



Review

Intelligent Load Balancing Techniques in Software Defined Networks: A Survey

Thabo Semong ^{1,*}, Thabiso Maupong ^{1,*}, Stephen Anokye ², Kefalotse Kehulakae ³, Setso Dimakatso ¹, Gabanthone Boipelo ¹ and Seth Sarefo ¹

¹ Department of Computer Science and Information Systems, Faculty of Science, Botswana International University of Science and Technology (BIUST), Palapye, Botswana; sd19100154@studentmail.biust.ac.bw (S.D.); gb19100155@studentmail.biust.ac.bw (G.B.); ss18100008@studentmail.biust.ac.bw (S.S.)

² Department of Computer Science and Engineering, University of Mines and Technology, Tarkwa, Ghana; sanokye@umat.edu.gh

³ Department of Electrical, Computer and Telecommunications Engineering, Faculty of Engineering, Botswana International University of Science and Technology (BIUST), Palapye, Botswana; kefalotse.kehulakae@studentmail.biust.ac.bw

* Correspondence: semongt@biust.ac.bw (T.S.); maupongt@biust.ac.bw (T.M.); Tel.: +267-7687-1031 (T.S. & T.M.)

Received: 4 June 2020; Accepted: 30 June 2020; Published: 3 July 2020



Abstract: In the current technology driven era, the use of devices that connect to the internet has increased significantly. Consequently, there has been a significant increase in internet traffic. Some of the challenges that arise from the increased traffic include, but are not limited to, multiple clients on a single server (which can result in denial of service (DoS)), difficulty in network scalability, and poor service availability. One of the solutions proposed in literature, to mitigate these, is the use of multiple servers with a load balancer. Despite their common use, load balancers, have shown to have some disadvantages, like being vendor specific and non-programmable. To address these disadvantages and improve internet traffic, there has been a paradigm shift which resulted in the introduction of software defined networking (SDN). SDN allows for load balancers that are programmable and provides the flexibility for one to design and implement own load balancing strategies. In this survey, we highlight the key elements of SDN and OpenFlow technology and their effect on load balancing. We provide an overview of the various load balancing schemes in SDN. The overview is based on research challenges, existing solutions, and we give possible future research directions. A summary of emulators/mathematical tools commonly used in the design of intelligent load balancing SDN algorithms is provided. Finally, we outline the performance metrics used to evaluate the algorithms.

Keywords: software defined networking (SDN); OpenFlow; network performance; survey; intelligent load balancing

1. Introduction

Technological advancement has had great impact on the amount of traffic on the internet. As a consequence, there has been a need to improve the intelligence, efficiency and reliability of the underlying network. With traditional network architecture, depicted in Figure 1, it is difficult to cope with the growing and evolving needs of the internet [1]. This failure to cope with the rapid growth stems from the fact that traditional networks are not designed to be robust enough to accommodate the fast growth [2]. Traditional networks suffer from network management issues because network hardware devices, such as switches, routers, and load balancers, are all vendor specific. Therefore,

such devices have tightly coupled data and control planes. Their functionality cannot be changed easily without the use of vendor specific network management systems [2].

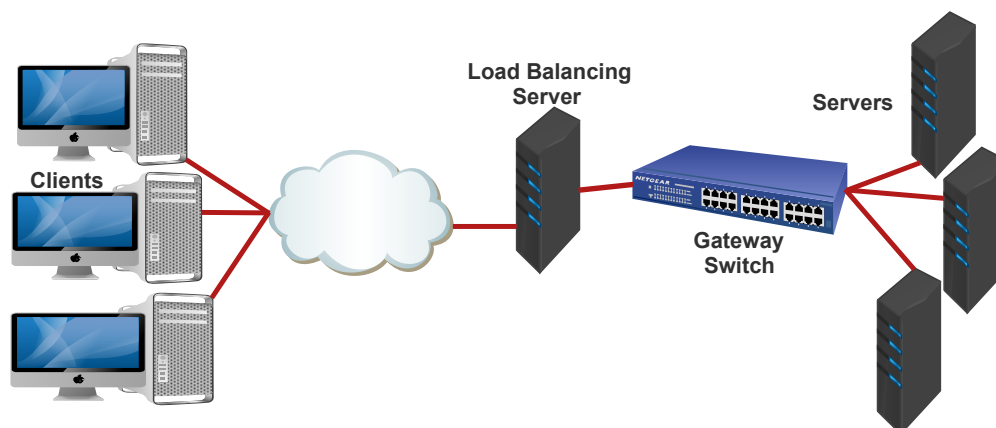


Figure 1. Traditional IP network load balancing model.

To mitigate the short comings of traditional networks, researchers have proposed solution such as software defined networking (SDN) [2–6], and network functions virtualization (NFV) [7–10]. Both SDN and NFV offer the ability for the network design and infrastructure to be abstracted in software and then implemented across different hardware and devices. It is this abstraction in software that enables the two technologies to cope with the failings of traditional networks. SDN further allows for the configuration and running of the network to be defined and modified via programming, thus adding programmability to the network [11–15].

SDN abstracts physical network devices and moves all the decision-making to the control plane, see Figure 2. It is in the control plane where all the network intelligence takes place including packet forwarding, and network management policies. It is this structure of SDN that allows for robust administration, and scalability of the network [5,14,16,17]. SDN controller enables the entire network to be controlled from a centralised point. Therefore, simplifying network design, and the network control becomes vendor-independent, which results in the utilization of simpler network devices.

An important part of designing and managing a network includes the process of effectively distributing traffic among network devices, i.e., load balancing [5,18,19]. An effective load balancer is one that optimizes network parameters such as latency, resource utilization, throughput, and fault tolerance with minimal power consumption. In traditional networks, load balancing is usually carried out by a dedicated server, see Figure 1. To cope with the ever changing network demands, dynamical load balancing servers have been implemented [18]. For load balancing in VFNs, we refer the reader to [9] and references within. In this paper, we focus on analysis and review of different load balancing techniques on SDNs. In order to have a comprehensive survey, we carry out the following:

- We highlight the key elements of SDN and OpenFlow.
- Review of effective load balancing technique in SDN.
- We highlight research challenges, existing solutions, and research directions for load balancing in SDN.
- Finally, we give a summary of the emulators/mathematical tools commonly used in the design and test of load balancing algorithms in SDN. We also outline the performance metrics used to evaluate the algorithms.

The rest of the paper is organized as follows: A brief overview of SDN architecture is covered in Section 2. In Section 3, we provide a comprehensive review of load balancing routing. Related works on intelligent load balancing in SDN are reviewed in Section 4, where we cover brief summaries of the existing methods. In Section 5, we thoroughly discuss and analyze results and provide future research prospects on SDN based load balancing techniques. We then finally conclude the paper in Section 6.

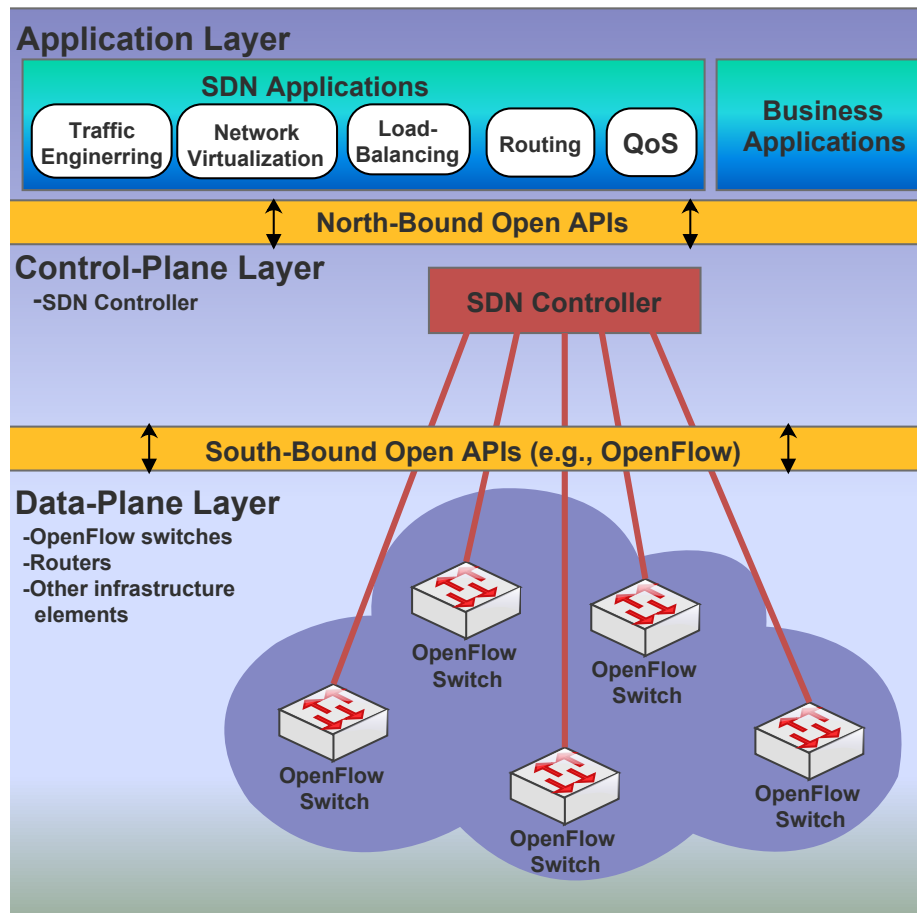


Figure 2. Overview of software defined networking (SDN) architecture with its main planes: data, control, and application planes, and the communication channels [15].

2. The SDN and OpenFlow Architectures

In order to set a foundation for a comprehensive review of load balancing in SDN, and for the completeness of this paper, we do an overview of SDN and OpenFlow architecture. Part of the overview is to provide a clear separation between SDN and OpenFlow, as the two have been mistakenly regarded as synonyms in some instances.

2.1. SDN Architecture

We review SDN main technical features, and we refer the reader to [14,20–24] for comprehensive details on SDN architecture. The commonly adopted SDN architecture was proposed by the Software-Defined Networking Research Group (SDNRG) [25], depicted in Figure 2. This architecture comprises three different planes, as follows.

Application plane: This layer is comprised of the business process and SDN applications, and provides for user interfaces.

Control plane: This is where the intelligence of SDN resides. The control plane comprises of SND controller, which is responsible for the running of the network. For example, all policies, relevant decisions regarding packet forwarding, and rules for the infrastructure are processed by the controller. Communication between the application and control planes is via the northbound application programming interfaces (APIs) [14]. These APIs are usually open sources which enables flexibility, since one can always modify them to suit their needs.

Data plane: This plane is also known as the infrastructure plane, and it consists of all network forwarding devices, i.e., routers, switches and bridges, etc. Communication between the data and control planes is via the southbound APIs and the most commonly adopted API is OpenFlow, see [26].

It is this SDN structure that provides for easiness and flexibility for coming up with new services and applications, such as load balancing, routing, policy enforcement, or any customized application that may be required for optimized network resource services [14].

2.2. OpenFlow Technology

OpenFlow facilitates for routing decisions, forwarding of rules like load-balancing, security and quality of service (QoS) from the SDN controller to be pushed onto devices in the data plane. It is the commonly adopted SDN protocol for communication between the network devices and the SDN controller, i.e., southbound API [26]. OpenFlow was first standardized by the open networking foundation (ONF), and it has since been adopted by information and communication technology (ICT) companies such as Cisco, HP, and Google [25,27]. OpenFlow protocol enables direct access and configurations of network devices such as routers and switches. As an example, OpenFlow enables access to the flow table and passing of instructions to the switches on how to forward network flows. This, along with programmability of the SDN controller, ensures that the network changes can be implemented with minimal delay.

There are two main types of OpenFlow-based switches: OpenFlow-hybrid and OpenFlow-only [28]. OpenFlow-hybrid switches make it possible to execute both OpenFlow operation and normal ethernet switching operation like traditional L2 ethernet switching, virtual local area network (VLAN) isolation, L3 routing (IPv4 routing, IPv6 routing), access control list (ACL), and QoS processing. On the contrary, OpenFlow-only switches allow for OpenFlow operations only, meaning that all packets are processed by the OpenFlow pipeline, and cannot be processed otherwise.

2.3. OpenFlow Architecture

OpenFlow controller, OpenFlow switch, and OpenFlow protocol constitute the main elements of an OpenFlow network architecture, Figure 3. In addition, how a software controller and switches share information is depicted in Figure 3. This sharing of information has abstraction of network elements and enabled by the OpenFlow protocol. OpenFlow architectures also provide a concentrated management of the global network; hence, the controller has total comprehension of the entire network, which makes it easier to implement topology recovery in case there are some eminent changes, e.g., during link failure. In case of load balancing, the architecture allows for a global view of the network and constant assessing of the load of several links. This allows for easy modification of forwarding rules such that the equality of the load is maintained. OpenFlow networks are based on the ability to conceptualize all traffic as flows, meaning that different flows will have different entries in a flow table and different rules can be stated for each entry.

The OpenFlow architecture, Figure 3, is not to be confused with SDN architecture, Figure 2, as SDN is an abstraction of the network in software. SDN allows for the network to be programmable, hence giving network engineers the flexibility to develop and deploy network management tools. On the other hand, OpenFlow is an API which enables communication between two SDN planes, being the control and data planes. To be more precise, OpenFlow protocol makes the sharing of information between the SDN controller and network devices in the data plane easier.

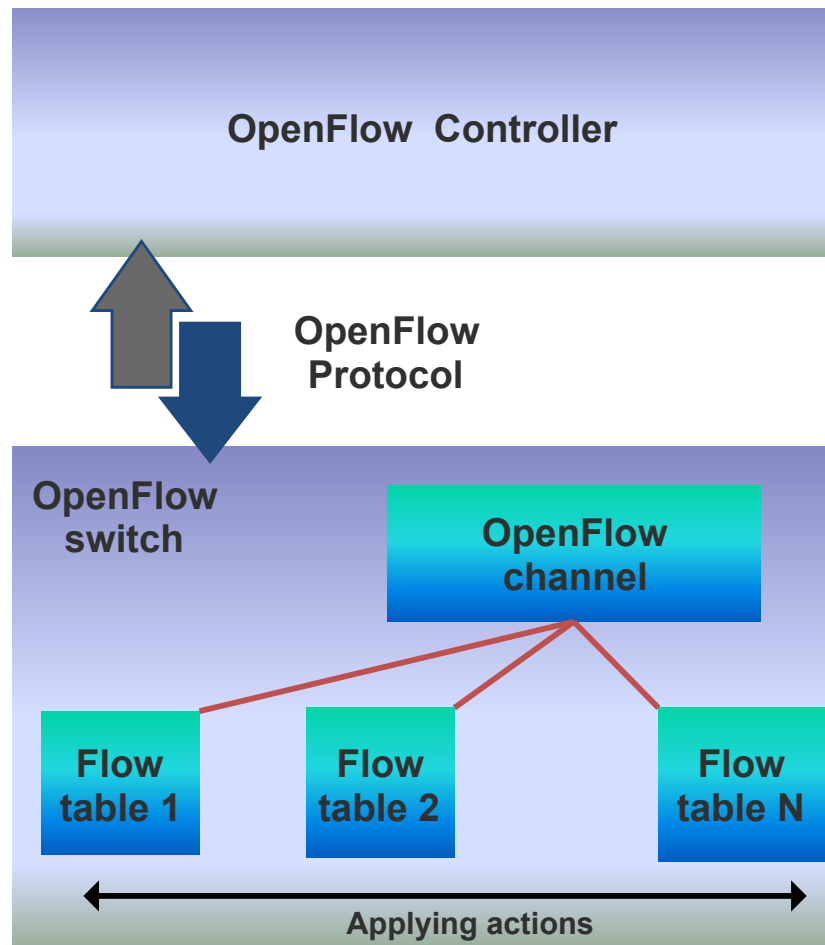


Figure 3. OpenFlow architecture.

3. Intelligent Load Balancing Routing

As already mentioned, intelligent load balancing has become a necessity in our networks because of continuous traffic volume increase, user demand growth, and applications' complexity. Most organizations need to build adaptable and robust networks that can withstand new challenges without compromising performance, security or service, hence the need for intelligent load balancers. Intelligent load balancing techniques split the network traffic into multiple flow paths to minimize congestion on any of the flow path [29]. Moreover, the intelligent load balancing technique allows for minimized response time, optimized traffic, and maximized throughput. Therefore, when using software intelligent load balancers, network managers can enjoy benefits like predictive analytics that make it possible to discover traffic bottlenecks before they even occur. This simply means that the software intelligent load balancers are also used to provide actionable insights to an organization. In this section, we look at how routing is done in SDN to facilitate for intelligent load balancing. In order to appreciate the power of intelligent load balancing routing in SDN and its advantages, we first cover a summary of load balancing routing in IP networks.

3.1. Load Balancing Routing in IP Networks

There is not much flexibility in routing when using traditional IP networks. Normally, IP network routers use common link state routing protocols like open shortest opath first (OSPF) [30,31] and intermediate system to intermediate system (IS-IS) [32] to acquire link metrics and the network topology. Algorithms that utilize the shortest paths between the source and destination node pairs is used in IP networks, and this can lead to congestion at some bottleneck links due to the static

routing path. Even though one can argue that changing the link weight will lead to change in routes, this can not be the best approach for intelligent load balancing in IP networks—the reason being that routers will have different receiving times when a link state change message is shared. This leads to a situation whereby the network states would become inconsistent when viewed from different routers. Whenever there are link metric updates, chances are that some transient traffic loops would be formed because there are no precise synchronization among the routers [33,34].

Therefore, it is very difficult to ensure that the load-balancing performance in the IP network is optimal [1], mainly due to the fact that there is no global view of the network. This lack of global network routing view also means that routing tables in IP networks contain destination network information only, hence routers can only know the next-hop network and the interface it must use.

Some of the well-known load balancing techniques that are IP based are discussed in [35–40]. Some works focused on the relative load on the different network instances [41], while some early works were based on round robin domain name system (RR-DNS) to allocate inbound connections towards a group of servers [42].

3.2. Intelligent Load Balancing Routing in SDN

Traditional load balancers are not programmable due to the fact that they are vendor locked by design, and this means that network managers and administrators are constricted from coming up with their own load balancing algorithms. The main contrast of SDN load balancers is that network managers and administrators are provided with a platform to design and build their own intelligent load balancing techniques, thus SDN load balancers are more flexible and programmable. Moreover, there is no separate hardware device needed in SDNs to act as a load balancer. In SDN, the network topology is used to make intelligent routing decisions between edge devices, which takes into account protection and bandwidth requirements for every service that is provisioned across the network. The outcome of the routing algorithm is a set of flow table entries that are installed in OpenFlow-capable switches to enable forwarding of packets from end to end.

The SDN architecture allows for the removal the routing protocols at the hardware level, thus resulting in an improved network management and diagnosis. The SDN controller, working together with the intelligent load balancing application via the OpenFlow protocol, makes data path control decisions without having to rely on algorithms defined by traditional network equipment. An SDN-based load balancer saves running time by having control over an entire network of application and web servers. Its main objective is to reduce overloads as optimally as possible, utilize network resources, and obtain minimum response time by distributing the network workload. Intelligent load balancing is also a basic problem in many practical systems in daily life; hence, various techniques based on SDN architecture to be discussed in Section 4 and references within have emerged.

The SDN controller is able to intelligently access and use the information of the state of the network at lower layers together with knowledge of the upper networking layers to make forwarding decisions. This means that information regarding the status of the network infrastructure, the applications running on them, and network traffic congestion all have effects on intelligent load balancing decision-making process. The SDN controller has the benefit of the global view of the network at the beginning of network construction, hence it can discover all possible flow paths between each source-destination node pairs by updating topology information of the global network. This global network views ability by the SDN and enables the load condition of every global path to be evaluated, resulting in a more optimized intelligent load balancing decision-making. This is demonstrated by the SDN network architecture shown in Figure 4.

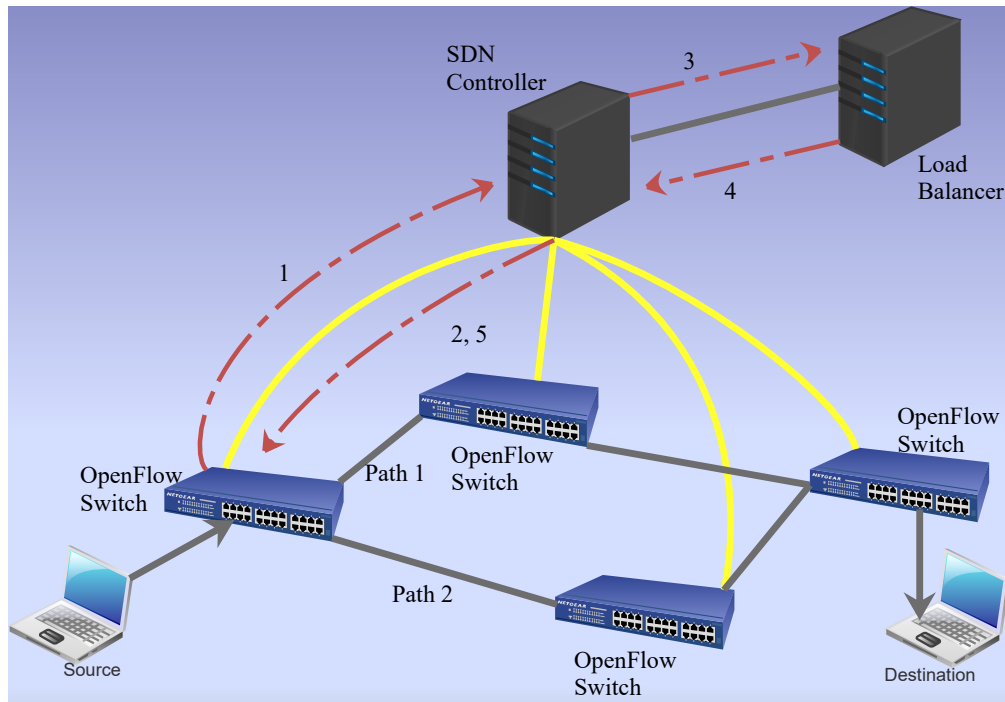


Figure 4. Network architecture for the SDN load balancing system.

4. Review of Load Balancing in SDN

Effective load balancers intelligently determine which device within the network farm is best to process an incoming data packet. Doing so requires algorithms that are geared towards distributing loads in a specific way. Algorithms vary widely, depending on whether a load is distributed on the network or application layer. Algorithm selection impacts the effectiveness of load distribution mechanisms and, consequently, performance and business continuity.

Optimizing network performance in SDN, like any other network, necessitates the need to find the best load balancing strategies. Various techniques have been proposed in literature to optimize network flow and to achieve intelligent load balancing among different links, servers, cloud, and controllers.

4.1. Controller Load Balancing

In order to increase capability of the control plane, sometimes it is essential to decentralize the SDN controller. The decentralization normally results in the control plane comprised of more than one controller. In the following, we review load balancing in the case where the control plane consists of several SDN controllers.

The reliable and load balance-aware multi-controller deployment (RLMD) scheme is proposed in [43]. According to the authors, the main objective of RLMD is to resolve the controller traffic load disparity and control plane accuracy issues that arise when the control and data planes split-up in the multi-controller SDN network. RLMD as a progressive scheme ensures that equitable allocation of controller traffic loads and effective dependable deployments of controllers are in place. The authors also came up with a multiple domain partition (MDP) algorithm that is able to match controllers and switches based on the load balancing rate of the controller and the node attractability. When tested against other typical strategies, their algorithm managed to balance the allocation of the controller traffic loads much better, and also enhanced the reliability of the control plane.

To solve the well-documented problem of load balancing in the SDN control plane, the authors of [44–47] focused on distributed controllers architecture for SDN control plane which can dynamically balance its traffic load. The researchers of these works were able to provide solutions to the problem of handling multiple controllers, hence providing a solution to open research on how to match each

controller to a switch and where to place the controllers. These works effectively split the switch requests among different controllers in order to avoid congestion.

A switch migration based strategy called *ElastiCon* is proposed in [48]. *ElastiCon* periodically balances the load of controllers in a pool, and ensures good network performance at all times irrespective of the traffic dynamics. The downside of *ElastiCon* is that in some cases it leads to a high network overhead. Some similar works that enhance efficiency in the SDN controller were presented in [48–50]. The focus of these research works were to select a switch to migrate as a process of load balancing. This process reduces overloaded controllers, hence avoiding degradation of the network performance.

The authors of [51] came up with a load balancing scheme based on switch groups of multiple controller architecture to improve time efficiency of the load balancing process. The proposed scheme uses a switch selection algorithm and target controllers selection algorithm to solve the load oscillation problem among controllers.

In summary, the main idea across the articles cited above is to implement a controller load balancing architecture that can utilize one of these controllers to partition control traffic and distribute it across different available controllers.

4.2. Server Load Balancing

We now consider sever based load balancing, in the case where a network has multiple servers.

In [52], the authors propose a load balancing strategy for distributing clients requests across multiple servers. The strategy ensures that the next request is forwarded to the least-loaded server. The authors evaluated their strategy with random-based and round-robin, and they also analyzed their effects on the overall network performance.

A genetic algorithm (GA) is introduced in [53] for server load balancing in the OpenFlow based system. GA is based on the biological process of natural selection whereby the favorable genetic traits are passed to the offspring. The algorithm has three types of operators, which are selection, crossover, and mutation. The researchers make an assumption that multiple clients make multiple requests to a load balancing pool of servers with varying loads. Their application of GA uses a roulette wheel as the selection method, single point crossover and single point mutation. Its fitness function is calculated based on the coefficient of variation given K servers with varying workload. The performance objective was to minimize the average for the coefficient of variance. The GA was evaluated against load based, random, and round robin algorithms and it gave better results.

To optimize the distribution of cloud consumer application requests, [54] proposed a partial swarm optimization (PSO) based load balancing technique. This scheme uses swarm intelligence to minimize average application response time, maximized throughput, and for better resource utilization. This proposed scheme actively configures and comes up with traffic flow paths needed, according to the network traffic load.

In [55], the authors propose a solution for mitigating against congestion in a pool of servers. The researchers used bandwidth and round-robin metrics to measure the performance of their proposed system. From the analysis of this work, the bandwidth strategy was more effective in terms of throughput and response time when compared to allocating servers based on round-robin strategy. Researchers in [56–61] studied similar approaches for allocating requests to servers.

The SDN based server load balancing strategies are mainly concerned with ensuring that requests are distributed efficiently. The common argument of adopting SDN technology among researchers is that it does not need any dedicated hardware when compared to the traditional load balancers. Moreover, OpenFlow protocol enables solutions that are more effective as servers are pressured to handle lot of access requests in a short time. The above studied literature demonstrated how SDN technology can be utilized to intelligently solve the congestion problem on server clusters due to the ever increasing number of users in our networks.

4.3. Load Balancing in Wireless Links

IEEE 802.11 does not come with an automotive load balancing in WLAN setup, hence load balancing in the existing wireless LANs can be a troubling issue [62]. Algorithms deployed on SDN controller can be utilized to dynamically balance access points (APs) load by choosing the less loaded ones from the sharing locality for clients association.

A two-tier architecture for managing load in access points (APs) is proposed in [63]. In this scheme, the SDN controller gets information from the APs and decides on the maximum allowed load level an AP can accept association requests without consulting the SDN controller. Their solution strikes a fair balance as it adjusts automatically the maximum load level an AP can act on its own. The proposed solution is also applicable to any wireless devices that support OpenFlow.

In [64], authors studied an adaptive load balancing technique for Wi-Fi APs. The proposed strategy is made up of two sections: arrival event detection and adaptive load balancing. APs send their traffic load and connection situation in real time to the SDN controller. Then, an algorithm is used to derive the optimal association relationship between APs utilizing the global vision of SDN controller. A similar strategy is also proposed in [65–67].

Variance analysis for massive mobile users based on an efficient SDN load balancing scheme was proposed in [68]. The proposed scheme has good scalability to support massive mobile users, and can effectively monitor dynamic workload in mobile communications. The authors also argued that their work has better load balancing when compared to existing solutions. The SDN technology simplifies the network, and it was utilized to enable intelligent traffic management from a global point of view for quick reaction to dynamic network conditions. The proposed scheme showed better throughput, better response time, and better overhead, but was faced with issues of low availability, high packet losses, and high energy consumption.

In [69], the authors proposed a software defined wireless sensor network (SDWSN) based load balancing technique. This work formulated a mathematical model using multi-path routing technology to reduce the maximum link utilization in the wireless sensor network. The proposed technique utilizes the advantage of the global view of the wireless sensor network provided by software defined network to implement an effective and flexible traffic scheduling. The focus was on how to intelligently manage the wireless sensor networks infrastructure in a smart city. In [70], an SDN technology was utilized to provide a solution to the problem of load balancing on heterogeneous wireless access networks. They designed an algorithm to consider both the network load and the user experience for effective load balancing on the radio access network (RAN). Their simulation results indicate that users can enjoy good connection experience and, at the same time, the RAN traffic load remain balanced.

4.4. Communication Path Selection Load Balancing

Effective load balancing of a network traffic across multiple links has been a common challenge faced by network engineers as networks increase in speed and size [71]. SDN enables schemes that can easily divide the traffic flow among similar paths in order to enhance the performance of the network in a more scalable and dynamic manner [72]. In data center networks that are designed for multi-path routing, proposed load balancing schemes should distribute the traffic effectively by ensuring that neither path is under-utilized nor over-utilized. Networks like data center are dynamic and traffic congestion can happen any time because of the nature of traffic variation.

In [73], the authors proposed a dynamic load-balanced path optimization (DLPO) scheme, which is applicable in various OpenFlow-based data center network topologies. The technique proposed enables for change of paths during flow transmissions. This ensures that effective traffic load balancing through various communication paths, and avoid congestion challenges in SDN data center networks. A technique called LoAd-BalancEd routing wIth OpenFlow (LABERIO) [1] was one of the earlier SDN based research works to propose a dynamic path optimization scheme to balance network traffic during flow transmissions. In this scheme, a traffic flow could be diverted to a new path during the transmission if the link loads of a network are unbalanced. The drawback of this

proposed technique is that, since its path optimization algorithm is dependent on network topology, the technique would have to be modified whenever there is a change in the network topology.

A load balancing routing algorithm for fat-tree networks that distributes traffic across multiple paths equally is introduced in [74]. This algorithm has delay drawback as it takes a long time to be executed. This drawback is addressed in [75], where the authors designed a dynamic load balancing algorithm with congestion control. They ensured that they looked into the weights of edges and nodes in order to find the shortest path between each nodes pair. The algorithm proposed in [75] also has its own drawback, however, as it is only suitable for small networks.

Strategies that optimize network load and at the same time reduce execution time, overheads, latency, and improve bandwidth utilization are covered in [76–78]. Even though these proposed techniques had some success in terms of improving the overall load balancing, they also suffered from packet loss and high energy consumption.

4.5. Artificial Intelligence Based Load Balancing

Researchers in [79–84], have utilized the advantages of SDN paradigm and integrated artificial neural networks (ANNs) to propose new load balancing techniques. ANNs are described as flexible mathematical structures that are capable of pin-pointing complex nonlinear connection between input and output data sets [85]. The model is found to be efficient and useful, especially in situations where the characteristics of the processes are not easy to explain using physical equations. ANNs are useful because they have no limits on input vectors, and due to the uncertainty of the traffic flow in network, a large amount of data can be collected to form a data-set. There is a high possibility that this formed complex traffic flow data-set follows no certain probability distribution, hence ANNs are most suitable for handling this uncertain network traffic data. The structure of the ANNs is depicted in Figure 5.

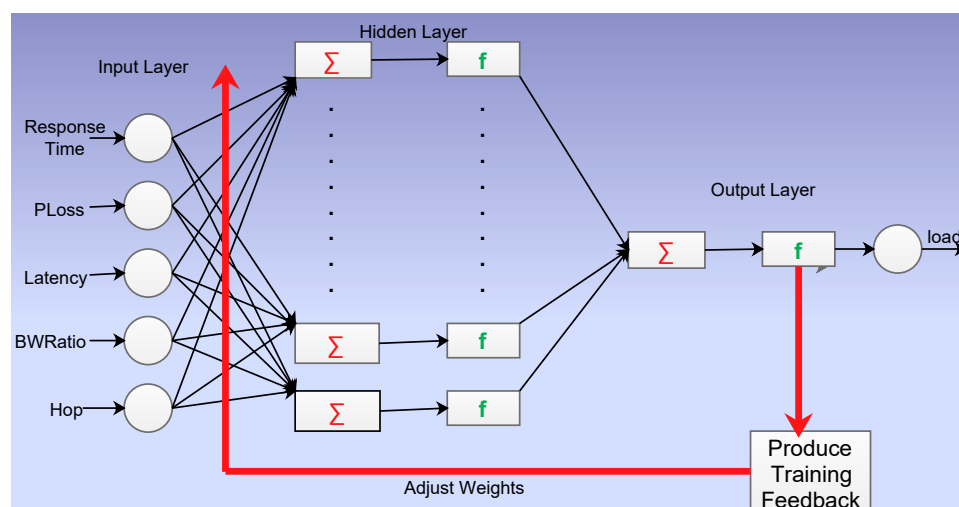


Figure 5. Load balancing neural network structure.

In [82], ANNs are used to overcome the challenge of overloaded or underloaded virtual machines (VMs) in cloud computing. The authors argued that the performance of applications eventually degrade due to overloaded and underloaded VMs resulting in poor resource utilization. To overcome this problem, they came up with an algorithm that can raise the processing speed, identify heavily loaded VMs, and optimize the resource utilization. The proposed algorithm performs better than the other two algorithms that it was compared with in terms of data migration time.

Reinforcement learning and deep neural networks are used to enhance both load balancing policies and routing in [86]. The scheme leverage based on the fact that, with the global view of the network provided by the SDN controller, it becomes easier to get enough information on network flow states like traffic, links, bandwidth, and hence the ability to establish an appropriate policy. This

scheme was compared to the shortest path, round robin, and ANN load balancing algorithms. Its performance was found to be better in terms of enhancing network utilization and minimizing delay.

Another load balancing strategy is covered in [87]. This strategy uses reinforcement learning to design a switch migration algorithm. The aim was to effectively select a set of switch migration queues, so that eventually the system can attain a global optimum from some local one.

As a way of ensuring that the least congested path is selected, one needs to confirm the path traffic load condition by intelligently analyzing the integrated path information. Traffic flow estimation and assessment is an important task that needs to be performed in order to ensure network quality of service and to enhance network performance. When ANNs are deployed on SDN, it ensures rationality and precision in terms of traffic flow path load condition, hence giving the controller an opportunity to make a more informed load balancing decisions.

4.6. A Summary of the Reviews

To conclude this section, a summary of widely adopted SDN load balancing algorithms is covered in Table 1. Here, we look at the performance objectives that the authors wanted to achieve as well as the selection criteria and mechanism used. Most of the literature works used centralized criteria as these works were based on SDN. Other works combined SDN and the traditional IP network into a hybrid network. This is because currently not many networks are fully SDN based. Hybrid setup can be the best way for moving into a fully SDN based network slowly, due to cost implications. We also give some summary of what the strategy used by the authors entails.

Furthermore, in Table 2, we outline possible future improvements and research direction on existing SDN based load balancing techniques. To do this, we outline the research challenge, the existing solution, and finally how the solution may be improved or the research challenge be addressed. For more on SDN based intelligent load balancing techniques, we refer the reader to [88–91].

Table 1. A summary of various intelligent load balancing techniques in the software defined networking (SDN).

Algorithm/Strategy	Performance Objective	Selection Criteria and Mechanism	Summary
Deep neural networks and Q-learning (DNQ) [12,86]	Reduction of packet loss rate with different load	Centralized intelligent center	In DNQ, intelligent techniques were used to carry out these three functions; path selection, the important nodes and flow forecasting.
Least connection [92,93]	Resource utilizations optimisation	Centralized and cooperative approach	A server with the least number of active connections is allocated more connection to balance the traffic.
Software-defined sensor network load balancing (SDSNLB) algorithm [69]	Throughput, link load jitter	Centralized	SDSNLB main objective is to allocate network traffic to different flow paths as a way of enhancing functionality. The strategy is to have balanced multipath allocation of bandwidth for optimal and productive network resource use.
Dynamic agent-based load balancing (DA-LB) [54,82]	Efficient and adaptive resource management.	Centralized intelligent center	The global visibility of SDN is used to efficiently migrate virtual machines in data center network.
Reliable and load balance-aware multi-controller deployment (RLMD) [43]	Node efficiency, node attractability, path quality, controller load balancing rate	Centralized and cooperative approach	RLMD proposed a scheme that managed to achieve effective deployment of controllers and can also balance loads successively among the controllers.
vSDN [94]	Average load, average delay, bandwidth utilization and throughput	Centralized intelligent approach	The strategy here is to utilize virtual SDN controller as a VNF to enhance load balancing. Under vSDN, a standby vSDN controller can be used to share the traffic load if the flow increases or become uneven.
Fuzzy synthetic evaluation mechanism (FSEM) [95]	Dynamically select the optimal path	Centralized	Network flow is sent to those flow paths under the OpenFlow switches, so that the SDN controller can utilizes its global view of the network and install the flow-handling.

Table 1. Cont.

Algorithm/Strategy	Performance Objective	Selection Criteria and Mechanism	Summary
Switch migration based decision-making (SMDM) [76]	Response time, load distribution and migration cost	Centralized approach	SMDM's primary objective is the process of choosing a master controller that can enhance load balancing factor. Based on low cost, a switch will be chosen for migration.
Adaptive load balancing Scheme [96]	Throughput and loss rate	Centralized and cooperative approach	This is a new adaptive technique used in data centers that leverage on SDN for load balancing.
Self-adaptive load balancing (SALB) [97]	Throughput testing, load balancing time, bandwidth utilization and loss rate	Centralized approach	The objective of SALB scheme is to ensure that at the same time load balancing and distance between devices are considered effective.
Double deep Q network based VNF placement algorithm (DDQN-VNFPA) [98]	Path delay, running time of VNFs, the number of VNFs and utilization ratio of VNFs	Centralized	This is a customized algorithm designed using gathered information to ensure that the network performance is optimized. This algorithm does enhance network performance when it comes to metrics like delay, throughput and load balancing.
Load variance-based synchronization (LVS) [99]	Loop-free forwarding, synchronization overhead	Centralized	LVS enhances load-balancing execution because it reduces synchronization frequency and also removes forwarding loops.

Table 2. A summary of intelligent load balancing techniques in the SDN based on research challenges, existing solutions and research directions.

Ref.	Research Challenge	Existing Solution	Research Direction
[100]	<ul style="list-style-type: none"> -Mission-critical network failure. -Aviation management problem. -Management and monitoring of energy consumption. 	<ul style="list-style-type: none"> -Migration of switches using active and dynamic traffic load re-balancing on SDN. -Using SDN to ensure that when a certain traffic load limit is exceeded by a controller, other controllers can help to share that traffic load. 	<ul style="list-style-type: none"> -Network visualization through SDN. -Designing of SDN based architecture for critical missions. -Energy load balancing strategies based of OpenFlow technology.
[101]	<ul style="list-style-type: none"> -The issue of degradation of the controller performance. -Optimization problem based on minimum cost bipartite task. -The challenge of discovering a many-to-one matching between the sets regarding the cost function. -Addressing the optimization problem of switch-to-controller assignment 	<ul style="list-style-type: none"> -Dynamic control devolution and dynamic switch-to-controller association. -WAN dynamic controller provisioning. -Addressing the SDN controller placement problem utilizing the idea of network partitioning. -Optimizing powerful decision-making based on game theory. 	<ul style="list-style-type: none"> -Managing a trade-off between the current loads on a controller and the switch-to-controller round trip time (RTT). -Understanding the idea of load-driven penalty. -Designing of environments that can handle multi-controller.
[97]	<ul style="list-style-type: none"> -Poor system performance due to poor traffic load management in the controller. -The problem of nmanaged heterogeneous traffic in the network. 	<ul style="list-style-type: none"> -Enhancing overall system performance and saving energy and time in mobile devices using a distributed algorithm. -Path load-balancing using the mechanism of fuzzy synthetic evaluation. -Utilization of server response time to balance traffic load. 	<ul style="list-style-type: none"> -Algorithm design for optimal switch-to-controller round trip time. -Load-driven distributed SDN controllers. -Robust and scalable multi-controller environments.
[69]	<ul style="list-style-type: none"> -Global resource balanced utilization problem in wireless sensor networks. -Requirements of wireless sensor network infrastructure for smart cities, based on high flexibility and adaptability. -Effects of traffic load imbalances on the network service quality 	<ul style="list-style-type: none"> -Designing of a load-balancing OpenFlow based mobile network architecture for achieving centralized control of energy-aware on demand. -Designing of an SDN enabled routing protocol for wireless multi-hop network. -Designing of an SDN-IoT integrated control system for smart cities. 	<ul style="list-style-type: none"> -Intelligent load balancing scheme for software defined wireless sensor network (SDWSN). -Intelligent and flexible network flow scheduling for SDN. -Traffic optimization routing on network links for smart and saver city sensor network.

Table 2. Cont.

Ref.	Research Challenge	Existing Solution	Research Direction
[102]	<ul style="list-style-type: none"> -End devices failure to mitigate against congestion due to the fact that traditional networks cannot use packets drop as an indication of congestion -Unavailability of the global view of the network topology from a centralized point 	<ul style="list-style-type: none"> -Hashing reroutes flows on multiple paths using ECMP. -Utilizing reinforcement learning technique for the adoption of a QoS-aware adaptive routing (QAR). -Using reinforcement learning technique to select the best traffic flow paths. 	<ul style="list-style-type: none"> -Utilizing SDN controller to design some intelligent methods for congestion control. -Reinforcement learning based rerouting algorithm based on SDN. -Rigorous evaluation of instant congestion control using fuzzy logic.
[103]	<ul style="list-style-type: none"> -Failure to dynamically optimize the customized strategies by intelligently changing the reward function. -Intelligent decision-making by software defined network controller. 	<ul style="list-style-type: none"> -Utilizing a multi-machine learning strategy based on a route pre-design scheme. -The implementation of quality of service adaptive routing in OpenFlow protocol using QoS-aware reward functions. -Replacement of Q-table in conventional Q-learning strategy using artificial neural network. 	<ul style="list-style-type: none"> -Utilizing deep reinforcement learning on OpenFlow technology. -Optimized mechanisms to enhance intelligent SDN packet flow. -Adoption of the most effective black-box in continuous time
[104]	<ul style="list-style-type: none"> -Routing optimization problem. -How to get near real-time optimal execution via a dynamic traffic flow scheme. 	<ul style="list-style-type: none"> -Enhancing media stream communication on end-to end HTTP through a Q-learning architecture. -Enhancing the global data traffic volume using quality of service information. 	<ul style="list-style-type: none"> -Deployment of a deep reinforcement learning strategy that can utilize recurrent neural network. -Deep reinforcement learning based on software defined networking.

5. Results and Comparisons

In this section, we discuss some open issues and observations made, as well as presenting a comparative analysis of different sampled intelligent load balancing metrics on SDN.

5.1. Performance Metrics and Implementation Tools

For this discussion, we selected 15 articles (Table 3) for qualitative metrics used to measure intelligent load balancing techniques in SDN. The selection is based on the fact that the articles cover some of the most used load balancing methods. From the selected articles, we discuss seven metrics used and seven different mathematical tools or emulator software adopted, as presented in Table 3.

Table 3. Qualitative comparison between different intelligent load balancing techniques in the SDN schemes based on the mathematical tools/emulators and the performance metrics adopted.

Ref.	Tools/Emulators	Performance Metrics						
		Throu-Ghput	Fairnes	QoS	Complexity	Congestion Control	Interference Mitigation	Delay
[105]	Mininet [106,107]	✓	×	✓	×	✓	×	✓
[103]	OMNET++ [108]	✓	×	✓	✓	✓	×	✓
[49]	Mininet	✓	✓	✓	✓	✓	×	✓
[109]	Mininet, Iperf [110]	×	✓	✓	✓	×	×	✓
[100]	Mininet	✓	×	✓	✓	✓	✓	✓
[15]	Mininet	✓	×	✓	✓	✓	×	✓
[102]	Mininet, Iperf	✓	×	✓	✓	✓	×	✓
[50]	Mininet	✓	✓	✓	×	✓	×	✓
[69]	Matlab	✓	✓	✓	✓	✓	×	×
[111]	OMNET++, Python	✓	×	✓	×	✓	✓	✓
[44]	.Net	✓	×	✓	×	✓	×	✓
[68]	Mininet	×	✓	✓	✓	×	✓	✓
[112]	Mininet, Iperf	✓	×	✓	✓	✓	×	✓
[113]	Mininet, MaxiNet [114]	✓	✓	✓	✓	✓	×	✓
[115]	Mininet	✓	×	✓	✓	×	✓	✓

Using Figure 6, one can deduce that, from the selected articles, 87% used throughput, 93% used delay, 40% used fairness, 100% used QoS, 73% used complexity, 80% used congestion control, and 27% used interference mitigation as qualitative metrics to measure the effectiveness of their proposed techniques. Some of these metrics are very critical when it comes to load balancing because they are attributes of good intelligent load balancing techniques, for example, QoS and delay. It is not surprising that these two metrics have been used by all or most of the researchers. Fairness and interference mitigation are the least used metrics by researchers from the selected articles. In most cases, a technique is designed to intelligently balance the network traffic with respect to a chosen metric. Moreover, such a technique would be regarded as a success even if it does not mitigate against interference and fairness, if it can effectively balance the the traffic flow. In our opinion, if more attention is not given to interference mitigation, this can eventually lead to degradation in performance of the technique. This is because interferences can crop up at anytime in the network. Other metrics used in the literature, not covered under this subsection, include utilization, latency, and run time. It is also worth noting that metrics adopted are based on the objective, as well as the techniques that are used to test the effectiveness of the proposed solution.

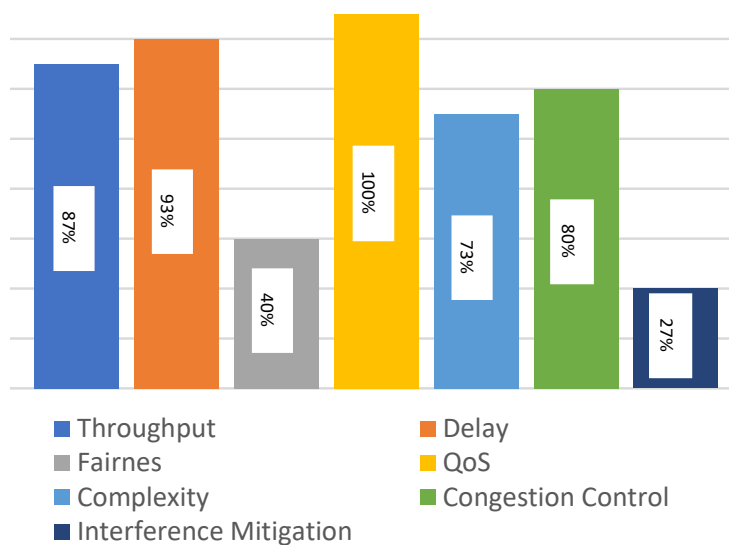


Figure 6. Percentage of load balancing metrics from sampled papers as illustrated by Table 3.

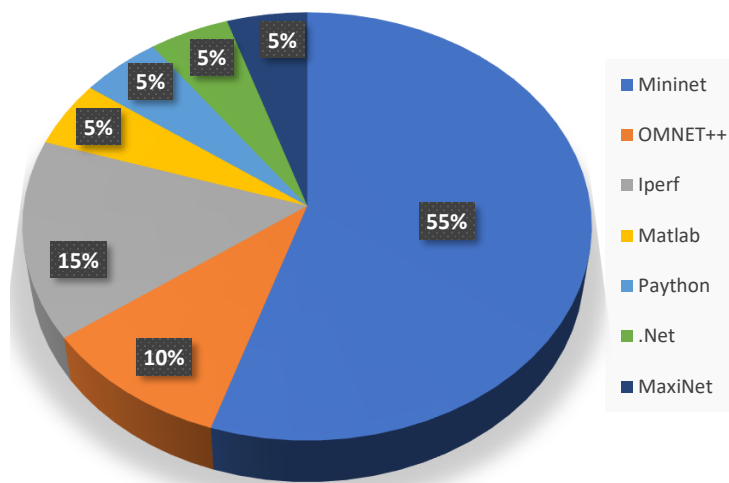


Figure 7. Distribution of the selected articles by the mathematical tool/emulator adopted as shown in Table 3.

We now look at the most used tools for the implementation of intelligent load balancing techniques in the SDN environment. As illustrated in Figure 7, 55% of the sampled researchers use Mininet as an emulator to test their proposed techniques. This is because Mininet is a powerful tool in terms of interactive development, demonstrations, and testing when using OpenFlow and SDN [106]. Iperf is used 15% of the time to generate network traffic, while OMNET++ is used 10%, and, finally, other tools like Matlab, Python, .Net, and MaxiNet are used a total of 20% of the time, with each tool contributing 5%. Some of the tools like Iperf and MaxiNet were used concurrently with Mininet to accomplish the needed SDN network simulations. OMNET++, Python were also used together in [111]. This was because the authors combined deep learning and reinforcement learning techniques, as they only learn from rewards by trial-and-error. The use of various tools demonstrates the fact that SDN gives the network administrator the flexibility to develop network administration methods that are not limited by the tools or by equipment vendors.

5.2. Open Issues

This subsection addresses some issues that have been left open in SDN intelligent load balancing. Some of the reviewed techniques did not pay much attention to energy saving. As lots of devices

connect to the internet, management and optimization of energy becomes paramount. When designing algorithms that manage traffic flow in the SDN network, the issue of energy conservation is of paramount importance and still remains open for future research.

Intelligent SDN load balancing techniques should also consider the issue of link or node failures. Incorporating failure recoveries management on some of the proposed intelligent load balancing techniques would be very promising for the future works.

Another open issue includes adaptation of the proposed SDN intelligent load balancing strategies in 5G networks. This is because 5G is regarded as a new revolutionary technology for mobile networks. Moreover, 5G networks are expected to be more efficient, achieve higher convenience, and handle a high number of users, applications like smart vehicles, video surveillance, smart cities and mobile industrial, and automation [116–120]—hence opening opportunities for the development of complex connections of powerful heterogeneous infrastructures with high traffic capacity. To cope with this level of complexity, 5G networks must be augmented with some level of intelligence load balancing, so as to ensure a successful deployment. SDN and NFV in particular would effectively offer such intelligence. Even though some work has been done in SDN load balancing for 5G, see [121,122], we believe more work still needs to be done, precisely in the development and testing of intelligent load balancers for 5G software defined networks. The potential works could focus on measures like congestion control, security, and QoS. This is because of the anticipated high traffic and number of devices in 5G networks.

6. Conclusions

In this paper, we presented a brief summary of SDN network, its architecture, and how the SDN has effectively improved communication in the fast and growing technology world. We discussed the OpenFlow technology, its architecture, and the role it plays in the effectiveness of SDN networks. The importance of SDN and NFV in terms of facilitating intelligent load balancing was also discussed. SDN and NFV enable network administrators to provide automatized network management and efficient resource allocation and provision. We conducted a review of the literature on intelligent SDN based load balancing by analyzing different techniques. The reviewed work was categorized according to the objective of the authors.

A summary of common intelligent load balancing techniques, their main performance objectives, and their selection criteria was presented in Table 1. Table 2 outlined some current research challenges, existing solutions, and future directions for load balancing techniques.

The performance metrics of some selected techniques were summarized in Table 3, and we also cover the tools used in the implementation and testing of these techniques. Mininet was found to be the most commonly used emulator by researchers because of its ability to simulate SDN based devices. In some cases, the Mininet emulator was used with other traffic generating tools like Iperf. From the sampled reviewed work in Table 3, most authors mainly tested their solutions against QoS, delay, and throughput metrics. Almost all effective solutions for intelligent load balancing are expected to perform exceptionally against these metrics.

Author Contributions: Conceptualization, T.S. and T.M.; writing—original draft preparation, T.S.; writing—review and editing, T.S., T.M., S.A., K.K., S.D., G.B., and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Office of Research, Development, and Innovation (ORDI), Botswana International University of Science and Technology.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- Long, H.; Shen, Y.; Guo, M.; Tang, F. LABERIO: Dynamic load-balanced routing in OpenFlow-enabled networks. In Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), Barcelona, Spain, 25–28 March 2013; pp. 290–297.
- Hakiri, A.; Gokhale, A.; Berthou, P.; Schmidt, D.C.; Gayraud, T. Software-defined networking: Challenges and research opportunities for future internet. *Comput. Networks* **2014**, *75*, 453–471.
- Erel, M.; Arslan, Z.; Ozcevik, Y.; Canberk, B. Software-defined wireless network (SDWN): A new paradigm for next generation network management. In *Modeling and Simulation of Computer Networks and Systems*; Elsevier: Amsterdam, The Netherlands, 2015; pp. 751–766.
- Abdelaziz, A.; Fong, A.T.; Gani, A.; Garba, U.; Khan, S.; Akhunzada, A.; Talebian, H.; Choo, K.K.R. Distributed controller clustering in software defined networks. *PLoS ONE* **2017**, *12*, e0174715.
- Neghabi, A.A.; Navimipour, N.J.; Hosseinzadeh, M.; Rezaee, A. Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature. *IEEE Access* **2018**, *6*, 14159–14178.
- Martinez-Julia, P.; Skarmeta, A.F. Empowering the internet of things with software defined networking. In *White Paper, IoT6-FP7 European Research Project*; 2014. Available online: <https://www.semanticscholar.org/search?q=Empowering-the-Internet-of-Things-with-Software&sort=relevance> (accessed on 3 May 2020).
- Mijumbi, R.; Serrat, J.; Gorricho, J.L.; Latré, S.; Charalambides, M.; Lopez, D. Management and orchestration challenges in network functions virtualization. *IEEE Commun. Mag.* **2016**, *54*, 98–105.
- Li, Y.; Phan, L.T.X.; Loo, B.T. Network functions virtualization with soft real-time guarantees. In Proceedings of the IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
- Herrera, J.G.; Botero, J.F. Resource allocation in NFV: A comprehensive survey. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 518–532.
- ETSI, N. Network functions virtualisation (nfv); management and orchestration. *NFV-MAN* **2014**, *1*, v0.
- Dao, N.N.; Kim, J.; Park, M.; Cho, S. Adaptive suspicious prevention for defending DoS attacks in SDN-based convergent networks. *PLOS ONE* **2016**, *11*, e0160375.
- Yu, C.; Zhao, Z.; Zhou, Y.; Zhang, H. Intelligent Optimizing Scheme for Load Balancing in Software Defined Networks. In Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, Australia, 4–7 June 2017; pp. 1–5.
- Huang, H.; Guo, S.; Li, P.; Liang, W.; Zomaya, A.Y. Cost minimization for rule caching in software defined networking. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *27*, 1007–1016.
- Kreutz, D.; Ramos, F.M.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2014**, *103*, 14–76.
- Semong, T.; Xie, K.; Zhou, X.; Singh, H.K.; Li, Z. Delay bounded multi-source multicast in software-defined networking. *Electronics* **2018**, *7*, 10.
- Jingjing, Z.; Di, C.; Weiming, W.; Rong, J.; Xiaochun, W. The deployment of routing protocols in distributed control plane of SDN. *Sci. World J.* **2014**, *2014*, 918536.
- Wang, L.; Li, Q.; Sinnott, R.; Jiang, Y.; Wu, J. An intelligent rule management scheme for Software Defined Networking. *Comput. Networks* **2018**, *144*, 77–88.
- Cybenko, G. Dynamic load balancing for distributed memory multiprocessors. *J. Parallel Distrib. Comput.* **1989**, *7*, 279–301.
- Liu, Z.; Lin, M.; Wierman, A.; Low, S.H.; Andrew, L.L. Greening geographical load balancing. *ACM SIGMETRICS Perform. Eval. Rev.* **2011**, *39*, 193–204.
- Yassine, A.; Rahimi, H.; Shirmohammadi, S. Software defined network traffic measurement: Current trends and challenges. *IEEE Instrum. Meas. Mag.* **2015**, *18*, 42–50.
- Xia, W.; Wen, Y.; Foh, C.H.; Niyato, D.; Xie, H. A survey on software-defined networking. *IEEE Commun. Surv. Tutor.* **2014**, *17*, 27–51.
- Mendiola, A.; Astorga, J.; Jacob, E.; Higuero, M. A survey on the contributions of software-defined networking to traffic engineering. *IEEE Commun. Surv. Tutor.* **2016**, *19*, 918–953.
- Farhady, H.; Lee, H.; Nakao, A. Software-defined networking: A survey. *Comput. Netw.* **2015**, *81*, 79–95.

24. Singh, S.; Jha, R.K. A survey on software defined networking: Architecture for next generation network. *J. Netw. Syst. Manag.* **2017**, *25*, 321–374.
25. Wickboldt, J.A.; De Jesus, W.P.; Isolani, P.H.; Both, C.B.; Rochol, J.; Granville, L.Z. Software-defined networking: management requirements and challenges. *IEEE Commun. Mag.* **2015**, *53*, 278–285.
26. Lara, A.; Kolasani, A.; Ramamurthy, B. Network innovation using openflow: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 493–512.
27. Jain, S.; Kumar, A.; Mandal, S.; Ong, J.; Poutievski, L.; Singh, A.; Venkata, S.; Wanderer, J.; Zhou, J.; Zhu, M.; et al. B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Comput. Commun. Rev.* **2013**, *43*, 3–14.
28. Jammal, M.; Singh, T.; Shami, A.; Asal, R.; Li, Y. Software defined networking: State of the art and research challenges. *Comput. Networks* **2014**, *72*, 74–98.
29. Lin, W.; Zhang, L. The load balancing research of SDN based on ant colony algorithm with job classification. In *2016 2nd Workshop on Advanced Research and Technology in Industry Applications (WARTIA-16)*; Atlantis Press: Paris, France, 2016.
30. Moy, J.T. *OSPF Version 2 (RFC 2328)*; Ascend Communications Inc.: Alameda, CA, USA, 1998.
31. Moy, J.T. *OSPF: Anatomy of an Internet Routing Protocol*; Addison-Wesley Professional: Boston, MA, USA, 1998.
32. Oran, D. OSI IS-IS Intra-domain Routing Protocol. *Network* **1990**. Available online: <https://dl.acm.org/doi/pdf/10.17487/RFC1142> (accessed on 10 May 2020).
33. Barreto, F.; Wille, E.C.; Nacamura, L., Jr. Fast emergency paths schema to overcome transient link failures in ospf routing. *arXiv* **2012**, arXiv:1204.2465.
34. Nelakuditi, S.; Lee, S.; Yu, Y.; Zhang, Z.L.; Chuah, C.N. Fast local rerouting for handling transient link failures. *IEEE/ACM Trans. Netw.* **2007**, *15*, 359–372.
35. Islam, S. Network Load Balancing Methods: Experimental Comparisons and Improvement. *arXiv* **2017**, arXiv:1710.06957.
36. Zou, S.; Huang, J.; Jiang, W.; Wang, J. Achieving high utilization of flowlet-based load balancing in data center networks. *Future Gener. Comput. Syst.* **2020**, *108*, 546–559.
37. Vijayakumar, V.; Joseph, K.S. Adaptive load balancing schema for efficient data dissemination in Vehicular Ad-Hoc Network VANET. *Alex. Eng. J.* **2019**, *58*, 1157–1166.
38. Adhikari, M.; Amgoth, T. Heuristic-based load-balancing algorithm for IaaS cloud. *Future Gener. Comput. Syst.* **2018**, *81*, 156–165.
39. Levin, A.; Lorenz, D.; Merlino, G.; Panarello, A.; Puliafito, A.; Tricomi, G. Hierarchical load balancing as a service for federated cloud networks. *Comput. Commun.* **2018**, *129*, 125–137.
40. Bestavros, A.; Crovella, M.; Liu, J.; Martin, D. Distributed packet rewriting and its application to scalable server architectures. In *Proceedings of the Sixth International Conference on Network Protocols*, Austin, TX, USA, 13–16 October 1998; pp. 290–297.
41. Aversa, L.; Bestavros, A. Load balancing a cluster of web servers: using distributed packet rewriting. In *Proceedings of the 2000 IEEE International Performance, Computing, and Communications Conference*, Phoenix, AZ, USA, 5–8 February 2000; pp. 24–29.
42. Katz, E.D.; Butler, M.; McGrath, R. A scalable HTTP server: The NCSA prototype. *Comput. Netw. ISDN Syst.* **1994**, *27*, 155–164.
43. Hu, T.; Yi, P.; Zhang, J.; Lan, J. Reliable and load balance-aware multi-controller deployment in SDN. *China Commun.* **2018**, *15*, 184–198.
44. Sufiev, H.; Haddad, Y.; Barenboim, L.; Soler, J. Dynamic SDN controller load balancing. *Future Internet* **2019**, *11*, 75.
45. Ma, Y.W.; Chen, J.L.; Tsai, Y.H.; Cheng, K.H.; Hung, W.C. Load-balancing multiple controllers mechanism for software-defined networking. *Wirel. Pers. Commun.* **2017**, *94*, 3549–3574.
46. Zhou, Y.; Zhu, M.; Xiao, L.; Ruan, L.; Duan, W.; Li, D.; Liu, R.; Zhu, M. A load balancing strategy of sdn controller based on distributed decision. In *Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, Beijing, China, 24–26 September 2014; pp. 851–856.
47. Hu, Y.; Wang, W.; Gong, X.; Que, X.; Cheng, S. Balanceflow: controller load balancing for OpenFlow networks. In *Proceedings of the 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, Hangzhou, China, 30 October–1 November 2012; Volume 2, pp. 780–785.

48. Dixit, A.; Hao, F.; Mukherjee, S.; Lakshman, T.; Kompella, R.R. ElastiCon; an elastic distributed SDN controller. In Proceedings of the 2014 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), Marina del Rey, CA, USA, 20–21 October 2014; pp. 17–27.
49. Yu, J.; Wang, Y.; Pei, K.; Zhang, S.; Li, J. A load balancing mechanism for multiple SDN controllers based on load informing strategy. In Proceedings of the 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Kanazawa, Japan, 5–7 October 2016; pp. 1–4.
50. Liang, C.; Kawashima, R.; Matsuo, H. Scalable and crash-tolerant load balancing based on switch migration for multiple open flow controllers. In Proceedings of the 2014 Second International Symposium on Computing and Networking, Shizuoka, Japan, 10–12 December 2014; pp. 171–177.
51. Zhou, Y.; Wang, Y.; Yu, J.; Ba, J.; Zhang, S. Load balancing for multiple controllers in SDN based on switches group. In Proceedings of the 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), Seoul, Korea, 27–29 September 2017; pp. 227–230.
52. Hamed, M.; ElHalawany, B.; Fouda, M.; Eldien, A.T. Performance analysis of applying load balancing strategies on different SDN environments. *Benha J. Appl. Sci. (BJSA)* **2017**, *2*, 91–97.
53. Chou, L.D.; Yang, Y.T.; Hong, Y.M.; Hu, J.K.; Jean, B. A genetic-based load balancing algorithm in OpenFlow network. In *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 411–417.
54. Govindarajan, K.; Kumar, V.S. An intelligent load balancer for software defined networking (SDN) based cloud infrastructure. In Proceedings of the 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 22–24 February 2017; pp. 1–6.
55. Arahunashi, A.K.; Vaidya, G.G.; Neethu, S.; Reddy, K.V. Implementation of Server Load Balancing Techniques Using Software-Defined Networking. In Proceedings of the 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 20–22 December 2018; pp. 87–90.
56. Kaur, S.; Singh, J. Implementation of server load balancing in software defined networking. In *Information Systems Design and Intelligent Applications*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 147–157.
57. Kavana, H.; Kavaya, V.; Madhura, B.; Kamat, N. Load balancing using SDN methodology. *Int. J. Eng. Res. Technol.* **2018**, *7*, 206–208.
58. Hamed, M.I.; ElHalawany, B.M.; Fouda, M.M.; Eldien, A.S.T. A new approach for server-based load balancing using software-defined networking. In Proceedings of the 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2017; pp. 30–35.
59. Montazerolghaem, A. SIP Server Load Balancing Based on SDN. *arXiv* **2019**, arXiv:1908.04047.
60. Zhang, H.; Guo, X. SDN-based load balancing strategy for server cluster. In Proceedings of the 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, Shenzhen, China, 27–29 November 2014; pp. 662–667.
61. Sabiya, J.S. Weighted round-robin load balancing using software defined networking. *Int. J. Adv. Res. Comput. Sci. Softw. Eng. (IJARCSSE)* **2016**, *6*, 621–625.
62. Yen, L.H.; Yeh, T.T.; Chi, K.H. Load balancing in IEEE 802.11 networks. *IEEE Internet Comput.* **2009**, *13*, 56–64.
63. Lin, Y.D.; Wang, C.C.; Lu, Y.J.; Lai, Y.C.; Yang, H.C. Two-tier dynamic load balancing in SDN-enabled Wi-Fi networks. *Wirel. Netw.* **2018**, *24*, 2811–2823.
64. Lin, C.Y.; Tsai, W.P.; Tsai, M.H.; Cai, Y.Z. Adaptive load-balancing scheme through wireless SDN-based association control. In Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan, 27–29 March 2017; pp. 546–553.
65. Abbasi, R.; Gilani, S.M.M.; Kabir, A.; Nawaz, Q.; Riaz, M.S. Load Balancing of SDN-enabled Wireless Network: Challenges, Technique and Evaluation. *J. Inf. Commun. Technol. Robot. Appl.* **2019**, *10*, pp. 80–92.
66. Bilaw, N.G.; Buslig, J.G.D.; Festin, C.A.M.; Tan, W.M. An SDN-based Dual AP Load Balancing Scheme for Wireless Networks. In Proceedings of the 2019 International Symposium on Multimedia and Communication Technology (ISMAT), Quezon City, Philippines, 19–21 August 2019; pp. 1–6.
67. MengHeang, H.; Gilani, S.M.; Hong, T.; Zhao, G.; B. Abdalla, H. Load Balancing in Wireless Networks using SDN-enabled Infrastructure: Traffic Analysis. In Proceedings of the 10th EAI International Conference on Mobile Multimedia Communications, 13–14 July 2017; pp. 34–40.
68. Zhong, H.; Lin, Q.; Cui, J.; Shi, R.; Liu, L. An efficient SDN load balancing scheme based on variance analysis for massive mobile users. *Mob. Inf. Syst.* **2015**, *2015*, 241732.

69. Cui, X.; Huang, X.; Ma, Y.; Meng, Q. A load balancing routing mechanism based on SDWSN in smart city. *Electronics* **2019**, *8*, 273.
70. Tang, W.; Liao, Q. An sdn-based approach for load balance in heterogeneous radio access networks. In Proceedings of the 2014 IEEE Symposium on Computer Applications and Communications, Weihai, China, 26–27 July 2014; pp. 105–108.
71. Boucadair, M.; Jacquenet, C. Software-defined networking: A perspective from within a service provider environment. In *RFC 7149*; 2014. Available online: <https://www.hjp.at/doc/rfc/rfc7149.html> (accessed on 3 May 2020).
72. Jamali, S.; Badirzadeh, A.; Siapoush, M.S. On the use of the genetic programming for balanced load distribution in software-defined networks. *Digit. Commun. Netw.* **2019**, *5*, 288–296.
73. Lan, Y.L.; Wang, K.; Hsu, Y.H. Dynamic load-balanced path optimization in SDN-based data center networks. In Proceedings of the 2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Prague, Czech Republic, 20–22 July 2016; pp. 1–6.
74. Li, Y.; Pan, D. OpenFlow based load balancing for fat-tree networks with multipath support. In Proceedings of the 12th IEEE International Conference on Communications (ICC' 13), Budapest, Hungary, 9–13 June 2013; pp. 1–5.
75. Bindhu, M.; Ramesh, G. Load balancing and congestion control in software defined networking using the extended Johnson algorithm for data centre. *Int. J. Appl. Eng. Res.* **2015**, *10*, 12911–12914.
76. Wang, C.; Hu, B.; Chen, S.; Li, D.; Liu, B. A switch migration-based decision-making scheme for balancing load in SDN. *IEEE Access* **2017**, *5*, 4537–4544.
77. Xu, H.; Li, X.Y.; Huang, L.; Du, Y.; Liu, Z. Partial flow statistics collection for load-balanced routing in software defined networks. *Comput. Netw.* **2017**, *122*, 43–55.
78. Tu, R.; Wang, X.; Zhao, J.; Yang, Y.; Shi, L.; Wolf, T. Design of a load-balancing middlebox based on SDN for data centers. In Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, China, 26 April–1 May 2015; pp. 480–485.
79. Xie, X.; Wu, J. Real-time flow identification based on neural network and OpenFlow over SDN. In Proceedings of the 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 25–26 August 2018; Volume 2, pp. 12–16.
80. Yang, C.T.; Chen, S.T.; Liu, J.C.; Su, Y.W.; Puthal, D.; Ranjan, R. A predictive load balancing technique for software defined networked cloud services. *Computing* **2019**, *101*, 211–235.
81. Rupani, K.; Punjabi, N.; Shamdasani, M.; Chaudhari, S. Dynamic Load Balancing in Software-Defined Networks Using Machine Learning. In *Proceeding of International Conference on Computational Science and Applications*; Springer: Singapore, 2020; pp. 283–292.
82. WilsonPrakash, S.; Deepalakshmi, P. Artificial Neural Network Based Load Balancing On Software Defined Networking. In Proceedings of the 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 11–13 April 2019; pp. 1–4.
83. Chen-Xiao, C.; Ya-Bin, X. Research on load balance method in SDN. *Int. J. Grid Distrib. Comput.* **2016**, *9*, 25–36.
84. Ruelas, A.M.; Rothenberg, C.E. A Load Balancing Method based on Artificial Neural Networks for Knowledge-defined Data Center Networking. In Proceedings of the 10th Latin America Networking Conference, Sao Paulo, Brazil, 3–4 October 2018; pp. 106–109.
85. Hsu, K.I.; Gupta, H.V.; Sorooshian, S. Artificial neural network modeling of the rainfall-runoff process. *Water Resour. Res.* **1995**, *31*, 2517–2530.
86. Huong, T.T.; Khoa, N.D.D.; Dung, N.X.; Thanh, N.H. A global multipath load-balanced routing algorithm based on Reinforcement Learning in SDN. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 16–18 October 2019; pp. 1336–1341.
87. Li, Z.; Zhou, X.; Gao, J.; Qin, Y. SDN controller load balancing based on reinforcement learning. In Proceedings of the 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 November 2018; pp. 1120–1126.
88. Chien, W.C.; Lai, C.F.; Cho, H.H.; Chao, H.C. A SDN-SFC-based service-oriented load balancing for the IoT applications. *J. Netw. Comput. Appl.* **2018**, *114*, 88–97.

89. Semong, T.; Xie, K. Efficient load balancing and multicasting for Uncertain-Source SDN: Real-Time link-cost monitoring. In *Computer Science Online Conference*; Springer: Cham, Switzerland, 2018; pp. 178–187.
90. Shahryari, S.; Hosseini-Seno, S.A.; Tashtarian, F. An SDN based framework for maximizing throughput and balanced load distribution in a Cloudlet network. *Future Gener. Comput. Syst.* **2020**, *110*, 18–32.
91. Zhao, Y.; Liu, C.; Wang, H.; Fu, X.; Shao, Q.; Zhang, J. Load balancing-based multi-controller coordinated deployment strategy in software defined optical networks. *Opt. Fiber Technol.* **2018**, *46*, 198–204.
92. Suwandika, I.P.A.; Nugroho, M.A.; Abdurahman, M. Increasing SDN performance using load balancing scheme on Web Server. In Proceedings of the 2018 6th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 3–5 May 2018; pp. 459–463.
93. Mustafa, M.; Ibrahim, A.M. Load balancing algorithms Round-Robin (RR), Least-Connection and least loaded efficiency. *GESJ Comput. Sci. Telecommun.* **2017**, *1*, 25–29.
94. Ejaz, S.; Iqbal, Z.; Shah, P.A.; Bukhari, B.H.; Ali, A.; Aadil, F. Traffic load balancing using Software Defined Networking (SDN) controller as virtualized network function. *IEEE Access* **2019**, *7*, 46646–46658.
95. Li, J.; Chang, X.; Ren, Y.; Zhang, Z.; Wang, G. An effective path load balancing mechanism based on SDN. In Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, Beijing, China, 24–26 September 2014; pp. 527–533.
96. Askar, S.K. Adaptive load balancing scheme for data center networks using software defined network. *Sci. J. Univ. Zakho* **2016**, *4*, 275–286.
97. Priyadarsini, M.; Mukherjee, J.C.; Bera, P.; Kumar, S.; Jakaria, A.; Rahman, M.A. An adaptive load balancing scheme for software-defined network controllers. *Comput. Netw.* **2019**, *164*, 106918.
98. Pei, J.; Hong, P.; Pan, M.; Liu, J.; Zhou, J. Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks. *IEEE J. Sel. Areas Commun.* **2019**, *38*, 263–278.
99. Guo, Z.; Su, M.; Xu, Y.; Duan, Z.; Wang, L.; Hui, S.; Chao, H.J. Improving the performance of load balancing in software-defined networks through load variance-based synchronization. *Comput. Netw.* **2014**, *68*, 95–109.
100. Hai, N.T.; Kim, D.S. Efficient load balancing for multi-controller in SDN-based mission-critical networks. In Proceedings of the 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, France, 19–21 July 2016; pp. 420–425.
101. El Kamel, A.; Youssef, H. Improving Switch-to-Controller Assignment with Load Balancing in Multi-controller Software Defined WAN (SD-WAN). *J. Netw. Syst. Manag.* **2020**, *28*, 553–575.
102. Zhao, J.; Tong, M.; Qu, H.; Zhao, J. An Intelligent Congestion Control Method in Software Defined Networks. In Proceedings of the 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN), Chongqing, China, 12–15 June 2019; pp. 51–56.
103. Yu, C.; Lan, J.; Guo, Z.; Hu, Y. DROM: Optimizing the routing in software-defined networks with deep reinforcement learning. *IEEE Access* **2018**, *6*, 64533–64539.
104. Sun, P.; Hu, Y.; Lan, J.; Tian, L.; Chen, M. TIDE: Time-relevant deep reinforcement learning for routing optimization. *Future Gener. Comput. Syst.* **2019**, *99*, 401–409.
105. Zhou, Y.; Ruan, L.; Xiao, L.; Liu, R. A method for load balancing based on software defined network. *Adv. Sci. Technol. Lett.* **2014**, *45*, 43–48.
106. Mininet Team. Mininet: An Instant Virtual Network on Your Laptop (or Other PC). Available online: <http://mininet.org/> (accessed on 1 May 2020).
107. Lantz, B.; Heller, B.; McKeown, N. A network in a laptop: Rapid prototyping for software-defined networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Monterey, CA, USA, 20–21 October 2010; pp. 1–6.
108. Varga, A.; Hornig, R. An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*; ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): Kolkata, India, 2008; p. 60.

109. Mallik, A.; Hegde, S. A novel proposal to effectively combine multipath data forwarding for data center networks with congestion control and load balancing using Software-Defined Networking Approach. In Proceedings of the 2014 International Conference on Recent Trends in Information Technology, Chennai, India, 10–12 April 2014; pp. 1–7.
110. Dugan, J.; Elliott, S.; Mah, B.A.; Poskanzer, J.; Prabhu, K. iPerf—The Ultimate Speed Test Tool for TCP, UDP and SCTP. Available online: <https://iperf.fr/> (accessed on 1 May 2020).
111. Huang, X.; Yuan, T.; Qiao, G.; Ren, Y. Deep reinforcement learning for multimedia traffic control in software defined networking. *IEEE Netw.* **2018**, *32*, 35–41.
112. Xue, H.; Kim, K.T.; Youn, H.Y. Dynamic Load Balancing of Software-Defined Networking Based on Genetic-Ant Colony Optimization. *Sensors* **2019**, *19*, 311.
113. Prakash, W.S.; Deepalakshmi, P. DServ-LB: Dynamic server load balancing algorithm. *Int. J. Commun. Syst.* **2019**, *32*, e3840.
114. Wette, P.; Dräxler, M.; Schwabe, A.; Wallaschek, F.; Zahraee, M.H.; Karl, H. Maxinet: Distributed emulation of software-defined networks. In Proceedings of the 2014 IFIP Networking Conference, Trondheim, Norway, 2–4 June 2014; pp. 1–9.
115. Cui, J.; Lu, Q.; Zhong, H.; Tian, M.; Liu, L. A load-balancing mechanism for distributed SDN control plane using response time. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 1197–1206.
116. Chin, W.H.; Fan, Z.; Haines, R. Emerging technologies and research challenges for 5G wireless networks. *IEEE Wirel. Commun.* **2014**, *21*, 106–112.
117. Cho, H.H.; Lai, C.F.; Shih, T.K.; Chao, H.C. Integration of SDR and SDN for 5G. *IEEE Access* **2014**, *2*, 1196–1204.
118. Wang, L.C.; Rangapillai, S. A survey on green 5G cellular networks. In Proceedings of the 2012 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 22–25 July 2012; pp. 1–5.
119. Gohil, A.; Modi, H.; Patel, S.K. 5G technology of mobile communication: A survey. In Proceedings of the 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), Gujarat, India, 1–2 March 2013; pp. 288–292.
120. Thompson, J.; Ge, X.; Wu, H.C.; Irmer, R.; Jiang, H.; Fettweis, G.; Alamouti, S. 5G wireless communication systems: Prospects and challenges [Guest Editorial]. *IEEE Commun. Mag.* **2014**, *52*, 62–64.
121. Sun, G.; Liu, F.; Lai, J.; Liu, G. Software defined wireless network architecture for the next generation mobile communication: Proposal and initial prototype. *J. Commun.* **2014**, *9*, 946–953.
122. Bouras, C.; Ntarzanos, P.; Papazois, A. Cost modeling for SDN/NFV based mobile 5G networks. In Proceedings of the 2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Lisbon, Portugal, 18–20 October 2016; pp. 56–61.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).