Reroute traffic flows	Load-balancing between links of switches	IEEE Transactions on Network and Service Management [81]
- Not considering controller overhead - Not using machine learning in a multi-controller scenario	+ Minimizing the impact of link failure + Better performance for delay-sensitive services + Improved throughput, energy consumption, delay	Efficient and reliable routing	Reliability-aware flows distribution	IEEE Transactions on Vehicular Technology [82]
- Data analysis of nodes with cloud technologies - Considering algorithms to compatible with 5G infrastructure	+ Throughput, delay, packet loss rate + Support wireless communication protocols + Time-sensitive prioritization	machine learning-based load-balancing	Distribution of nodes to base stations	IEEE Internet of Things Journal [88]
- Single point of failure controller	+ Improved throughput, round-trip delay, packet loss rate	Scheduling to calculate rerouting	Load balance of link traffic	International Conference on Measuring Technology and Mechatronics Automation (IEEE) [97]

Table 12 (continued)

Disadvantages	Advantages	Key Contribution	Main Subject	Conference/Journal
- Non-consideration of the packet processing priority - Controller bottleneck - Non-use of a combination of transmission paths for optimization of load-balancing - Testing non-extremity of fixed pockets/non-fixed pockets	+Classification of tasks by type of service +Improved data transfer time and load-balancing +Optimal local prevention	Service-Oriented SDN-SFC	Programming data transfer routes	Journal of Network and Computer Applications (Elsevier) [42]
- The need to minimize the cost of fulfilling requests	+Improving throughput and load-balancing +Considering communication delays and calculations +Maximum acceptance of requests	Cloudlet network framework on the mobile edge	Resource management and load-balancing	Future Generation Computer Systems (Elsevier) [7]
- Need for achieving complete network control among fog nodes with data layer Programming	+End-to-end routing +Reliable (bandwidth guarantee) +Improved throughput and response time +Efficient for large-scale systems +Increased system availability +Reduced delay in finding the offloading node	Dynamic offloading service between fog nodes	Finding the optimal node to handle tasks	Future Generation Computer Systems (Elsevier) [29]
- Non-implementation of network traffic based on real-world applications	+Improved packet delivery rate, packet loss, and delay	Admission control	Network flow management and congestion reduction	Computer Networks (Elsevier) [101]
- Higher communication overhead - The lack of identity and prevention of security attacks - The need for load balance between heterogeneous devices	+Reduced rotation and waiting time +Improved processing performance and use of network resources	Hierarchical architecture of controllers	Network management and load-balancing among devices	Journal of Network and Computer Applications (Elsevier) [22]
- The lack of resource efficiency	+Reduced delay and energy consumption +Solving resource pricing problems between the user and the edge resource provider	Energy-aware resource allocation	Improved QoS in edge computing	Sustainable Computing: Informatics and Systems (Elsevier) [41]
- The lack of evaluation of energy consumption, network lifetime, and packet delivery rate	+Meeting scalability and delay requirements +Improved response time, packet loss rate, and processing time	SDN network programming	Load-balancing for the Fog of Things Platforms	Journal of King Saud University—Computer and Information Sciences (Elsevier) [17]
- Migration overhead	+Prevent control plane overhead and distribute traffic efficiently +Reduce response time and cost of migration	Dynamic switch migration	Load-balancing among controllers	Computer Networks (Elsevier) [91]

Table 12 (continued)

Disadvantages	Advantages	Key Contribution	Main Subject	Conference/Journal
- Need for high privacy in a decentralized model - Achieving online task offloading and resource allocation with cooperating massive IoT networks	+Improved reliability, delay, energy +Privacy-preserving, security, and confidentiality by blockchain +Higher throughput and lower overhead	Blockchain-based Deep Reinforcement Learning	Energy-aware task scheduling and offloading	Future Generation Computer Systems (Elsevier) [96]
- Combining the presented approach with security-aware scheduling approaches	+Improved load-balancing, delay +Meeting the security requirements of IoT devices +Reduce response time	Security-aware workflow scheduler	Joint security and performance optimization	Journal of Information Security and Applications (Elsevier) [98]
- Need for optimization algorithms to load-balancing at the data plane - Need for hybrid machine learning algorithms for packet analysis	+Improved bandwidth, response time, delay, and packet loss +Considering security metrics such as detection accuracy and authentication time	Using honeypots, blockchains, and vSwitches	Providing secure multi-controller load-balancing	Future Generation Computer Systems (Elsevier) [94]
- Extend on dynamic network	+Optimizing packet delivery ratio, average latency, network lifetime, and energy consumption	Traffic flow optimization	Energy efficient routing	Sustainable Energy Technologies and Assessments (Elsevier) [84]
- Extending the proposed framework to a more large-scale SDN - Non-compliance of distributed architecture with security frameworks	+Optimization of migration time, response time, and controller load +Improved CPU usage, latency, communication cost, and throughput	Switch migration	Multi-domain SDN slave controller load balancing	Journal of King Saud University—Computer and Information Sciences (Elsevier) [43]
- Need to apply machine learning techniques - Non-implementation of Fog and Edge computing	+Increased security +Improved throughput, delay, response time, and resource utilization +Improved the durability, stability, and load balancing	Blockchain-SDN-based secure architecture	Traffic load management of real-time applications	Digital Communications and Networks (Elsevier) [15]
- Single-point failure controller used	+Reduced concerns about resource scarcity +Network congestion elimination +Improved delay, resource efficiency, and throughput +Less number of handovers	Data offloading and load-balancing	Reduced short-term resource shortages and network congestion	Journal on Wireless Communications and Networking (Springer) [49]
- Non-consideration of other aspects of QoS - Need for implementation of the algorithm in the real SDC-FN platform - The lack of evaluation of energy consumption	+Mobility support +Improved delay and response time	Cloud / Fog network architecture	Reduced real-time service delay	International Conference on Communication and Networking in China (Springer) [86]

Table 12 (continued)

Disadvantages	Advantages	Key Contribution	Main Subject	Conference/Journal
- Need for practical application and performance analysis - Interaction of unauthorized users with each other - Fault to check fault tolerance	+Improved delay and throughput +Improved load-balancing, scalability, accessibility, integrity, and network security +Heterogeneity support	Virtualization of network functioning	SDN-based distributed IoT network	Cyber Security and Computer Science (Springer) [10]
- Increase transfer time, and packet loss rate	+Improved response time and Throughput	Load-balancing optimization	Load distribution between SDN controllers in IoT application	Wireless Personal Communications (Springer) [36]
- Failure points of switches and controllers - Delaying the load-balancing function with multiple migrations - High cost of migration in a large-scale environment	+Increased response time, resource efficiency +Improved fault tolerance and reliability for migration	Monitoring and classification of the service	SDN-based load-balancing service	Wireless Personal Communications (Springer) [25]
- Improper management of multiple attacks - Need for the deployment of distributed blockchain technology for confidential data management and security	+Dealing with the epidemic damage of the Covid-19 virus in the industry +Ensuring security and reliability +Improved throughput, response time, and packet loss rate	SDN-based IoT architecture with NFV	Productivity of industry potentials in the Covid-19 pandemic	Cluster Computing (Springer) [3]
- Evaluation of load-balancing and traffic-based decisions for green cloud computing	+Improved throughput, bandwidth utilization, response time	Machine Learning for routing and server selection	Load-balancing in DCN Servers	Arabian Journal for Science and Engineering (Springer) [83]
- The need for scheduling with the load-balancing of flight nodes	+Improved throughput, packet delivery rates, and end-to-end delays +Increased network lifetime and traffic balancing	Computational load distribution between nodes	Distributed traffic congestion control	Electronics (MDPI) [65]
- Unstable performance - Need for checking other goals, such as reliability - Need for other searching criteria in the optimization algorithm - Non-evaluation of the selection of non-dominated solutions based on angle or distance	+Improved energy consumption, cost, and run time	Using multi-objective optimization	Load-balancing in cloud computing	Sensors (MDPI) [6]
- Need to expand the security parameters and more performance - Not testing the proposed technique in a real test-bed environment	- Improved Response time, energy consumption, and communication delay	Secure and energy-aware fog computing architecture	Load-balancing to improve utilization of resources	Sustainability (MDPI) [87]

Table 12 (continued)

Disadvantages	Advantages	Key Contribution	Main Subject	Conference/Journal
- Increased transfer delay - Lack of evaluation of energy consumption	+Improved E2E delay, resource efficiency +Achieving a fair allocation of resources +Maximization of profitability of service providers +Improved Quality of Experience (QoE)	Hierarchical architecture of controllers	Assigning requests to cloud data centers	Multimedia Systems Conference (ACM) [47]
- Single point failure central controller - Starvation in tasks with lower priority - Challenges the cloud for long distances with the user	+Improved response time, throughput +Assigning CPU resources to high-priority tasks	Task classification	Load-balancing in cloud network links	Workshop on Advanced Research and Technology in Industry Applications (Atlantis Press) [77]
- Inefficient use of resources - Non-consideration of QoS	+Minimization of the overall cost of communication	Controller placement based on clustering	Load-balancing between multiple controllers	Scalable Computing [56]
- Super-controller bottleneck - Non-consideration of migration costs and the distance between controllers and switches in overload controllers - Non-consideration of resource efficiency	+Reduced delay +Improved load-balancing	Real-time delay-based load-balancing	Simultaneous overload of multiple controllers	Computers, Materials & Continua (Tech Science Press) [12]
- Not testing the proposed strategy in real scenarios - Non-consideration of other performance criteria such as energy consumption and response time	+Improved load-balancing, +Reduced number of controllers and average delay and delay	Controller placement	Load-balancing and reduced packet release delay	Computers, Materials & Continua (Tech Science Press) [102]
- Non-evaluation of energy consumption, loss rate, and packet delivery	+Minimized delays +Reduced completion time of tasks +Potential for mobility and location-awareness	Cloud/edge computing architecture	Improved load-balancing and performance in latencies	Conference Proceedings (AIP) [93]

Table 13 Load-balancing QoS metrics in the reviewed techniques

Ref	Load-balancing	Scalability	Response /Execution time	Resource utilization	Delay/ Latency	Overhead	Throughput	Jitter	Packet loss rate	Packet delivery rate	Energy consumption	Accessibility	Processing time	Reliability	Waiting / transfer time	Cost	Security	Network lifetime
[25]			✓	✓														
[37]			✓			✓												
[89]			✓	✓														
[38]					✓		✓											
[29]			✓				✓9											
[14]					✓			✓	✓									
[101]					✓				✓	✓								
[16]	✓				✓						✓	✓						
[10]					✓		✓											
[26]			✓		✓		✓			✓								
[43]			✓	✓	✓											✓		
[36]			✓				✓											
[22]													✓		✓			
[42]	✓														✓			
[11]	✓			✓														
[41]					✓						✓							
[44]							✓		✓									
[17]			✓						✓				✓					
[32]			✓															
[28]			✓											✓				
[45]		✓			✓													
[7]	✓						✓											
[99]					✓					✓								
[79]					✓							✓					✓	
[52]	✓		✓													✓		
[15]			✓	✓	✓		✓											
[2]		✓															✓	
[49]				✓	✓		✓											
[33]					✓													
[86]					✓													
[30]			✓						✓									
[100]				✓		✓												
[77]			✓				✓											
[65]					✓		✓			✓								
[102]	✓				✓													
[12]	✓				✓													

Table 13 (continued)

Ref	Load-balancing	Scalability	Response /Execution time	Resource utilization	Delay/ Latency	Overhead	Throughput	Jitter	Packet loss rate	Packet delivery rate	Energy consumption	Accessibility	Processing time	Reliability	Waiting / transfer time	Cost	Security	Network lifetime
[56]						✓												
[3]			✓				✓		✓									
[6]			✓								✓					✓		
[92]	✓																	
[80]				✓	✓		✓											
[47]				✓	✓													
[5]					✓													
[20]			✓	✓														
[9]					✓		✓	✓										
[48]	✓																	
[67]											✓							✓
[93]					✓													
[90]			✓	✓			✓											
[91]			✓													✓		
[96]					✓						✓			✓				
[81]				✓		✓	✓											
[87]			✓		✓						✓							
[82]					✓		✓				✓							
[88]					✓		✓		✓									
[83]			✓	✓			✓											
[97]					✓		✓		✓									
[84]					✓					✓	✓							✓
[85]				✓	✓													
[98]	✓				✓													
[94]			✓	✓	✓				✓									
[95]			✓	✓							✓					✓		
Total	10	2	23	16	31	4	19	2	8	5	9	2	2	2	2	5	2	2

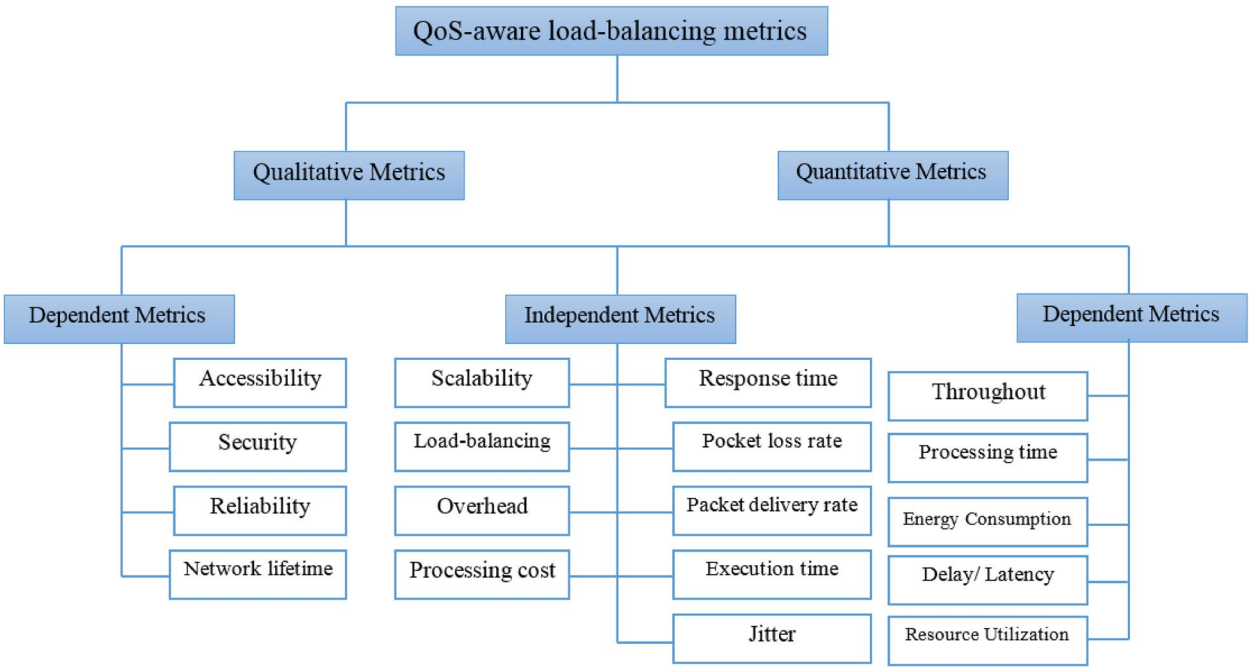

Fig. 9 Taxonomy of load balancing metrics

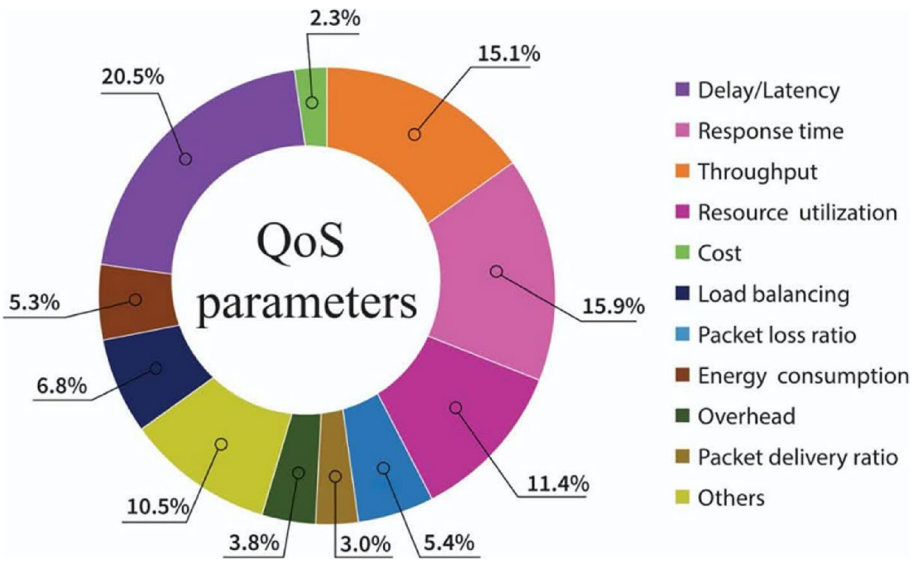

Fig. 10 Percentage of QoS criteria in the studied techniques

human life. Also, the load-balancing techniques used for each application are shown in Table 15.

RQ5. Identifying different simulation tools helps network developers to test load-balancing algorithms to improve QoS. A simulator is a framework that provides a virtual environment to test performance, deployment models, resource allocation, and load balancing. Simulators play a major role in testing and validating, before deploying on real hardware. Every simulator has some unique features and is used to evaluate different parameter performance. Table 16 shows the various load-balancing algorithms/ frameworks, their simulation tools and testbeds, the type of controller and data set, the load-balancing layer, the number of architectural layers, and QoS space dimensions. The studied load-balancing algorithms mainly focus on the selection of fog and edge resources

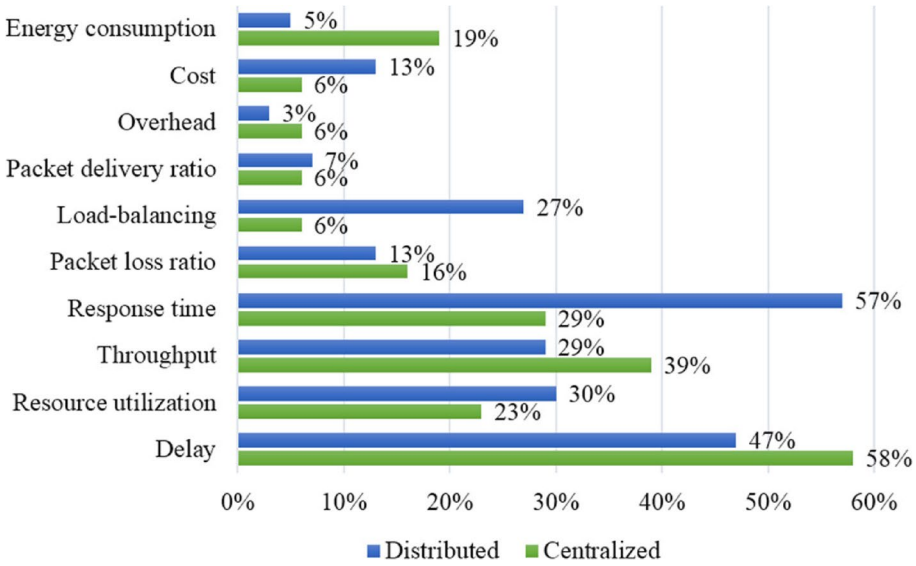

Fig. 11 Percentage of QoS criteria in centralized and distributed techniques

in the control plane to allocate input tasks. Mininet and Matlab tools are mainly used to evaluate the simulation results of load-balancing techniques in the reviewed studies.

In the following, some of the studied controllers are discussed concerning the load-balancing problem and achieving the desired QoS in SD-IoT networks. Table 17 shows of the Characteristics SDN controllers proposed by the researchers. Moreover, the percentage of evaluation tools/testbeds used in the reviewed articles are shown in Figs. 14 and 15, along with the controllers.

As can be seen in Fig. 16, Load Balancing algorithms can be single objectives, meaning they focus on one performance parameter whereas multi-objectives can focus on two or three parameters and many objectives on four or more than four parameters. Most algorithms reviewed in this literature are multi-objective. In Fig. 17, the load balance layers based on studies are shown.

Discussion

One of the most significant applications of SDN is load balancing in IoT, which is the process of distributing traffic evenly across multiple servers to optimize resource utilization and ensure high availability of services. SDN-based load-balancing techniques have gained popularity in recent years due to their flexibility, scalability, and cost-effectiveness. However, these techniques also face several challenges and issues that need to be addressed for their wider adoption and improved performance.

Aiming to identify the better mechanism for the load-balancing problem and to distinguish its pros and cons, the load-balancing mechanisms should be evaluated and compared to each other. The optimal migration policy depends on the number of migrations and produced traffic, current QoS parameter values, and SLA. Migrations reduction can minimize the number of active resources, which is effective in QoS parameters. Migration may have some problems such as security, delay, and energy consumption, as well as the complexities of deciding to choose a proper service provider. A good routing can satisfy the QoS requirement and minimize the energy consumption of the entire network. It also can balance distributing traffic loads over network links and increase the network lifetime. Rerouting is related to changing the selected route due to QoS requirements. It prevents network congestion and improves QoS, but may cause routing overhead. Network architecture is designed or evolved based on network parameters such as QoS and load-balancing, which is a time-consuming process. The policy is used during the network design process where there is a possibility of unbalanced resource loading. The offloading technique is done at the time of execution and according to the users' requirements.

Clustering is used in some issues such as routing, security, and quality of service. It can lead to load-balancing, increasing throughput, and network stability. The formation of stable clusters in an overloaded and dynamic network is necessary to reduce bandwidth consumption and prevent congestion. Classification is used for load balancing based on QoS and efficient use of resources. It reduces the waiting time for tasks and increases load-balancing. The resource allocation strategy guarantees the needs of the applications based on the provider's

Table 14 Directions of QoS parameters

Ref	Parameter	Directions			Description
		User	Service provider	Infrastructure provider	
[22]	Processing time	x	✓	x	The duration of packet execution on the processor increases the success rate of the network
[10, 16, 26, 41] [49, 99, 101]	Delay	✓	✓	x	The time required for packets to go from the source to the destination nodes. This delay includes a set of transmission, queue, processing, and release delays
[6, 16, 19, 41]	Energy consumption	x	✓	✓	Energy is consumed in the network when transferring data from source to destination. Most IoT devices are battery-powered, so routing protocols with suitable power consumption are preferred to extend the lifetime of IoT networks
[6, 52]	Cost	✓	✓	x	The user is charged for receiving the service
[26, 65, 99, 101]	Packet delivery rate	X	✓	x	The ratio of the total packets received by the destination node to the number of packets transmitted by the source node
[29, 38, 44, 77] [80]	Throughput	x	x	✓	The number of bytes transmitted over the network from the source to the destination node is determined in time
[14, 101]	Jitter	x	x	✓	Deviation from the average data reception delay resulting from changes in data arrival time over the network is due to congestion and changes in data packet paths
[20, 25, 43, 47] [49, 89]	Resource efficiency	x	✓	✓	The maximum amount of resources used (e.g., CPU, memory, and bandwidth) in the network
[28]	Reliability	x	x	✓	It checks the correct and timely performance of the tasks of the final devices as well as the continuity of service delivery
[2, 45, 78]	Scalability	x	✓	✓	It indicates the ability of the network to provide service to the increase of final devices and the amount of network load changes
[37, 56, 100]	Communication overhead	x	x	✓	The ratio of whole requests submitted by the network to the total incoming requests
[7, 11, 12, 16, 42] [52, 92, 102]	Load-balancing	x	✓	✓	Fair distribution of workload among network resources to increase network efficiency
[2, 79]	Security	✓	✓	x	A set of policies and arrangements are developed to enforce unauthorized access, prevent changes, and restrict access, attacks, and threats to available network resources
[16, 79]	Stability /accessibility	x	✓	✓	It refers to the accessibility and continuity of the service based on the maximum efficiency of the resources available in the network
[3, 6, 15, 32, 77]	Response time	✓	x	x	The transfer time, waiting, and processing to perform the task sent in the network

infrastructure resources. It aims for load-balancing and QoS and may have the possibility of resource overload.

An admission control mechanism can control the acceptance of users and avoid load imbalance of task flows. The aggregation of tasks is used to increase the lifetime of the network and reduce the number of tasks sent and received by the network nodes. However, in applications that require end-to-end confidentiality, task aggregation becomes a challenging practice. Virtualization technology through minimizing the operational costs satisfies the user needs in the shortest possible time. The position and location of the controller can increase the acceptance rate of tasks, lack bottlenecks, and improve QoS. At the same time, controller placement delay can be high in large-scale networks.

The purpose of flow changing is to split traffic into multiple paths with minimal congestion for optimal use of resources and reduced response time in high-scale networks. Demand response with developing a demand side management program to control and schedule the user's tasks are used for traffic routing and load balancing in the entire network. It can lead to a lower

Fig. 12 QoS parameters from different directions

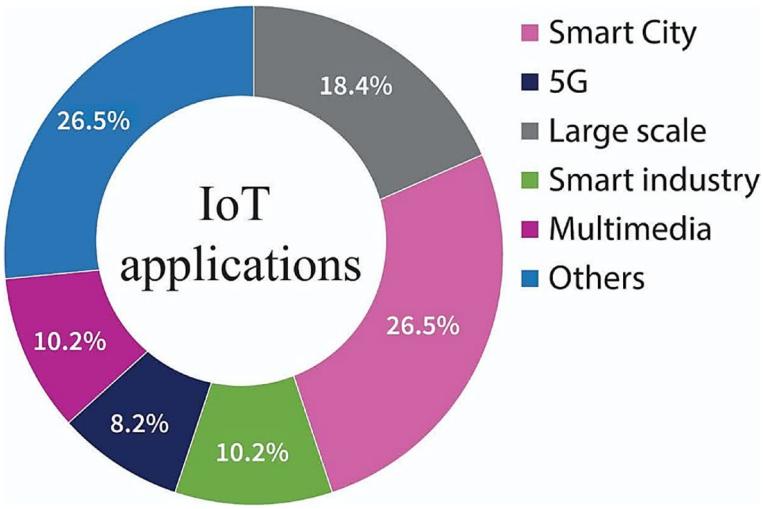

Fig. 13 IoT applications studied

Table 15 Application-based load-balancing techniques

Tech App	Routing	Architecture	Offloading	Allocation	Classification	Clustering	Scheduling	Migration	Others
Smart city		✓	✓	✓		✓		✓	✓
5G			✓	✓	✓				✓
Large scale	✓	✓					✓		
Industry	✓	✓							✓
Multimedia	✓				✓				
Real-time								✓	

Table 16 Common SD-IoT load-balancing tools/simulations

Controller	Dataset	Tools/Testbed	Dimensions	No. layers	Layer	Algorithm/Framework	Ref
ONOS, Open Daylight	-	Mininet[a]	Multi-objective	Three	Control	SMBLB algorithm (Greedy)	[25]
Floodlight	Iperf[a] Cbench	Mininet	Multi-objective	Three	Control	ESMLB framework	[37]
Floodlight	HTTPerf	OpenStack	Multi-objective	Three	Application	SBLB algorithm	[89]
Floodlight	Iperf	Kaa	Multi-objective	Three	Application	Future load forecasting based on fuzzy logic	[38]
Ryu	Iperf	Mininet	Multi-objective	Three	Control	Innovative algorithm	[29]
Ryu	D-ITG[a]	Mininet/WiFi	Multi-objective	Five	Infrastructure	AQRA algorithm	[14]
-	Random	Mininet/WiFi	Multi-objective	Three	Control	Ant Colony Optimization Algorithm (Opt-ACM)	[101]
-	Random	Hypervolume	Many-objective	Three	Application/ Control	Innovative algorithm	[16]
-	Random	Mininet/WiFi	Multi-objective	Four	Control	Distributed controller architecture	[10]
-	Random	Matlab	Multi-objective	Three	Control	Secure framework	[26]
ONOS	Iperf	Mininet	Many-objective	Three	Control	ESCALB algorithm	[43]
-	Random	NS2	Multi-objective	Three	Control	Spider monkey algorithm (LB-SMOA)	[36]
Floodlight	Caltech	Mininet/WiFi	Multi-objective	Three	Infrastructure	Hierarchical architecture	[22]
-	-	C# Environment	Multi-objective	Three	Application	Greed-based Service-oriented algorithm (GSOA)	[42]
-	-	AMPL/CPLEX	Multi-objective	Three	Application/ Control	Distributed architecture	[11]
-	-	CloudSimSDN	Multi-objective	Four	Control	Distributed edge computing architecture	[41]
Ryu	-	Mininet/NS3/ Wi-Fi	Multi-objective	Three	Control	QALB algorithm	[44]
Pox	Iperf	Mininet	Multi-objective	Three	Control	FoT based platform	[17]
Ryu	-	Real testbed	Single-objective	Three	Application	TALB algorithm	[32]
-	-	Matlab	Multi-objective	Three	Control	MCDM algorithm	[28]
-	Random	Matlab	Multi-objective	Three	Control	HECSDN architecture	[45]
Floodlight	-	Mininet/WiFi	Multi-objective	Three	Control	Architecture	[7]
Floodlight	Iperf	OpenStack	Multi-objective	Three	Infrastructure	OpenAMI	[99]
Pox	Iperf	Mininet	Multi-objective	Three	Infrastructure	Middlebox-Guard Framework	[79]
Ryu	D-ITG	Mininet	Multi-objective	Four	Control	Modified Greedy algorithm	[52]
Floodlight	Wireshark	Mininet-Wifi	Many-objective	Three	Application	Architecture	[15]
Ryu	UNSW-NB15	Mininet	Multi-objective	Three	Control	IDPS framework	[2]
-	Random	Matlab	Multi-objective	Three	Infrastructure	SDN-based LB	[49]
-	-	Real testbed	Single-objective	Four	Application/ Control	Modified particle swarm (MPSO-CO) / SDCFN architecture	[33]
-	-	Matlab	Single-objective	Four	Application/ Control	Fireworks algorithm/ SDC-FN architecture	[86]
Pox	Iperf	Mininet	Multi-objective	Four	Control	Proposed algorithm	[30]
-	Random	OMNeT + +	Multi-objective	Three	Control	TE Framework	[100]
Open DayLight	-	Mininet	Multi-objective	Three	Application	Ant colony algorithm	[77]
-	Cbrgen	NS2	Multi-objective	Three	Application	Firefly algorithm	[65]
-	-	Matlab/ Simulink	Multi-objective	Four	Control	Sunflower algorithm	[102]
Ryu	Iperf	Mininet	Multi-objective	Three	Control	SMLBAL algorithm	[12]
-	-	Matlab	Single-objective	Three	Control	Grey Wolf algorithm	[56]
-	-	Mininet/WiFi	Multi-objective	Three	Control	EdgeSDNI4COVID architecture	[3]
-	Random	Matlab	Multi-objective	Three	Control	Particle swarm algorithm	[6]

Table 16 (continued)

Controller	Dataset	Tools/Testbed	Dimensions	No. layers	Layer	Algorithm/Framework	Ref
-	Random	Matlab	Single-objective	Three	Control	Greedy approach	[92]
-	Random	Matlab	Multi-objective	Three	Control	Framework based on SDN and EC	[80]
Pox	Iperf	Mininet	Multi-objective	Three	Application	MOSP algorithm	[47]
-	-	C# Environment	Single-objective	Three	Application/ Control	SODA Framework	[5]
Open DayLight	-	Mininet	Multi-objective	Three	Control	Vertical architecture	[20]
Floodlight	-	Real testbed, OMNeT + +	Multi-objective	Three	Infrastructure	UbiFlow framework	[9]
-	-	Real testbed	Single-objective	Three	Application	Approximate algorithms	[48]
-	-	NS2	Multi-objective	Three	Infrastructure	Clustering algorithm	[67]
-	-	Matlab	Single-objective	Three	Application/ Control	Whale algorithm	[93]
-	-	Python environment	Multi-objective	Three	Infrastructure	LBSMT algorithm	[90]
Ryu	-	Mininet	Multi-objective	Three	Control	DSMLB framework	[91]
Ryu	-	Mininet	Multi-objective	Four	Control	DRL algorithm	[96]
Ryu	Iperf	Mininet	Multi-objective	Three	Infrastructure	L2RM framework	[81]
-	Bitbrain	COSCO	Multi-objective	Three	Control	Proposed architecture	[87]
-	Random	Mininet	Multi-objective	Three	Control	RAFDA algorithm	[82]
-	-	Riverbed Modeler	Multi-objective	Three	Infrastructure	MLA algorithms	[88]
Floodlight	Iperf	Mininet	Multi-objective	Three	Application	MRBS algorithm	[83]
Floodlight	Iperf	Mininet	Multi-objective	Three	Infrastructure	Proposed algorithm	[97]
-	-	Matlab	Many-objective	Three	Infrastructure	LSOA algorithm	[84]
Floodlight	-	Mininet/WiFi	Multi-objective	Four	Infrastructure	SDBlockEdge algorithm	[85]
Open DayLight	CIC	IoTSim-Osmosis	Multi-objective	Three	Control	S-FoS algorithm	[98]
Pox	-	NS-3	Many-objective	Three	Control	HBO algorithm	[94]
-	Random	Matlab	Many-objective	Three	Application/ Control	GWO algorithm	[95]

end-to-end delay and higher delivery ratio. By utilizing demand response approaches, it is possible to reduce or shift energy consumption from peak hours to periods of less demand. Scheduling the process of mapping tasks to available resources is somewhat based on user requirements. Task Scheduling is important to increase resource utilization by considering the balance between performance and QoS.

To conclude this section, a summary of widely adopted SD-IoT load-balancing techniques along with a description of the approach, layer used, and advantages and disadvantages of each technique is covered in Table 18. Most of the load-balancing techniques have focused on the cloud, and fog/control layers in the SD-IoT network. Mechanisms that are decided before starting the network and are used continuously over time are called static load-balancing techniques and are shown in blue color and Mechanisms that are used and changed depending on the conditions and QoS parameters during the network execution are called dynamic load-balancing techniques are shown in green color.

Table 19 contains a list of research questions along with conclusions to plan the survey on load balancing and to determine the current issues on load balancing. These questions contain the basic idea of this article.

Future research trends and opportunities

In SD-IoT load balancing mechanisms, there are still many issues and challenges that need to be discussed and resolved in the future by extra development and optimization of research. In the following, open research topics in the SD-IoT field will be discovered to answer Question 6 of the study.

Future Direction 1. Considering new QoS parameters: Examining the existing articles in the field of load balancing, it can be concluded by collecting and analyzing data that some QoS parameters such as availability, security, fault tolerance, reliability, network survival, and traffic patterns have been ignored in almost all reviewed articles. Consequently, the adoption of these parameters in load balancing can be an efficient roadmap for future researchers and increase the efficiency of current methods.

Future Direction 2. Multi-directional many-objective QoS: To maintain the QoS from different directions, load

Table 17 Characteristics of the discussed SDN controllers

Controller	Centralized	Distributed/ Flat	Open Source	Programming language		Scalability	Reliability	Consistency	MultiThreading	Modular	Platforms		
				Java	Python						Lin	Mac	Win
Floodlight	✓		✓	✓				✓	✓	✓	✓	✓	✓
OpenDaylight		✓	✓	✓		✓	✓	✓	✓	✓	✓		
ONOS		✓	✓	✓		✓	✓		✓	✓	✓		
RYU	✓		✓		✓	✓	✓		✓	✓	✓		
POX	✓		✓		✓			✓		✓	✓	✓	✓

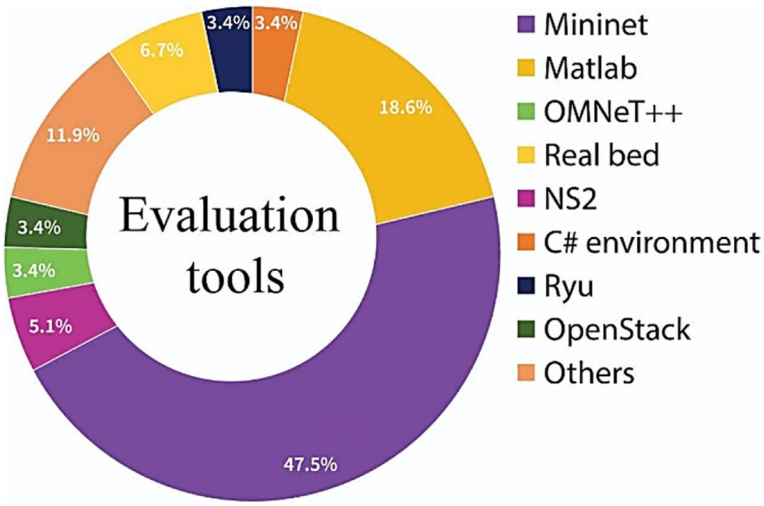

Fig. 14 Evaluation tools/techniques used in the reviewed articles

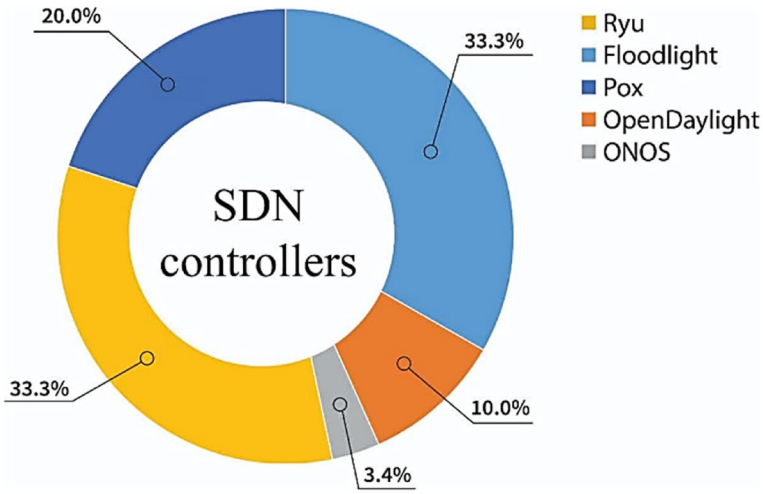

Fig. 15 Controllers were used in the reviewed articles

balancing is required for the proper use of provisioned resources. It is important to improve QoS by considering multiple combinations of parameters from different directions. A well-designed resource allocation mechanism is a significant issue for users, service providers, and infrastructure providers to keep improving QoS as many objective issues.

Future Direction 3. Multi-constraint many-objective QoS: Examining constraints with improvements in QoS parameters will be attractive as many-objective problems related to the network. With the arrival of new tasks to receive the service, the optimal trade-off is achieved between the objectives, and solutions stated in which the constraints will determine the acceptance of the solutions to the problem. Designing efficient traffic engineering algorithms that can handle diverse traffic patterns is a challenging task. Adopting new optimization algorithms can be very motivating for future studies. Optimization algorithms such as greedy [92], particle swarm [6], Ant colony [101], Lion Swarm [84], and grey wolf [56] are effective for many-objective problems in load-balancing.

Future Direction 4. Application-based load-balancing: Although the smart city is a widely used application in research papers and real-world scenarios, some other applications, such as healthcare and industry have been less studied. Exploration of data and extraction of information in such applications can be considered an interesting open topic for future studies. The type of application can influence the choice of load-balancing

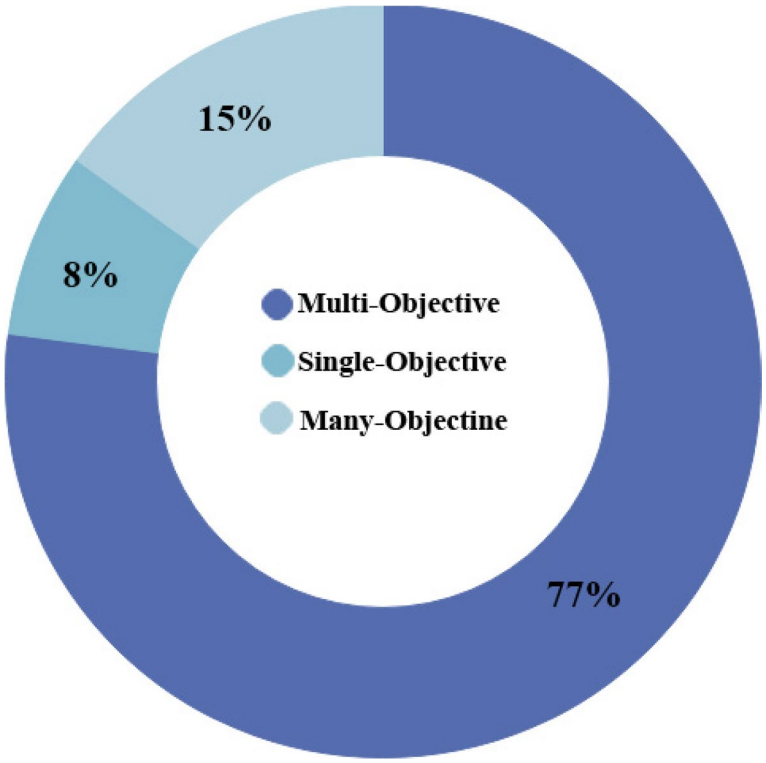

Fig. 16 Type of Reviewed Algorithms

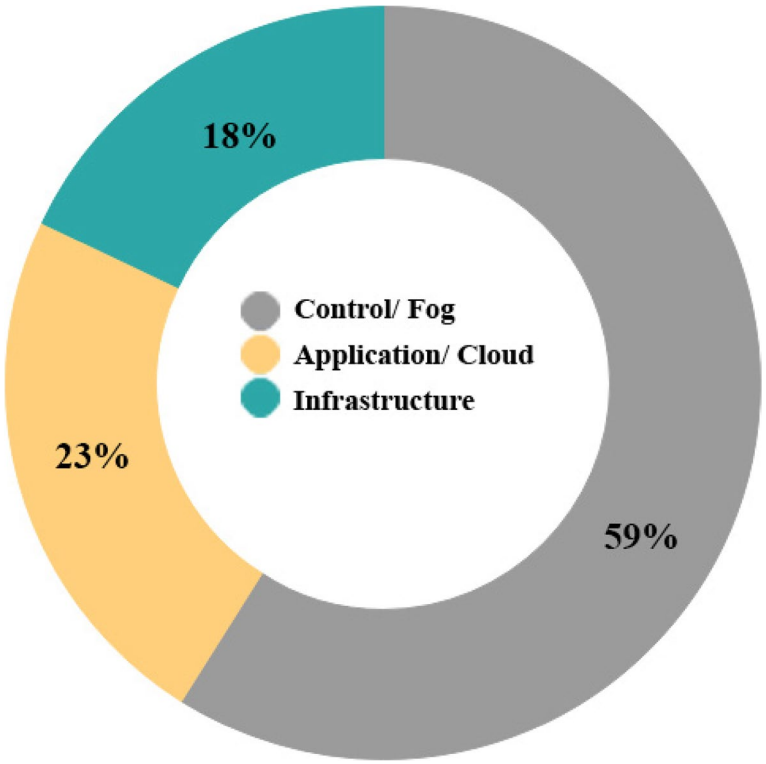

Fig. 17 Load balancing layer in the studied techniques

Table 18 Comparison of load-balancing techniques in SD-IoT

Disadvantages	Advantages	Layer	Function	Technique
- Communication overhead - Security challenges	+ Support user mobility + Minimizing the number of active servers + Reduce the possibility of rejecting tasks due to resource constraints + Energy management and response time reduction + Improving resource utilization	Cloud / Fog / Control	Services movement between resources	Migration
- Challenges in finding a reliable route in mobile networks - Routing overhead	+ Traffic management + Improvement of some QoS parameters	Cloud / Fog	Finding the best path to transfer tasks and their data	Routing
- Rerouting overhead	+ Low overhead and failure probability by choosing the right path + Prevent network congestion + Improving reliability, energy, delay, throughput, packet delivery rate, and resource utilization	Cloud / Fog	Finding alternative routes	Rerouting
- Architectural design costs	+ Improving the quality of service + Workload management + Reducing complexity and increasing efficiency of control units + Achieving distributed control of flows for scalability and reliability	Cloud / Fog / Control	Changing the centralized control layer to a distributed one	Architecture
- Possibility of overloading and underloading resources	+ Policies related to cache, congestion control, queuing, scheduling, green computing, and security + Improving the QoS	Cloud / Fog / Control	Change in network configuration and management	Policy
- Probability of violating resource capacity threshold - Lack of scalability	+ Improving response time, throughput, and delay + Network congestion control + Support of mobile applications + Maximizing request acceptance rate and optimizing the use of resources	Fog	Delivery of tasks to other resources for network balance and stability	Offloading
- Challenges in determining the number of clusters - Local minimum problem - Computational cost	+ Increasing stability + Effective resource management + Minimum communication cost between tasks + Improving load-balancing, scalability, availability, integrity, and security	Cloud / Fog / Control	Group tasks and resources based on their similarities	Clustering

Table 18 (continued)

Disadvantages	Advantages	Layer	Function	Technique
- Lack of trade-off between speed and accuracy of the traffic classification mechanism	+ Possibility of reliable communication + Network traffic forecasting + Improving load-balancing, response time, throughput, and resource utilization + Providing QoS in routers + Separation of traffic in different streams + Filtering and intrusion detection + Allocation of appropriate levels of QoS to different applications	Cloud / Infrastructure	Classification of traffic based on the type of tasks and their requirements based on predefined rules	Classification
- Possibility of resource overhead	+ Balance between performance and QoS + Creating justice on network nodes + Maximize network capacity	Cloud / Fog / Control	Mapping tasks on available resources Based on resource status	Allocation
- Increase in rejection of tasks	+ Network congestion control + Improving delay, packet delivery ratio, and packet loss ratio	Fog	Responsible for ensuring the authentic network load	Admission control
- Increased delay	+ Reducing network traffic load + Extending the lifetime of the network in energy-constraint networks + Reducing the number of packets sent in the network + Reducing communication costs and accurate data recovery + Reduce data redundancy	Cloud / Edge	Collecting and combining data from different sources	Aggregation
- Temporal mismatch between traffic load and resource availability - Balance and orchestration of virtual resources	+ Increase utilization and reduce costs + Efficient resource management + Increase security + Reducing energy consumption	Cloud / Fog	Implementing virtual functions on physical resources	Virtualization
- Taking too much time on large-scale controller placement - Link setup latency for a switch to controller communications - Possible delay in placing controllers	+ Development of receptive capacity + Increased network fault tolerability + Management flexibility + Improve overall network performance + Minimize runtime and delay + Choosing the best number of controllers	Control / Edge	Deployment of the controller in a suitable place	Controller placement
- Reduced security - Inefficiency in saturation scenarios	+ Congestion control + Reduced response time + Ensuring a proportional share of traffic for better use of resources	Fog	Splitting traffic into multiple paths	Flow change
- User categorization based on the application requirement - Prediction of user behaviour and energy requirements	+ Reducing energy costs for end users + Reducing peak energy consumption + Reducing the congestion of transmission lines	Infrastructure	Users' energy consumption management in response to resource conditions	User demand management
- Need to analyze the temporal-spatial - Mobility management in heterogeneous networks	+ Dynamic resource management + Improving task completion time, throughput, and delay	Fog	Efficient and appropriate assignment of resources based on task execution time	Scheduling

Table 19 Review research questions and conclusion

No. Questions	Conclusion										
Question 1 (load balancing techniques)	QoS-aware load-balancing techniques based on maximum use of centralized and distributed architectures are shown in the table below. Also, the table number related to the characteristics of the technique is mentioned										
		Routing	**Architecture**	**Offloading**	**Allocation**	**Classification**	**Clustering**	**Scheduling**	**Migration**	**Others**	
	Centralized	✓		✓		✓		✓		✓	
	Distributed		✓		✓		✓		✓		
	No. Table	2	4	3	7	5	9	8	6	10	
	The following table shows the use of load-balancing techniques in SD-IoT network layers. In general, further need for load balancing at the fog/controller layer is discussed										
		Routing	**Architecture**	**Offloading**	**Allocation**	**Classification**	**Clustering**	**Scheduling**	**Migration**		**Others**
	Application	✓				✓					
	Control/fog		✓	✓	✓		✓	✓	✓		✓
	Infrastructure	✓		✓				✓			
Question 2 (QoS parameters)	The load balancing techniques that lead to the maximum improvement of QoS parameters are shown in the table below										
	QoS Techniques	**Delay**	**Response time**	**Throughput**	**Resource Utilization**	**Cost**	**Packet loss ratio**	**Packet delivery ratio**	**Energy consumption**	**Security**	**Load balancing**
	Routing	✓	✓	✓	✓				✓	✓	
	Architecture	✓	✓	✓	✓		✓				
	Offloading	✓		✓	✓						
	Allocation	✓	✓		✓						
	Classification		✓								
	Clustering	✓									
	Scheduling	✓		✓					✓		
	Migration	✓	✓		✓	✓					✓
	Others	✓					✓	✓			
	Table shows that most researchers have concentrated on delay, followed by response time, throughput, and resource utilization										
Question 3 (multi-directions QoS)	As explained in Table 14, according to the QoS directions and SD-IoT network layers, three horizontal and vertical layers can be proposed, and nine parameters can be obtained from their combination										
	Directions layers			**User**		**Service provider**		**Infrastructure provider**			
	Application/Cloud			Cost		Throughput		Resource utilization			
	Control/Fog			Delay		Response time		Reliability			
	Infrastructure			Accessibility		Energy consumption		Energy consumption			

Table 19 (continued)

No. Questions	Conclusion
Question 4 (applications)	Figure 11 shows that most researchers have concentrated on smart city applications, followed by large-scale, industry, and multimedia
Question 5 (simulation tools)	General and specialized simulation tools and real beds can be used for simulation. Most researchers have used Matlab and Mininet tools to evaluate their techniques. Figure 12 shows the evaluation tools used in different research articles
Question 6 (future research)	Future research is described in Sect. 6, which is divided into three general categories • Application-based load-balancing • Controller-Architecture-based load-balancing technique • Multi-constraint many-objective QoS in multi-direction

technique. The use of load-balancing techniques in different applications is shown in Table 15.

Future Direction 5. Simulation tools observation: To answer question 5, such future research can be defined. Adopting appropriate simulation tools will increase the quality of the study. Almost all proposed load-balancing techniques are tested in the simulation environment. So there is a need to implement load balancing in the real environment. In addition to simulation tools, no single and comprehensive data set is observed in the reviewed articles. Some authors have accepted the randomly generated data as their data set. Appropriate reference datasets gathered in this field can be very useful for future studies.

Future Direction 6. Adaptive and load tolerant load-balancing (trade-off between load-threshold and usual QoS Parameters): As the number of tasks increases and resource saturation challenges require efficient load-balancing approaches to accepting maximum tasks. In this article, multi-directional QoS is discussed from the user, service provider, and infrastructure directions, which leads to many-objective QoS. By trade-off between workload and determining load threshold, two goals of QoS and load threshold are considered, which can be in conflict.

Future Direction 7. Controller-Architecture-based load-balancing technique: Centralized architecture involves a single controller that manages the entire network. While this approach may be effective for small-scale networks, it becomes more challenging to manage larger networks due to communication delays, bottlenecks, and reliability. Decentralized architecture, on the other hand, allows each controller to control its local area, making it a more feasible and economical option for larger networks. Distributed architecture combines controllers of both centralized and decentralized architecture, allowing local controllers to communicate with each other and with a central controller to achieve a global solution. This approach provides a more flexible and adaptable network, making it a promising area for future research. In multi-controller architectures, there is a need for a dynamic load-balancing mechanism that can handle burst traffic and adjust controller loads without compromising traffic balancing. Based on the studies, for centralized controllers, the routing technique is suitable, for multi-controllers, the architecture policy technique, and for distributed controllers, the migration technique is suitable.

Future Direction 8. Blockchain technology: To maintain, update, monitor, and exchange information between controllers to achieve load balance, blockchain architecture can be used. Blockchain technology can provide a secure and decentralized platform for load balancing, allowing for greater transparency and accountability in network management.

Future Direction 9. Prediction-based load-balancing: Depending on the behaviour of service requesters, the status of servers can be used to predict workload and classification of incoming traffic for efficient server allocation and QoS improvement based on several parameters, using Machine learning techniques. Machine learning techniques can help optimize SDN-based load balancing and traffic engineering algorithms by predicting traffic patterns and resource utilization.

Future Direction 10. Heterogeneous in QoS optimization: Most of the studied works have considered the available resources to be homogeneous for the simplicity of the problem, while resources such as CPU can have different capacities, costs, and energy consumption.

Conclusion

IoT applications and produced data volume and dynamic data flow requests have increased over time. It may lead to network overload and congestion, instability in network nodes (switches, controllers, and servers), and lowering QoS. Using SDN architecture, load-balancing methods can transfer the burden across resources and improve the QoS in IoT environments. SDN is a programmable and powerful solution for data flow control in the heterogeneous IoT network; it offers opportunities to design the network and improve QoS by separating the data layer from the control layer. So, in this paper, we described the properties of combined IoT and SDN networks, along with the architecture and effective role of SDN in IoT to meet QoS needs in different directions. A review of the load-balancing literature in SD-IoT networks and their typical features, the available solutions along with their advantages and disadvantages, related QoS parameters, and appropriate tools and testbeds are covered in this review.

To the best of the authors' knowledge, there is no comprehensive research that considered all QoS directions in the field of load-balancing. Also, some parameters such as availability, fault tolerance, and reliability are highlighted as influent QoS parameters. Furthermore, this paper discusses and compares load-balancing technics to show an overview of the latest approaches for upcoming works in this scope.

Authors' contributions

The authors was responsible for the data collection, analysis of the literature review, and paper organization. The authors read and approved the final manuscript.

Funding

This research received no specific grant from any funding agency.

Availability of data and materials
The data has been gathered from research papers and articles that are mentioned in Tables 2, 3, 4, 6, and 8.

Declarations

Ethics approval and consent to participate
All procedures are performed by the ethical standards. The present study is part of a Ph.D. dissertation.

Competing interests
The authors declare no competing interests.

Received: 19 February 2023 Accepted: 3 April 2024
Published online: 13 April 2024

References

1. Sheikh A, Ambhaikar A, Kumar S (2021) The Analysis of QoS parameters for IoT networks. Open J Sci Technol 4(1):40–50
2. Shafi Q, Basit A, Qaisar S, Koay A, Welch I (2018) Fog-Assisted SDN Controlled Framework for Enduring Anomaly Detection in an IoT Network. IEEE Access 6:73713–73723
3. Rahman A et al (2022) SDN–IoT empowered intelligent framework for industry 4.0 applications during COVID-19 pandemic. Cluster Comput 25(4):2351–2368
4. Muthanna MSA et al (2021) Cognitive control models of multiple access IoT networks using LoRa technology. Cogn Syst Res 65:62–73
5. Liu Y, Zeng Z, Liu X, Zhu X, Bhuiyan MZA (2020) A Novel Load Balancing and Low Response Delay Framework for Edge-Cloud Network Based on SDN. IEEE Internet Things J 7(7):5922–5933
6. Albowarab MH, Zakaria NA, Zainal Abidin Z (2021) Directionally Enhanced Binary Multi-Objective Particle Swarm Optimisation for Load Balancing in Software Defined Networks. Sensors 21(10):3356
7. Shahryari Sh, Hosseini-Seno SA, Tashtarian F (2020) An SDN based framework for maximizing throughput and balanced load distribution in a Cloudlet network. Future Generat Comput Syst 110:18–32
8. Qi Q, Tao F (2019) A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing. IEEE Access 7:86769–86777
9. Wu D et al (2020) Towards Distributed SDN: Mobility Management and Flow Scheduling in Software Defined Urban IoT. IEEE Trans Parall Distrib Syst 31(6):1400–1418
10. Mukherjee BK, Pappu SI, Islam MJ, Acharjee UK (2020) An SDN Based Distributed IoT Network with NFV Implementation for Smart Cities. in: Proceedings of the International Conference on Cyber Security and Computer Science. Springer, Dhaka, 325, pp 539–552
11. Maswood MMS, Rahman MR, Alharbi AG, Medhi D (2020) A Novel Strategy to Achieve Bandwidth Cost Reduction and Load Balancing in a Cooperative Three-Layer Fog-Cloud Computing Environment. IEEE Access 8:113737–113750
12. Babbar H et al (2021) Load balancing algorithm for migrating switches in software-defined vehicular networks. Computers, Materials & Continua 67(1):1301–1316
13. Akbar A, Ibrar M, Jan MA, Bashir AK, Wang L (2021) SDN-Enabled Adaptive and Reliable Communication in IoT-Fog Environment Using Machine Learning and Multiobjective Optimization. IEEE Internet Things J 8(5):3057–3065
14. Deng GC, Wang K (2018) An Application-aware QoS Routing Algorithm for SDN-based IoT Networking. in: Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC). IEEE, Natal, pp 00186–00191
15. Rahman A, Islam MJ, Band SS, Muhammad G, Hasan K, Tiwari P (2023) Towards a blockchain-SDN-based secure architecture for cloud computing in smart industrial IoT. Digital Communications and Networks 9(2):411–421
16. Cao B, Sun Z, Zhang J, Gu Y (2021) Resource Allocation in 5G IoV Architecture Based on SDN and Fog-Cloud Computing. IEEE Trans Intell Transport Syst 22(6):3832–3840
17. Batist E, Figueiredo G, Prazeres C (2022) Load balancing between fog and cloud in fog of things based platforms through software-defined networking. Journal of King Saud University-Computer and Information Sciences 34(9):7111–7125
18. Fernández-Fernández A, Cervelló-Pastor C, Ochoa-Aday L (2017) A Multi-Objective Routing Strategy for QoS and Energy Awareness in Software-Defined Networks. IEEE Commun Lett 21(11):2416–2419
19. Mohammadi R, Nazari A, Abdoli H, Nassiri M (2023) EQAFR: an energy and QoS aware fuzzy routing for internet of underwater things using SDN. Earth Sci Inf 16(4):3563–3577
20. Zhong X, Zhang L, Wei Y (2019) Dynamic Load-Balancing Vertical Control for a Large-Scale Software-Defined Internet of Things. IEEE Access 7:140769–140780
21. Ahammad I, Khan MAR, Salehin ZU (2021) QoS performance enhancement policy through combining fog and SDN. Simul Model Pract Theory 109:102292
22. Eghbali Z, Lighvan MZ (2021) A hierarchical approach for accelerating IoT data management process based on SDN principles. J Netw Comput App 181:103027
23. Shi Y, Zhang Y, Chen J (2020) Cross-layer QoS enabled SDN-like publish/subscribe communication infrastructure for IoT. China Commun 17(3):149–167
24. Zhou Y, et al (2014) A Load Balancing Strategy for SDN Controller based on Distributed Decision. in: Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, Beijing, pp 851–856
25. Kumar S, Cengiz K, Vimal S, Suresh A (2022) Energy efficient resource migration based load balance mechanism for high traffic applications IoT. Wireless personal communications 127(1):385–403
26. Li J et al (2020) A Secured Framework for SDN-Based Edge Computing in IoT-Enabled Healthcare System. IEEE Access 8:135479–135490
27. Javadzadeh G, Rahmani AM (2020) Fog Computing Applications in Smart Cities: A Systematic Survey. J Wireless Netw 26(2):1433–1457
28. Banaie F, Hossein Yaghmaee M, Hosseini SA, Tashtarian F (2020) Load-Balancing Algorithm for Multiple Gateways in Fog-Based Internet of Things. IEEE Internet Things J 7(8):7043–7053
29. Phan LA, Nguyen DT, Lee M, Park DH, Kim T (2021) Dynamic fog-to-fog offloading in SDN-based fog computing systems. Future Generat Comput Syst 117:486–497
30. Batista E, Figueiredo G, Peixoto M, Serrano M, Prazeres C (2018) Load Balancing in the Fog of Things Platforms Through Software-Defined Networking. in: Proceedings of the 2018 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). IEEE, Halifax, pp 1785–1791
31. Happ D, Wolisz A (2017) Towards gateway to Cloud offloading in IoT publish/subscribe systems. in: Proceedings of the 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, Valencia, pp 101–106
32. Chen YJ et al (2018) SDN-Enabled Traffic-Aware Load Balancing for M2M Networks. IEEE Internet Things J 5(3):1797–1806
33. He X, Ren Z, Shi C, Fang J (2016) A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles. China Commun 13(2):140–149
34. Li Y, Chen M (2015) Software-Defined Network Function Virtualization: A Survey. IEEE Access, special section on ultra-dense cellular netw 3:2542–2553
35. Abuarqoub A (2020) A Review of the Control Plane Scalability Approaches in Software Defined Networking. Future Internet 12(3):49
36. Mayilsamy J, Rangasamy DP (2021) Load Balancing in Software-Defined Networks Using Spider Monkey Optimization Algorithm for the Internet of Things. Wireless Pers Commun 116(1):23–43
37. Sahoo KS et al (2020) ESMLB: An Efficient Switch Migration based Load Balancing for Multi-Controller SDN in IoT. IEEE Internet Things J 7(7):5852–5860
38. Montazerolghaem A, Hossein Yaghmaee M (2020) Load-Balanced and QoS-Aware Software-Defined Internet of Things. IEEE Internet Things J 7(4):3323–3337

39. Ali J, Roh B, h, (2022) A Novel Scheme for Controller Selection in Software-Defined Internet-of-Things (SD-IoT). Sensors 22:3591
40. Tariq M, Naeem F, Vincent Poor H (2022) Toward Experience-Driven Traffic Management and Orchestration in Digital-Twin-Enabled 6G Networks. arXiv preprint arXiv:2201.04259
41. Singh A, Aujla GS, Bali RS (2021) Container-based load balancing for energy efficiency in software defined edge computing environment. Sustainable Computing: Informatics and Systems 30:100463
42. Chien WC et al (2018) A SDN-SFC-based Service-oriented Load Balancing for the IoT Applications. J Netw Comput App 114:88–97
43. Ali J, Jhaveri RH, Alswailim M, Roh BH (2023) ESCALB: An effective slave controller allocation-based load balancing scheme for multi-domain SDN-enabled-IoT networks. Journal of King Saud University-Computer and Information Sciences 35(6):101566
44. Manzoor S, Chen Z, Gao Y, Hei X, Cheng W (2020) Towards QoS-Aware Load Balancing for High Density Software Defined Wi-Fi Networks. IEEE Access 8:117623–117638
45. Lin FP, Tsai Z (2020) Hierarchical Edge-Cloud SDN Controller System with Optimal Adaptive Resource Allocation for Load-Balancing. IEEE Syst J 14(1):265–276
46. Wu D, Nie X, Deng H, Qin Z (2021) Software Defined Edge Computing for Distributed Management and Scalable Control in IoT Multinetworks. arXiv preprint arXiv:2104.02426
47. Amiri M, Al Osman H, Shirmohammadi S (2020) Resource Optimization through Hierarchical SDN-Enabled Inter Data Center Network for Cloud Gaming. Proceedings of the 11th ACM Multimedia Systems Conference. pp 166–177
48. Xu S, Wang X, Yang G, Ren J, Wang S (2020) Routing optimization for cloud services in SDN-based Internet of Things with TCAM capacity constraint. J Commun Netw 22(2):145–158
49. Duan X, Muhammad Akhtar A, Wang X (2015) Software-defined networking-based resource management: data offloading with load balancing in 5G HetNet. J. Wireless Commun, Netw, p 1
50. Bannour F, Souihi S, Mellouk A (2018) Distributed SDN Control: Survey, Taxonomy, and Challenges. IEEE Commun Surv Tutorials 20(1):333–354
51. Ahmed HG, Ramalakshmi R (2018) Performance Analysis of Centralized and Distributed SDN Controllers for Load Balancing Application. in: Proceedings of the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, Tirunelveli, pp 758–764
52. Min Z, Sun H, Bao S, Gokhale AS, Gokhale SS (2021) A Self-Adaptive Load Balancing Approach for Software-Defined Networks in IoT. in: Proceedings of the 2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS). IEEE, Washington, pp 11–20
53. Santos MAS, et al (2014) Decentralizing SDN's control plane. 39th Annual IEEE Conference on Local Computer Networks. IEEE, Edmonton, pp 402–405
54. Sarmiento DE, Lebre A, Nussbaum L, Chari A (2021) Decentralized SDN Control Plane for a Distributed Cloud-Edge Infrastructure: A Survey. IEEE Commun Surv Tutorials 23(1):256–281
55. Hamdan M et al (2020) A comprehensive survey of load balancing techniques in software-defined network. J Netw Comput App 174:102856
56. Keshari SK, Kansal V, Kumar S (2021) A Cluster based Intelligent Method to Manage Load of Controllers in SDN-IoT Networks for Smart Cities. Scal Comput: Pract Exper 22(2):247–257
57. Li L, Xu Q (2017) Load balancing researches in SDN: A survey, in: Proceedings of the 2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC). IEEE, Macau, pp 403–408
58. Huang V, Fu Q, Chen G, Wen E, Hart J (2017) BLAC: A Bindingless Architecture for Distributed SDN Controllers. in: Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks (LCN). IEEE, Singapore, pp 146–154
59. Punitha V, Mala C (2021) Traffic classification for efficient load balancing in server cluster using deep learning technique. J Supercomput 77(8):8038–8062
60. Pinto Neto EC, Callou G, Aires F (2017) An Algorithm to Optimise the Load Distribution of Fog Environments. in: Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, Banff
61. Neghabi AA, Jafari Navimipour N, Hosseinzadeh M, Rezaee A (2018) Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature. IEEE Access 6:14159–14178
62. Chahlaoui F, Dahmouni H (2020) A Taxonomy of Load Balancing Mechanisms in Centralized and Distributed SDN Architectures. SN Comput Sci 1(5):1–16
63. Neghabi AA, Jafari Navimipour N, Hosseinzadeh M, Rezaee A (2019) Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network. Int J Commun Syst 32(4):e3875
64. Jafarnejad Ghomi E, Rahmani AM, Nasih Qader N (2017) Load-balancing algorithms in cloud computing: A survey. J Netw Comput App 88:50–71
65. Kaur M, Prashar D, Rashid M, Khanam Z, Alshamrani SS, AlGhamdi AS (2022) An Optimized Load Balancing Using Firefly Algorithm in Flying Ad-Hoc Network. Electronics 11(2):252
66. Saeedi S, Khorsand R, Ghandi Bidgoli S, Ramezanpour M (2020) Improved Many-objective Particle Swarm Optimization Algorithm for Scientific Workflow Scheduling in Cloud Computing. Comput. Indust, Eng, p 147
67. Hajian E, Khayyambashi MR, Movahhedinia N (2022) A Mechanism for Load Balancing Routing and Virtualization Based on SDWSN for IoT Applications. IEEE Access 10:37457–37476
68. Pourghebleh B, Hayyolalam V (2020) A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things. Cluster Comput 23(2):641–661
69. Belgaum MR, Musa S, Alam MM, Su'ud MM, (2020) A Systematic Review of Load Balancing Techniques in Software-Defined Networking. IEEE Access 8:98612–98636
70. Li L, Rong M, Zhang G (2014) An Internet of Things QoS Estimate Approach based on Multi-Dimension QoS, in: Proceedings of the 2014 9th International Conference on Computer Science & Education. IEEE, Vancouver, pp 998–1002
71. Liu L, Chang Z, Guo X, Mao S, Ristaniemi T (2018) Multi-objective Optimization for Computation Offloading in Fog Computing. IEEE Internet Things J 5(1):283–294
72. Bouloukakis G et al (2021) PrioDeX: a Data Exchange middleware for efficient event prioritization in SDN-based IoT systems. ACM Trans Internet Things 2(3):1–32
73. Li J et al (2019) Service Migration in Fog Computing Enabled Cellular Networks to Support Real-Time Vehicular Communications. IEEE Access 7:13704–13714
74. Ashipala A, Abolarinwa JA (2023) SDN-Enabled Data Offloading and Load Balancing in WLAN and Cellular Networks. Proceedings of International Conference on Information systems and Emerging Technologies
75. Anoushee M, Fartash M, Akbari Torkestani J (2023) An intelligent resource management method in SDN based fog computing using reinforcement learning. Computing 106:1051–1080
76. Kamarudin IE, Ameedeen MA, Umar Ong MI, Zabidi A (2023) QSroute : A QoS Aware Routing Scheme for Software Defined Networking. IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS). Penang, Malaysia, pp 388–391
77. Lin W, Zhang L (2016) The load balancing research of SDN based on ant colony algorithm with job classification. in: Proceedings of the 2016 2nd Workshop on Advanced Research and Technology in Industry Applications (WARTIA-16). Atlantis Press. pp 472–476
78. Ghomi EJ, Rahmani AM, Qader N (2017) Load-balancing algorithms in cloud computing: A survey. J Netw Comput App 88:50–71
79. Liu Y, Kuang Y, Xiao Y, Xu G (2018) SDN-Based Data Transfer Security for Internet of Things. IEEE Internet Things J 5(1):257–268
80. Li X, Li D, Wan J, Liu C, Imran M (2018) Adaptive Transmission Optimization in SDN-Based Industrial Internet of Things With Edge Computing. IEEE Internet Things J 5(3):1351–1360
81. Wang YC, You SY (2018) An Efficient Route Management Framework for Load Balance and Overhead Reduction in SDN-Based Data Center Networks. IEEE Trans Netw Serv Management 15(4):1422–1434
82. Ibrar M, Wang L, Shah N, Rottenstreich O, Muntean GM, Akbar A (2023) Reliability-Aware Flow Distribution Algorithm in SDN-Enabled Fog Computing for Smart Cities. IEEE Trans Vehicular Technol 72(1):573–588

83. Begam GS, Sangeetha M, Shanker NR (2022) Load Balancing in DCN Servers through SDN Machine Learning Algorithm. Arab J Sci Eng 47(2):1423–1434
84. Keshari SK, Kansal V, Kumar S, Bansal P (2023) An intelligent energy efficient optimized approach to control the traffic flow in Software-Defined IoT Networks. Sustainable Energy Technol Assess 55:102952
85. Latif Z, Lee C, Sharif K, Helal S (2022) SDBlockEdge: SDN-Blockchain Enabled Multihop Task Offloading in Collaborative Edge Computing. IEEE Sensors J 22(15):15537–15548
86. Shi C, Ren Z, He X (2016) Research on load balancing for software defined cloud-fog network in real-time mobile face recognition. Proceedings of the International Conference on Communications and Networking in China. Springer, Cham, pp 121–131
87. Singh J, Singh p, Amhoud EM, Hedabou M, (2022) Energy-Efficient and Secure Load Balancing Technique for SDN Enabled Fog Computing. Sustainability 14(19):12951
88. Cicioğlu M, Çalhan A (2022) A Multiprotocol Controller Deployment in SDN-Based IoMT Architecture. IEEE Internet of Things J 9(21):20833–20840
89. Abdelltif AA, Ahmed E, Fong AT, Gani A, Imran M (2018) SDN-Based Load Balancing Service for Cloud Servers. IEEE Commun Magazine 56(8):106–111
90. Babbar H, Rani S, Bashir AK, Nawaz R (2022) LBSMT: Load Balancing Switch Migration Algorithm for Cooperative Communication Intelligent Transportation Systems. IEEE Trans Green Commun Netw 6(3):1386–1395
91. Zafar S, Lv Z, Zaydi NH, Ibrar M, Hu X (2022) DSMLB: Dynamic switch-migration based load balancing for software-defined IoT network. Comput Netw 214:109145
92. Basu D, Jain A, Rao AS, Ghosh U, Datta R (2022) Multi-cluster Dynamic Load Balancing in vSDN-enabled 5G Heterogeneous Network for Smart Cyber-Physical Systems. in: Proceedings of the 2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS). IEEE, Bangalore, pp 871–876
93. Darade SA, Akkalakshmi M, Pagar N (2022) SDN based load balancing technique in internet of vehicle using integrated whale optimization method. AIP Conf Proc 2469(1):020006
94. Abdulqadder IH, Zou D, Aziz IT (2023) The DAG blockchain: A secure edge assisted honeypot for attack detection and multi-controller based load balancing in SDN 5G. Future Generat Comput Syst 141:339–354
95. Rostami M, Goli-Bidgoli S (2023) TMaLB: A Tolerable Many-Objective Load Balancing Technique for IoT Workflows Allocation. in IEEE Access 11:97037–97056
96. Sellami B, Hakiri A, Yahiac SB (2022) Deep Reinforcement Learning for energy-aware task offloading in join SDN-Blockchain 5G massive IoT edge network. Future Generat Comput Syst 137:363–379
97. Wang Y, Liu R, Li Y, Chen Z, Zhang N, Han B (2022) SDN Controller Network Load Balancing Approach for Cloud Computing Data Center. 2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). IEEE, Changsha, pp 242–246
98. Javanmardi S, Shojafar M, Mohammadi R, Persico V, Pescapè A (2023) S-FoS: A secure workflow scheduling approach for performance optimization in SDN-based IoT-Fog networks. J Inf Sec App 72:103404
99. Montazerolghaem A, Yaghmaee Moghaddam MH, Leon-Garcia A (2018) OpenAMI: Software-Defined AMI Load Balancing. IEEE Internet Things J 5(1):206–218
100. Qu K, Zhuang W, Ye Q, Shen X, Li X, Rao J (2020) Traffic Engineering for Service-Oriented 5G Networks with SDN-NFV Integration. IEEE Network 34(4):234–241
101. Kumar R, U V, Tiwari V, (2021) Opt-ACM: An Optimized load balancing based Admission Control Mechanism for Software Defined Hybrid Wireless based IoT (SDHW-IoT) network. Comput, Netw, p 188
102. Hans S, Ghosh S, Kataria A, Karar V, Sharma S (2022) Controller Placement in Software Defined Internet of Things Using Optimization Algorithm. CMC-Computers, Materials & Continua 70(3):5073–5089
103. Soo WK, Ling TCh, Maw AH, Win ST (2018) Survey on Load-Balancing Methods in 802.11 Infrastructure Mode Wireless Networks for Improving Quality of Service. ACM Comput Surv. 51(2):1–21
104. Kumar P, Kumar R (2019) Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey. ACM Comput Surv 51(6):1–35
105. Semong T et al (2020) Intelligent Load Balancing Techniques in Software Defined Networks: A Survey. Electronics 9(7):1091
106. Kaur M, Aron R (2021) A systematic study of load balancing approaches in the fog computing environment. J Supercomput 77(8):9202–9247
107. Haghi Kashani M, Mahdipour E (2022) Load Balancing Algorithms in Fog Computing: A Systematic Review. IEEE Trans, Serv Comput. (Early Access)
108. Gures E, Shayea I, Ergen M, Azmi MH, El-Saleh AA (2022) Machine Learning-Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey. IEEE Access 10:37689–37717
109. Zhu R, Wang H, Gao Y, Yi S, Zhu F (2015) Energy Saving and Load Balancing for SDN Based on Multi-objective Particle Swarm Optimization. Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, pp 176–189

Publisher's Note