Journal of Al-Qadisiyah for Computer Science and Mathematics Vol. 15(3) 2023 , pp Comp. 164–182

University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

A State-of-the-Art Survey and Taxonomy of Classification, Algorithms, and Techniques for Load Balancing in SDN

Maghrib Abidalreda Maky Alrammahi [a], Wesam S. Bhaya [b]

[a] ITRDC, University of Kufa, Najaf, Iraq and College of IT, University of Babylon Babylon, Iraq, maghrib.alramahi@uokufa.edu.iq

[b]Information Networks Department, College of Information Technology, University of Babylon, Iraq, wesambhaya@uobabylon.edu.iq

ARTICLE INFO

Article history:

Received: 15 /07/2023

Rrevised form: 08 /09/2023

Accepted : 12 /08/2023

Available online: 30 /09/2023

Keywords:

Software Defined Networking (SDN),

Traditional Network,

load balancing (LB),

OpenFlow Protocol,

SDN Controller,

LB-Classification,

LB-Algorithms,

LB-Techniques

ABSTRACT

The management of traditional networks has become increasingly complex due to the expansion of the network and the development of new technologies such as cloud computing, the Internet of Things (IoT), and big data. Therefore, it is imperative to transition from operating within conventional networks to utilizing advanced networks capable of effectively managing modern technology. One of the most significant advancements in networking is the implementation of software-defined networks (SDN). SDNs aim to decouple the control plane, which controls network functions, from the data plane, which handles data transmission. This separation enhances the flexibility of network management. The distribution of traffic inside SDN networks plays a crucial role in enhancing network performance and response. Implementing Load Balancing (LB) enhances overall system performance and guarantees the efficient and dependable utilization of network resources. This research aims to comprehensively analyze recent research studies and the taxonomy of LB in SDN, such as classification, algorithms, and techniques. This research provides a comprehensive, state-of-the-art survey of LB in SDN according to LB-Classification, LB algorithms, and LB-Techniques. This research proposed a modern taxonomy for LB-Classification based on two factors: scheduling and models. Also, it proposed a new taxonomy of LB-Algorithms based on three types (static, dynamic, and hybrid) and a taxonomy for a third type (hybrid) consisting of three kinds (hybrid-LB, hybrid dynamic-LB, and hybrid static-LB). Finally, this research proposed a modern classification of LB-Techniques based on six types: (Controller -LB, Server -LB, Path Selection and Re-route - LB, Scheduling Management and Queue -LB, Artificial Intelligence -LB, and Wireless and Wi-Fi-LB).

MSC..

https://doi.org/10.29304/jqcm.2023.15.3.1273

∗Corresponding author : Maghrib Abidalreda Maky Alrammahi

Email addresses: maghrib.alramahi@uokufa.edu.iq

Communicated by 'sub etitor'

1. Introduction:

Technological improvements have exerted a significant impact on the amount of internet traffic. As a result, there is an increasing demand to enhance the network's cognitive capabilities, operational effectiveness, and dependability [1]. The current constraints associated with adjusting to rapid expansion in traditional networks necessitate the development of more robust infrastructure to accommodate this increase [2]. The management of networks often presents issues in traditional network systems as a result of the distinctive properties of network hardware components, such as switches, routers, and load balancers, which are exclusive to particular vendors. Hence, these devices demonstrate a correlation between their data and control planes. The level to which their functionality can be changed is limited if they utilize network management systems specific to the particular vendor [3].

Figure 1 thoroughly represents both traditional and software-defined network topologies inside a unified framework. Software-defined networking (SDN) facilitates separating network services from vendor-specific hardware devices, separating the control plane, which manages the actual forwarding of data packets, from the data plane, which decides on network traffic [4]

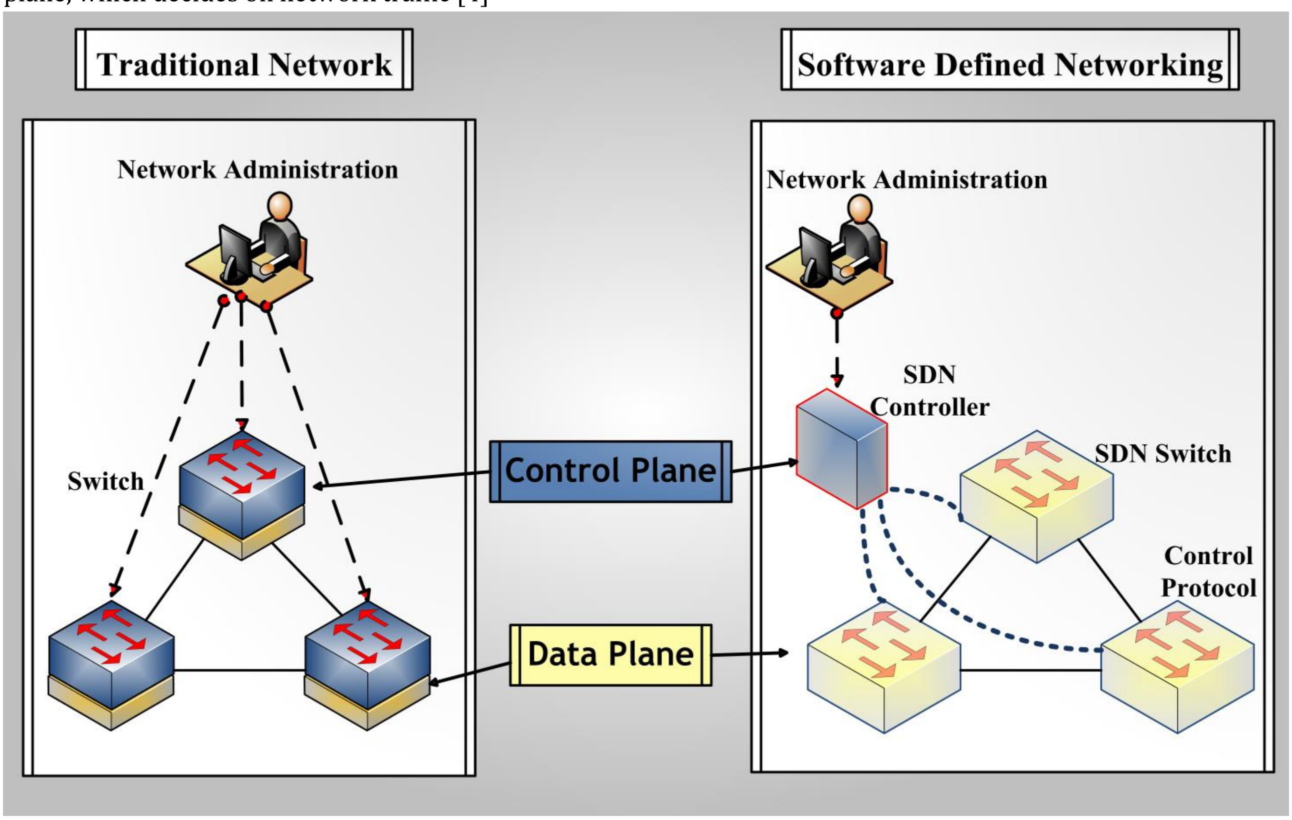

Fig 1- ***SDN vs. conventional networks explained [4].***

SDN (Software-Defined Networking) [5, 6] and NFV (Network Functions Virtualization) [7, 8] have been suggested as solutions to traditional network issues.

Using Software abstractions such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV) have the potential to be implemented across a wide range of devices and hardware platforms inside the network architecture and infrastructure. Due to this software abstraction, the two technologies can avoid the limitations inherent in conventional networks. SDN facilitates the ability to programmatically define and modify the configuration and operation of a network. Additionally, it allows for the network's programming [9, 10].

SDN controllers provide centralized administration of the entire network [11]. The concept of SDN involves transferring decision-making processes to the control plane, providing a level of abstraction for the physical network devices [12]. The control plane is responsible for all network intelligence, encompassing tasks such as packet forwarding and network management policies [13]. The structure of SDN facilitates efficient management and scalability of the network [14]. Given that network design and management do not rely on certain vendors, employing less complex network equipment is possible.

The practice of load balancing, which involves the equitable distribution of network traffic among various network devices, is a critical component in the design and administration of networks [15]. An adept load balancer improves several network metrics, such as latency, response time, resource consumption, and throughput. Conventional networks often utilize a dedicated server to facilitate load balancing [16].

Figure 2 depicts traditional networks that commonly employ a dedicated server for load balancing. Consequently, conventional networks must enhance the precision of their load-balancing mechanisms to effectively accommodate the growing demands of the network [17]. The programmability of SDN load balancing enables precise and efficient manipulation of rules inside the flow table, facilitating the addition or deletion of rules. Most SDN load-balancing flow tables rely on load balancing algorithms [18].

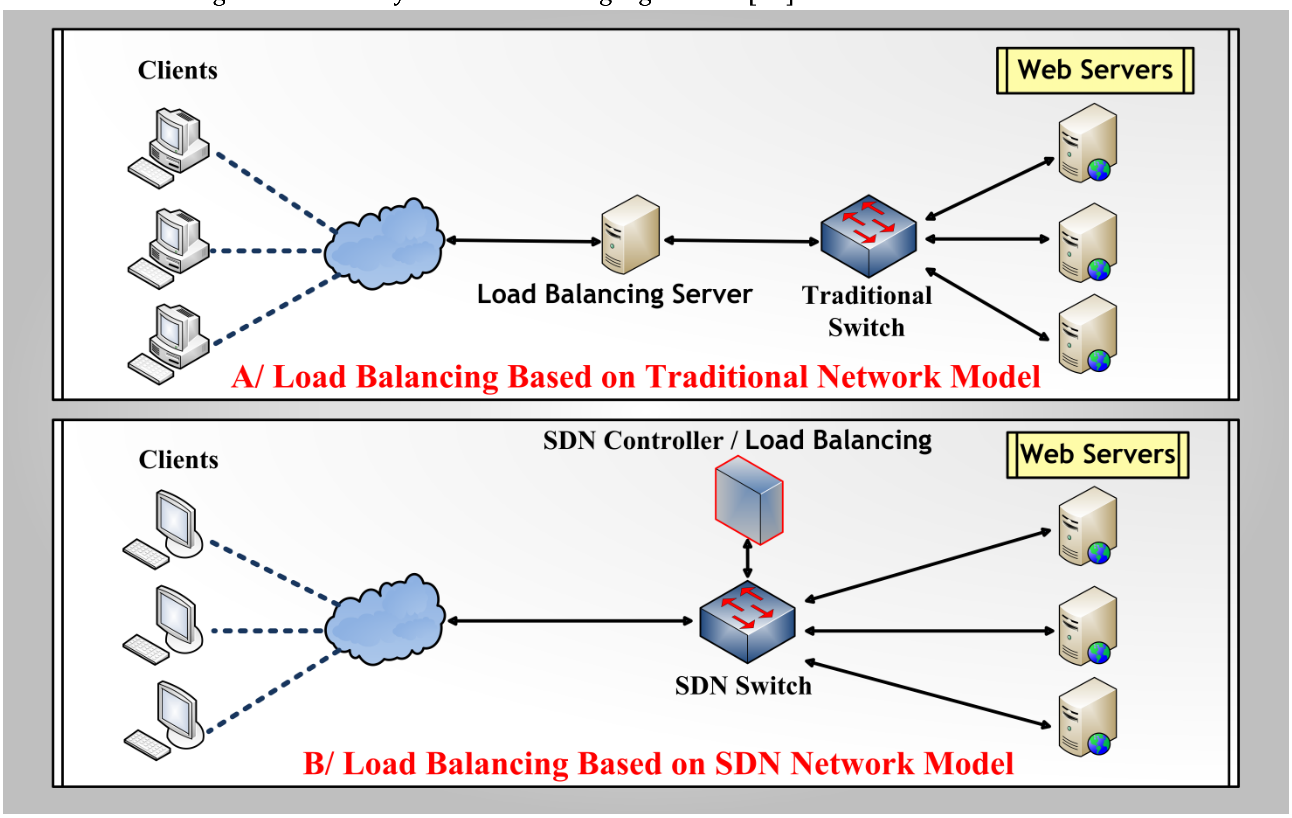

Fig. 2 - ***load Balancing Models [18] A/Traditional Network B/ SDN Network***

Load balancing is one of the most critical concerns in SDN [19], and cloud computing [20], i.e., load balancing ensures that services and paths continue to function normally in the event of network congestion or the failure of some routes or devices [21].

Load balancing in the Software-Defined Networking (SDN) framework refers to the organized distribution of network traffic across various paths and assets to optimize network resources and mitigate network congestion. Enhancing the network's overall performance, responsiveness, and reliability is critical [22]. It will also prevent some servers from overloading while others are inactive or performing minimal work [23].

Load balancing improves both the overall performance and resource consumption of a system. Every resource is distributed effectively, which helps enhance the system's overall performance. This load balancing operating mechanism contributes to excellent user satisfaction. This section will clarify the most critical parts related to loading balancing in SDN networks, namely classifications, algorithms, and techniques, with suggestions provided by this paper to review the state-of-the-art.

This research provides a complete analysis of the existing survey on load balancing in SDN. Figure 3 depicts the approach utilized in the execution of the survey. The primary aim of this research is to investigate the present condition of load balancing classification in SDN networks and present a comprehensive analysis of the most advanced techniques and algorithms utilized in load balancing within the framework of SDN.

Furthermore, this research study presents an innovative taxonomy of the methodologies employed in load balancing for SDN. This research includes many essential contributions:
- Highlight the key features and the difference between traditional and modern networks
- Review the state-of-the-art classification for load balancing in SDN.
- A comprehensive study of the latest SDN load-balancing algorithms.
- A comprehensive study of the latest SDN load balancing techniques.

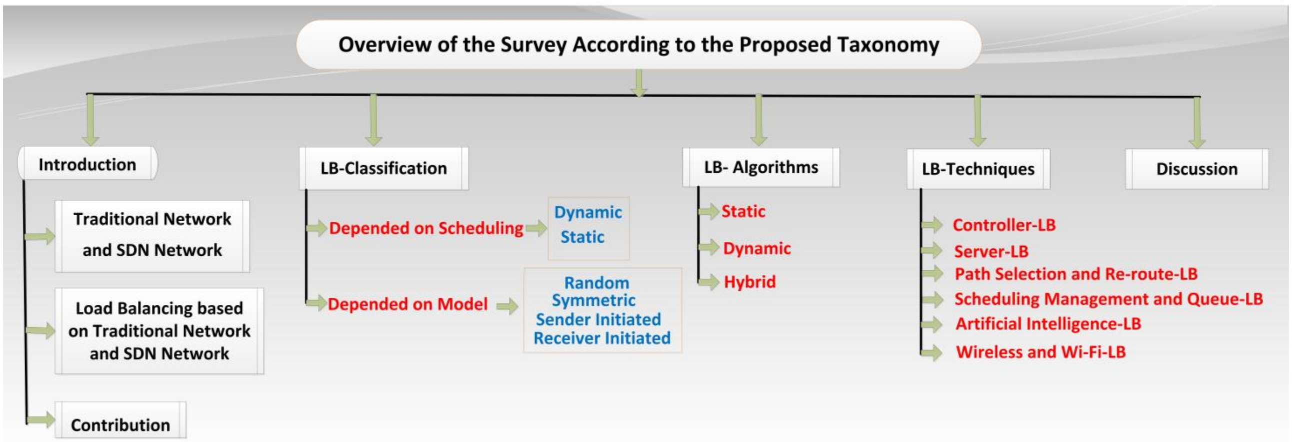

Fig. 3 - ***Overview of the Survey According to the Proposed Taxonomy for paper***

The subsequent sections of the paper are organized in a structured manner. Section 2 presents a comprehensive analysis of load-balancing classification. Section 3 presents a state-of-the-art classification of load balancing algorithms. Section 4 presents a taxonomy for load balancing techniques in SDN. Section 5 explains the discussion and research gap. Lastly, the paper concludes in Section 6.

2. Load Balancing Classification (LB-Classification):

In the following section, the dissertation will discuss the current assessment of load balancing. Based on a comprehensive analysis conducted on LB-Classification, as evidenced by the results presented in Table 1 and the proposal, the state-of-the-art study critically reviews the current knowledge of LB-Classification, as illustrated in Figure 4.

Table 1- Essential Research in LB-Classification

No.	Ref	Year	Main Parts	Sub-Parts	Diagram	Summary
1	24	2015	5	×	×	Include five parts (Static, Dynamic, Centralized, Distributed, and Hierarchical)
2	25	2016	2	8	√	(Distributed, Non-distributed, Deterministic, Probabilistic, Cooperative, Non-Cooperative, Centralized, Semi-centralized)
3	26	2017	2	15	√	(Static, Dynamic, Optimal, Sub-optimal, Approximate, Heuristic, Distributed, Non-Distributed, Cooperative, Non-Cooperative, Centralized, semi-centralized, Sender Initiated, Receiver Initiated, Symmetric)
4	Proposed	2023	2	20	√	(Static, Dynamic, Optimal, Sub-optimal, Approximate, Heuristic, Distributed, Non-Distributed, Cooperative, Non-Cooperative, Centralized, Semi-centralized, Stochastic, Deterministic, Classical, Nature Inspired, Random, Sender Initiated, Receiver Initiated, Symmetric)

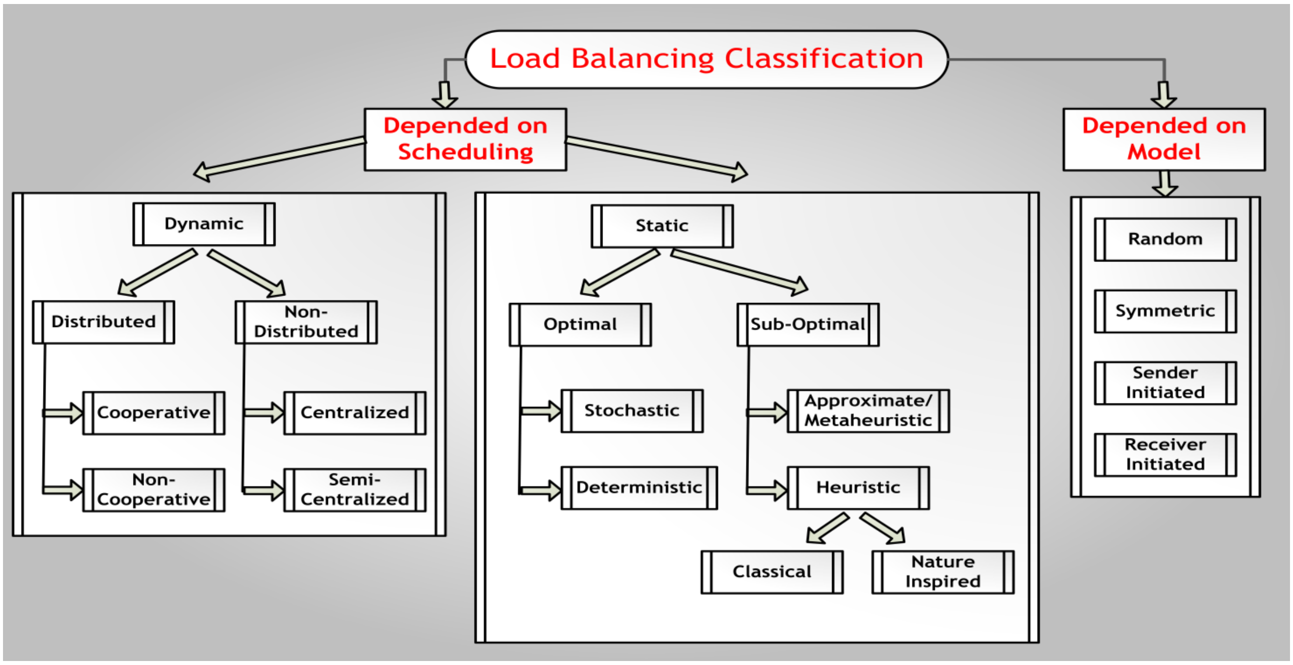

Fig. 4 - ***State-of-the-Art for LB-Classification***

2.1 Depending on the Scheduling

(A) Depending on Scheduling, it is separated into static and dynamic Scheduling. In Static Scheduling, the assignment of tasks occurs offline before the real-time job is officially scheduled for execution on the processor. The task's allocation and scheduling time on the processor do schedule, and the pre-scheduling plan executes the study after the task has officially started implementation [27]. Its advantage is that it can plan the deployment to reduce overhead in the task scheduling process. In dynamic Scheduling, task assignment and schedulability testing are performed online while the system runs. In a real-time system, many tasks are not on the processor in a periodic manner, especially non-periodic jobs that arrive at the system randomly and are dynamically scheduled for execution [24].

(B) Two distinct classifications are used for static Scheduling: optimal and sub-optimal Scheduling. Optimal Scheduling is a job scheduled according to the optimal method; the job scheduler has access to all relevant information regarding the job and the resource, allowing for the most effective allocation decisions to be made in a reasonable amount of time [28]. Introduce sub-optimal Scheduling for large-scale situations of the same problem. This strategy is based on a methodology that divides the original problem into two sub-problems [29].

(C) There are two kinds of optimal Scheduling: stochastic and deterministic. In deterministic Scheduling, the additional workload of a particular node is always transmitted to another specific node [30]. In stochastic Scheduling, each node distributes its other tasks with probability P to a node and probability 1-P to another node [25].

(D) There are two kinds of suboptimal Scheduling: approximate and heuristic. In approximate Scheduling, the technique stops when it finds a "good" one instead of going through all possible solutions. The next step is to use an objective function to evaluate the answer. Based on the evaluation results, the job scheduler will decide whether to use this method for the most recent jobs [31]. Heuristic Scheduling uses the most realistic assumptions about tasks and resources to develop a "reasonable" solution that is neither limited by beliefs nor evaluated by an objective function. So they can make more adaptable and flexible choices in a reasonable amount of time [31].

(E) There are two kinds of heuristic Scheduling: nature-inspired and classical schedules. The nature-inspired scheduling methods are subdivided into evolutionary and swarm-based algorithms [32]. In contrast, classical Scheduling refers to the job scheduling issue often posed with the restriction that specific jobs must be carried out exclusively [33].

(F) Two distinct classifications are used for dynamic Scheduling: distributed and non-distributed Scheduling. In distributed Scheduling, if a task is not always suitable to execute on a specific resource, it can be rescheduled for a different resource. Therefore, it is possible to distribute requests to multiple resources [34]. When using a non-distributed scheduling method, all decisions on dynamic Scheduling are centralized at the level of request and resource management [34].

(G) There are two kinds of distributed Scheduling: cooperative and non-cooperative. In cooperative Scheduling, Multiple resources will collaborate to make dynamic and distributed selections on job scheduling. Multiple resources determine who should perform each task to increase throughput [31]. Each processor cooperates to maximize scheduling and system efficiency [34]. In a non-cooperative, individual processors function independently from one another and make their choices without considering how those choices would affect the remainder of the system [31]. Decisions on Scheduling jobs are made independently by each resource [34].

(I) There are two kinds of non-distributed Scheduling: centralized and semi-centralized Scheduling. Centralized Scheduling is a conceptually single controller, specialized scheduler (or scheduling system) that manages all events across many locations [35]. In contrast, a semi-centralized scheduling controller does consider moderate-scale networks. This technique divides the network into domains or clusters, with a single SDN controller in charge of each. The behaviour of these controllers within an environment is the same as that of a centralized network [36].

2.2 Depended on Models

Each node has a local monitor that links to it. Each monitor is responsible for gathering information and keeping it current on the status of the local node. The main benefits of utilizing this paradigm include excellent performance, availability, and extensibility at an affordable price [37]. It is divided into four categories [31], which are as follows:

- **Sender-initiated:** load-distributing action is triggered when an overflowing node (the sender) attempts to delegate a job to a node operating with a lower-than-expected load (receiver). It is practical to use techniques that use remote invocation [37].
- **Receiver-initiated:** load distribution starts with a node (the receiver) currently under load. This node then tries to retrieve a job from a node that is now overloaded (sender) [31].
- **Symmetrically**: Initiated algorithms take advantage of these two approaches by making it necessary for both senders and receivers to search for suitable domains [38].
- **A random** policy will randomly pick the target node in a distributed system from among the system's nodes. A considerable performance improvement was achieved Via this direct tactic [39].

3. Load Balancing Algorithms (LB-Algorithms):

In the subsequent section, this dissertation will discuss the present evaluation of load balancing. Based on extensive research conducted on LB-Algorithms, as shown by the findings presented in Table 2, and also the proposal made by this thesis to review the state-of-the-art about LB-Algorithms, as depicted in Fig. 5.

Table 2 - Essential Research in LB-Algorithms

No.	Ref/Year	Main Parts	Algorithms	Diagram	Related Works	Summary
1	31 (2014)	×	15	√	×	(Max-Min, Min-Min, Bidding, Ant Colony, Adaptive, RR, Genetic based, Hybrid, Scatter Based, Capacity Based, Branch and Bound, Pollen, Neighbor based, Drafting, Hierarchical Based)
2	38 (2015)	2	34	×	×	(Round Robin, Shortest Job, Min-Min, Max-Min, Two-phase, CLBVM, PALB, FAMLB, Throttled, Honeybee, Active Clustering, Biased Random Sampling, Generalized Priority, Join-Idle-Queue, Genetic Algorithm, Ant Colony, Hill Climbing, Decentralized Content, Server-Load Balancing, ...)

3	18 (2015)	2	8	×	√	(Round-Robin, Random Selection, Weighted Round-Robin, Weighted Random Selection, Least Loaded, Least Link Load, Least Connections, Dynamic Weighted Random Selection)
4	Proposed	3	20	√	√	(Round Robin, Random, Shortest Job First, Min-Min, Max-Min, PQ, FIFO, LC, Token Routing, Genetic Algorithm, Ant Colony, DT, SVM, Power Aware, SHL, AC, hybrid-LB, hybrid dynamic-LB, and hybrid static-LB)

This paper proposes a new classification of LB algorithms based on three types: static, dynamic, and hybrid. Then explain seven types of algorithms for the first type (static) and ten types of algorithms for the second type (dynamic); a proposal for a third type (hybrid) consists of three kinds (hybrid-LB, hybrid dynamic-LB, and hybrid static-LB). Each class contains a variety of other algorithms, with a schema for all algorithms. Table 3 shows a showing LB algorithms in current studies containing (topology, LB-Algorithms, Algorithms, and Details).

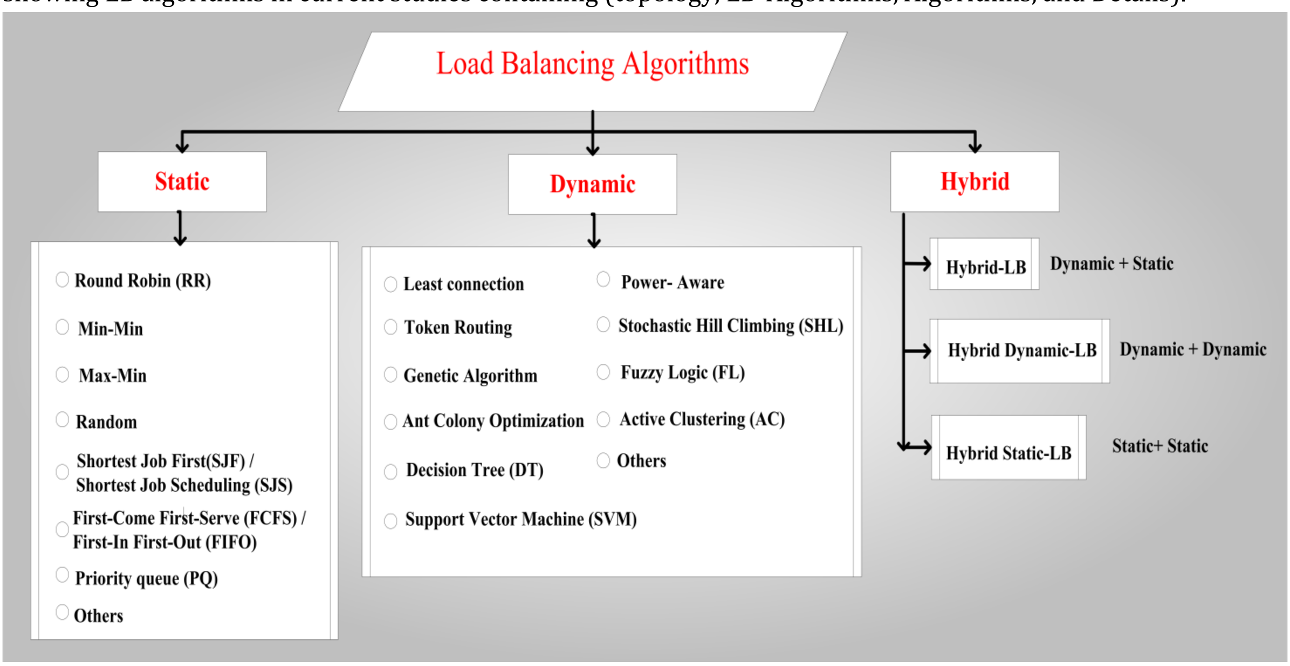

Fig. 5 - *State-of-the-Art for LB-Algorithms*

3.1 Static - LB

Static load balancing methods don't depend on how the system runs. In static algorithms, it is essential to know past design information, such as the time it takes to communicate, the resources a task needs, the processing power of the system nodes, the storage media, and the memory size [40]. The main benefit of static algorithms is that they are easy to implement, but the chance of getting the balance wrong is high [41]. So, the main problem with static load-balancing algorithms is that they need to consider the system's current state. This type makes them unsuitable for plans whose load status can't be predicted in advance [42]. It includes the following algorithms:

1. Round Robin (RR): All processors take part in the execution of tasks. In a round-robin fashion, each process is assigned a CPU. Although processors' workload distributions are identical, the processing times for various functions are not. Others may be idle because specific nodes may be under heavy strain at any given time [43]. Use the ring as its work queue. Queue jobs all have the same execution time and execute sequentially. Employees who do not finish within their turn will be returned to the queue and await the next available turn [44].

2. Min-Min: Initially, the minimum execution time of all jobs is determined. Then, it selects the task with the shortest execution time among all functions. Algorithmically, the work assigns the resource with the shortest completion time. The Min-Min algorithm repeats the same approach until all parts have been scheduled [45]. This algorithm has the benefit of executing jobs with the shortest execution times. The disadvantage of this approach is that some jobs may be starvation [46].

3. Max-Min: Essentially identical to the min-min approach, except after determining the minimum finish time of jobs, the highest values are chosen [47]. The machine with the shortest completion time for all tasks is chosen. The specified node and job are then connected. Then, the node's availability time is updated by adding the task's execution time. A procedure that requires the most time is changed one by one [48].

4. Randomized: A process that a specific node can manage with varying probabilities. The process allocation order for each processor is maintained independently of distant processor allocation. This method performs optimally when all processes are equally loaded. The randomized approach does not preserve a deterministic strategy [49].

5. Shortest Job Scheduling / Shortest Job First: The minor executable task is chosen first. The idea is to finish the execution of small works to utilize the resources to complete significant activities. The shortest work benefited from decreased waiting time for procedures, which made it an effective strategy [38]. It is divided into Preemptive and Non-Preemptive Scheduling. Non-preemptive means that once a processor's allotted time has passed, another cannot seize the processor until the process has finished [50].

6. First-Come-First-Serve (FCFS) / First-In-First-Out (FIFO): If data arrives in the queue, it is processed. The FCFS method needs to be better because some cases or most data, like sensors, must be split up based on their importance or urgency [51]. The FCFS algorithm decides the order of things in the non-safety queue [52].

7. Priority queue (PQ): A solution to the inadequacies of the FIFO algorithm that enables the inclusion of data to be processed before other data. The PQ technique has four queues categorized as "high," "medium," "normal," and "low." It processes all queues belonging to the class with the highest priority first, followed by the classes with decreasing priority until the queues are empty. For these classes, the PQ technique requires at least four queues [51]. Some situations are subdivided into classes based on the dataset type: three classes (high, medium, low) or only two (high, normal).

3.2 Dynamic - LB

It balances the loads upon each job's arrival by continuously assessing the system load in a real-time environment with a set policy [31]. During runtime, the workload is split up across several processors. Based on the newly acquired knowledge, the master assigns new tasks to the slaves [53]. It depends on the current situation. It is unnecessary to have any prior understanding of the system [54]. The performance, accuracy, and running time of dynamic LB algorithms are much better than those of static LB algorithms [32]. Dynamic load balancing solutions require a higher CPU and RAM usage to examine the current state of all servers within the cluster and execute computations and calculations [55]. It includes the following algorithms:

1. Least connection (LC): It must dynamically count the number of connections for each server to predict load [49]. The load balancing depends on each server's connections, calculates which server has the fewest, and then allocates the link to that server [56].

2. Token Routing: This method generates a quick and accurate routing decision. It is to reduce the cost of the system by transferring tokens around it. This technique allows agents to know their global status and neighbors' workloads without knowing their global status or neighbors' workloads. Before passing the token, they build a foundation of knowledge. The tokens are the source of this collection of information [47].

3. Genetic Algorithm: Use when more than one controller controls a network, and the load needs to be spread out and rebalanced. It keeps the controller and switches from not working at their best in an SDN environment and makes the network more reliable [57]. It does help find optimal solutions for routing and scheduling issues [58]. Reducing the time it takes to complete a task can be based on strategy using certain principles and randomization dependent on the network load for cloud computing [59].

4. Ant Colony Optimization (ACO): A meta-heuristic technique is used for load balancing. The heuristic method guarantees an optimal solution for any number of tasks and machines [60]. To accomplish a job, the ants in the ACO model create a network and communicate with one another, and The method uses this to build a VM allocation strategy [61].

5. Decision Tree (DT): It is a machine learning algorithm implemented to handle regression and classification and used to make decisions. It is an applied algorithm inside the controller plane in the SDN network to get load balancing through generated flows, which occurs through training and testing the performance, predicting the response generated flows, and calculating running time [62]. It is used to assess the dynamic performance of the system. The candidate attributes of the Decision Tree are chosen through a data mining process [63]. It trains and then classifies the kernel for a suitable device, i.e., a CPU or GPU [64].

6. Support Vector Machine (SVM): It is a binary supervised classifier that machine learning uses. It is an applied algorithm inside the Ryu controller in the SDN network to get load balancing through generated flows. It occurs through training and testing the performance, predicting the response generated flows, and calculating running time [62]. Identifying and classifying information services for various traffic web applications is achieved using an SVM-dependent network analysis method and its implementations in SDN networks [65].

7. **Power–Aware:** It distributes real-time traffic load, saving power and keeping higher-priority traffic from slowing down [66]. The percentage of each computing node's use is estimated. Then, the method tells us how many computing nodes are running and how many are not working [67]

8. Stochastic Hill Climbing (SHL): It uses load balancing to allocate incoming jobs to the servers or virtual machines (VMs) [68]. It is a mathematical query optimization approach. It uses a random method to locate accessible virtual machines. This method loops until the optimal solution to a problem is discovered. The technique is used with service broker policy to route request traffic [69].

9. Fuzzy Logic (FL): Utilizing a dynamic load balancing method also considers various metrics such as memory, bandwidth, and disk usage. It accurately forecasts the location of the virtual machine depending on the next job that will be scheduled [70]. It is used to do link analysis for interconnections for traffic management. Because fuzzy design has fewer intervals, the study shows that multilevel design is more energy-saving for load balancing in fog zones [71].

10. Active Clustering (AC): The algorithm, as a method for self-aggregation, is used to reorganize the network. This technique aims to group comparable instances regarding their service type [72].

3.3 Hybrid – LB

These algorithms are developed and presented to avoid the shortcomings of dynamic and static load-balancing techniques. They are used to combine the benefits of static and dynamic methods to construct a new model [41]. It indicates that the benefits of two or more existing dynamic or static algorithms can be combined to create a new algorithm [73], proposing a new classification of hybrid LB algorithms based on three types: (hybrid LB, hybrid dynamic LB, and hybrid static-LB).

In some cases, researchers use two or more algorithms of the static type to generate a new model for load balancing, and this type is known as a hybrid static LB. Some researchers use two or more algorithms of the dynamic type to develop a new model for load balancing, known as hybrid dynamic LB, and finally, create a hybrid type by merging two or more algorithms between dynamic and static algorithms, known as hybrid LB.

Table 3 - LB-Algorithms in Current Studies

Ref	Year	Topology	LB-Algorithms	Algorithms	Details
[62]	2021	SDN Technology	Hybrid Dynamic-LB	NN	Utilize neural network (NN) to combine (shortest path selection based + request processing) to optimize the router request time to decrease the delay.
[74]	2022	SDN Technology	Hybrid Dynamic-LB	DT+SVM	Utilize machine learning to develop hybrid algorithms (Decision Tree and Support Vector Machine) for conflict flow detection and categorization (load balancing, route).
[75]	2020	Cloud Computing Technology	Hybrid Dynamic-LB	K-means+ cockroach swarm optimization	This paper uses two techniques (K-means clustering and cockroach swarm optimization (MCSO)). Utilize the K-means clustering method to break data into small parts (heuristic). Utilize the recommended cockroach swarm optimization (MCSO)) method to estimate the related load level subject, including CPU usage, memory utilization, and circular IO occupancy rate.

[76]	2021	SDN Technology	Dynamic-LB	switch-aware reinforcement learning load balancing (SAR-LB)	Use reinforcement learning-based switch and controller selection strategy for switch migration his name switch-aware reinforcement learning load balancing (SAR-LB). Because of correct switch migration decisions, the proposed technique produced better (almost equal) load distribution among SDN controllers.
[77]	2022	Cloud Computing Technology	Dynamic-LB	Hill-Climbing	Based on the hill-climbing technique to reduce reaction time, the author recommended employing two mathematical optimization models to execute dynamic resource allocation to virtual machines and job scheduling. The first algorithm is a hill-climbing algorithm to create pseudo-code. The second method is for work schedule tasks to pick a virtual machine
[78]	2021	DCN	Dynamic-LB	Link Load Balance Route (LLBR)	Propose a technique for link load balance route (LLBR) algorithm for DCNs dynamically shifts throughout time based on the original ST-ResNet algorithm. Enhance resource usage, provide equitable sharing, and assure the functioning of diverse applications
[79]	2013	Cloud Computing Technology	Static-LB	Min-Min Original, LBIMM, (PA-LBIMM	Proposed Load Balance Improved Min-Min (LBIMM) scheduling algorithm based on the characteristic of the Min-Min scheduling algorithm and proposed user-priority awarded load balance improved min-min scheduling algorithm (PA-LBIMM) based on (LBIMM) to make sure that the tasks and resources match the user's priorities.
[80]	2015	Cloud Computing Technology	Hybrid Static-LB	Min-Min + Max-Min Algorithmes, ELBMM Algorithm	Proposed Enhanced load balancing Min Min Algorithm (ELBMM) based on Min-Min and Max-Min algorithms. It uses the advantages of two algorithms to achieve task scheduling for maximum completion time and well-utilized resources.
[81]	2016	Cloud Computing Technology	Hybrid-LB	GA+ Min-Min, ILBMM Algorithme	This paper proposed an Improved Load Balanced Min-Min (ILBMM) algorithm based on genetic algorithms (GA) and Min-Min algorithm to achieve task scheduling, reduce the makespan, and maximize resource usage.
[82]	2022	5G Technology	Hybrid Dynamic-LB	Hybrid Deep Learning (Artificial intelligence and machine learning algorithms)	Propose a hybrid deep learning model from convolution neural network (CNN) and long short-term memory (LSTM). The CNN calculates resource allocation and slice selection, whereas the LSTM utilizes statistical information to balance the load.

4. Load Balancing Techniques (LB- Techniques):

In the subsequent section, this dissertation will discuss the present evaluation of load balancing. Based on extensive research conducted on LB- Techniques, as shown by the findings presented in Table 4, and also the proposal made by this thesis to review the state-of-the-art about LB-Techniques, as depicted in Fig. 6. Table 5 presents an overview of current studies on LB-Techniques.

Table 4 - Essential Research in LB- Techniques

No.	Ref	Year	Techniques	Name of Techniques
1	84	2020	2	1. Conventional load balancing techniques 2. Artificial intelligence-based load balancing techniques
2	83	2020	4	1. Controller Load Balancing, 2. Server Load Balancing 3. Communication Path Selection Load Balancing 4. Load Balancing in Cloud Computing
3	1	2020	5	1. Controller Load Balancing 2. Server Load Balancing 3. Load Balancing in Wireless Links 4. Communication Path Selection Load Balancing 5. Artificial Intelligence Based Load Balancing
4	Proposed	2023	6	1. Controller-LB 2. Server-LB 3. Path Selection and Re-route-LB 4. Scheduling Management and Queue-LB 5. Artificial Intelligence-LB 6. Wireless and Wi-Fi-LB

This paper proposed a new classification of LB - Techniques based on six types: (Controller-LB, Server-LB, Path Selection and Re-route-LB, Scheduling Management and Queue-LB, Artificial Intelligence-LB, and Wireless and Wi-Fi-LB), with a state-of-the-art taxonomy.

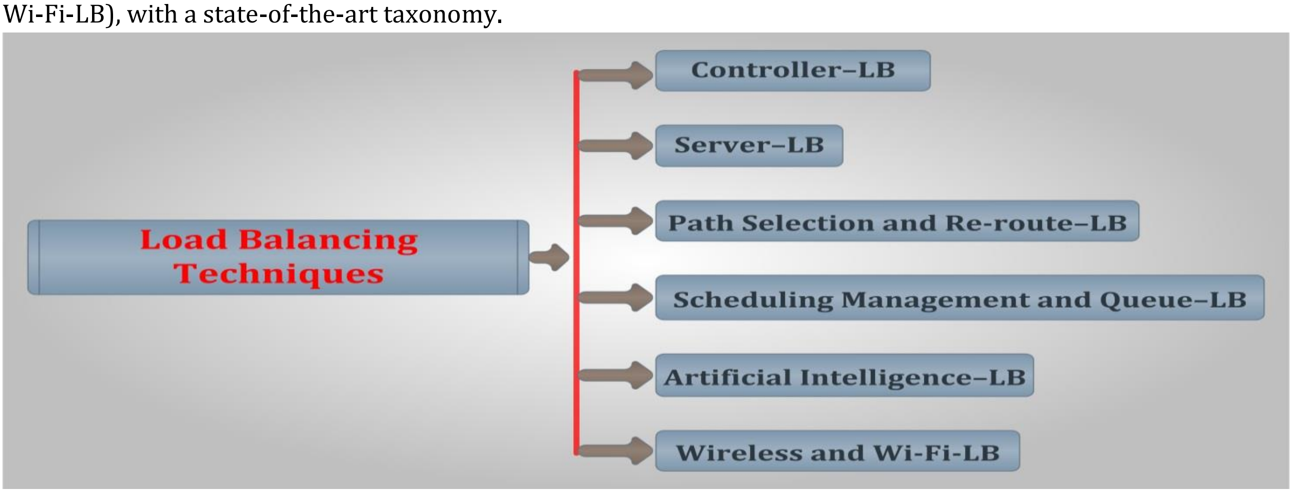

Fig. 6 ***A State-of-the-Art for LB-Techniques***

1. Controller–LB: This load-balancing method does not depend on a singular or centralized controller unit. However, the system utilizes decentralized controllers that consist of multiple controllers. This approach incorporates the concept of controller distribution, enabling load balancing between controllers. In specific scenarios, switch migration is employed to achieve workload balancing.

2. Server–LB: This load-balancing technique involves the utilization of multiple servers, or server clustering, also known as server pools, to distribute the workload across the servers equally. It employs a range of methods and algorithms to distribute the load effectively. Certain cases depend on the state of the processor or memory in a given workload. In contrast, others employ intelligent algorithms for prediction, while others utilize dynamic distribution for load balancing.

3. Path Selection and Re-route–LB: This particular load-balancing method depends on utilizing multiple paths during network congestion. Therefore, achieving equal load distribution across the various routes or redirecting traffic to enhance network performance and minimize congestion is necessary

4. Scheduling Management and Queue–LB: This type of load balancing employs bandwidth management or resource management techniques, such as allocating processor or memory resources within networks. A queueing model is employed in load balancing, in which a single-level or multi-level feedback queue is utilized as a case study. Additionally, it is recommended to employ a dynamic load scheduling module to distribute the workload effectively.

5. Artificial Intelligence–LB: This load-balancing method utilizes optimization techniques based on artificial intelligence (AI). This system employs a range of optimization algorithms, including Convolutional Neural Networks (CNN) and deep learning, as well as machine learning algorithms, to effectively distribute the workload and enhance the quality of service.

6. Wireless and Wi-Fi-LB: This approach to load balancing utilizes WiFi networks, 5G technology, Radio Access Networks (RAN), and Wireless Software-Defined Networking (SDN) techniques to handle network load imbalances among users and devices while determining optimal routing paths. These technologies have been designed and implemented for Software-Defined Networking (SDN) networks to solve load distribution issues.

Table 5: LB- techniques in current studies

LB –Tech.	Ref.	Topology	Strategy	Algorithm	Platform	Summary and Result
Controller -LB	[85]	Software-Defined Optical Networks (SDON)	Multi-Controller	Switch Migration Based Controller Placement (SMBCP) algorithm	Simulation based on Python	Proposed an approach (SMBCP) fared better than CSM and MUSM when it came to loading balancing among controllers in SDON networks, and it lowered the costs associated with switch migration. It does expect that using SMBCP will reduce the load disparity by 50 %. The maximum amount of reduction that CSM and MUSM are capable of is 12.4% and 22.3%, respectively.
	[86]	SDN	Switch selection for migration	Controller Adaptation and Migration Decision (CAMD) algorithm	Mininet emulator	This method (CAMD) outperformed Elasticon and DALB methods in load balancing across controllers in SDN networks by using the OpenFlow protocol to reduce the response time and decrease switching migration costs. The improvement is about 15%.
	[87]	SDN	Multiple controllers are distributed over a cluster	Algorithm based Wardrop equilibrium and game theory	Simulation based on MATLAB	The proposed load balancing technique does base on Wardrop equilibrium and game theory compared with a static switch-controller association strategy. The proposed algorithm achieved an increase in total throughput while simultaneously reducing the control connection latency.
	[88]	SDN	Switch migration	Migration competency -based load balancing (MCBLB) algorithm	Simulation based on MATLAB	Proposes an MCBLB his work (shift-move) and compares it with online controller load balancing (OCLB) and MCBLB-shift-only algorithms. Result: Improve load balancing by up to 14 % compared to previous efforts
	[89]	SDN	Single server for load balancing	Load Balancing by Server Response	floodlight controller is based JAVA	Suggest an LBBSRT, which, compared to the Round Robin and Random algorithms, performs a significantly better job of

Server -LB				Time (LBBSRT) algorithm		balancing the load and selecting the server with the shortest or most steady response time. During the process of the computation involving the CPU and memory
	[90]	SDN	Server cluster	Round-Robin and Least-Connections algorithms	Mininet emulator	Comparing the Round-Robin and Least-Connections algorithms for counting the number of active server connections by requesting flow entry information from three servers and measuring response time. The OpenFlow Least-Connections algorithm's higher than that of the OpenFlow Round-Robin method in terms of performance.
	[91]	SDN	Dynamic server LB	Ant Colony System (ACS) algorithm	Mininet emulator	Propose an Ant Colony System (ACS) algorithm and compare it to the Round Robin and random algorithms. This algorithm uses the controller's data on server load and network statistics to find the best server and the best path for traffic. ACS algorithm is better than other algorithms because network throughput has increased while network latency has decreased.
Path Selection and Re-route – LB	[92]	SDN	1. Two-level fast re-routing	Methods of optimization for mathematical modeling	Use mathematical models in SDN architecture	Suggested techniques for mathematical modeling optimization to split the hierarchical tiers of the functions for determining main (lower level) and backup (higher level) routes, as well as a specific recording of previously known conditions for network bandwidth security. It offers a two-level solution for quick rerouting with load balancing in SDN, including link, node, route, and bandwidth protection strategies.
	[93]	SDN	Flow re-routing	partial flow statistics collection (PFSC) algorithm	Mininet emulator	A comparison of the proposed PFSC and a pull-based collecting mechanism to reroute traffic. Compared to a previous pull-based collecting technique, the suggested method reduces statistic collection overhead by roughly 56%. However, the load-balancing performance does barely influenced by 3%–5%.
	[94]	Edge-Cloud Network	Middle Route Layer	Novel services orchestrati	C++ is used to check the	Proposed a (SODA) and compared it to the shortest routing path scheme (TSRP) to reduce the time

		based on SDN	(MRL) adjusts packet routing	on and data aggregation framework (SODA) algorithm	results of experiments	it takes for services to respond, , and make the most efficient use of energy. There are three primary levels in the network. (1) Data centers layer (DCL), middle routing layer (MRL), and vehicle network layer (VNL).
	[95]	Cloud center based on SDN	Dynamic load scheduling module	Novel hybrid load balance is SDN-LB algorithm	Mininet emulator	The SDN-LB algorithm, a new type of hybrid load balance, is compared with three versions of traditional static load balance. The SDN-LB comprises four main modules (algorithms): traffic detection, load calculation, dynamic load scheduling, and flow management. The results show that SDN-LB has a throughput that is 3% better than the traditional.
	[96]	SDN)-enabled Clouds	bandwidth management and priority queues	priority-aware VM allocation (PAVA) algorithm	CloudSim SDN	Propose priority-aware resource placement algorithms known as priority-aware VM allocation (PAVA) algorithms and compare them with other algorithms (FFD, FFD+DF, BWA). PAVA comprises three main modules (algorithms): Bandwidth allocation applications (BWA), First-fit decreasing for bandwidth requirement (FFD), and Dynamic flow scheduling algorithm (DF). The result is a reduction for large-scale complicated application cases by 43.2% .)
Scheduling Management and Queue –LB	[97]	5G based on SDN	Resource management and queuing model	SDN-based load balancing mechanism (partial data offloading algorithm	MATLAB simulation	The proposed SDN-based load balancing technique (partial data offloading algorithm) compares favorably to classical load balancing (LB) and distributed mobility load balancing (DLB). The result can save primary resources, reduce the probability of missing a threshold by 20 to 50%, and demonstrate that SDN-based LB improves than baseline techniques by decreasing the frequency of handovers by 50%.
	[98]	Data Centers Based on SDN	multilevel feedback queue (MLFQ)	DSCSD, a dynamic traffic scheduling and congestion control scheme across data centers	Mininet emulator	It compares DSCSD to conventional methods. First, it suggested a new traffic scheduling mechanism that could dynamically route a freshly incoming flow based on real-time network information. DSCSD improves connection efficiency by using the shortest pathways, unlike conventional techniques

				based on SDN algorithm		and SDN-based threshold schemes. Second, depending on using a multilevel feedback queue system of congestion control appropriate for diverse flows and could identify anomalies by preventing malicious flows from taking bandwidth for lengthy periods. Experiment and analysis improved network usage across data centers by classifying and diverting traffic.
Artificial Intelligence -LB	[99]	SDN	Novel intelligent SDN	DNQ (hybrid from Deep Neural Networks (DNNs) and Q-learning algorithms	Not clear	DNQ was proposed and compared to Dijkstra's and Q-learning algorithms. It improves the network efficiency and implements the new load balancing technique more intelligent and efficient by lowering the time for path reselection. When the proposed algorithm does compare to Dijkstra's algorithm and the Q-learning method based on bandwidth, the transmission time does reduce by 38.10% and 19.10%, respectively. The rate of packet loss is lower by 10.81% and 5.25%.
	[100]	Datacenter based on SDN	Scheduling of server load balancing	Load Balancing Scheme based on Fuzzy Logic (LBSFL) algorithm	Mininet emulator	Propose an LBSFL and compare it with LBBSRT (Load Balancing by Server Response Time) algorithm. The SDN controller organizes current network requests to accomplish server load balancing using a fuzzy logic framework. Under the presumption of ensuring network performance, this technique has the potential to enhance server load distribution significantly. Compared to LBBSRT, the response time does rise by roughly 15%, and the system load can be maintained between 30% and 50% of its maximum capacity.
	[101]	SDN	Prevent overflow of flow table	Machine learning algorithm called Decision tree (Iterative Dichotomiser 3-ID3) algorithm	Mininet emulator	The suggested method is called Decision Tree-based Entries Reduction (DTER), and it uses the robust machine learning technique called Decision tree (ID3) to prevent flow table overflow. Compared to other current schemes such as STAR and AC7 + OSPF. DTER reduces 99.99 percent of the controller's

						overhead, reducing controller effort and average end-to-end packet latency with high throughput, and also entrances are reduced to 99 percent of their original size, allowing more room for new flows.
Wireless and Wi-Fi-LB	[102]	Software Defined WiFi networks (SD-WiFi)	novel multi-controller load balancing	Hybrid from (Support Vector Machine (SVM) and Type-2 Fuzzy based Particle Swarm Optimization (TFPSO) algorithm	OMNeT++ simulator	SVM algorithm and TFPSO method were presented to balance the load in SD-WiFi and compared to MPSO-CO. WiFi APs use SVM to identify incoming flow requests as HP or LP. TFPSO migrates an overcrowded local controller (LC) to another LC. Markov Chain Model (MCM) used crowded, and underloaded transition states in global controllers to estimate future LC load based on present load. The outcome improves throughput by 33% and workload by 70%. MPSO-CO vs.
	[103]	Wireless SDN	Traffic load and AP users	Adaptive Load-Balancing (ALB) algorithm	NS3 simulation	Propose a modeling simulation to investigate AP load, user traffic, and throughput. Adaptive Load-Balancing (ALB) methodology compared to "Strongest-Signal-First (SSF), Least-Load-First (LLF), and Cell-Breathing approaches." The results indicate that the average AP load of the present scheme is 11 ~ 28 % higher than SSF and 26 ~ 346% more increased than Cell-Breathing and that the average user throughput is 16 ~ 26% more increased than SSF and23 ~ 377% more increased than Cell-Breathing.
	[104]	SD-LTE-RAN	QoS and Priority scheduler	Novel QoS Aware Load Balance (QALB) algorithm	NS-3 simulator	Proposed a (QALB) algorithm and compare it with HLFB and MinTHT. Resolve network load imbalances, enhance QoS, and reduce overall GBR satisfaction while also reducing network congestion. Compared to conventional load balance methods, QALB reduces network overload by 15% and reduces network overload by 10% in mobility circumstances.

5. Discussion

This section provides an overview of current surveys and discusses the distinctive aspects of this study in comparison. The present working paper focuses on current studies and a comprehensive study of previous studies about load balancing in SDNs.

Section 1: Introduction: This section compares traditional and SDN networks' architectural differences and load distribution mechanisms. Additionally, it explains the concept of load balancing and underscores its significance within SDN networks. Also, comprehensively explain the critical components of load balancing in SDN networks, including classifications, methods, and techniques. Figures 1, 2, and 3 perform a theatrical production.

Figure 1 illustrates the architectural distinction between conventional networks and SDN networks. It shows the segregation of the control plane from the data plane, explaining its advantages. Additionally, the figure highlights the programmability aspect of the SDN controller.

Figure 2 depicts the visual representation of load balancing and highlights the different load distributions between traditional and SDN networks.

Figure 3 provides a comprehensive depiction of the survey, organized by the proposed taxonomy outlined in the paper.

Section 2: LB-Classification: This section presents the suggested method for LB-Classification and supports it with a comparison analysis of three significant prior studies [24, 25, and 26]. It includes the inclusion of Figure 4 and Table 1 to enhance the presentation of the research proposal.

Figure 4 presents novel methodologies for classifying LB, which is contingent upon two key factors: (depending on scheduling and dependent on models). Furthermore, it organizes these techniques into distinct subcategories using a modern paradigm. This paper provides a comprehensive analysis of each component, comprising 20 sub-components.

Table 1 presents a comparative analysis of the three most significant previous research studies, with the proposed classification of LB in this study paper and the critical enhancements and recommendations incorporated. The table herein comprises various components: the main parts, sub-parts, diagram, and summary.

Section 3: LB-Algorithms: Which are evaluated by comparing them to three significant prior studies [18, 31, and 38]. This section includes Figure 5 as well as Tables 2 and 3.

The proposed classification in Figure 5 introduces a novel categorization of LB algorithms, encompassing three distinct types: static, dynamic, and hybrid. Subsequently, this discourse will clarify the categorization of algorithms into seven distinct types under the first classification, namely static algorithms. An extensive examination of ten algorithmic types falling under the second classification, referred to as dynamic algorithms, will next be provided. Furthermore, a proposal for a third classification, denoted as hybrid algorithms, will be presented, including three specific subtypes: hybrid-LB, hybrid dynamic-LB, and hybrid static-LB. Every class comprises a diverse range of additional algorithms.

Table 2 compares the three most significant prior works and the proposed research paper's contribution to the LB algorithms. Additionally, it includes vital enhancements and suggestions that have been incorporated. The table presented herein comprises several components: Main Parts, Algorithms, diagrams, Related Works, and Summary.

Table 3 presents an overview of load balancing (LB) algorithms employed in recent studies. The table contains topology, LB algorithms, algorithms, and details.

Section 4: LB-Techniques: This section presents the suggested approach to LB-Techniques, which is evaluated by comparing it with three key prior studies [1, 83, 84]. The section includes Figure 6, as well as Tables 4 and 5.

This study introduces a novel categorization of LB Techniques in Figure 6, including six distinct types: Controller -LB, Server -LB, Path Selection and Re-route -LB, Scheduling Management and Queue -LB, Artificial Intelligence LB, and Wireless and Wi-Fi-LB. A detailed description accompanies each category.

Table 4 presents a comparative analysis of the three most significant previous studies, along with the proposed approach outlined in this research paper, which introduces an additional load-balancing technique known as Scheduling Management and Queue-LB. The table comprises two main sections, namely techniques and names of techniques.

Table 5 presents an overview of the LB techniques employed in recent studies. The table includes information on the LB techniques utilized (LB techniques, Topology, Strategy, Algorithm, Platform, Summary and Result).

The research gap in this context pertains to the areas within load balancing in SDNs that have yet to be extensively explored by previous studies. The paper highlights the need for a comprehensive study that covers architectural differences, load distribution mechanisms, and the significance of load balancing in SDNs. Also, introducing new ways to classify load-balancing algorithms and a detailed categorization of LB techniques suggests that there may be a need for more standardized ways to classify things and more research into different load-balancing techniques.

Furthermore, comparing the proposed LB algorithms and techniques with existing studies indicates a need for more comprehensive evaluations of the effectiveness and performance of various load-balancing approaches. This suggests a research gap in evaluating the proposed methodologies against established ones and addressing load-balancing challenges in wireless and Wi-Fi networks.

6. Conclusion:

This research summarizes the concept of traditional networks, their differences from SDN networks, and their differences in load-balancing distribution. This paper systematically studies state-of-the-art load balancing in SDNs, classifications, algorithms, and techniques. Proposed new approaches for LB classification based on two factors (depending on scheduling and dependent on models) and divided it into other sub-parts with a modern scheme. Also, explain in detail each part, which consists of 20 sub-parts, and propose a new taxonomy of LB-Algorithms based on three types: (static, dynamic, and hybrid) with a proposal for a third type (hybrid) consisting of three kinds (hybrid-LB, hybrid dynamic-LB, and hybrid static-LB). Also, it proposed a new classification of LB-Techniques based on six types: (Controller-LB, Server-LB, Path Selection and Re-route-LB, LB, Scheduling Management and Queue-LB, Artificial Intelligence-LB, and Wireless and Wi-Fi-LB), and with an explanation of each type. Table 2 shows LB techniques in current studies.

SDN is a new architecture that aims to improve network speed and quality of service (QoS) by efficiently distributing the load. The target distribution of services with constrained resources has to be further studied. Additionally, the paper recommends conducting a comprehensive study on load balancing in SDN for heterogeneous and homogeneous environments in the data centre. This paper proposes a feasible path for additional research into load balancing in SDN.

Acknowledgements

The authors thank Kufa University and Babylon University, College of Information Technology, Information Networks Department, for their endless support and for completing this research.

Reference

[1] T. Semong et al., "Intelligent Load Balancing Techniques in Software Defined Networks: A Survey," Electronics, vol. 9, no. 7, p. 1091, 2020/07/03 2020.

[2] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-Defined Networking: Challenges and research opportunities for Future Internet," Computer Networks, vol. 75, pp. 453-471, 2014/12 2014.

[3] D. B. Rawat and S. R. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey," IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 325-346, 2017.

[4] M. Priyadarsini and P. Bera, "Software defined networking architecture, traffic management, security, and placement: A survey," Computer Networks, vol. 192, p. 108047, 2021/06 2021.

[5] O. Blial, M. Ben Mamoun, and R. Benaini, "An Overview on SDN Architectures with Multiple Controllers," Journal of Computer Networks and Communications, vol. 2016, pp. 1-8, 2016.

[6] A. Abdelaziz et al., "Distributed controller clustering in software defined networks," (in eng), PloS one, vol. 12, no. 4, pp. e0174715-e0174715, 2017.

[7] J. Gil Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 518-532, 2016/09 2016.

[8] I. Alam et al., "A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV," ACM Computing Surveys, vol. 53, no. 2, pp. 1-40, 2021/03/31 2021.

[9] A. Malik, R. de Fréin, and B. Aziz, "Rapid Restoration Techniques for Software-Defined Networks," Applied Sciences, vol. 10, no. 10, p. 3411, 2020/05/14 2020.

[10] M. Garrich, F.-J. Moreno-Muro, M.-V. Bueno Delgado, and P. Pavon Marino, "Open-Source Network Optimization Software in the Open SDN/NFV Transport Ecosystem," Journal of Lightwave Technology, vol. 37, no. 1, pp. 75-88, 2019/01/01 2019.

[11] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey," IEEE Access, vol. 7, pp. 107346-107379, 2019.

[12] A. Malik, B. Aziz, M. Adda, and C.-H. Ke, "Smart routing: Towards proactive fault handling of software-defined networks," Computer Networks, vol. 170, p. 107104, 2020/04 2020.

[13] B. Görkemli, S. Tatlıcıoğlu, A. M. Tekalp, S. Civanlar, and E. Lokman, "Dynamic control plane for SDN at scale," IEEE Journal on Selected Areas in Communications, vol. 36, no. 12, pp. 2688-2701, 2018.

[14] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN)," Computer Networks, vol. 112, pp. 279-293, 2017/01 2017.

[15] M. Adil et al., "Enhanced-AODV: A Robust Three Phase Priority-Based Traffic Load Balancing Scheme for Internet of Things," IEEE Internet of Things Journal, vol. 9, no. 16, pp. 14426-14437, 2022/08/15 2022.

[16] H. Babbar, S. Parthiban, G. Radhakrishnan, and S. Rani, "A genetic load balancing algorithm to improve the QoS metrics for software defined networking for multimedia applications," Multimedia Tools and Applications, vol. 81, no. 7, pp. 9111-9129, 2022/01/08 2022.

[17] L. Hui, S. Yao, G. Minyi, and T. Feilong, "LABERIO: Dynamic load-balanced Routing in OpenFlow-enabled Networks," presented at the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), 2013/03, 2013. Available: http://dx.doi.org/10.1109/aina.2013.7

[18] M. Qilin and S. Weikang, "A Load Balancing Method Based on SDN," presented at the 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation, 2015/06, 2015. Available: http://dx.doi.org/10.1109/icmtma.2015.13

[19] H. Xue, K. T. Kim, and H. Y. Youn, "Dynamic Load Balancing of Software-Defined Networking Based on Genetic-Ant Colony Optimization," (in eng), Sensors (Basel, Switzerland), vol. 19, no. 2, p. 311, 2019.

[20] B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," presented at the 2009 Fifth International Joint Conference on INC, IMS and IDC, 2009. Available: http://dx.doi.org/10.1109/ncm.2009.218

[21] M. Suchara, D. Xu, R. Doverspike, D. Johnson, and J. Rexford, "Network architecture for joint failure recovery and traffic engineering," presented at the Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems - SIGMETRICS '11, 2011. Available: http://dx.doi.org/10.1145/1993744.1993756

[22] V. D. Chakravarthy and B. Amutha, "Path based load balancing for data center networks using SDN," International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 4, p. 3279, 2019/08/01 2019.

[23] H. Zhong, Y. Fang, and J. Cui, "Reprint of “LBBSRT: An efficient SDN load balancing scheme based on server response time”," Future Generation Computer Systems, vol. 80, pp. 409-416, 2018.

[24] G. Rastogi and R. Sushil, "Analytical literature survey on existing load balancing schemes in cloud computing," presented at the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015/10, 2015. Available: http://dx.doi.org/10.1109/icgciot.2015.7380705

[25] M. Mesbahi and A. M. Rahmani, "Load balancing in cloud computing: a state of the art survey," Int. J. Mod. Educ. Comput. Sci, vol. 8, no. 3, p. 64, 2016.

[26] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," Journal of Network and Computer Applications, vol. 88, pp. 50-71, 2017.

[27] H. Zhang, F. Tang, and L. Barolli, "Efficient flow detection and scheduling for SDN-based big data centers," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 5, pp. 1915-1926, 2018/04/26 2018.

[28] K. Qureshi and P. Manuel, "Adaptive pre-task assignment scheduling strategy for heterogeneous distributed raytracing system," Computers & Electrical Engineering, vol. 33, no. 1, pp. 70-78, 2007/01 2007.

[29] M. Abu Sharkh, A. Shami, and A. Ouda, "Optimal and suboptimal resource allocation techniques in cloud computing data centers," Journal of Cloud Computing, vol. 6, no. 1, 2017/03/09 2017.

[30] R. Strausz, "Deterministic versus stochastic mechanisms in principal–agent models," Journal of Economic Theory, vol. 128, no. 1, pp. 306-314, 2006/05 2006.

[31] N. Rathore and I. Chana, "Load Balancing and Job Migration Techniques in Grid: A Survey of Recent Trends," Wireless Personal Communications, vol. 79, no. 3, pp. 2089-2125, 2014/08/20 2014.

[32] S. Afzal and G. Kavitha, "Load balancing in cloud computing – A hierarchical taxonomical classification," Journal of Cloud Computing, vol. 8, no. 1, 2019/12 2019.

[33] V. Muresan, W. Xiaojun, V. Muresan, and M. Vladutiu, "A comparison of classical scheduling approaches in power-constrained block-test scheduling," presented at the Proceedings International Test Conference 2000 (IEEE Cat. No.00CH37159), Available: http://dx.doi.org/10.1109/test.2000.894299

[34] M. Padole and A. Shah, "Comparative Study of Scheduling Algorithms in Heterogeneous Distributed Computing Systems," in Advanced Computing and Communication Technologies, ed: Springer Singapore, 2017, pp. 111-122.

[35] R. Ruby, S. Zhong, B. M. ElHalawany, H. Luo, and K. Wu, "SDN-Enabled Energy-Aware Routing in Underwater Multi-Modal Communication Networks," IEEE/ACM Transactions on Networking, vol. 29, no. 3, pp. 965-978, 2021/06 2021.

[36] S. Kuklinski, Y. Li, and K. T. Dinh, "Handover management in SDN-based mobile networks," presented at the 2014 IEEE Globecom Workshops (GC Wkshps), 2014/12, 2014. Available: http://dx.doi.org/10.1109/glocomw.2014.7063430

[37] N. Rathore and I. Chana, "A sender initiate based hierarchical load balancing technique for grid using variable threshold value," presented at the 2013 IEEE International Conference on Signal Processing, Computing and Control (ISPCC), 2013/09, 2013. Available: http://dx.doi.org/10.1109/ispcc.2013.6663440

[38] N. Kumar and N. Mishra, "Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing- A Systematic Review," International Journal of Computer Applications, vol. 131, no. 18, pp. 11-19, 2015/12/17 2015.

[39] K. Lu and A. Y. Zomaya, "A Hybrid Policy for Job Scheduling and Load Balancing in Heterogeneous Computational Grids," presented at the Sixth International Symposium on Parallel and Distributed Computing (ISPDC'07), 2007/07, 2007. Available: http://dx.doi.org/10.1109/ispdc.2007.4

[40] A. Akbar Neghabi, N. Jafari Navimipour, M. Hosseinzadeh, and A. Rezaee, "Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network," International Journal of Communication Systems, vol. 32, no. 4, p. e3875, 2019/01/13 2019.

[41] A. S. Milani and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends," Journal of Network and Computer Applications, vol. 71, pp. 86-98, 2016/08 2016.

[42] A. A. Rajguru and S. Apte, "A comparative performance analysis of load balancing algorithms in distributed system using qualitative parameters," International Journal of Recent Technology and Engineering, vol. 1, no. 3, pp. 175-179, 2012.
[43] R. Soumya, "Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment," International Journal on Cloud Computing: Services and Architecture, vol. 2, no. 5, pp. 1-13, 2012/10/31 2012.
[44] D. K. SAROJ HIRANWAL, "Adaptive round robin scheduling using shortest burst approach based on smart time slice," 2011.
[45] C. Huankai, F. Wang, N. Helian, and G. Akanmu, "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing," presented at the 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH), 2013/02, 2013. Available: http://dx.doi.org/10.1109/parcomptech.2013.6621389
[46] T. Kokilavani and D. G. Amalarethinam, "Load balanced min-min algorithm for static meta-task scheduling in grid computing," International Journal of Computer Applications, vol. 20, no. 2, pp. 43-49, 2011.
[47] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, "Availability and load balancing in cloud computing," in International Conference on Computer and Software Modeling, Singapore, 2011, vol. 14, pp. 134-140: sn.
[48] Y. Mao, X. Chen, and X. Li, "Max–Min Task Scheduling Algorithm for Load Balance in Cloud Computing," in Advances in Intelligent Systems and Computing, ed: Springer India, 2014, pp. 457-465.
[49] O. Tkachova, U. Chinaobi, and A. R. Yahya, "A load balancing algorithm for SDN," Scholars J Eng Technol (SJET), vol. 9, pp. 25-36, 2016.
[50] R. K. Mondal, E. Nandi, and D. Sarddar, "Load Balancing Scheduling with Shortest Load First," International Journal of Grid and Distributed Computing, vol. 8, no. 4, pp. 171-178, 2015/08/31 2015.
[51] C. H. Lee and J. S. Park, "An SDN-based packet scheduling scheme for transmitting emergency data in mobile edge computing environments," Hum. Cent. Comput. Inf. Sci, vol. 11, p. 28, 2021.
[52] M. Adnan et al., "On the Design of Efficient Hierarchic Architecture for Software Defined Vehicular Networks," (in eng), Sensors (Basel, Switzerland), vol. 21, no. 4, p. 1400, 2021.
[53] B. Yagoubi and Y. Slimani, "Task Load Balancing Strategy for Grid Computing," Journal of Computer Science, vol. 3, no. 3, pp. 186-194, 2007/03/01 2007.
[54] H. Bheda and H. Bhatt, "An overview of load balancing techniques in cloud computing environments," International journal of Engineering and Computer Science, vol. 4, pp. 9874-9881, 2015.
[55] O. H. Jader, S. Zeebaree, and R. R. Zebari, "A state of art survey for web server performance measurement and load balancing mechanisms," International Journal of Scientific & Technology Research, vol. 8, no. 12, pp. 535-543, 2019.
[56] T. Yong Meng and R. Ayani, "Comparison of Load Balancing Strategies on Cluster-based Web Servers," SIMULATION, vol. 77, no. 5-6, pp. 185-195, 2001/11 2001.
[57] S.-B. Kang and G.-I. Kwon, "Load balancing of software-defined network controller using genetic algorithm," Contemporary Engineering Sciences, vol. 9, pp. 881-888, 2016.
[58] E. Pekel, "A simple solution to technician routing and scheduling problem using improved genetic algorithm," Soft Computing, vol. 26, no. 14, pp. 6739-6748, 2022/04/09 2022.
[59] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing," Procedia Technology, vol. 10, pp. 340-347, 2013.
[60] P. Xu, G. He, Z. Li, and Z. Zhang, "An efficient load balancing algorithm for virtual machine allocation based on ant colony optimization," International Journal of Distributed Sensor Networks, vol. 14, no. 12, p. 155014771879379, 2018/12 2018.
[61] L. Pettersson and C. Lundell Johansson, "Ant Colony Optimization-Optimal Number of Ants," ed, 2018.
[62] M. H. H. Khairi et al., "Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms," IEEE Access, vol. 9, pp. 76024-76037, 2021.
[63] E. M. Voumvoulakis, A. E. Gavoyiannis, and N. D. Hatziargyriou, "Decision trees for dynamic security assessment and load shedding scheme," presented at the 2006 IEEE Power Engineering Society General Meeting, 2006. Available: http://dx.doi.org/10.1109/pes.2006.1709418
[64] U. Ahmed, J. C.-W. Lin, G. Srivastava, and M. Aleem, "A load balance multi-scheduling model for OpenCL kernel tasks in an integrated cluster," Soft Computing, vol. 25, no. 1, pp. 407-420, 2020/07/15 2020.
[65] C.-C. Liu, Y. Chang, C.-W. Tseng, Y.-T. Yang, M.-S. Lai, and L.-D. Chou, "SVM-based Classification Mechanism and Its Application in SDN Networks," presented at the 2018 10th International Conference on Communication Software and Networks (ICCSN), 2018/07, 2018. Available: http://dx.doi.org/10.1109/iccsn.2018.8488219
[66] A. Fernandez-Fernandez, C. Cervello-Pastor, L. Ochoa-Aday, and P. Grosso, "An Online Power-Aware Routing in SDN with Congestion-Avoidance Traffic Reallocation," presented at the 2018 IFIP Networking Conference (IFIP Networking) and Workshops, 2018/05, 2018. Available: http://dx.doi.org/10.23919/ifipnetworking.2018.8696820
[67] J. M. Galloway, K. L. Smith, and S. S. Vrbsky, "Power aware load balancing for cloud computing," in proceedings of the world congress on engineering and computer science, 2011, vol. 1, pp. 19-21.
[68] B. Mondal, K. Dasgupta, and P. Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach," Procedia Technology, vol. 4, pp. 783-789, 2012.
[69] M. Zahid, N. Javaid, K. Ansar, K. Hassan, M. KaleemUllah Khan, and M. Waqas, "Hill Climbing Load Balancing Algorithm on Fog Computing," in Advances on P2P, Parallel, Grid, Cloud and Internet Computing, ed: Springer International Publishing, 2018, pp. 238-251.
[70] M. S. Q. Zulkar Nine, M. A. K. Azad, S. Abdullah, and R. M. Rahman, "Fuzzy logic based dynamic load balancing in virtualized data centers," presented at the 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2013/07, 2013. Available: http://dx.doi.org/10.1109/fuzz-ieee.2013.6622384
[71] S. P. Singh, A. Sharma, and R. Kumar, "Design and exploration of load balancers for fog computing using fuzzy logic," Simulation Modelling Practice and Theory, vol. 101, p. 102017, 2020/05 2020.

[72] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," presented at the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, 2010. Available: http://dx.doi.org/10.1109/waina.2010.85

[73] S. Hamadah, "A survey: a comprehensive study of static, dynamic and hybrid load balancing algorithms," International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN, pp. 2249-9555, 2017.

[74] S. Begam, S. M, and S. N. R, "Load Balancing in DCN Servers through SDN Machine Learning Algorithm," ed: Research Square Platform LLC, 2021.

[75] G. Senthilkumar and M. P. Chitra, "A Novel hybrid heuristic-metaheuristic Load balancing algorithm for Resource allocationin IaaS-cloud computing," presented at the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020/08, 2020. Available: http://dx.doi.org/10.1109/icssit48917.2020.9214280

[76] S. Yeo, Y. Naing, T. Kim, and S. Oh, "Achieving Balanced Load Distribution with Reinforcement Learning-Based Switch Migration in Distributed SDN Controllers," Electronics, vol. 10, no. 2, p. 162, 2021/01/13 2021.

[77] A. Aghdashi and S. L. Mirtaheri, "Novel dynamic load balancing algorithm for cloud-based big data analytics," The Journal of Supercomputing, vol. 78, no. 3, pp. 4131-4156, 2021/08/23 2021.

[78] Y. Liu, J. Zhang, W. Li, Q. Wu, and P. Li, "Load Balancing Oriented Predictive Routing Algorithm for Data Center Networks," Future Internet, vol. 13, no. 2, p. 54, 2021/02/22 2021.

[79] Y.-W. Ma, J.-L. Chen, Y.-H. Tsai, K.-H. Cheng, and W.-C. Hung, "Load-Balancing Multiple Controllers Mechanism for Software-Defined Networking," Wireless Personal Communications, vol. 94, no. 4, pp. 3549-3574, 2016/10/27 2016.

[80] G. Patel, R. Mehta, and U. Bhoi, "Enhanced Load Balanced Min-min Algorithm for Static Meta Task Scheduling in Cloud Computing," Procedia Computer Science, vol. 57, pp. 545-553, 2015.

[81] S. S. Rajput and V. S. Kushwah, "A Genetic Based Improved Load Balanced Min-Min Task Scheduling Algorithm for Load Balancing in Cloud Computing," presented at the 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), 2016/12, 2016. Available: http://dx.doi.org/10.1109/cicn.2016.139

[82] S. Khan, S. Khan, Y. Ali, M. Khalid, Z. Ullah, and S. Mumtaz, "Highly Accurate and Reliable Wireless Network Slicing in 5th Generation Networks: A Hybrid Deep Learning Approach," Journal of Network and Systems Management, vol. 30, no. 2, 2022/01/27 2022.

[83] S. Rout, S. S. Patra, P. Patel, and K. S. Sahoo, "Intelligent Load Balancing Techniques in Software Defined Networks: A Systematic Review," presented at the 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), 2020/12/16, 2020. Available: http://dx.doi.org/10.1109/isssc50941.2020.9358873

[84] M. R. Belgaum, S. Musa, M. M. Alam, and M. M. Su'ud, "A Systematic Review of Load Balancing Techniques in Software-Defined Networking," IEEE Access, vol. 8, pp. 98612-98636, 2020.

[85] Y. Zhao, C. Liu, H. Wang, X. Fu, Q. Shao, and J. Zhang, "Load balancing-based multi-controller coordinated deployment strategy in software defined optical networks," Optical Fiber Technology, vol. 46, pp. 198-204, 2018.

[86] K. S. Sahoo and B. Sahoo, "CAMD: a switch migration based load balancing framework for software defined networks," IET Networks, vol. 8, no. 4, pp. 264-271, 2019/07 2019.

[87] F. Cimorelli, F. Delli Priscoli, A. Pietrabissa, L. R. Celsi, V. Suraci, and L. Zuccaro, "A distributed load balancing algorithm for the control plane in software defined networking," presented at the 2016 24th Mediterranean Conference on Control and Automation (MED), 2016/06, 2016. Available: http://dx.doi.org/10.1109/med.2016.7535946

[88] F. Al-Tam and N. Correia, "On Load Balancing via Switch Migration in Software-Defined Networking," IEEE Access, vol. 7, pp. 95998-96010, 2019.

[89] S. Yilmaz, A. M. Tekalp, and B. D. Unluturk, "Video streaming over software defined networks with server load balancing," presented at the 2015 International Conference on Computing, Networking and Communications (ICNC), 2015/02, 2015. Available: http://dx.doi.org/10.1109/iccnc.2015.7069435

[90] Z. Hailong and G. Xiao, "SDN-based load balancing strategy for server cluster," presented at the 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, 2014/11, 2014. Available: http://dx.doi.org/10.1109/ccis.2014.7175817

[91] S. Sathyanarayana and M. Moh, "Joint route-server load balancing in software defined networks using ant colony optimization," presented at the 2016 International Conference on High Performance Computing & Simulation (HPCS), 2016/07, 2016. Available: http://dx.doi.org/10.1109/hpcsim.2016.7568330

[92] O. Lemeshko and O. Yeremenko, "Enhanced method of fast re-routing with load balancing in software-defined networks," Journal of Electrical Engineering, vol. 68, no. 6, pp. 444-454, 2017/11/01 2017.

[93]n H. Xu, X.-Y. Li, L. Huang, Y. Du, and Z. Liu, "Partial flow statistics collection for load-balanced routing in software defined networks," Computer Networks, vol. 122, pp. 43-55, 2017.

[94] Y. Liu, Z. Zeng, X. Liu, X. Zhu, and M. Z. A. Bhuiyan, "A Novel Load Balancing and Low Response Delay Framework for Edge-Cloud Network Based on SDN," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 5922-5933, 2020/07 2020.

[95] Y. Wang, X. Tao, Q. He, and Y. Kuang, "A dynamic load balancing method of cloud-center based on SDN," China communication, 2016.

[96] J. Son and R. Buyya, "Priority-Aware VM Allocation and Network Bandwidth Provisioning in Software-Defined Networking (SDN)-Enabled Clouds," IEEE Transactions on Sustainable Computing, vol. 4, no. 1, pp. 17-28, 2019/01/01 2019.

[97] X. Duan, A. M. Akhtar, and X. Wang, "Software-defined networking-based resource management: data offloading with load balancing in 5G HetNet," EURASIP Journal on Wireless Communications and Networking, vol. 2015, no. 1, 2015/06/23 2015.

[98] D. Sun, K. Zhao, Y. Fang, and J. Cui, "Dynamic Traffic Scheduling and Congestion Control across Data Centers Based on SDN," Future Internet, vol. 10, no. 7, p. 64, 2018/07/09 2018.

[99] C. Yu, Z. Zhao, Y. Zhou, and H. Zhang, "Intelligent Optimizing Scheme for Load Balancing in Software Defined Networks," presented at the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), 2017/06, 2017. Available: http://dx.doi.org/10.1109/vtcspring.2017.8108541

[100] L. Guoyan, W. Xinqiang, Z. Zhigang, C. Yadong, and L. Shudong, "A Scalable Load Balancing Scheme for Software-Defined Datacenter Networks based on Fuzzy Logic," International Journal of Performability Engineering, vol. 15, no. 8, p. 2217, 2019.

[101] P. Nallusamy, S. Saravanen, and M. Krishnan, "Decision Tree-Based Entries Reduction scheme using multi-match attributes to prevent flow table overflow in SDN environment," International Journal of Network Management, 2020/10/28 2020.

[102] S. Manzoor, X. Hei, and W. Cheng, "A Multi-controller Load Balancing Strategy for Software Defined WiFi Networks," in Cloud Computing and Security, ed: Springer International Publishing, 2018, pp. 622-633.

[103] C.-Y. Lin, W.-P. Tsai, M.-H. Tsai, and Y.-Z. Cai, "Adaptive Load-Balancing Scheme through Wireless SDN-Based Association Control," presented at the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), 2017/03, 2017. Available: http://dx.doi.org/10.1109/aina.2017.16

[104] A. K. Rangisetti, T. V. Pasca S, and B. R. Tamma, "QoS Aware load balance in software defined LTE networks," Computer Communications, vol. 97, pp. 52-71, 2017/01 2017.